P56 — Instructions For Final (Second) Lab Report

Prof Andrew W. Moore

Lent 2023/2024

1 Submission

The remainder of assessment for this module is constructed from two pieces of work. Firstly, the final report and secondly, a presentation in the final lab/lecture slot. Details on the presentation will be provided separately. The split will be 20%/80% as described in the original prospectus. The talk is not marked directly however, no attempt at the talk will result in penalty.
Assignment 2 Submission Deadline: 13 Mar 2024 12:00 Noon
Submission: submit through moodle, as a single **pdf** file.

This, the second lab report, is intended to summarize your experience over the last three labs only. This report does not require discussion of the work exploring your nominated application. Students are required to prepare their reports individually.

2 Structure

For each of the following sections, make clear the Lab exercises you are discussing.

I would recommend brief answers interspersed with figures for Lab 3, and Lab 4, whereas a more formal structure, e.g., goals, methods, results, and conclusion will be more suitable for Lab 5.

A maximum target of 4000 words is recommended for this report, fewer words may be sufficient.

2.1 Lab 3 - Critical software sections, Locks, and Database interactions: Network and Disk

Report on your results for each of the five Chapter 7 exercises; in each case briefly discussing any points or differences of note.

In each of the book questions, I expect you to use as many figures as needed. Discussion can be brief, limited for 7.2-7.5 to commenting briefly on any differences you observe

- the recreation of the book images, but in your own environment, is the core task. If the images are unclear, please take the time to create informative images, for example zooms and descriptive annotations as appropriate.

2.2 Lab 4, Kernel Tracing

4 Disk and trace

Focussing upon the first trace example of Chapter 25, use exercise 25.1 as a guideline. I've paraphrased the question here.

Find a unique anomaly in the HTML files for **each** the four experiments (sequential and random writing, sequential and random reading) in this chapter. Operations like sync() force operating system behavior that can also help identify, illustrate, and understand an anomaly in read or write. Alongside each figure in the report, describe the anomaly (including its location), and your thoughts and reasoning about what likely is going on. Briefly describe a likely approach to improving the performance.

Network

For each of the 5 Network experiments detailed in the handout (four from the book, and an additional one involving a single disk server and three remote clients), provide a snapshot/figure of the system under trace. Seek to identify where possible a unique anomaly present in the trace. If no anomaly is present, describe why you believe this is the case.

Be selective in what you wish to show; the figures can easily be very large so it is incumbent upon you to focus (zoom) into the relevant details.

2.3 Lab 5, Performance with Antagonists

Lab 5 specifically intends for you to compare and contrast the behaviour of a known workload when subject to an antagonist.

Furthermore, by using KUtrace together with the application's own logging, you can make clear identification of how the application is affected by the antagonistic workloads.

Discuss with the aid of figures, (e.g., from KUtrace) each of the (network) experiments 3, 4, and 5 when these experiments are subject to one or more of the *cpu hog* antagonist. Look for differing behaviour between loads with and without an antagonist and then by using KUtrace alongside the application's own measurements identify the cause of the new behaviour. Describe clearly the root cause of these differences, justifying your discussion with measurements, the study of the KUtrace data, and so-forth.

Discuss how the two mechanisms (application-based measurement and KUtrace measurements) can, together with intentional resource starvation due to antagonistic workloads, reveal specific behaviours, identify potential bottlenecks, and potential for both

misbehaviour and poorer performance. Draw conclusions on how best each of these tools may be used; both independently and together.
A few things to remember:

- Include references to all figures within the report; and attempt to force figures to be adjacent (I personally prefer figures above the first reference) to the first reference in the text (try using the argument `\begin{figure}[h]` **h** for **here**).
- Do not make assertions, instead make claims supported by evidence from your experiments or citations.
- Correct typos and grammar mistakes.
- If/where necessary, annotate your figures; yes use an editor and draw an arrow/circle/large letter A as needed. This makes a reader's life much easier as you can refer to *the big A on Figure XYZ*.

3 End

Try always to keep a sense of wonder, try stuff, and don't be hipster Flanders. Thank you all.