Advanced Operating Systems: Lab 1 – Getting started with kernel tracing

Lecturelet 1

Prof. Robert N. M. Watson

2023-2024

Lab 1 objectives

- The purpose of Lab 1 is to teach skills needed in Labs 2 + 3:
  - Log into the RPi4 cluster
  - Use DTrace for some basic OS instrumentation
  - Manage and analyse the data coming out of DTrace using Python in JupyterLab
  - Create some typical plots using matplotlib and flame graphs
  - Ensure that you know how to submit the results via Moodle
- Admin notes on assessment and deadlines
  - Submit on time so that we have time to review and return comments before Lab 2 starts
  - For Part II students, this lab is optional
  - For Part III/ACS students, the lab has low weighting
  - As discussed in lecture, you want to learn these skills before we start on Lab 2, or you'll be playing catchup

Lab 1 format

- The format is “short answer” and all students will submit PDF output from JupyterLab using Moodle
  - Generate the PDF using JupyterLab’s “File → Export Notebook As → PDF” feature
  - Be aware that this builds the notebook using LaTeX, which takes an incredibly long time – (30+ seconds) – on the RPi4, so please be patient!
  - Try out the export function early – if you have trouble, get in touch
- After experimenting in our sample notebook, create a fresh notebook containing only the requested solutions to submit your work

Documents on the module/unit websites

- We have provided several documents you will need from the module website, which apply to all labs:
  - Lab Setup Guide (Everyone)
  - DTrace Quick Start (Everyone)

- These documents are specific to Lab 1:
  - Lab 1 – Kernel Tracing – Assignment (Everyone)
  - Lab 1 – JupyterLab notebook (Everyone)

The workload: dd(1)

- Not so much of a benchmark as a standard UNIX I/O tool:

```
DD(1)            FreeBSD General Commands Manual            DD(1)

    NAME
      dd – convert and copy a file

    SYNOPSIS
      dd [operands ...]

    DESCRIPTION
      The dd utility copies the standard input to the standard output.  Input
      data is read and written in 512-byte blocks.  If input reads are short,
      input from multiple reads are aggregated to form the output block.  When
      finished, dd displays the number of complete and partial input and output
      blocks and truncated input records to the standard error output.
    …
```

The workload: Command-line arguments

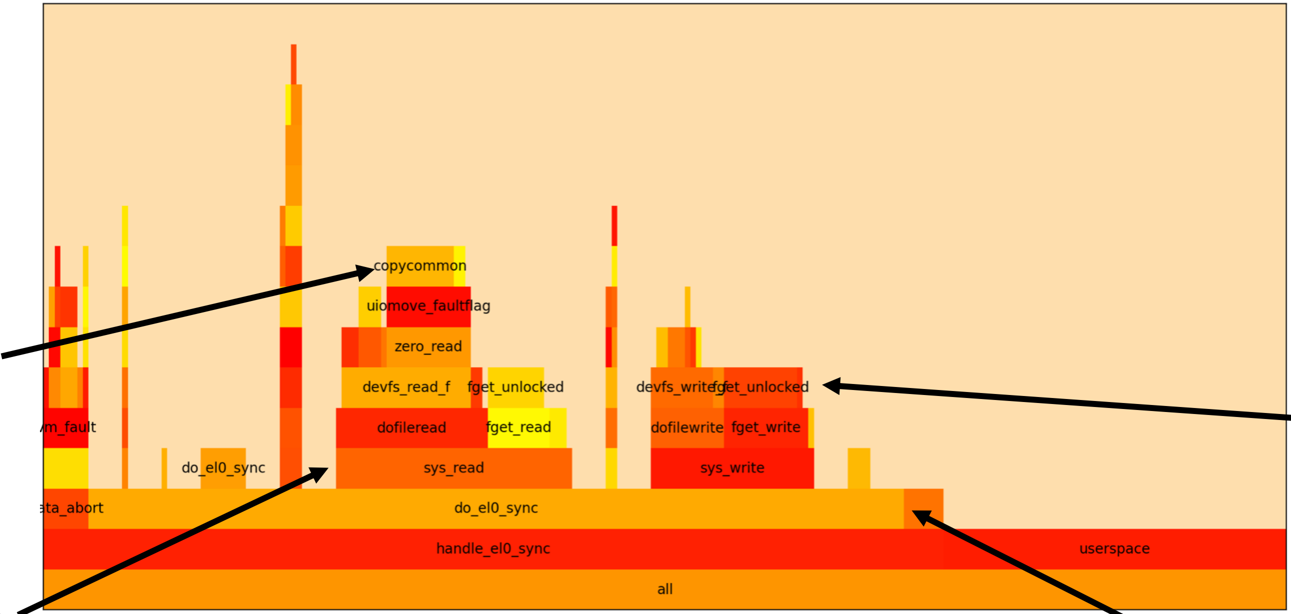
- We will use dd(1) to copy blocks from `/dev/zero` (source of all zeroes) to `/dev/null` (bottomless sink for data):

```
  dd if=/dev/zero of=/dev/null bs=8K count=5000
```

- The block size is 8,192 bytes, and we copy 5,000 blocks – i.e., a bit less than 40MiB
- This is a kernel-centric benchmark that does nothing productive, but causes the kernel to do quite a lot of work to achieve that nothing
- We will explore where the CPU cycles go!

Flame graphs

- You will already be familiar with most of the plot types we use in this course – scatter plots, histograms, etc.
- But we will also use **flame graphs**, used to render collections of stack traces captured via profiling [etc]
  - X axis represents (unordered) execution time (**or another unit**)
  - Y axis captures call chain – “top” entries are on CPU



Y axis: Call graph

X axis: Proportion of overall execution time

copycommon() is a significant user of CPU time when called (indirectly) from **sys_read**()

The highest bar, at any point on the X axis, is “on CPU”. E.g.:

do_el0_sync() is spans most execution time, but actual CPU time is mostly spent in other functions; e.g., **fget_unlocked**().

See Gregg, CACM 59(6) / ACMQ April 2016

How to get started

- Log in from the CUDN or over the Cambridge/CL VPNs
  - Acquire login credentials to your RPi4 (see email)
  - SSH into it setting up TCP port 8080 forwarding as described in the lab setup guide and prerecorded Lecture 2
- Following the Lab 1 assignment material:
  - (Optionally under tmux) launch JupyterLab
  - Open the first lab
  - Step through cell by cell, trying out and experimenting with the examples we've given
  - Create a fresh Notebook to hold only your solutions
  - When done, generate a PDF using "Export Notebook as … PDF"
  - Submit via Moodle
- Do check well before the deadline that you can reach the Moodle page

Other notes

- Come to the first in-person lab session so that we can offer help and advice as you work through the lab
- The course instructor cannot extend assignment deadlines; this must be done via the undergraduate or graduate education offices / your college tutor
    - On the whole, we are not fussed about extensions, we just can’t approve them ourselves

How to contact us

- Preferred: Course slack
  - advopsys.slack.com

- Also possible: Email to the lecturer
  - robert.watson@cl.cam.ac.uk