

P56 - Lab 1, Measuring: CPU and Memory

Prof Andrew W. Moore

Lent 2023/2024

The goal of this lab is to gain experience measuring two computer sub-systems: two of the five principle contributors to (poor) systems performance. You will tackle a seemingly straightforward problem, estimating the time each will approach, refine your approach, and interpret the measurements you have made.

As part of this lab you will be following the P56 textbook, you are keenly encouraged to record your lab progress; you may wish to use a blank lab notepad, or a Jupiter notebook to record your results, annotating with figures as appropriate.

This lab will require you build and extend a measurement-focussed codebase. The code base can be retrieved from <https://www.cl.cam.ac.uk/teaching/current/P56/ucamonly/book-user-code.zip>

We will closely follow Chapter 2, 3, and 4 (*CPU*, *Memory*, and *CPU and Memory Interaction* detailing the measurement of a CPU and memory, along with cache interactions.

1 Before you begin

After you pull the P56 repository, there is a script `P51a/setup/setuplab.sh` it will create a `$CRSID` directory and another subdirectory for this Lab.

Run the script (create the directory) and `cd` to this directory.

Retrieve the zip file www.cl.cam.ac.uk/teaching/current/P56/ucamonly/book-user-code.zip and unpack it in this directory. `wget` should work just fine.

The zip file will unpack into a subdirectory `aw_files/book-user-code`, to build the code, `cd` here and run the script `compile_all_user.sh`. The build will take about 5-7 minutes.

Textbook tools create html viewable using a browser; to extract svg files (for your lab reports) you can use the tool SVG crowbar 2 from <https://nytimes.github.io/svg-crowbar/>

2 Notes for Chapter 2 - CPU

Follow the chapter closely; when working through Chapter 2 examples consider inspecting the assembler code produced by the compiler.

Additionally, use the <http://www.godbolt.org> to explore compiler options further. Attempt Exercises 2-1 to 2-8, 2-9 is entirely optional.

Once finished with Chapter 2, do not measure performance of unoptimized code – too much of the time is wasted on things that should not be there, and the results are unrelated to deployed, production, optimized code. Mostly measures memory access time!

3 Notes for Chapter 3 - Memory

Beware the code supplied is incomplete - you must fill in sections as guided by the Chapter text, most notably for 3-8

Attempt Exercises 3-1 to 3-8.

4 Notes for Chapter 4 - CPU and Memory interaction

This chapter focusses upon estimation in the face of CPU/Memory interactions building upon results using the code `matrix.cc`

Note carefully that this code

Exercise 4-1 is optional, however carefully note results, inconsistencies and steps to resolve them as you make your progress through this chapter. We will revisit this example later in the course.

5 Saving Your Experiments

Make sure to back up your experiments. Remember that multiple teams may use the same test machines, so be careful when handling data.

All the measurements are saved under your crsid folder, so backing up the entire folder is a good idea. To copy a remote directory onto your local machine:

`sftp 151@<hostname>.nf.cl.cam.ac.uk and get -r <directory>.`

There are also other ways to copy a remote directory, you are welcome to use those as well. You may wish to compress results files in order to save space.

Please do not push any changes, data or results directly to the P56 repository. You can fork the repository to your own user and push changes there. If you would like to suggest a correction or an enhancement to a notebook or a script, please use pull-requests.

6 Understanding Your Measurements

A single lab report will be required for the first few labs. Instructions for the lab report will be provided separately.

The following items are intended to help you understand your results, and may provide supporting evidence for your report. However, they are just suggestions - feel free to approach the data differently!

- Follow closely the chapters where appropriate, in this way the learning outcome becomes clear.
- This module leans heavily on the Exercises of the textbook.

Yet, you should always look for odd or surprising results, and try to explain them. Note that sometimes exceptional results may indicate a problem in your setup or scripts.