ELEN 511
Advanced Computer Architecture

Homework #1
gem5

Prof. Hoeseok Yang
Santa Clara University

Fall Quarter 2022

Homework #1

- In this homework, you will practice the basics of gem5 (a full computer system simulator)
  - Disclaimer: this homework is based on the gem5 tutorial available here: https://www.gem5.org/documentation/learning_gem5/introduction/

- Tasks – 15 + 2 points
  - Part 0: setup – 2 points
    - Building gem5
  - Part 1: setup – 2 points
    - Creating a simple configuration
  - Part 2 write a test application – 3 points
    - Write a C-program (Sieve of Eratosthenes) and have it running on gem5
  - Part 3 Analysis – 2 points
    - Analyze the statistics
  - Part 4: customizing configuration – 3 points
    - Change the configuration and analyze the statistics in comparison with the original configuration
  - Part 5: cache configuration – 2 points
  - Report - 3 points

- Submission
  - Upload your report (in pdf) and source files (configuration and C source files) to Camino
  - Due: Nov. 4th, 11:59pm

What is gem5?

- gem5 is a modular discrete event driven computer system simulator platform.

  1. gem5's components can be **rearranged, parameterized, extended** or replaced easily to suit your needs.

  2. It **simulates the passing of time** as a series of discrete events.

  3. Its intended use is to simulate one or more computer systems in various ways.

  4. It's more than just a simulator; it's a simulator platform that lets you use as many of its premade components as you want to **build up your own simulation system**.

Linux

- Recommended environment
  - Ubuntu on X86/X64
  - Tested on Ubuntu 22.04 ARM64

- If you don't have a Linux machine
  - You may use a (free) virtual machine
    - https://www.virtualbox.org/

Part 0: setup – Building gem5 (2 points)

- https://www.gem5.org/documentation/learning_gem5/part1/building/

- Required packages
  - Git, gcc, scons, python, protobuf, boost

- Getting the code
  - git clone https://gem5.googlesource.com/public/gem5

- Build
  - python3 `which scons` build/X86/gem5.opt -j9
    - Try without –j9 if not successufl

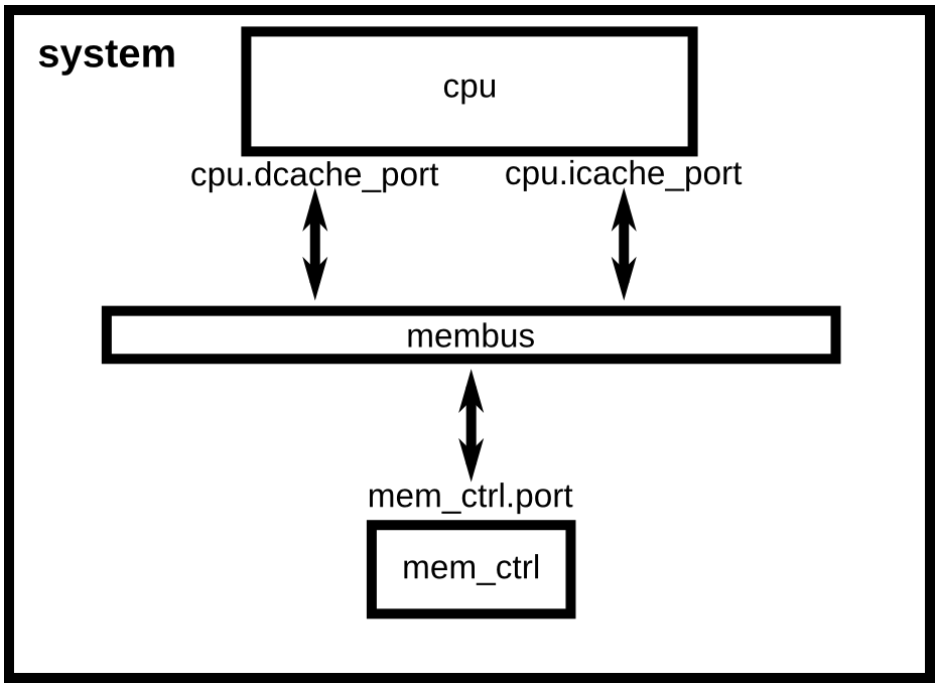
- Output:
  - Build/X86/gem5.opt

Part 0: setup – Building gem5 - screenshot

```
Variables file /local.chinook/gem5/gem5-tutorial/gem5/build/variables/X86 not found,
  using defaults in /local.chinook/gem5/gem5-tutorial/gem5/build_opts/X86
scons: done reading SConscript files.
scons: Building targets ...
 [ISA DESC] X86/arch/x86/isa/main.isa -> generated/inc.d
 [NEW DEPS] X86/arch/x86/generated/inc.d -> x86-deps
 [ENVIRONS] x86-deps -> x86-environs
 [     CXX] X86/sim/main.cc -> .o
 ....
 .... <lots of output>
 ....
 [   SHCXX] nomali/lib/mali_midgard.cc -> .os
 [   SHCXX] nomali/lib/mali_t6xx.cc -> .os
 [   SHCXX] nomali/lib/mali_t7xx.cc -> .os
 [      AR]  -> drampower/libdrampower.a
 [   SHCXX] nomali/lib/addrspace.cc -> .os
 [   SHCXX] nomali/lib/mmu.cc -> .os
 [  RANLIB]  -> drampower/libdrampower.a
 [   SHCXX] nomali/lib/nomali_api.cc -> .os
 [      AR]  -> nomali/libnomali.a
 [  RANLIB]  -> nomali/libnomali.a
 [     CXX] X86/base/date.cc -> .o
 [    LINK]  -> X86/gem5.opt
scons: done building targets.
```

Part 1: creating a simple configuration (2 points)

- Follow
  - https://www.gem5.org/documentation/learning_gem5/part1/simple_config/



Part 1: creating a simple configuration – Running gem 5 (2 points)

- Build/X86/gem5.opt configs/tutorial/part1/simple.py

```
gem5 Simulator System.  http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 21.0.0.0
gem5 compiled May 17 2021 18:05:59
gem5 started May 17 2021 22:05:20
gem5 executing on amarillo, pid 75197
command line: build/X86/gem5.opt configs/tutorial/part1/simple.py

Global frequency set at 1000000000000 ticks per second
warn: No dot file generated. Please install pydot to generate the dot file and pdf.
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb: listening for remote gdb on port 7005
Beginning simulation!
info: Entering event queue @ 0.  Starting simulation...
Hello world!
Exiting @ tick 490394000 because exiting with last active thread context
```

Part 2: Write and run your own test application (3 points)

- Write a program that implements
  - Sieve of Eratosthenes
    - https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes
- and outputs one single integer at the end:
  - the number of prime numbers <= 100,000,000.
- Compile your program as a static binary.
  - The output should be: 5761455
- Run it on gem5
  - Run your sieve program in gem5 instead of the 'hello' example

Part 3: Analyze the statistics (3 points)

- Refer to the following page to understand the gem5 statistics and output
- Analyze the cache behavior of your simulation
- In addition to any information which your simulation script prints out, after running gem5, there are three files generated in a directory called m5out:
  - **config.ini**
    - Contains a list of every SimObject created for the simulation and the values for its parameters.
  - **config.json**
    - The same as config.ini, but in json format.
  - **stats.txt**
    - A text representation of all of the gem5 statistics registered for the simulation.

Part 3: Analyze the statistics - screenshot

```
system.clk_domain.clock                          1000                       # Clock period in ticks (Tick)
system.clk_domain.voltage_domain.voltage            1                       # Voltage in Volts (Volt)
system.cpu.numCycles                            57467                       # Number of cpu cycles simulated (Cycle)
system.cpu.numWorkItemsStarted                      0                       # Number of work items this cpu started (Count)
system.cpu.numWorkItemsCompleted                    0                       # Number of work items this cpu completed (Count)
system.cpu.dcache.demandHits::cpu.data           1941                       # number of demand (read+write) hits (Count)
system.cpu.dcache.demandHits::total              1941                       # number of demand (read+write) hits (Count)
system.cpu.dcache.overallHits::cpu.data          1941                       # number of overall hits (Count)
system.cpu.dcache.overallHits::total             1941                       # number of overall hits (Count)
system.cpu.dcache.demandMisses::cpu.data          133                       # number of demand (read+write) misses (Count)
system.cpu.dcache.demandMisses::total             133                       # number of demand (read+write) misses (Count)
system.cpu.dcache.overallMisses::cpu.data          133                       # number of overall misses (Count)
system.cpu.dcache.overallMisses::total             133                       # number of overall misses (Count)
system.cpu.dcache.demandMissLatency::cpu.data    14301000                       # number of demand (read+write) miss ticks (
system.cpu.dcache.demandMissLatency::total     14301000                       # number of demand (read+write) miss ticks (Tic
system.cpu.dcache.overallMissLatency::cpu.data    14301000                       # number of overall miss ticks (Tick)
system.cpu.dcache.overallMissLatency::total     14301000                       # number of overall miss ticks (Tick)
system.cpu.dcache.demandAccesses::cpu.data         2074                       # number of demand (read+write) accesses (Count
system.cpu.dcache.demandAccesses::total          2074                       # number of demand (read+write) accesses (Count)
system.cpu.dcache.overallAccesses::cpu.data         2074                       # number of overall (read+write) accesses (Cou
system.cpu.dcache.overallAccesses::total         2074                       # number of overall (read+write) accesses (Count)
system.cpu.dcache.demandMissRate::cpu.data     0.064127                       # miss rate for demand accesses (Ratio)
system.cpu.dcache.demandMissRate::total      0.064127                       # miss rate for demand accesses (Ratio)
system.cpu.dcache.overallMissRate::cpu.data     0.064127                       # miss rate for overall accesses (Ratio)
system.cpu.dcache.overallMissRate::total     0.064127                       # miss rate for overall accesses (Ratio)
system.cpu.dcache.demandAvgMissLatency::cpu.data 107526.315789                       # average overall miss latency ((Cycle/C
system.cpu.dcache.demandAvgMissLatency::total 107526.315789                       # average overall miss latency ((Cycle/Cour
system.cpu.dcache.overallAvgMissLatency::cpu.data 107526.315789                       # average overall miss latency ((Cycle/
system.cpu.dcache.overallAvgMissLatency::total 107526.315789                       # average overall miss latency ((Cycle/Cou
```

Part 4: Alternative Configurations (3 points)

- Change the CPU model
  - from TimingSimpleCPU to MinorCPU
    - Hint: you may want to add a command line parameter to control the CPU model.
- Vary the CPU clock from 1 GHz to 3 GHz (in steps of 500 MHz) with both CPU models
  - Hint: again, you may want to add a command line parameter for the frequency.
- Change the memory configuration
  - from DDR3_1600_x64 to DDR3_2133_x64 (DDR3 with a faster clock) and LPDDR2_S4_1066_x32 (low-power DRAM often found in mobile devices).
- Make comparison with the previous configuration

Part 5: Bonus points (2 points)

- See here:
  - https://www.gem5.org/documentation/learning_gem5/part1/cache_config/

- Try at least 3 different cache configurations with your test program
  - Different set associativity
  - Different block size
  - Different levels of caches

- Make comparative evaluations of the cache statistics between them

Report

- Must include
  - Overall procedure of your homework
  - Source codes
    - C (your test application)
    - Configuration files
  - Screenshot of important results
    - Building result
    - Simulation result
    - Statistics
  - Analysis
    - In particular for Part 3 and Part 4