

Parallel Programming hw3 ----APSP(CPU)

學號：109062639 姓名：葉哲欣

Implementation:

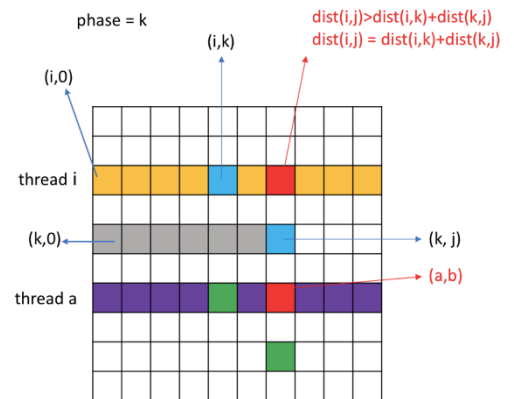
(a) Which algorithm do you choose?

Algorithm : Foly-Warshall

(b) How to implement?

Thread framework: OpenMP

採用老師上課所教的類似 cuda 平行處理方式，執行每個 phase(k)時，由於每個階段是循序被執行的，並不會有 data dependency，故以平行的方式每個 thread 處理一行下去比較。直到所有的 row 被 thread 執行完成，換下一個 phase。如右圖所示，k 固定情況下，thread i 會跟每一個(i,j)之 dist 值做比較，當 $\text{dist}(i,j) > \text{dist}(i,k) + \text{dist}(k,j)$ 則交換，跑完所有 phase 即可完成。



(c) Time complexity

原本 Foly-warshall algorithm time complexity: $O(|V|^3)$

初值設定：

Edge 讀取： $O(3 * |E| / p)$

無邊權重設定： $O(|E|^2 / p)$

Phase: $O(|V| - 1) = O(|V|)$

OpenMp :每個 thread 執行一行，假設有 p 個 threads,

每個 thread 共要 n/p 次,故 Time complexity: $O(|V|/p * |V|) = O(|V|^2/p)$

Time Complexity : $O(3 * |E| / p) + O(|E|^2 / p) + O(|V|) * O(|V|^3/p) = O(|V|^3/p)$

(d) Design and generate testcase

採取 complete graph 方式，以 random 方式生成：

$|V| = 6000$, $|E| = |V| * |V - 1|$ E 之 range : (500,1000)

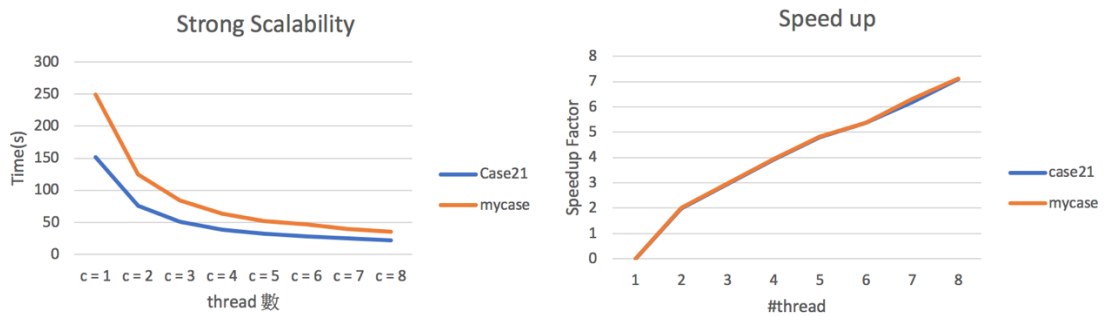
Experiment & Analysis

(a.) System Spec:

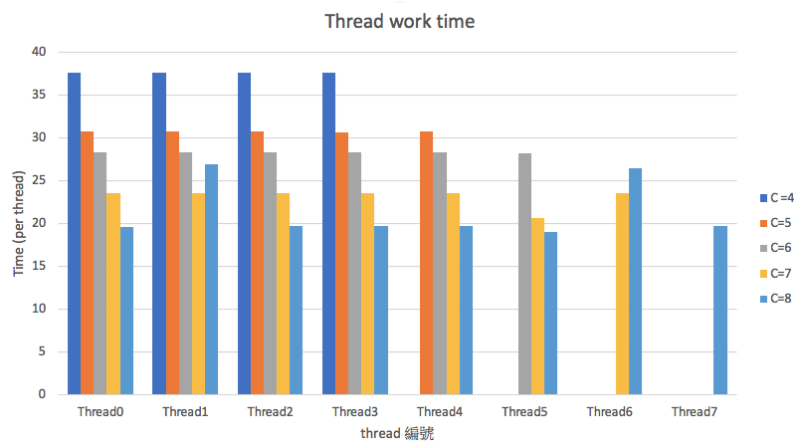
使用 apollo system

(b.) Strong Scalabilitys

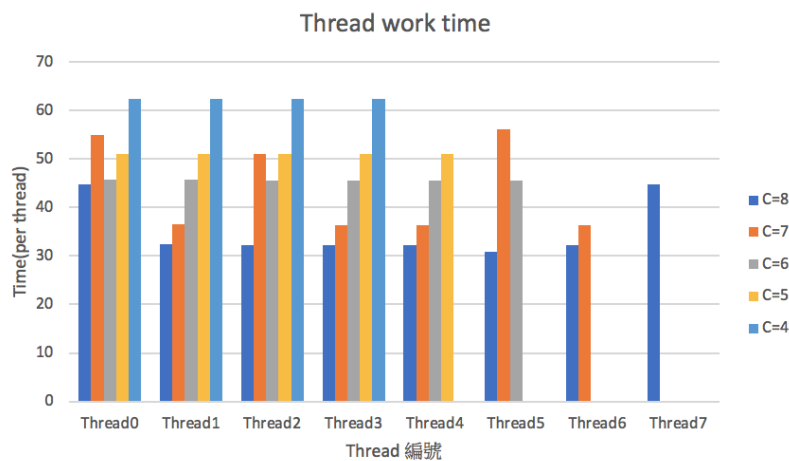
(1.) case: (case21,mycase)



(2.) Thread work time (case:21)



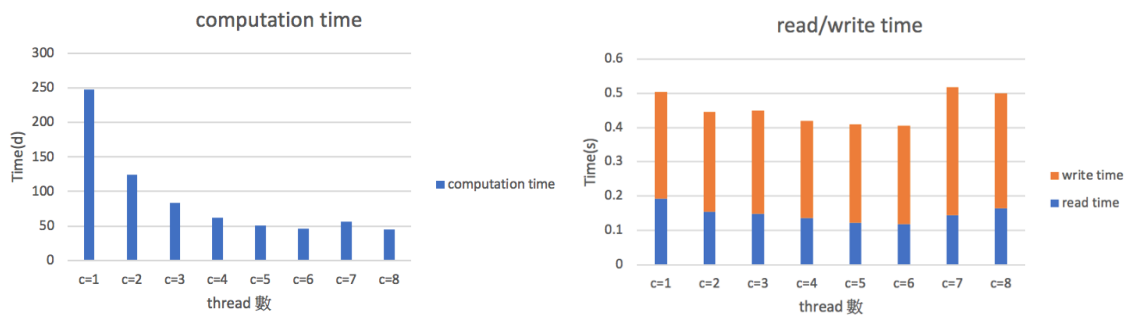
Thread work time (mycase)



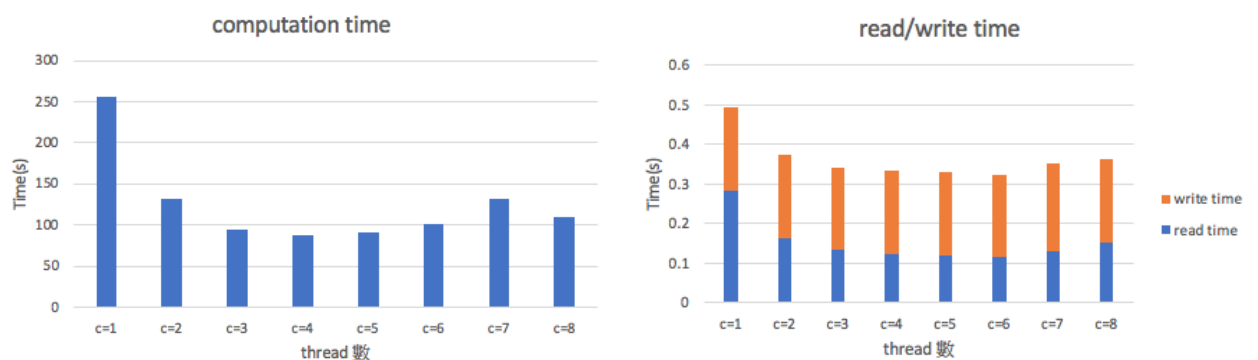
從上圖(2.)我們可以發現：此方法並不能有效地平均分配工作，thread 執行時

間不一，造成有些 thread 執行時間過長，而其他的 thread 必須等待。

(3.)case:mycase



(4.) method: collapse



Experiment & Conclusion

a.

本次作業我學習到 Folyd-warshall 如何實現平行運算，採取 openMp 實作方式，初期以為 collapse 版本效能會比一般 parallel for 效能好;但實際實驗並非如此。Collapse 將迴圈攤平給不同的 thread 執行，反而增加了 thread 溝通上與傳遞資料的問題，造成效能不如預期。

另外此次作業也學習到如果想要多個 phase 同時進行，必須解決 data dependency 問題，也是 hw4 會時做到的如何將 distance matrix 拆成多個 Block 運算。