Advanced Operating Systems:
Lab 2 – IPC
Part II Assignment

Prof. Robert N. M. Watson

2021-2022

Your lab assignment will explore the behaviour of UNIX pipe IPC across a range of buffer sizes, and the impact of VM optimisation.

Approach

The following questions are with respect to a fixed total IPC size of 16MiB (the default for the benchmark). As with Lab 1, take measurements across a spectrum of powers-of-two buffer buffer sizes between 32 bytes and 16MiB. Use `2proc` mode, and the `-i pipe` IPC type in all experiments. To explore the impact of virtual-memory optimisation, also vary the value of `kern.ipc.pipe_mindirect` across the buffer-size space, considering how the optimisation affects each potential buffer size.

Submitting your completed assignment

Please submit your solution in the form of a single PDF interleaving written answers, plots, tabular data, and source-code excerpts, generated from your JupyterLab notebook. The easiest way to do this is via the Print menu option on JupyterLab's File menu. All submissions are via the course's Moodle page.

Experimental questions

1. Performance

First, gather data on the performance of pipe IPC with various configurations of the VM optimisation feature:

Update: 2022-03-02

- Create a plot illustrating how pipe IPC performance changes across a range of buffer sizes, using the default `kern.ipc.pipe_mindirect` threshold.
- Create a plot showing the performance of two data sets across a range of buffer sizes:
  - **One case in which we measure performance with VM optimisation enabled for every data point by setting the mindirect threshold to 32B.**
  - **A second case in which we measure performance with VM optimisation disabled at every data point by setting the mindirect threshold to 64MiB.**

  Other sensible and justified mindirect threshold values are also acceptable.

2. OS behaviour

Explore pipe performance using OS-based techniques to understand the performance curves in these plots:

- Create two plots drawing on message sent and received information gathered using `getrusage(2)` across a range of buffer sizes: one with, and the other without, VM optimisations.
- Partition and analyse the IPC performance plots; then use DTrace profiling and, as necessary, tracing (e.g., of system calls, context switches), to explore and explain OS behaviors across a range of buffer sizes.

3. Performance counters

Now extend your analysis to include hardware performance counters as well as the results of DTrace analysis:

- Explore how varying buffer sizes, as well as configuration of VM optimisation, affects the architectural and micro-architectural aspects of cache and memory behaviour.

4. Hypotheses

Using the data you have collected, explain for each of our hypotheses whether and why it is supported by your experimental results:

1. Larger pipe buffer sizes improve IPC performance.
2. Page-borrowing virtual-memory optimisations:
   (a) Are well tuned to current microarchitectures; and
   (b) Always achieve a performance improvement, when enabled at or above the default 8KiB threshold, over the unoptimised baseline.
3. *Not assigned for Part II.*