ECE 5755 Modern Computer Systems and Architecture, Fall 2023
Assignment 0: Getting to know your tools

This warm up assignment is meant to help you get acquainted with the tools that will be used for the labs in this course. You will learn how to access the course servers, run programs, and use performance profiling tools on a simple sorting program.

Set Up

Logging into the course VM

Note: You will need to use CU VPN (https://it.cornell.edu/cuvpn) if you are not connecting from the Cornell network.

You can log into the course server as follows:

```
$ ssh <YOUR-NET-ID>@ece5755.ece.cornell.edu
```

The first time you log in it will ask for approval (enter ‘yes’). Use your Cornell password to log into the server. The first time you log in, run the following command to set your home folder private.

```
$ chmod -R og-rwx ~
```

Download **lab0.zip** from Canvas and extract the contents. Copy the contents to the server from your local machine via SCP (or any other method). *The following command is to be executed where you have lab0 on your local machine.*

```
$ scp -r lab0
<YOUR-NET-ID>@ece5755.ece.cornell.edu:/home/<YOUR-NET-ID>/.
```

Navigate to the lab0 folder on the course server

```
$ cd lab0
```

Bubble Sort

The first part of the lab involves writing a simple bubble sort program. The program template has been written for you in the file **bubble.c**

Modify the bubble sort function so it performs bubble sort on an integer array of size n and returns a pointer to a smallest-to-largest sorted array of size n.

If you need a refresher on bubble sort: https://www.geeksforgeeks.org/bubble-sort/ .

Compile with gcc and run the program to verify the output.

```
$ gcc bubble.c -o bubble
$ ./bubble 100 input_100
```

You may verify your code by printing the output array or using the included verifying function, but make sure to comment out the verifying code when profiling.

Profiling Bubble Sort

You will be using the toplev tool from pmu-tools to perform Top-down microarchitectural performance analysis. Top-down analysis breaks down the cause of program performance bottlenecks into four categories: frontend bound, bad speculation, backend bound, and retiring. You won't need to know what these mean for this lab, but as you continue on to later labs and you learn more about processor microarchitecture over the course, the measurements provided by toplev will become useful in helping you understand performance bottlenecks in your program.

The following are good references/further reading:

- Toplev manual: https://github.com/andikleen/pmu-tools/wiki/toplev-manual
- Intel guide to Top-down analysis: https://www.intel.com/content/www/us/en/docs/vtune-profiler/cookbook/2023-0/top-down-microarchitecture-analysis-method.html
- Ahmad Yassin's original ISPASS 2014 paper: http://bit.ly/tma-ispass14

Before profiling your program, run the following so that your program will run on bare-metal

```
$ srun -p ece5755 -n 1 --mem=1g -w slurm-instructional-compute-01
--pty /bin/bash
```

Profile bubble sort using the pre-installed pmu-tools

```
$ python3 /usr/local/pmu-tools/pmu-tools/toplev.py --core S0-C0 -l1
```

```
-v --no-desc taskset -c 0 ./bubble <input_size> <input_file>
```

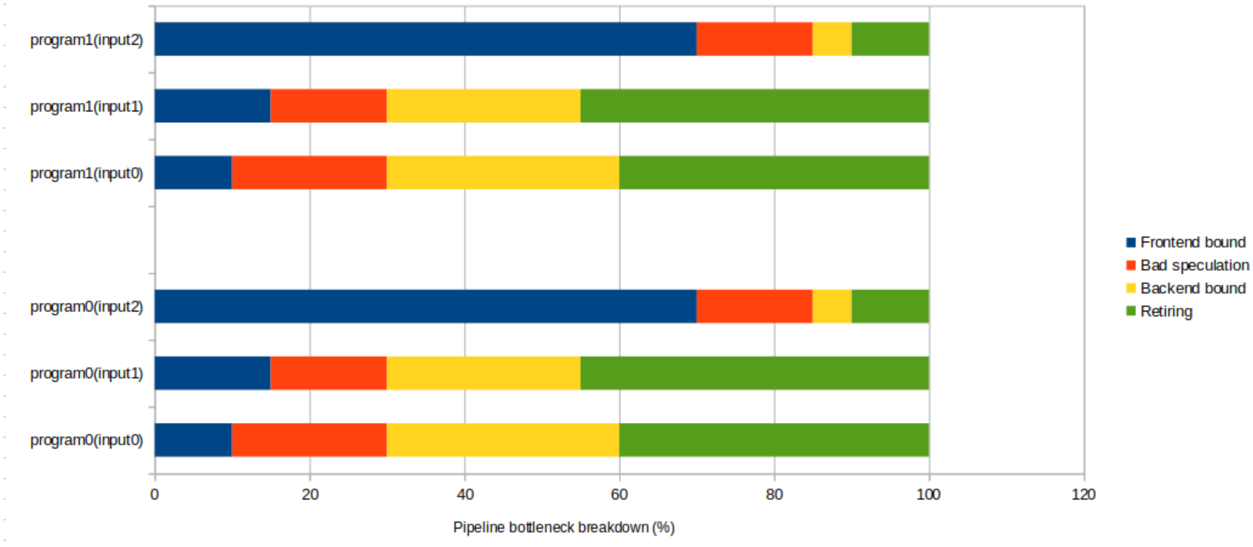
Profile and collect data for input_100, input_1000, and input_5000. You may need to run it several times for results to stabilize.

Profile a Sorting Algorithm of Your Choice

Choose another sorting algorithm besides bubble sort and implement it in **mysort.c**. Profile it using the same steps as you did for bubble sort and collect Top-down analysis data.

Deliverables

You will be submitting both your code and a **short (< 1-page) report** containing your Top-down analysis data you collected using pmu-tools. The report must include a horizontal segmented bar chart of the data you collected similar to the one below. Also include a short description of what you did for your own sorting implementation, any observations from the profiling data, and any issues you encountered during the lab. To submit your code, zip the entire lab0 folder including your modified code. **If you copied/borrowed code that you did not write yourself, you must cite the source.**



Sample bar chart

The lab will be **due on Tuesday September 5 at 11:59 pm.** You will be submitting two files on Canvas.

1. **lab0.zip**
2. **lab0.pdf**