Fox Maikovich
CSCI 3753
14 November 2016
Programming Assignment 4

Linux Task Scheduler

Abstract

In this assignment the Linux task scheduler was tested in order to determine the strengths and weaknesses of the differing scheduling policies. This is useful as it can allow us to understand how much time differing processes take, and which scheduling method is best for different situations. Overall, each of the scheduling algorithms produced similar results when it came to running time. What proved to be the difference was context switches. These context switches created a noticeable difference between each of the different scheduling algorithms.

Introduction

Different Linux schedulers will use different scheduling algorithms: the default Linux scheduler (CFS), first in first out (FIFO), and round robin (RR). Some scheduling will perform better than others under different circumstances. Each scheduling algorithm was tested across the following three representative process types: CPU bound, I/O bound, mixed. Within each of these types the number of running process per test was split into 3 categories: low (5-10 simultaneous process instances), medium (10s of simultaneous process instances), and high (100s of simultaneous process instances). The benchmark testing gathers wall time, CPU time, number of context switches, and CPU usage.

The three scheduling methods all use different steps when determining scheduling. RR rotates the CPU among the processes in the ready queue, providing each with a time slice to complete their task. This method is thus circular and continues rotating until the processes complete. Due to these characteristics RR is simple and fair but wait times can be long. CFS, the default scheduler on Linux, was created with fairness in mind. CFS uses a time ordered red-black tree instead of a queue to organize the ready processes. FIFO, or first in first out, prioritizes mainly on when the process was added to the ready queue. The process then continues to run until it voluntarily yields the process, or is preempted by a higher-priority task.

Method

For each process type a different scheduling program was used to run the benchmark tests. For the CUP bound tasks, the pi-sched.c program was used. For the IO bound tasks, the rw.c program was used. Then for our mixed tasks a combination of both was used in the mixed.c program. Pi-sched.c is a simple program for statistically calculating pi using a specific scheduling policy. Rw.c is a small i/o bound program to copy N bytes from an input file to an output file.

Each program is used to test FIFO, Round Robin, and CFS, with vary amount of concurrent processes. The Linux time command is used to gather the benchmark data and is ran through a bash script. The results of these tests were then averaged. The tests were run in the virtual machine Linux environment.

Results

CPU

SCHED_OTHER

Light

wall=0.31 user=0.58 system=0.00 CPU=186% i-switched=153 v-switched=21
wall=0.35 user=0.65 system=0.00 CPU=184% i-switched=190 v-switched=24
wall=0.31 user=0.58 system=0.00 CPU=185% i-switched=166 v-switched=24

Medium

wall=1.96 user=3.55 system=0.03 CPU=182% i-switched=1155 v-switched=97
wall=1.48 user=3.65 system=0.06 CPU=149% i-switched=2139 v-switched=107
wall=1.52 user=2.80 system=0.04 CPU=186% i-switched=933 v-switched=104

Heavy

wall=4.39 user=8.44 system=0.01 CPU=192% i-switched=2327 v-switched=301
wall=4.90 user=9.37 system=0.06 CPU=192% i-switched=2634 v-switched=302
wall=5.35 user=10.31 system=0.02 CPU=192% i-switched=2898 v-switched=303

SCHED_FIFO

Light

wall=0.32 user=0.60 system=0.00 CPU=189% i-switched=4 v-switched=16
wall=0.34 user=0.65 system=0.00 CPU=189% i-switched=7 v-switched=16
wall=0.28 user=0.53 system=0.00 CPU=188% i-switched=5 v-switched=16

Medium

wall=1.63 user=3.04 system=0.00 CPU=186% i-switched=4 v-switched=60
wall=1.47 user=2.69 system=0.00 CPU=183% i-switched=6 v-switched=56
wall=1.53 user=2.86 system=0.00 CPU=186% i-switched=4 v-switched=56

Heavy

wall=4.79 user=8.84 system=0.01 CPU=184% i-switched=16 v-switched=158
wall=4.60 user=8.40 system=0.02 CPU=182% i-switched=14 v-switched=156
wall=4.97 user=9.20 system=0.00 CPU=185% i-switched=12 v-switched=156

SCHED_RR

Light

wall=0.32 user=0.61 system=0.00 CPU=189% i-switched=7 v-switched=16
wall=0.29 user=0.54 system=0.00 CPU=189% i-switched=4 v-switched=16
wall=0.28 user=0.53 system=0.00 CPU=186% i-switched=2 v-switched=19

Medium

wall=1.47 user=2.76 system=0.00 CPU=188% i-switched=4 v-switched=56
wall=1.60 user=2.91 system=0.00 CPU=182% i-switched=6 v-switched=57
wall=1.34 user=2.51 system=0.00 CPU=187% i-switched=5 v-switched=56

Heavy

wall=4.47 user=8.19 system=0.01 CPU=183% i-switched=9 v-switched=158
wall=4.54 user=8.47 system=0.00 CPU=186% i-switched=11 v-switched=156
wall=4.75 user=8.70 system=0.00 CPU=183% i-switched=14 v-switched=156

I/O

SCHED_OTHER

Light

wall=0.39 user=0.00 system=0.23 CPU=59% i-switched=1259 v-switched=2376
wall=0.38 user=0.00 system=0.18 CPU=49% i-switched=1390 v-switched=2465
wall=0.59 user=0.00 system=0.15 CPU=26% i-switched=1272 v-switched=2523

Medium

wall=1.34 user=0.02 system=0.65 CPU=50% i-switched=7109 v-switched=10127
wall=1.38 user=0.00 system=0.59 CPU=43% i-switched=7535 v-switched=10191
wall=2.01 user=0.01 system=0.76 CPU=38% i-switched=7508 v-switched=10609

Heavy

wall=2.96 user=0.04 system=1.24 CPU=43% i-switched=22727 v-switched=28955
wall=3.01 user=0.03 system=1.80 CPU=30% i-switched=21746 v-switched=30852
wall=3.72 user=0.04 system=2.08 CPU=57% i-switched=25554 v-switched=36538

SCHED_FIFO

Light

wall=0.24 user=0.00 system=0.19 CPU=78% i-switched=1 v-switched=2878
wall=0.25 user=0.00 system=0.19 CPU=76% i-switched=0 v-switched=2693
wall=0.31 user=0.00 system=0.17 CPU=54% i-switched=0 v-switched=2757

Medium

wall=0.85 user=0.00 system=0.86 CPU=101% i-switched=0 v-switched=17287
wall=0.89 user=0.01 system=0.80 CPU=91% i-switched=0 v-switched=17354
wall=0.93 user=0.01 system=0.90 CPU=97% i-switched=1 v-switched=17051

Heavy

wall=3.12 user=0.02 system=3.02 CPU=97% i-switched=2 v-switched=52846
wall=2.98 user=0.06 system=2.84 CPU=97% i-switched=2 v-switched=54613
wall=2.94 user=0.06 system=2.78 CPU=96% i-switched=1 v-switched=53191

SCHED_RR

Light

wall=0.23 user=0.00 system=0.23 CPU=98% i-switched=1 v-switched=2698
wall=0.32 user=0.00 system=0.18 CPU=58% i-switched=0 v-switched=2917
wall=0.19 user=0.00 system=0.17 CPU=94% i-switched=0 v-switched=2736

Medium

wall=0.98 user=0.00 system=1.13 CPU=115% i-switched=0 v-switched=17292
wall=0.97 user=0.01 system=1.08 CPU=113% i-switched=0 v-switched=18963
wall=0.96 user=0.00 system=1.10 CPU=115% i-switched=2 v-switched=17539

Heavy

wall=2.97 user=0.04 system=3.22 CPU=109% i-switched=0 v-switched=51864
wall=3.17 user=0.03 system=3.55 CPU=112% i-switched=1 v-switched=51943
wall=3.48 user=0.01 system=3.48 CPU=100% i-switched=1 v-switched=50511

Mixed

SCHED_OTHER

Light

wall=1.04 user=0.72 system=0.31 CPU=98% i-switched=3056 v-switched=2398
wall=1.03 user=0.70 system=0.21 CPU=87% i-switched=4088 v-switched=2318
wall=0.87 user=0.59 system=0.20 CPU=90% i-switched=2328 v-switched=2426

Medium

wall=3.99 user=3.12 system=0.80 CPU=98% i-switched=8434 v-switched=10661
wall=3.86 user=2.92 system=0.76 CPU=95% i-switched=10974 v-switched=9673
wall=4.50 user=3.44 system=0.90 CPU=96% i-switched=14393 v-switched=9951

Heavy

wall=14.10 user=10.02 system=2.17 CPU=86% i-switched=21709 v-switched=31265
wall=13.54 user=10.44 system=2.27 CPU=93% i-switched=22933 v-switched=32157
wall=18.38 user=14.50 system=2.73 CPU=93% i-switched=29974 v-switched=31608

SCHED_FIFO

Light

wall=0.63 user=0.78 system=0.17 CPU=150% i-switched=1 v-switched=2993
wall=0.66 user=0.76 system=0.22 CPU=149% i-switched=0 v-switched=3043
wall=0.75 user=0.98 system=0.18 CPU=153% i-switched=0 v-switched=2744

Medium

wall=3.38 user=4.43 system=1.10 CPU=163% i-switched=2 v-switched=17195
wall=3.17 user=3.97 system=1.11 CPU=160% i-switched=2 v-switched=16840
wall=2.93 user=3.68 system=1.06 CPU=161% i-switched=2 v-switched=16400

Heavy

wall=9.36 user=11.38 system=3.26 CPU=156% i-switched=6 v-switched=51978
wall=9.04 user=10.57 system=3.65 CPU=157% i-switched=8 v-switched=50858
wall=8.73 user=10.43 system=3.04 CPU=154% i-switched=5 v-switched=51185

SCHED_RR

Light

wall=0.60 user=0.72 system=0.17 CPU=148% i-switched=2 v-switched=2886
wall=0.62 user=0.71 system=0.15 CPU=138% i-switched=1 v-switched=2795
wall=1.28 user=0.65 system=0.16 CPU=63% i-switched=2 v-switched=3248

Medium

wall=4.81 user=3.54 system=0.93 CPU=93% i-switched=7 v-switched=17664
wall=5.69 user=3.54 system=0.97 CPU=79% i-switched=8 v-switched=17693
wall=4.83 user=3.81 system=1.06 CPU=100% i-switched=14 v-switched=17314

Heavy

wall=9.87 user=10.62 system=2.91 CPU=137% i-switched=52 v-switched=51295
wall=9.48 user=11.84 system=3.18 CPU=158% i-switched=47 v-switched=51888
wall=9.20 user=11.10 system=3.09 CPU=154% i-switched=59 v-switched=50972

Analysis

After looking at the data, it appears that FIFO preformed the best out of the three different scheduling methods when it came to wall time across the differing process types. In most tests the FIFO scheduling method appears to have a lower wall time. This is slightly odd as I thought that CFS would produce the best results. CFS theoretically should produce the best run time as it has a low turnaround time and thus a quick response time when multi-tasking.

Other than with the CPU process type, CFS uses less CPU than the other two scheduling methods. For both the I/O and mixed process type the CFS scheduler used a much lower amount of the CPU.

The downside to CFS is how many context switches occur. The CFS data contains substantially more involuntary switches than FIFO and RR. I found this odd as I would predict that RR would contain more context switches as it constantly moves from one process to another. FIFO on the other hand should produce the lowest amount of context switch overhead as processes voluntarily give up CPU.

When it came to the heavy I/O bound process types, FIFO preformed slightly better than its competitors, but all still preformed relatively the same.

Overall, my initial thoughts about what the data would show were not proven. CFS is a great overall scheduling method which incorporates properties of both FIFO and RR. This unfortunately was not very apparent in the data as CFS preformed about the same as the other methods. This could be due to a few different factors. The benchmark tests were performed inside of the virtual machine, which sits on top of other operating system, and thus could create inaccurate and sporadic results. This could be avoided by using a machine with a native Linux operating system.

Conclusion

The results of the benchmark test did not pronounce one scheduler better than the others. All of the schedulers preformed about the same overall with a few differences.