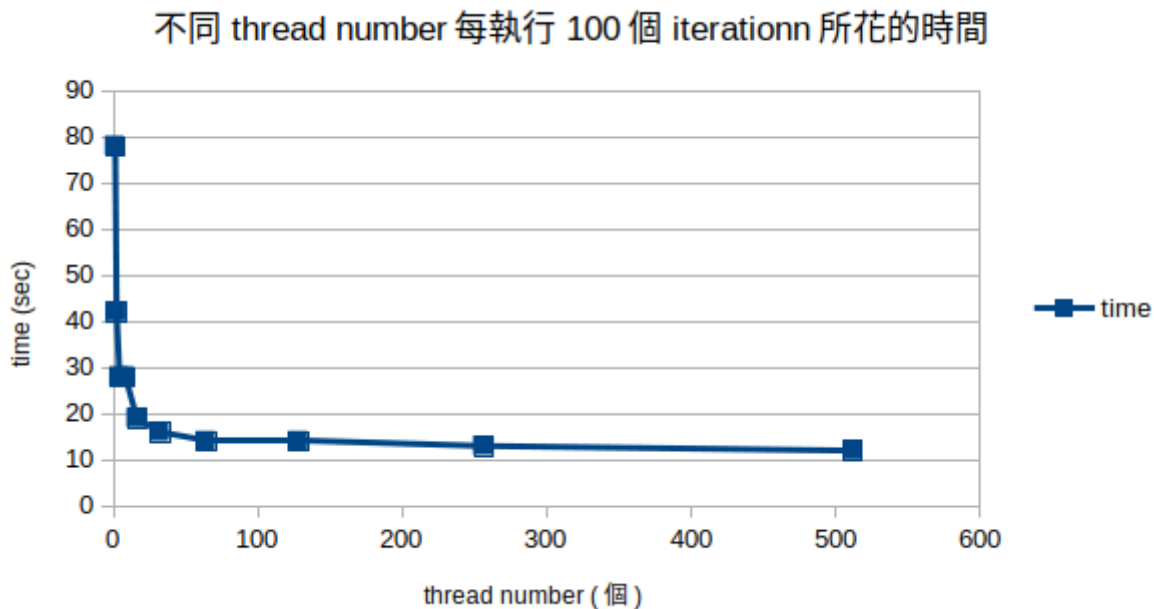


SP HW4 report

b07902048 資工二 李宥霆

3

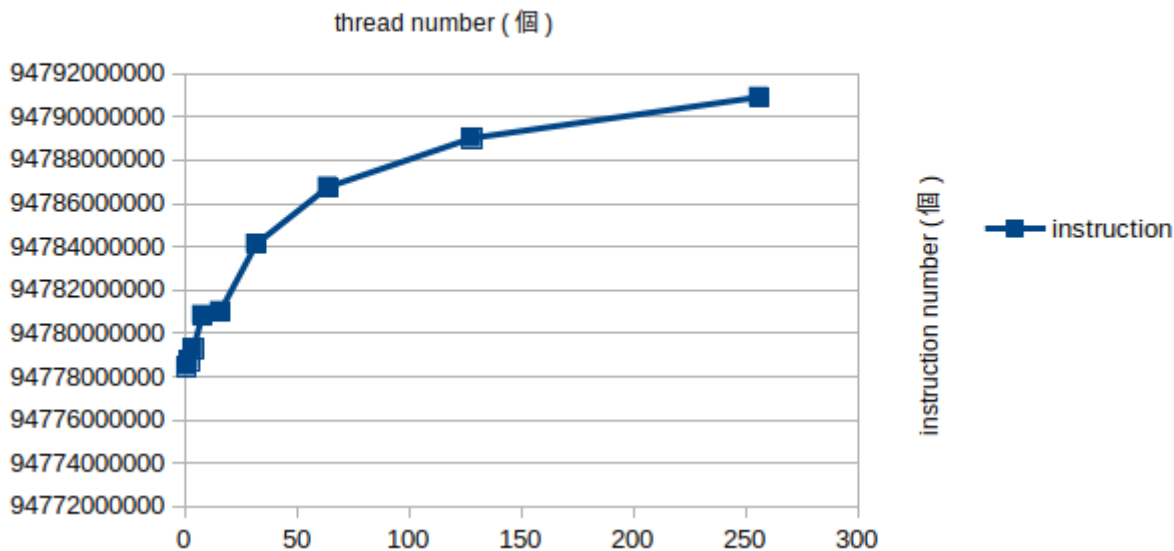


測試環境：linux10 工作站 (24 cpu core)

當thread number上升到約40左右時，執行的時間表現不再顯著地上升，這是因為可使用的CPU只有24個，當thread大於一定數目時，CPU隨時都會保持滿載的情況，此時增加thread並沒有辦法使執行時間變得更快，反之，當生成的thread數太多，因為生成越多的thread所需要的cost也越高，反而有可能會造成執行時間回升

4

不同 thread number 所花的 instruction number



測試環境：linux10 工作站 (24 cpu core)

因為CPU會對所有thread進行流程控制，決定當前要跑那一個thread，因此使用越多的thread所需要manage的thread數越多，instruction number也越高

5 Bonus

another method: 使用column parallel的方式

對於cache來說，使用cache-friendly的方式來進行矩陣運算會使得performance變得更好，與row parallel相比，column parallel的performance較差，因為對於cache來說矩陣的資料是一個row一個row讀取，當我們使用column parallel的方式時，cache跑每個迴圈都需要再load下一個row的資料，造成performance變差，以下是兩者在不同thread下執行40個iteration的performance比較

number	method	thread number	performance
1	row parallel	1	13.922s
2	row parallel	100	7.200s
3	column parallel	1	26.333s
4	column parallel	100	17.507s

執行時間比較結果: $2 < 1 < 4 < 3$

Bonus file name :

another_way.c

Compile way:

make (與hw4.c在同一個makefile裡) 或者

```
gcc another_way.c -lm -pthread -O3 -o another_way
```

execute:

```
./another_way [X_train] [y_train] [X_test] [number of threads]
```