L41 - Advanced Operating Systems:
Lab 2 - IPC

Dr Robert N. M. Watson

2020-2021

This is L41 Lab 2. If you are a Part II student, please see the other lab variant.

Your lab report will compare several configurations of the IPC benchmark, exploring (and explaining) performance differences between them. Do ensure that your experimental setup suitably quiesces other activity on the system, and also uses a suitable number of benchmark runs.

Hypotheses

In this lab, we provide you with three hypotheses that you will test and explore through benchmarking:

1. *Larger IPC buffer sizes improve IPC performance regardless of IPC model selected.*
2. *Page-borrowing virtual-memory optimisations in the pipe implementation are always effective in achieving better performance than a more conventional copy-based socket implementation.*
3. *The probe effect associated with HWPMC is negligible.*

We will test these hypotheses by measuring net throughput between two IPC endpoints in two different threads. We will use DTrace and HWPMC to establish the causes of divergence from these hypotheses, and to explore the underlying implementations leading to the observed performance behavior. As we have considered the probe effect associated with DTrace in our prior lab, we will consider only the probe effect of HWPMC in this lab.

Approach

The following questions are with respect to a fixed total IPC size (the default 16MiB). As with Lab 1, take measurements across a spectrum of powers-of-two buffer buffer sizes between 1 byte and the total size. Use `2thread` mode in all experiments. You will use both pipe IPC and local socket IPC for your experiments, configuring using the `-i` argument to one of two modes:

pipe IPC When passed the `-i pipe` argument, IPC is performed using FreeBSD's `pipe(2)` implementation, which utilises VM-based page-borrowing techniques.

local socket IPC When passed the `-i local` argument, IPC is performed using local (UNIX) domain sockets, which implement local IPC on the sockets networking API. You will use only the default automatic socket-buffer sizing variant, and not enable manual configuration via `-s`.

Experimental questions

First, begin your investigation from a OS-design perspective using DTrace:

- Does increasing IPC buffer size uniformly change performance across IPC models; if not, why not?

- How do the dynamics of the two IPC models differ from one another, and how might that explain arising performance behaviours?

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

- How does changing the IPC buffer size affect the architectural and micro-architectural aspects of cache and memory behaviour – and why?
- Can we reach causal conclusions about the relative scalability of the kernel's pipes and local socket implementations given additional evidence from processor performance counters?
- Explore the impact of the probe effect on your causal investigation; how may have HWPMC changed the behavior of the benchmark?

For the purposes of this lab, you may wish to consider using the following types of data:

- Benchmark-reported `getrusage(2)` coounters: `msgsnd` and `msgrcv`.
- DTrace-reported measurements of `read(2)` and `read(3)` returned lengths, analysed as distributions.
- Memory-focused performance counters such as architectural or speculative load, store, and instruction fetch; L1I, L1D, and L2 cache hits and misses; ITLB and DTLB misses; and memory and accesses. You may disregard other couonters, such as those relating to branch prediction and exception handling.
- Other DTrace-reported data, such as profiling results or traces of context switching.

Notes

Graphs and tables should be used to illustrate your measurement results. Ensure that, for each question, you present not only results, but also a causal explanation of those results – i.e., why the behaviour in question occurs, not just that it does. For the purposes of graphs in this assignment, use achieved bandwidth, rather than total execution time, for the Y axis, in order to allow you to more directly visualise the effects of configuration changes on efficiency. All plots in this lab should use a log X axis.