HW11 報告

資工三 408410098 蔡×祥

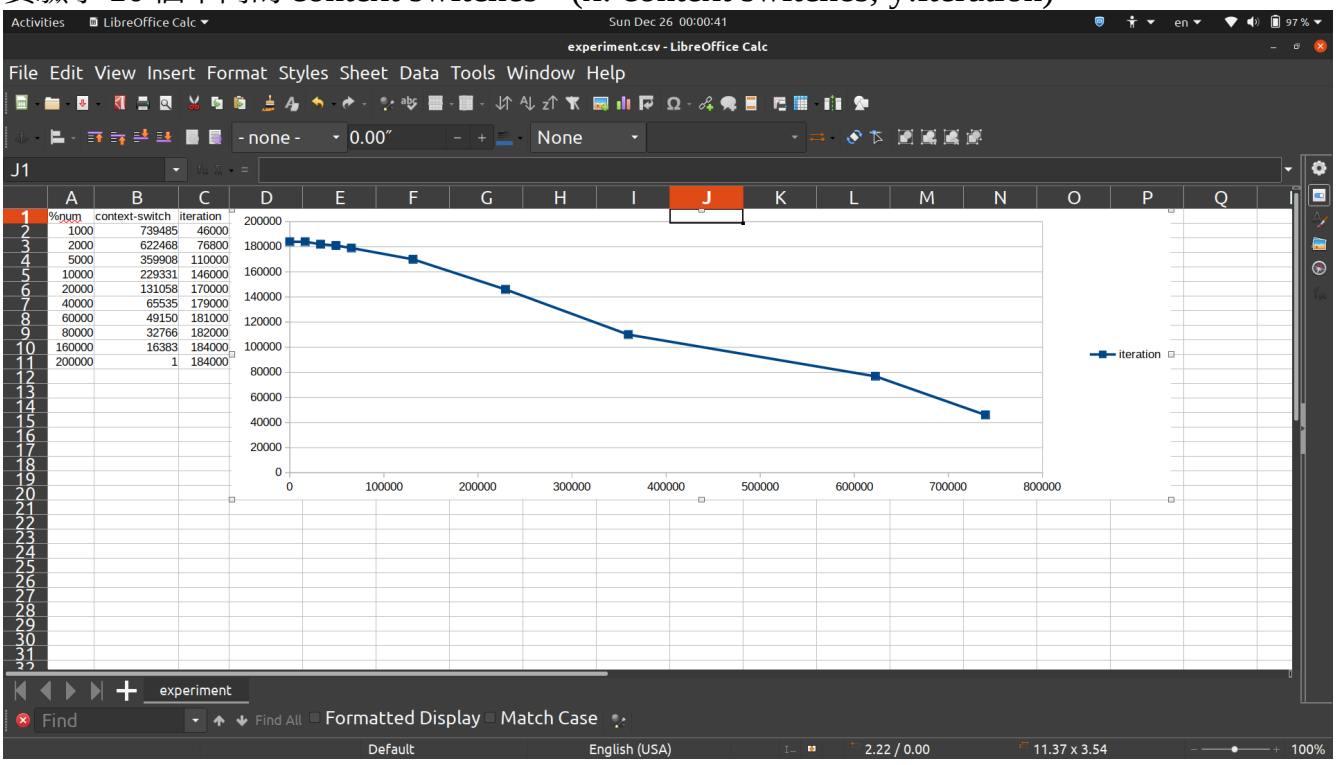
1.
sched_latency_ns:每一個 task 的出現週期。
sched_min_granularity_ns:一個 task 多久要 context switch

本次實驗，系設定:
sched_min_granularity_ns :100000

experiment 1.
固定 sched_latency_ns = 18000000，調整 sleep(0)的呼叫次數(透過 iteration % 不同的數字來達成)，即為 voluntary context switch 的改變作為操作變因

執行： ./reportChildStat ./cpu_with_sleep
(cpu_with_sleep.c 裡面的 iteration%num num 要手動改)

實驗了 10 個不同的 context switches：(x: context switches; y:iteration)



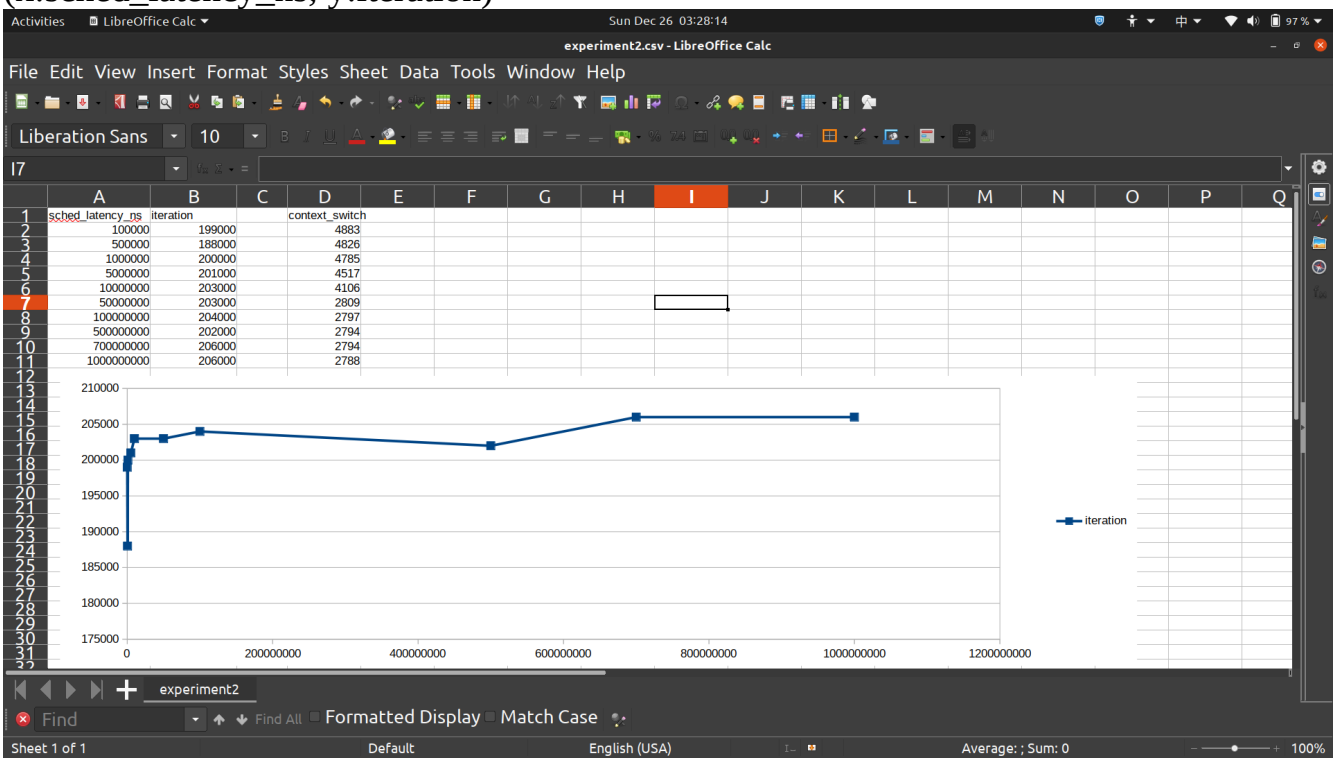
可以看到，voluntary context switches 越多次，iteration 的次數越少
（即為效能越差，因為時間相同）

其中，可以發現當 voluntary context switches 到 65535 次數以下時（-x 軸方向），效能增加逐漸緩和。

experiment 2.
固定 vountary context switches 次數，(控制為 1，讓它不要執行 sleep(0))，增加 user app（透過一次執行 4 個 cpu_no_sleep.c 程式）並改變 sched_latency_ns 來觀察 OS user mode 之間的 context switches 與效能的關係。

執行:
./cpu_no_sleep& ./cpu_no_sleep& ./cpu_no_sleep& ./reportChildStat ./cpu_no_sleep

(x:sched_latency_ns; y:iteration)



可以看到 sched_latency_ns 越大，效能就越好。而當 sched_latency_ns > 1000000 之後，效能成長就趨於飽和了。

再做實驗的時候，我發現當 cpu 功率提昇後，好像會有一段時間它都處於活躍狀態，但之後會進入一段...自動降功率? 的階段，跑出來的數據感覺跟理論有些差異。