Lab 2 **Designing an Experiment with Sniper**

EE156 Advanced Computer Architecture
Prof. Mark Hempstead — Hypothesis and Experimental Plan Due Date: 2/24/23
TA: Parnian Mokri, Robert Costa and Bharat Kesari — Report Due Date: 3/3/23

1 Overview

This assignment will give you the freedom to design your own computer architecture experiment using Sniper. The assignment consists of the following parts:

- **Part a:** Propose a hypothesis and experimental plan based on the options listed below.
- **Part b:** Run your experiment and analyze the results.

Note Due dates are different for part a and part b

2 Part A: Propose a Hypothesis

In this part of the lab, your job is to propose a hypothesis and design an experiment to test that hypothesis. You must satisfy the following requirements in your proposal:

- Use at least 5 splash2 or npb benchmarks
- Identify what you are sweeping (select from the knobs listed below)
- Sample entire space (e.g., in lab 1 we swept last level cache size; sampling the entire space in this context would be simulating every possible cache size or at least a subset of sizes which are evenly spaced or otherwise accurately represent the entire sample space)
- Estimate simulation time (**note:** this may require running some preliminary simulations)

Potential Knobs:

- cache size
- cache associativity
- number of threads (including threads per core and total number of threads)
- commit width and number of reservation station entries ('rs_entries' in the config file)
  (**hint:** use rob.cfg in /YOUR_PATH_TO_SNIPER/config)
- prefetching (**hint:** use prefetcher.cfg in /YOUR_PATH_TO_SNIPER/config)
- or propose an additional knob that is of interest to you!

2.1 Example: Writing a Hypothesis and Drafting an Experimental Plan

Your hypothesis should be clear, concise, and should dictate the idea that you plan to test. You need to justify your hypothesis. The following are a few example hypotheses:

- Increasing number of threads for multi-threaded workloads will decrease total execution time because...
- Adding a prefetcher to the last level cache will increase IPC and reduce system energy consumption because...
- System power consumption is more sensitive to last level cache size than associativity because...

Your experimental plan should provide a clear method for testing your hypothesis and should meet the requirements listed above. For example, if your hypothesis is "Increasing number of threads for multi-threaded workloads will decrease total execution time," you should address:

- Pick 5 multi-threaded workloads
- Sweep the number of threads in each benchmark (1 to 2n) where n is the number of cores in your configuration. **Note:** To specify number of threads for benchmarks, do **-i large-4**, 4 represents the number of threads in this case. **-n 2** in the run_sniper command represents a double core architecture.
- Does your observation change when you change cache size? Which level of cache would have the most impact? Plot the results for IPC.
- Does cache associativity or the size of cache blocks affect your hypothesis?
- Would prefetching help in this case? Plot the results of both cases and explain them.

2.2 Submission Part 2A

Upload a report in PDF format **to Canvas** which includes a 2-sentence hypothesis and a description of your experimental plan which meets the above requirements. Your experimental plan should also include a diagram showing the architecture you will be working with. Feel free to use a different architecture than that which was used in lab 1, but please stick to one of the configurations already in /YOUR_PATH_TO_SNIPER/config.

3 Lab2: Part B: Run Your Experiments and Analyze the Results

In part B, you will run your experiment and analyze your results. Sometimes this requires running multiple experiments if you see interesting trends that you decide to look at more closely. In this part, you will write scripts to run experiments and process data. These scripts will be included in the final submission so that someone skilled in the art, who already has sniper running, could recreate your experiment.

3.1 Submission Part B

You should submit any scripts and data, along with your report in PDF format through provide using the command below. **It is highly recommended that you use LaTeX for your report.** You should write your report from the perspective that you are a computer architect working in industry who is trying to inform his or her colleagues. Your report should include: an introduction, experimental setup, results and analysis, and conclusion. Include topology, McPAT, and CPI diagrams. You should explain how varying the parameters affect your results (performance and power/energy). Then offer your analysis as to why you see the results you do. Do certain workloads perform better (in terms of both performance and power/energy) than others for certain configurations? Why do you think that is? Make sure to address the following questions wherever it is relevant in your report:

1. Describe at a high level the Splash2 workloads you chose to run.
2. Describe what it is that you are sweeping. For example, if you are sweeping prefetcher type, describe what a prefetcher is and describe the implementations of the prefetchers you are testing.
3. What performance metrics have you used to evaluate your architecture and why? What power/energy metrics have you used and why?
4. Is your experiment conclusive? If not, conduct a brief follow-up experiment which investigates why your first experiment was inconclusive.

Note: You might want to use metrics that combine both power and performance, e.g., energy-delay-product (EDP); if this is the case, you need to state why you did that and how it helps you to evaluate your topology.

3.2 Provide

1. Your lab report as a single PDF file lab2.pdf. Note that submitted files shouldn't include space in their name.

2. README file describing the filename and content of the files you have submitted. List all the files, and make sure it can be used to reproduce your experiments easily. You can simplify, buthere is a good tutorial on how to write a README.
3. Your submit script(s) used to run experiments, this can be either a shell script or python script
4. Your post-processing / data analysis scripts this could be either MATLAB, python or Excel.

- ssh to homework
- provide ee156 lab2 lab2.zip

4 Useful Notes

- Use the /data partition on the red-giant for your simulations. /data is shared between everyone, so make your own directory with your name and copy your results there. This directory is not backed up by IT, so use it as temporary storage. Copy the final results (your csvs) in your home directory that is backed up.
- README needs to explain all the files. For the scripts, specify the input, output, path, how to run, and what it does. This includes any decoding decisions you made as well. For example, if your csv doesn't have a heading, say why and how you addressed it in the README.
- We strongly recommend you not to hardcode paths in your scripts and use variables. This good practice and makes it easier for others to use your code.
- Add comments to your scripts so that by looking at them, a reviewer can grasp the algorithm. Here is an example on adding comments to your code. And also here for general guidelines.
- Adding column headers to your csvs make them more readable and is recommended
- All figures need to be referenced in the text. We will take points off for each figure that is not explained in the text.
- Figures need to be captioned; with a number that you'd refer to in the text to the number. You should also try and annotate interesting data points on the plots.
- Explain the figures in the text thoroughly. This is a bit tricky with small workloads. But try to see if there are trends or interesting data points that you can point out to.
- Format needs to be clear. Have sections in your report that sets up your story. The papers you read are great examples of writing good reports.
- Flow: For a full grade, each paragraph needs an interesting intro sentence that shows the takeaway. The paragraph needs a flow and should end with a conclusion. This is a skill that you will only acquire it by practicing. This class, is your safe space. We want to see the next generation of leaders in our field. You can use free tools like Grammarly, or use Tufts writing center. Please talk to your TAs and speak up if you have further questions about this. And don't let perfect get in the way of good. Getting a 98 or 99 is just as good as 100. But I encourage all of you to take this opportunity and improve your scientific writing.