Parallel Programming Exercise 8-10

Author:	李子筠(b06901145@ntu.edu.tw)
Student ID	b06901145
Department	Electrical Engineering

(If you and your team member contribute equally, you can use (co-first author), after each name.)

1 Problem and Proposed Approach

(Brief your problem, and give your idea or concept of how you design your program.)

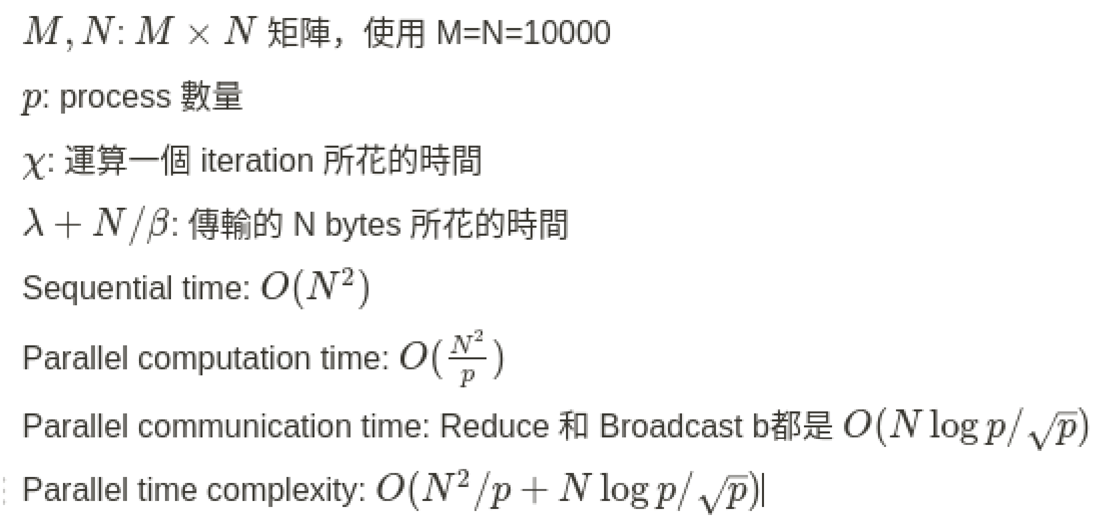
題目要求實做 checkerboard block decomposition 版本的 matrix vector multiplication。

我修改課本的方法，直接用第一個 row 的最後一個 process 將 b 分散在 第一個 row，再向整個 column broadcast。讀 matrix 的方法是用最後一個 process 向其他 process 用 MPI_Send/MPI_Recv 來傳它需要的部份。

最後每個 process 獨自算部份乘法，用 MPI_Reduce 加總每個 col 的結果，就得到最後的乘積。

2 Theoretical Analysis Model

(Try to give the time complexity of the algorithm, and analyze your program with iso-efficiency metrics)



M, N: $M \times N$ 矩陣，使用 M=N=10000

p: process 數量

χ: 運算一個 iteration 所花的時間

$\lambda + N/\beta$: 傳輸的 N bytes 所花的時間

Sequential time: $O(N^2)$

Parallel computation time: $O(\frac{N^2}{p})$

Parallel communication time: Reduce 和 Broadcast b都是 $O(N \log p/\sqrt{p})$

Parallel time complexity: $O(N^2/p + N \log p/\sqrt{p})$

Communication complexity: $\Theta(n \log p/\sqrt{p})$

$$\text{Isoefficiency metric: } n^2 \geq Cn \log p/\sqrt{p} \implies n \geq C \log p/\sqrt{p}$$

$$M(n) = n^2$$

$$\text{Scalability function: } \frac{M(C\sqrt{p}\log p)}{p} = \frac{C^2 p \log^2 p}{p} = C^2 \log^2 p$$

3 Performance Benchmark

(Give your idea or concept of how you design your program.)

Table 1. The execution time

Processors	1	2	3	4	5	6	7	8
Real execution time	0.1397	0.0757	0.0557	0.0459	0.0393	0.0359	0.0328	0.0301
Estimate execution time	0.1397	0.0699	0.0466	0.0349	0.0280	0.0233	0.0200	0.0175
Speedup	1	1.8449	2.5070	3.0440	3.5556	3.8890	4.2654	4.6416
Karp-flatt metrics	x	0.0841	0.0983	0.1047	0.1016	0.1086	0.1069	0.1034

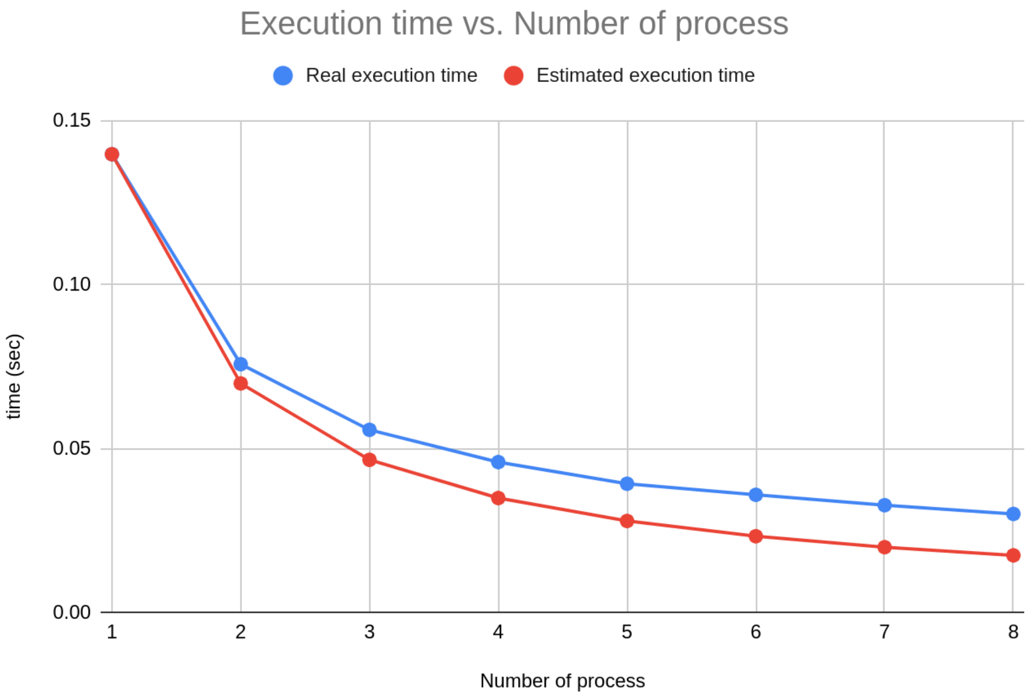


Figure 1. The performance diagram

4 Conclusion and Discussion

(Discuss the following issues of your program

1. What is the speedup respect to the number of processors used?
2. How can you improve your program further more
3. How does the communication and cache affect the performance of your program?
4. How does the Karp-Flatt metrics and Iso-efficiency metrics reveal?

)

1. speedup 隨著 processor 數量上升
2. 可能可以改用 vector instruction 算內積
3. 從複雜度可以看出來，communication 不是主要影響的部份，computation 影響較大。
4. Karp-Flatt metrics 大致上維持定值，代表 Sequential 計算佔了很大一部份。Iso-efficiency metrics 顯示這個程式有很好的 Scalability。

Appendix(optional):

(If something else you want to append in this file, like picture of life game)