

HW11 了解 context-switch 的 overhead

系級：資工三 學號：409410114 姓名：周 x 君

1. 解釋 sched_latency_ns

表示行程最少運行時間，防止頻繁的切換，對於交互系統（如桌面），該值可以設置得較小，這樣可以保證交互得到更快的響應，預設值為 1500000。

2. 解釋 sched_min_granularity_ns

表示一個 run queue 內所有行程運行一次的週期，預設值為 12000000。

```
shiwulo@vm:~/osdi/hw11-context-switch$ sudo cat /sys/kernel/debug/sched/min_granularity_ns
1500000
shiwulo@vm:~/osdi/hw11-context-switch$ sudo cat /sys/kernel/debug/sched/latency_ns
12000000
```

3. 實驗設計

● 實驗環境

- Intel(R) Core(TM) i5-1035G4 CPU @ 1.10GHz 1.50 GHz，二核心
- Linux version 5.15.0-56-generic (buildd@lcy02-amd64-102)
- gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0

● 實驗方式

- 預設 sched_min_granularity_ns = 10000
- 修改 sched_latency_ns & iteration
- 各五組實驗分別進行 10 次
- 執行方式 ./reportChildStat ./cpu

● 實驗數據

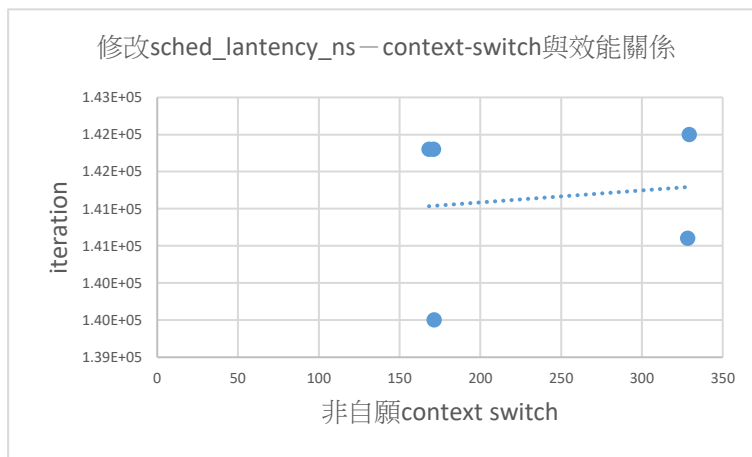
■ sched_lantency_ns

sched_latency_ns=10000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.43E+05	1.37E+05	1.40E+05	1.41E+05	1.45E+05	1.38E+05	1.44E+05	1.44E+05	1.44E+05	1.42E+05	1.42E+05
context switches(invol)	168	169	174	185	133	164	162	161	190	176	168.2
sched_latency_ns=100000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.23E+05	1.39E+05	1.40E+05	1.47E+05	1.42E+05	1.41E+05	1.41E+05	1.39E+05	1.44E+05	1.39E+05	1.40E+05
context switches(invol)	134	160	200	200	170	196	182	165	157	150	171.4
sched_latency_ns=1000000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.44E+05	1.44E+05	1.42E+05	1.40E+05	1.43E+05	1.42E+05	1.45E+05	1.38E+05	1.44E+05	1.36E+05	1.42E+05
context switches(invol)	133	145	123	236	186	117	205	173	165	227	171
sched_latency_ns=10000000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.35E+05	1.40E+05	1.33E+05	1.33E+05	1.33E+05	1.42E+05	1.43E+05	1.47E+05	1.46E+05	1.54E+05	1.41E+05
context switches(invol)	275	206	476	252	318	289	262	547	411	247	328.3
sched_latency_ns=100000000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.38E+05	1.31E+05	1.44E+05	1.45E+05	1.40E+05	1.42E+05	1.46E+05	1.50E+05	1.42E+05	1.42E+05	1.42E+05
context switches(invol)	285	385	251	438	458	429	469	125	202	251	329.3

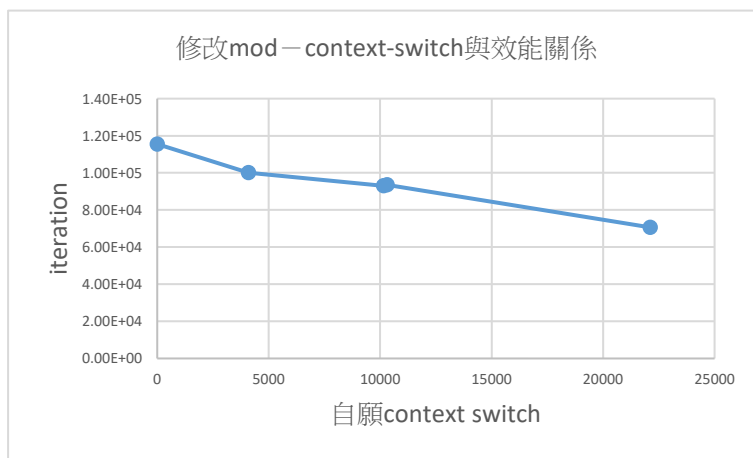
iteration

mod = 30000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	7.50E+04	7.50E+04	7.50E+04	6.55E+04	7.43E+04	6.83E+04	7.50E+04	7.00E+04	6.44E+04	6.40E+04	7.07E+04
context switches(vol)	24480	23331	20693	20373	20996	20115	33019	18318	20384	19430	22113.9
mod = 50000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	8.91E+04	9.64E+04	9.89E+04	9.80E+04	9.43E+04	8.75E+04	1.00E+05	9.70E+04	8.74E+04	8.68E+04	9.35E+04
context switches(vol)	10151	10864	10439	10137	9886	9661	12462	10242	9085	10202	10312.9
mod = 70000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	8.90E+04	9.03E+04	8.41E+04	1.11E+05	8.84E+04	9.17E+04	9.07E+04	9.41E+04	9.55E+04	9.54E+04	9.30E+04
context switches(vol)	10725	9768	9122	10876	9922	10371	10090	10284	10507	9820	10148.5
mod = 100000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05
context switches(vol)	6116	5240	6525	3951	1297	877	4303	5958	2903	3749	4091.9
mod = 2000000											
	1	2	3	4	5	6	7	8	9	10	avg
iteration	1.07E+05	1.16E+05	1.20E+05	1.18E+05	1.19E+05	1.07E+05	1.22E+05	1.15E+05	1.20E+05	1.11E+05	1.16E+05
context switches(vol)	1	1	1	1	1	1	1	1	1	1	1

實驗結果



隨著 iteration 增加，非自願 context-switch 增加，因為 sched_latency_ns 變小。



隨著 iteration 增加，自願 context-switch 減少，因為 sleep 頻率減少。

- 預估當 context-switch 的數量到達多少的時候，程式碼的執行時間會超過一個小時

```
shiwulo@vm:~/osdi/hw11-context-switch$ time ./reportChildStat ./cpu
start
6891, iteration = 1.11e+05
voluntary context switches 10876
involuntary context switches 116
soft page faults 580
hard page faults 0
stime 1
utime 10

real    0m15.290s
user    0m10.751s
sys     0m1.810s
```

隨機擷取其中一測試資料如上圖，context switch 增加 10876 的時間約為 15.3 秒
若要超過一小時，context switch 數量約需達 2559059。

參考資料：

<https://www.twblogs.net/a/5b886fff2b71775d1cdc67da>

<https://unix.stackexchange.com/questions/698603/no-sched-min-granularity-ns-in-kernel-parameters>

<https://access.redhat.com/solutions/177953>