Bandwidth and Latency Measurement on Inter-Process Communications

Kedar Bastakoti, Ruohui Wang

Abstract

There are multiple ways to do inter-process communications(IPC). For example, pipe is often used to communicate between processes, whereas TCP and UDP can also be used to communicate both locally and remotely. Due to their differences in design, they tend to have different latencies and througputs. Experiments to show

1 Clock Resolution

To begin with, how can we measure the time? We need clocks to do so. And with all the ways to measure, people care about criteria that makes measurements good. One of the criteria is how precise they can be. It is worth noting that accuracy is also a key factor to consider, which we will leave to the discussion.

The Linux operating system provides multiple system calls for measurements, in different resolutions.

As per the POSIX manual[1], the function `gettimeofday` is obsolete, and the switch to `clock_gettime` is recommended, so we used `clock_gettime` instead.

Also, the x86 CPU provides `rdtscp` instruction to give the CPU timestamp, in terms of TSC frequency.

The resolution is the smallest possible increase of the clock. In order to measure the resolution, we tried to create minimal possible differences between two time measurements. We create such difference by inserting an line of assembly `inc r12` into the code. And the result shows we are producing

However, along with `clock_gettime`, function `clock_getres` is provided for user to query the resolution of time. As we will demonstrated, it produces the same result as ours.

2 Pipe

3 TCP

4 UDP

5 Evaluation

References

[1] IEEE and Open Group. The open group base specifications issue 7, 2018 edition, 2017.