HW11

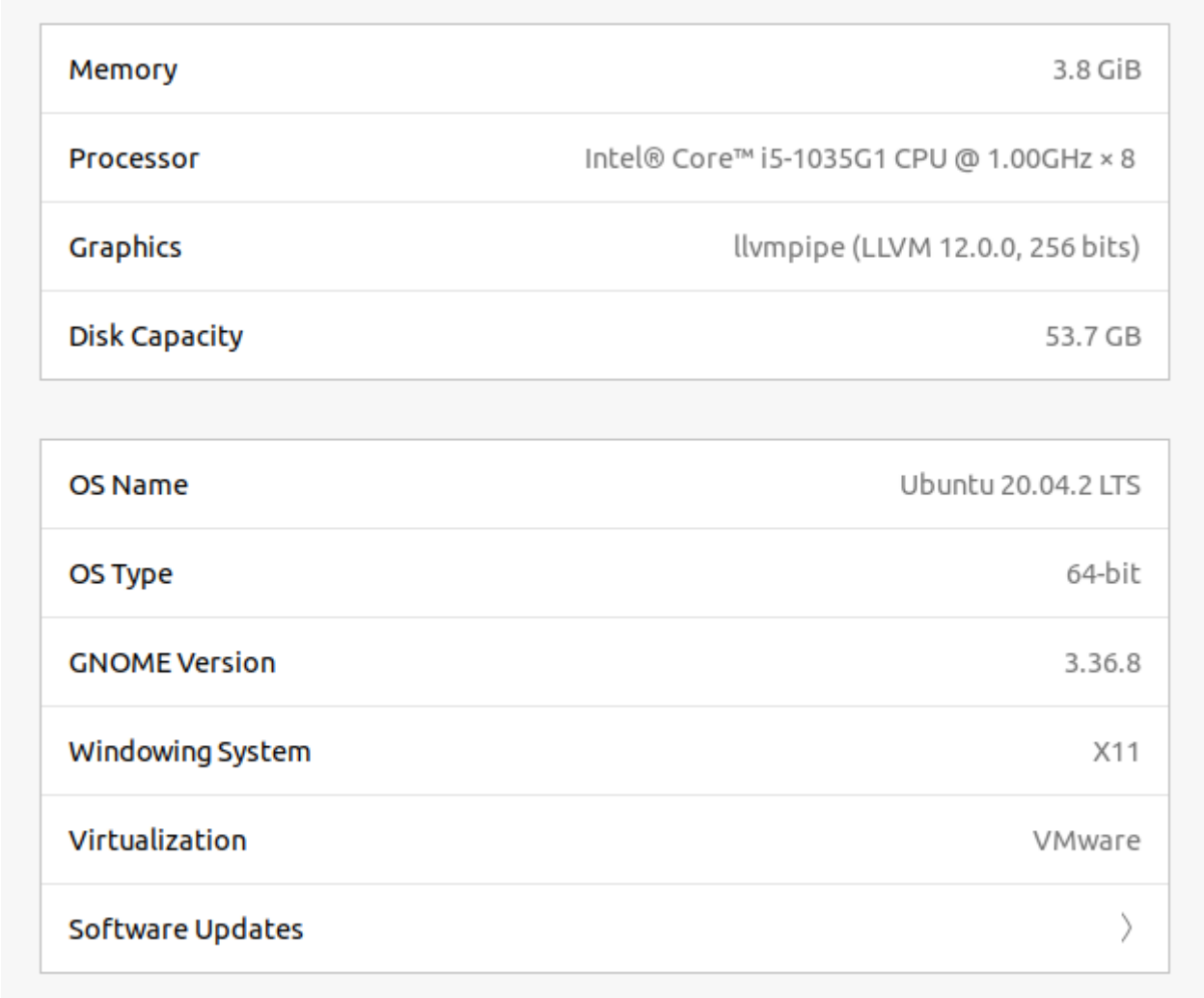
資工三 曹O萱 409410082

(1)解釋sched_latency_ns和sched_min_granularity_ns

sched_latency_ns:表示一個運行隊列所有進程運行一次的周期，當前這個與運行隊列的進程數有關。
sched_latency_ns
是調度程序週期的初始值。調度器週期是一個時間段，在此期間，所有可運行的任務都應該被允許至少運行一次。
sched_min_granularity_ns:表示進程最少運行時間，防止頻繁的切換。如果設定的太小會頻繁造成context-switch，如果設定的太大可能會造成不公平，所以要低延遲的話就設定小一點，工作量比較大的話就設定大一點。

(2)設計實驗，說明context-switch的次數與效能的關係

- 系統環境



- 預設的情況下，sched_latency_ns = 24000000，sched_min_granularity_ns = 3000000

1. 自願性context-switch

- 利用改變iteration%的值(r)來控制sleep的次數

```
long randAccess() {
    while(1) {
        iteration++;
        for (int i=0; i<cacheSize; i+=64) {
            mem[i] += 1;
            if (iteration % r == 0)
                sleep(0);
        }
    }
}
```

- 設定sched_latency_ns = 10000
- 執行方式

```
time ./reportChildStat ./cpu
```

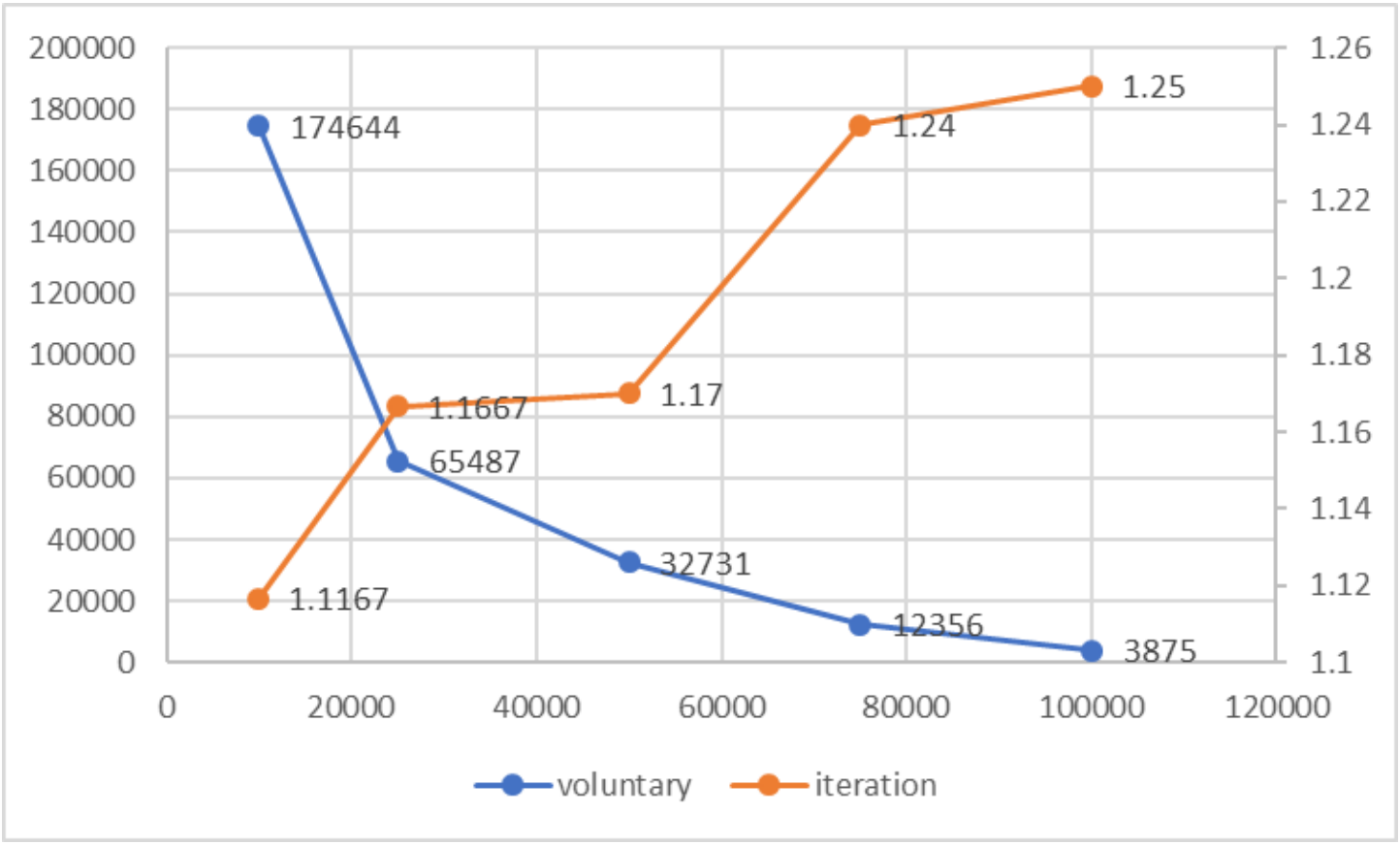
```
shiwulo@vm:~/HW/hw11$ time ./reportChildStat ./cpu
start
28075, iteration = 5.13e+04
voluntary context switches 81739
involuntary context switches 110
soft page faults 578
hard page faults 0
stime 0
utime 13

real    0m49.192s
user    0m13.166s
sys     0m0.064s
```

- 實驗數據(10次測試後取平均)

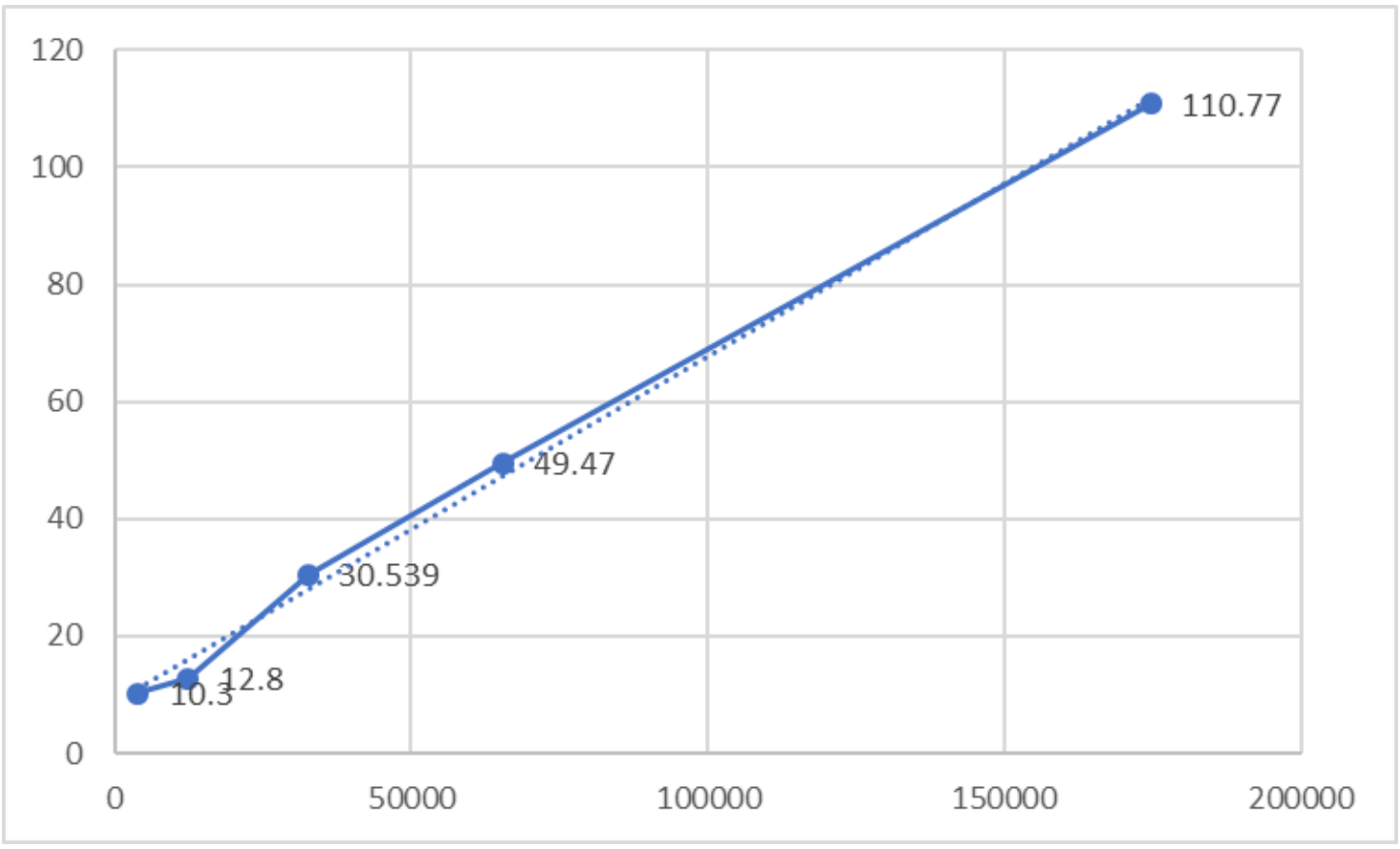
r	10000	25000	50000	75000	100000
voluntary	174644	65487	32731	12356	3875
iteration(單位:e+05)	1.1167	1.1667	1.17	1.24	1.25
time(單位:s)	110.77	49.47	30.539	12.79	10.277

context-switch次數與效能的關係



r越大，sleep的次數越少，自願性context-switch越少，效能(iteration)越好，但r大到一定程度之後context-switch次數就逐漸飽和，後面就看不出太大的變化了

context-switch次數與時間的關係



context-switch次數越多時間越長，大致上呈線性成長，趨勢線的方程式為 y = 0.0006x + 8.598 ，將y = 3600(一小時)帶入後，可以得到要執行一小時需要5,985,670次context-switch

2. 非自願性context-switch兩組

- 設定sched_latency_ns

```
$ sudo -s
# echo > 10000 /sys/kernel/debug/sched/latency_ns
```

- 執行方式

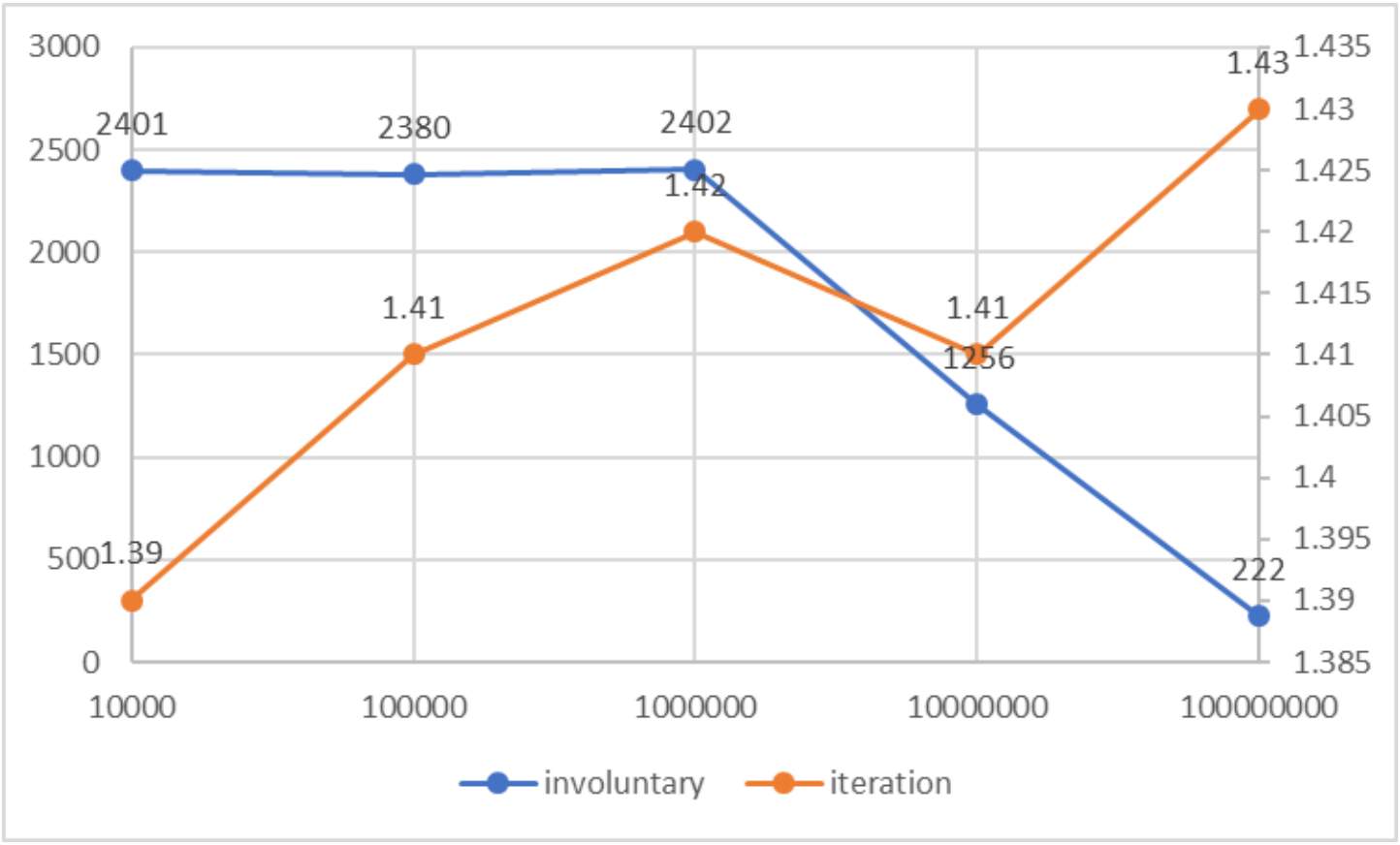
```
./cpu&  time ./reportChildStat ./cpu
```

```
shiwulo@vm:~/HW/hw11$ ./cpu& ./reportChildStat ./cpu
[1] 28584
start
start
28584, iteration = 1.39e+05
28586, iteration = 1.39e+05
voluntary context switches 1
involuntary context switches 2401
soft page faults 580
hard page faults 0
stime 0
utime 10
[1]+  Done                    ./cpu
shiwulo@vm:~/HW/hw11$
```

- 實驗數據

sched_latency_ns	**10000**	**100000**	**1000000**	**100000000**	**100000000**
involuntary	2401	2380	2402	1256	222
iteration(單位:e+05)	1.39	1.41	1.42	1.41	1.43

(iteration單位:e+05)



latency數值越大，非自願性context-switch的數量便會減少，效能上也會些微上升但不明顯