Assignment#4

CSE 530, Fall 2024

Note:

- Please include your name and PSUID on the first page.
- Submit all files on Canvas.
- The assignment must be submitted on Canvas before the due date (midnight).
- No single line answers are accepted in the submission.
- Refer to the syllabus for late submission policies.
- No kind of collaboration is allowed unless specifically mentioned in the assignment.
- All source materials must be cited. The University Academic Code of Conduct will be strictly enforced.
- All queries related to Assignment should have a subject line **CSE530: Assignment#4**
- Please use CANVAS MESSAGES for any queries. Make sure to send your queries to both the TAs for a faster response.

DO NOT USE CHATGPT OR ANY OTHER SIMILAR TOOLS FOR THE COMPLETION OF THIS ASSIGNMENT!! ANY UTILIZATION OF SUCH RESOURCES TO COMPLETE THE ASSIGNMENT WOULD BE CONSIDERED AS AN ACT OF PLAGIARISM

Goal:

Experiment and investigate various performance impacts of different out-of-order core configurations for a benchmark.

Tools required: GEM5[1]
Benchmark - Pick one benchmark from /home/other/CSE530-FA2022/cse530_a4/benchmarks based on the first letter of your last name.

Letters	Benchmark
A - F	Rodinia[1]: openmp/bfs
G - L	Rodinia[1]: openmp/hotspot
M - S	Stanford[5]: quicksort
T- Z	Stanford[5]: bubblesort

Using GEM5 tool: Refer to Canvas – Assignment 3 [2, 3]

The GEM5 is built in "/home/other/CSE530-FA2022/gem5/build_a4/build/X86/gem5.opt"

Experiment details:

The OoO core configuration parameters of interest for this assignment are the pipeline width, branch predictor and number of entries in the reorder buffer.

You are given two OoO CPU configurations at the extreme – LargeCore and SmallCore. The LargeCore has pipeline width of 8, 192 ROB entries and LTAGE branch predictor [5]. The SmallCore has pipeline width of 2, 16 ROB entries and a 2-bit branch predictor.

You will be required to add more configurations by varying the three parameters in between LargeCore and SmallCore for experimentation, as needed – the goal is to pick a core design between the cost of the SmallCore and the LargeCore which gives most of the performance benefits of the large core but with fewer resources.

Details on simulation setup: Required files are in the folder: /home/other/CSE530-FA2022/cse530_a4

- Copy the cse530_a4 directory from /home/other/CSE530-FA2022/cse530_a4 to your grads folder: /home/grads/psu_userid.

  cp -r /home/other/CSE530-FA2022/cse530_a4/ /home/grads/psu_userid/cse530_a4

  Then make sure the necessary files are executable.

  chmod -R 777 /home/grads/psu_userid/cse530_a4
- The file a4CPUConfig.py (lines 26-37) has the hardcoded settings for LargeCore and SmallCore.
- cse530_a4_sys_config.py has a parameter named "cse530-core-config" which takes a string (large or small – lines 26 & 27).
- You can modify the string parameter in cse530_a4_sys_config.py to an integer and add more if/elif conditions in a4CPUConfig.py to add core configurations.
- You can change runbinaries.sh to suit your sweeps.
- You need to change **only** the three core parameters highlighted above. You need not change the cache/memory configurations.
- Execution: add the –binargs argument when executing the simulator. Example: Bubblesort benchmark
  /home/other/CSE530-FA2022/gem5/build_a4/build/X86/gem5.opt --outdir=/home/grads/psu_userid/Desktop/m5out_Bubblesort --redirect-stdout --redirect-stderr /home/grads/psu_userid/cse530_a4/cse530_a4_sys_config.py --cse530-core-config=small --binfile=/home/grads/psu_userid/cse530_a4/benchmarks/stanford/Bubblesort --binargs "" --l1i_size "4kB" --l1d_size "4kB" --l2_size "32kB"

Report Expectations:

- 4 pages maximum, please follow word limit while answering the questions
- The report should have two parts,
  1. Experiments you conducted to arrive at GoodCore configuration. You can arrive at GoodCore configuration in different ways. Here's one way,

- Define three or four configurations that are between SmallCore and LargeCore, say C1, C2, C3.
- Fix two parameters (say, branch predictor and #pipeline width) for all the configurations (SmallCore, C1, C2, C3, LargeCore) and vary #ROB entries alone.
- Observe the trends with respect to some related metric (please think and choose the right CPU performance metric, there can be multiple correct metrics). TAs can help you in figuring out the right metric but won't say an exact answer.
- Based on the observed trends, make a smart choice of the #ROB entries for the GoodCore.
- Rinse and repeat for the other two parameters to arrive at configuration for GoodCore

2. Answer the questions below based on your experiments.

Questions:

1. What is the benchmark you picked? Also paste the link to the zip folder for relevant simulation outputs & scripts. (**1 line**)
2. Which core parameter(s) have the most impact on performance? Are there any dependencies between the parameters (e.g., do you need to change two or more parameters at once to have an impact on the performance)? (**Max 150 words**)
3. What are the parameters you chose? Let us call this GoodCore. (**3 lines**)
4. What are the IPC improvements and execution time speedups of LargeCore and GoodCore with respect to the SmallCore? (**3 lines** / 1 table)
5. How did the benchmark characteristics affect your GoodCore design decisions? Look at the benchmark code (both the .c and .s files may be useful). (**Max 150 words**)
6. How did you reduce the search space? (**Max 50 words**)

References:

1. https://www.gem5.org/documentation/
2. https://www.gem5.org/documentation/learning_gem5/part1/building/
3. https://jlpteaching.github.io/comparch/modules/gem5/assignment3/
4. TAgged GEometric history length predictor https://jilp.org/vol8/v8paper1.pdf

FAQs:

Contact:

If your query is not CSE530-specific: Please type your query in Google and see if there are answers present in websites like stackoverflow.

Otherwise:

1. Read this document.
2. If you do not find an answer: Please include the following details so we can find solutions to your issues faster.
   a. Please mention what you have tried and what worked/did not work in your CANVAS email.
   b. Please explain the error (I am hoping in the process you will figure out the issue yourself). Do NOT just copy-paste the error. Do NOT just attach the image of the error without an explanation of what the error is.

GEM5:

- Please issue the runs ahead of time. You can use TMUX or other similar environments to let the run happen in the background. No support will be provided related to these please read about these online.

Grading:

- The grading will be relative however the points distribution is fixed -- curved based on class performance. Meaningful observations (that correlates the benchmark aspects with the observed simulation data) will get points. The rubric on the assignment page will show the points distribution.
- The report should be well-written with reasonings and legible graphs with axes labels. Do NOT provide absolute numbers. Please pick a valid baseline and normalize with respect to that (you will be penalized if you show absolute numbers).
- Don't forget to add relevant data to your report (this may include data from changing any other system knobs not listed in the assignment description) – that had an impact on the choices you made in choosing/skipping a configuration.
- Pick proper (first order metrics) metrics. On the report, the reasonings should involve why you picked the metric and how the variations of its statistics correlate with your understanding of the benchmark.

Q & A:

1. Do I need to copy home/other/CSE530-FA2022/cse530_a4/ to my machine?
   a. Yes. Copy cse530_a4 folder to your machine.
2. I get an error “NameError: name ‘Cse530O8FUPool’ is not defined.

    a. Please read the instructions. Use the proper GEM5 build for this assignment - build_a4/build/X86/gem5.opt
3. What metrics do I use?
    a. If you change branch predictor, see the stats that are associated with that module and pick the relevant metric.
4. Do we need to add m5_reset_stats(0,0) and m5_dump_stats(0,0) in the benchmark?
    a. Adding these would help in correlating the stats with the region of interest better.
5. Is it enough to use the four predictions - 2bit, tournament, bimode and ltage? Or we must choose more predictions from gem5/src/cpu/pred?
    a. You have all options in gem5/src/cpu/pred to choose from.
6. Do we need to obtain a non-zero warmup tick for this project as well?
    a. Yes, since usually you cannot rely on stats when the system is not warmed up.
    b. Try to decrease the cache size to L1:4KB and L2:32KB.
    c. Some of the benchmarks in A4 are compute intensive. If the above step does not warm up for such benchmarks, I would increase the number of iterations and set of my ROI to the latter half. In this case, it is alright if the warmup tick is not reached.