Advanced Operating Systems:
Lab 2 – IPC
Part II Assignment

Prof. Robert N. M. Watson

2023-2024

Your lab assignment will explore the behaviour of UNIX pipe and shared memory IPC across a range of buffer sizes, as well as the potential probe effect of tools you use in analysing that behaviour.

Approach

The following questions are with respect to a fixed total IPC size of 16MiB (the default for the benchmark). Take measurements across a spectrum of powers-of-two buffer buffer sizes between 128 bytes and 16MiB. Use `2proc` mode, and only the `-i pipe` and `-i shmem` IPC types in your experiments.

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. Use Jupyter's File → Export Notebook As → PDF. **This takes quite a long time on a Raspberry Pi – 30 seconds or more – please be patient while you wait for the PDF to download**. All submissions are via the course's Moodle page.

Experimental questions

1. Performance

Gather and explore data on the performance of pipe and shared memory IPC:

- Create a single plot illustrating how pipe and shared memory IPC performance changes across a range of buffer sizes.
- Compare the two IPC types: Where do the individual performance lines increase and decrease? Where are the key inflection points? Where do the data sets differ? Update: 2024-02-12

2. Architectural and microarchitectural work

Use hardware performance counters to explore how work is represented in the architecture and microarchitecture. Create plots illustrating:

- The architectural behaviour of both IPC types using the `arch` counter set.
- The microarchitectural behaviour of both IPC types using the `tlbmem` counter set.

Using performance-counter data on the total number of instructions, the number of load instructions, the number of store instructions, the number of function returns, and the number of bus accesses, analyse this data.

- Partition and analyse the two data sets in the plot (pipe IPC and shared memory IPC), describing for each of these counters how they may relate to the performance results we see. Update: 2024-02-12
- Compare counter data between the two workloads; what explanations might it help give for the comparative performance results?

3. OS-based analysis

Now extend your analysis using DTrace to understand the OS behaviours associated with our workloads. Pipe IPC experiences an unexpected drop in performance part way into its initial exponential bandwidth growth (between buffer sizes of 2^{12} and 2^{13}).

- Explore this issue by reviewing the `sys_pipe.c` kernel pipe implementation, and using DTrace's profiling support.
- Explain why this performance 'hiccup' may occur in terms of OS and microarchitectural behaviour.

4. Assess our hypotheses

Using the data you have collected, explain whether our hypothesis whether it is supported by your experimental results, and why you think that.

5. Optional task (unmarked)

Correct the pipe performance issue, and generate a plot illustrating the resulting performance change. What recommendation(s) would you make for the OS designer to improve performance?