

Blog

Written words that are often helpful

Performance Testing: Threads vs. Processes

Posted by chathaway on April 22, 2018, 9:30 p.m.

Recently, I spotted a line in *Distributed Systems: Principles and Paradigms* that caught my interest because it ran counter to my understanding of thread performance on a Linux system.

Instead of using processes, an application can also be constructed such that different parts are executed by separate threads. Communication between those parts is entirely dealt with by using shared data. Threadswitching can sometimes be done entirely in user space, although in other implementations, the kernel is aware of threads and schedules them. The effect can be a dramatic improvement in performance. (Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Prentice-Hall.)

The sense I got from this statement, and other such comments through the chapter, suggested to me that we should expect better performance from a multi-threaded system than the same system using multiple processes. At my day job, I work on a database which handles parallelism through IPC communication with multiple processes; anytime the topic of threading comes up, the gain gets questioned. So with these two opposing thoughts, I decided testing was order.

The program

I opted to write a simple $3n+1$ implementation (https://en.wikipedia.org/wiki/Collatz_conjecture) to test this hypothesis. In addition to the base program, I wrote two methods for spawning workers; one using `pthread_create`, and one using `fork`.

The source code can be found at: https://github.com/chuck211991/thread_testing

The results

Before I wrote a good scaffolding to generate graphs, I wanted to see if there was a drastic difference to determine what my parameters should be. Here are the results of some basic testing:

	Threads	Limit	Time (threads)	Time (procs)
10		50000000	1m39.287s	1m39.859s
100		50000000	2m55.208s	2m44.556s
500*		50000000	1m57.329s	1m57.837s
5000		50000000	2m23.804s	2m24.206s

- Prior to this run, I increased the Docker container from 2 CPU's to 5 (my physical host has 6).

It is clear that there is no difference here. Why?

Like all good programmers, off to stackoverflow I went (<https://stackoverflow.com/a/809049>). In essence, the Linux kernel doesn't differentiate between threads and processes.

Verdict: it doesn't matter which method you use.