Gem5 Lab 3: Cache Optimization

Fall, 2020

Due: This lab is to be completed during your time slots. After you are done, submit everything to github classroom.

The purpose of this assignment is to supplement your knowledge about trade-offs behind the design of caches, alongside intuition about their relative performance.

Cloud Access

You can connect to the cloud Linux server (**2440.cs.uh.edu**) with any ssh client during your time slot. Alternatively, you can use your own computer with Oracle Virtualbox. Keep in mind in the Linux server that we provide, gem5 is installed under /opt/gem5. In the Oracle Virtualbox disk image, it is installed under /home/gem5/gem5.

Windows ssh client
https://mobaxterm.mobatek.net/download.html
https://www.putty.org/
Mac OS
Mac OS has a built-in SSH client.
Android
https://android.md/2019/05/10/5-best-android-ssh-clients/

Cache design trade-off

You will use bzip2 as the benchmarks in this lab.

The execution of gem5 provides the output file "**stats.txt**" under folder "**m5out**" (default, or into your defined directory). We can find **miss rates** of L1 DCache, L1 ICache and L2 Cache in the file, for instance,

system.cpu.dcache.overall_miss_rate::total 0.002790 # miss rate for overall accesses
system.cpu.icache.overall_miss_rate::total 0.000004 # miss rate for overall accesses
...
From statics above, the L1 DCache has a 0.2790% miss rate, L1 ICache 0.0004%, and etc.

You can also find stats about number of cache accesses, hits and misses in “stats.txt”.

Use config.ini to find cache configuration information such as latency for both Icache and Dcache. For instance, below shows icache latency of 2 cycles.

[system.cpu.icache]
type=Cache
...
assoc=2
...
data_latency=2

Deliverable: Given an L1 miss penalty of 6 cycles, L2 miss penalty of 50 cycles, and one cycle cache hit/instruction execution, use configuration parameter as above and calculate the CPI for each benchmark by following equation.

$$CPI = 1 + \frac{(IL1.miss\_num + DL1.miss\_num) \times 6 + L2.miss\_num \times 50}{Total\_Inst\_num}$$

Optimize CPI for bzip2 benchmark

If we repeat the previous experiment, with a different configuration, we will find that many factors can influence the performance of program and cache hierarchy. By exploring the design space and trying different configurations, we will try to find an optimal configuration of cache which provides the best performance (i.e. **lowest CPI**).

You should explore the trade-offs between different factors, **which include associativity, block size, and size allocation for L1 instruction cache and L1 data cache (they may have unequal size).**

Given a two-level cache hierarchy, 512KB available for L1 cache (for L1 d-cache and i-cache together) and 4MB available for L2 cache.

Keep in mind from the class note, L1 caches often have relatively small size and are optimized for speed. L2 cache often has much larger size and higher associativity. Caches with larger size and higher associativity are slower (hit latency).

The experiments should be done with simulation of **100 million** instructions. It may take about 15 minutes to complete simulation of 100 million instructions using the server that we provide. Only in case it takes too long to complete (for instance, more than 40 minutes)), you can pick a different number. It is preferred to be more than 50 million. If you use a number different from 100 million, please state it in the lab report.

Experiment with cache *associativity, block size, and size allocation for L1 instruction cache, and L1 data cache*. Run Gem5 simulation with following cache configurations.

		L1 icache	L1 dcache	L2 cache
Setting 1	**Large L1, direct map**	256 KB, direct map	256 KB, direct map	8MB, 8 way associate
Setting 2	**Small L1, direct map**	32 KB, direct map	32 KB, direct map	8MB, 8 way associate
Setting 3	**L1, 2 way**	64KB, 4 way	64KB, 4 way	4MB, 16 way
Setting 4	**L1, 8 way**	64 KB, 8 way	64 KB, 8 way	4MB, 16 way

	Total Instruction count	IL1 misses	DL1 misses	L2 misses	L1 miss penalty	L2 miss penalty	IPC
Setting 1	100000000	1207	572577	172388	6	50	0.988709
Setting 2	100000000	3321	1350643	172395	6	50	0.878854
Setting 3	100000000	1189	822126	180249	6	50	0.950393
Setting 4	100000000	1189	788656	180248	6	50	0.954127

Q1: Which configuration achieve the lowest CPI for the benchmark?

CPI=1+((IL1.miss_num+DL1.miss_num)*6+L2.miss_num*50)/total_inst_numUsingthetableabove:setting1:CPI=1.12(

Submit

A report to the questions.
- A report containing the answers for the questions, named as, **gem5_lab3_cache_report.pdf**
- The statistics output files generated by the simulator (**stats.txt**)
- **config.ini** under m5out

Note that you must submit stats.txt and config.ini files for all your experiments. To make TA's grading job easier, label the files that you have submitted using the table below:

	Name of stats.txt	**Name of config.ini**
Experiment for setting 1	stats_setting1.txt	config_setting1.ini
Experiment for setting 2	stats_setting2.txt	config_setting2.ini
Experiment for setting 3	stats_setting3.txt	config_setting3.ini
Experiment for setting 4	stats_setting4.txt	config_setting4.ini

Command line for submitting files to github

```
cd [your github repository location]
git add "*"
git commit -m "Logisim lab3"
git push origin master
```