P51a - Lab 2, Measurements and Interpretation

Prof Andrew W. Moore

Lent 2022/23

The goal of this lab is to gain experience with measuring behaviour of two computer sub-systems, estimate the time activities will take, and interpreting the measurements you have made.

As part of this lab you will be following the textbook rather than working through a pre-configured Jupyter notebook although if you wish to compile your results into a convenient location; a blank booklet is made available to you in the P51 repository.

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/2223/P51/ucamonly/book-user-code.zip`

We will closely follow Chapter 5 and 6 detailing the measurement of a disk system and of one side of a network client-server.

1 Before you begin

Update your pull of the P51a repository, there is a script `P51a/setup/setuplab2.sh` it will create a $CRSID directory and another subdirectory for this Lab2.

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

Retrieve the zip file `www.cl.cam.ac.uk/teaching/2223/P51/ucamonly/book-user-code.zip` and unpack it in this directory.

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 5 - Disk

Follow the chapter closely; including the Exercises 5.1 to 5.11 as appropriate.

On each R.Pi there are two disks the root is an sd-card, while /flash/ is a USB thumb drive based upon classic flash memory.

Furthermore, on the first machine of each cluster is a faster drive, an NVMe / SSD disk. It is mounted at /nvme

This gives you several different disk technologies. No spinning disks are available on the R.Pi cluster at the time this was written.

Compile the mystery3 code with the commandline:

```
g++ -O2 -o mystery3 mystery3.cc -lrt
```

Note that this code tries to strip the extension name naively, looking for the last dot. This means that the path to use must either contains a file that has an extension name, or it should not include any dot. E.g. ../../../nvme_testfile will cause problems, but 2 /nvme/speedTestFile will not.

When Processing, for the sort make sure

```
export LC_ALL=C
```

is set (otherwise the sort will not function predictably.)

An alternative approach is in this example

```
cat testfile_read_times.json | LC_ALL=C sort | \
./makeself show_disk.html > testfile_disk_read.html
```

This puts LC_ALL=C near sort to avoid changing environment variables globally.

Programmes such `makeself` expect to be run in the `book-user-code` directory.

The html files may be viewed using the browser runing on the R.Pi

```
chromium-browser testfile_disk_read.html testfile_disk_write.html
```

You can also copy the html files locally for faster response.

Note Well
As indicated in Exercise 5.9 you will need to write some code;

5.9 Complete the missing part of TimeDiskWrite() . Mine is seven more lines, setting the block current times. This will be easy if you have followed what the strategy is, and a bit harder if you have been only skimming this text and the code. But when you are done, you will better understand what is going on.
Search in mystery3.cc for "You get to fill in this part !!"

Don't forget to recompile and if you get it wrong (or the code doesn't exist) the write graph will appear to have taken no time at all – a vertical line on the graph in *testfile*_disk_write.html

3 Notes for Chapter 6 – Networks

Like the previous, follow Chapter 6 and attempt the Exercises at the end. While no code needs to be written; like the code of Chapter 5, the executables that process data expect to be run in the `book-user-code` directory. You will need two machines for this chapter so will need to have the code (or at least the server executable) on each machine too.

A few typo's remain in the textbook so Gary Guo has provided some example commands on the web page `https://hackmd.io/@nbdd0121/ByAvcuMai` I've reproduced it here:
Compiled with ./compile_all_user.sh (alternatively Gary supplied a makefile, posted on his website).

On your_server_machine

```
./server4
```

On your_client_machine

```
./client4 your_server_machine 12345 write -key "kkkkk" -value "vvvvv" 1000000
./client4 your_server_machine 12345 quit
```

Each time you run `client4`, it will display the log filename it writes to. You need only remove .log and assign it to a var for convenience. Each command will write a new logfile; you are interested in the log from the write command, not the quit command.

For example

LOGSTEM=client4_20230209_141229_l51-pi888.nf.cl.cam.ac.uk_19102

Processing:

```
  ./dumplogfile4 "Write 1MB" $LOGSTEM.log > $LOGSTEM.json
cat $LOGSTEM.json | ./makeself show_rpc.html > $LOGSTEM.html
```

Once again you can display the html files using chromium on the R.Pi, or copy them locally.

4 Saving Your Experiments

Make sure to back up your experiments, including (but not limited to) Jupyter notebooks, dump files and scripts. 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 root@<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.
Exporting a Notebook as `.tex` will save graphs as separate files, which you can then include in your lab report.

Please do not push any changes, data or results directly to L50 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.

5 Understanding Your Measurements

A single lab report will be required for the first three 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!

- Discuss the methodology of the measurement tools.
- Explain how the limitations of are mitigated in this lab.
- Explore the limitations of the experiments conducted in this lab, and explain where the quality of the experiment (e.g., setup, methodology) could have been improved.

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.