多處理機平行設計

Homework 5 Report

姓名：陳聲發  
學號：F74057049

**Homework 5.1**

檔案：

hw5\_1.c

編譯：

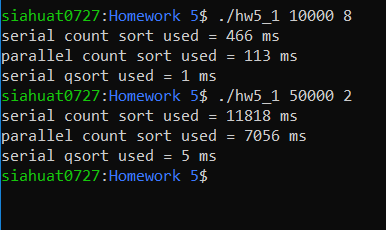
make 或 gcc -g -Wall -fopenmp -o hw5\_1 hw5\_1.c

運行：

./hw5\_1 NUM\_TO\_SORT NUM\_OF\_THREAD

e.g. ./hw5\_1 10000 8

運行結果：



回答：

1. Private: i, j, count Shared: a, n, temp
2. No. 因為每跑一圈只會改變shared memory中temp的一個元素，且這元素由始至終也只會被賦值一次
3. memcpy直接平行化沒意義，但可以做一些小改動來達到更好的效果，類似改成memcpy(a+local\_i, temp+local\_i, local\_n \* sizeof(int))
4. see hw5\_1.c
5. Performance: serial qsort > parallel Count\_sort > serial Count\_sort

(see figure above)

**Homework 5.2**

檔案：

hw5\_2.c

編譯：

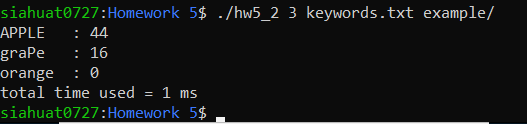
make 或 gcc -g -Wall -fopenmp -o hw5\_2 hw5\_2.c

運行：

./hw5\_2 NUM\_OF\_THREAD PATH\_TO\_KEYWORD\_FILE PATH\_TO\_DIRS

e.g. ./hw5\_2 8 keywords.txt example/

運行結果：



回答：

這次作業要注意的事項比較多，有盡可能減少critical section的大小。比如critical section只有設在enqueue 和 dequeue的最小範圍 ，還有readdir內部應該有可能發生race condition， 因此也設了critical，但這個與queue操作的是互不影響的，因此有設成不同的critical section。而對於shared variable計數累積，用了兩個atomic，一個是為了確保每個producer都讀完檔的producer\_done++，另一個是確定和keyword match之後的KEYWORD[i].count++。

另外，對於threads個數n, producer個數會是floor(n/2) ，consumer個數會是ceil(n/2)。若keyword重複，這程式碼的行為會是重複計算。（也就是顯示Apple 和 apple 的個數均為k）