B03902052　顏廷宇　資工二

SP_HW4 Report

	100	10000	1000000	10000000
n/1	R: 0.07sec U:0.000sec S:0.000sec	R: 0.67sec U:0.008sec S:0.004sec	R: 1.54sec U:0.706sec S:0.016sec	R: 8.90sec U:5.704sec S:0.116sec
n/2	R: 0.06sec U:0.000sec S:0.000sec	R: 0.39sec U:0.000sec S:0.008sec	R: 2.90sec U:0.752sec S:0.024sec	R: 10.70sec U:7.244sec S:0.212sec
n/5	R: 0.06sec U:0.000sec S:0.000sec	R: 0.18sec U:0.016sec S:0.004sec	R: 1.51sec U:1.120sec S:0.064sec	R: 13.21sec U:10.076sec S:0.396sec
n/10	R: 0.07sec U:0.000sec S:0.000sec	R: 0.34sec U:0.016sec S:0.000sec	R: 2.10sec U:1.204sec S:0.044sec	R: 15.19sec U:11.376sec S:0.480sec
n/25	R: 0.06sec U:0.000sec S:0.004sec	R: 0.12sec U:0.008sec S:0.008sec	R: 2.00sec U:1.560sec S:0.080sec	R: 19.86sec U:13.488sec S:0.544sec
n/100	R: 0.06sec U:0.000sec S:0.012sec	R: 0.21sec U:0.044sec S:0.000sec	R: 2.79sec U:0.112sec S:1.904sec	R: 20.03sec U:17.140sec S:0.620sec

1. Real time 分析 ： 照理來說，real time 應該會因 segment size 從 n/1 到 n/100 而成正比變多，但實驗結果顯示，時間是變多了，但不成正比，只有成長一點點，這就是因為用 multithread，平行化讓整體運算時間變少了。

2. User time 分析 ：user time 是 amount of CPU time spent in user-mode. 所以當 test data 小的時候，user time 自然就很小。但當測資大，且 segment size 越小時(ex : n/100)，user time + sys time 就越趨近於 real time，甚至，在 (10000000, n/100)時，user time + sys time > real time，而這就是 multithread 的好處，讓程式平行化加快，但也會讓 CPU 的 loading 變重。

*這是我做的另一份實驗，我把 segment size 設很小

	100	10000	1000000	10000000
100	R: 0.05sec U:0.000sec S:0.000sec	R: 0.21sec U:0.036sec S:0.008sec	R: 6.53sec U:3.992sec S:1.380sec	R: 71.34sec U:48.876sec S:14.856sec
25	R: 0.13sec U:0.000sec S:0.000sec	R: 0.31sec U:0.048sec S:0.068sec	R: 10.21sec U:5.276sec S:4.752sec	R: 121.55sec U:64.492sec S:48.196sec
10	R: 0.29sec U:0.000sec S:0.004sec	R: 0.43sec U:0.064sec S:0.132sec	R: 19.01sec U:7.728sec S:11.488sec	R: 236.83sec U:91.892sec S:125.728sec
5	R: 0.11sec U:0.000sec S:0.004sec	R: 0.50sec U:0.060sec S:0.280sec	R: 29.91sec U:10.832sec S:21.704sec	R: 347.94sec U:116.036sec S:227.932sec
2	R: 0.11sec U:0.000sec S:0.008sec	R: 0.76sec U:0.152sec S:0.588sec	R: 57.49sec U:18.944sec S:53.684sec	R: 403.01sec U:165.220sec S:410.680sec
1	R: 0.12sec U:0.004sec S:0.020sec	R: 1.75sec U:0.276sec S:1.208sec	R: 122.76sec U:33.492sec S:107.896sec	R: 972.78sec U:306.180sec S:971.448sec

1. Real time 分析 ： 若固定 segment size，看 segment size = 1 的時候，real time 成長的比例跟 test data 成正比；但在 segment size = 100 時，real time 成長的幅度卻不大明顯。這是因為 segment size 小的時候，要進行運算的 segment 數就會變多，所以，用 multithread 就可以讓運算速度有線性的成長；但 segment size 數大，test data 數也很小時，所需要的 segment 數也跟著變少，multithread 就浪費了他平行化的功能了，沒有成功加快運算速度。

2. User time 分析 ： 在 segment size = 100 時，user time + sys time < real time，這代表 multithread 沒有很大的幫助，這是因為當 segment size 大時，跑的 segment 數就少，讓 multithread 無用武之地。而當 segment size = 1 時，user time + sys time > real time，這就是因為 multithread，讓程式在 CPU 跑的時間會比真實時間多，multithread 這時對程式的加速就幫助很大。

3. Sys time 分析 ： 我的 sys time 會這麼久的原因是因為我的程式裡面用了大量的 malloc，才會導致 sys time 上升。