Parallel Programming Exercise 4 – 12

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.)

$$\pi = \int_0^1 \frac{4}{1+x^2}\,dx$$
$$\approx \frac{1}{3n}\left[f(x_0) - f(x_n) + \sum_{i=1}^{n/2}(4f(x_{2i-1}) + 2f(x_{2i}))\right],\ x_i = \frac{i}{n}$$

用 simpson's method 估算定積分，得到 pi 的近似值

主要可以平行做的部份是後面的 sum，將 n 個 term 平分給 p 個 process 最後用 `MPI_Reduce` 加總。

2 Theoretical Analysis Model

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

p: processor 數量

χ: 每次兩個數字相加所需時間

λ: processor 間傳訊息所需時間

$$\chi\left\lceil\frac{n}{p}\right\rceil + \lceil\log p\rceil\,(\lambda+\chi)$$

需要在 process 內部算出 n/p 個項的和，以及 reduce p 個 process 所需的時間。

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	1.86E-4	1.06E-4	7.87E-5	7.58E-5	6.26E-5	6.17E-5	5.54E-5	5.92E-5
Estimate execution time	1.71E-4	8.72E-5	5.97E-5	4.61E-5	3.81E-5	3.29E-5	2.92E-5	2.64E-5
Speedup	x	1.75	2.37	2.46	2.98	3.02	3.36	3.15
Karp-flatt metrics	x	0.14	0.13	0.21	0.17	0.20	0.18	0.22

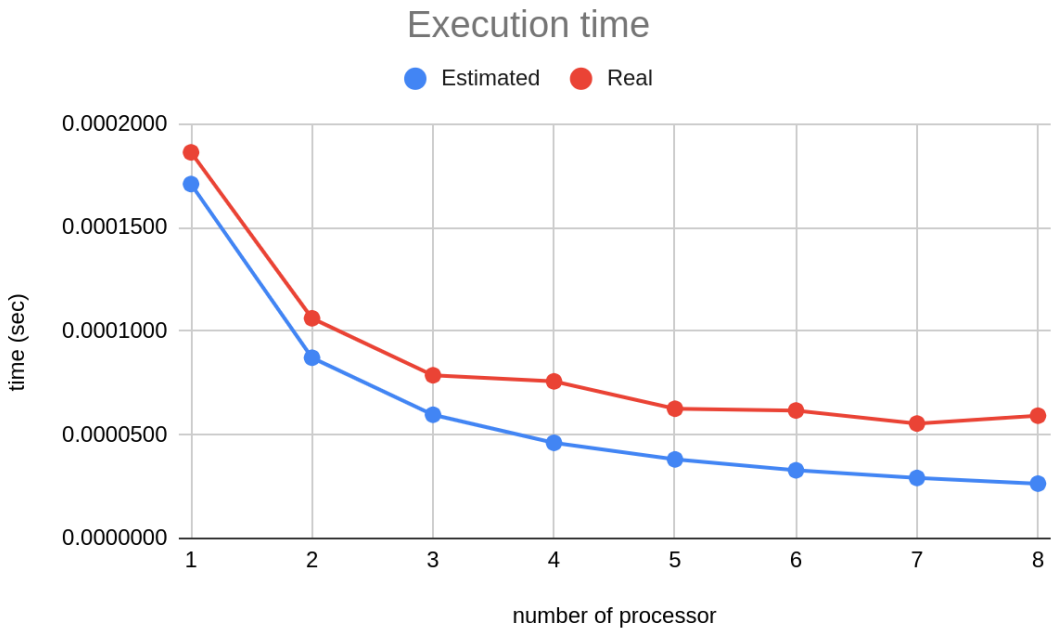

Figure 1. The performance diagram

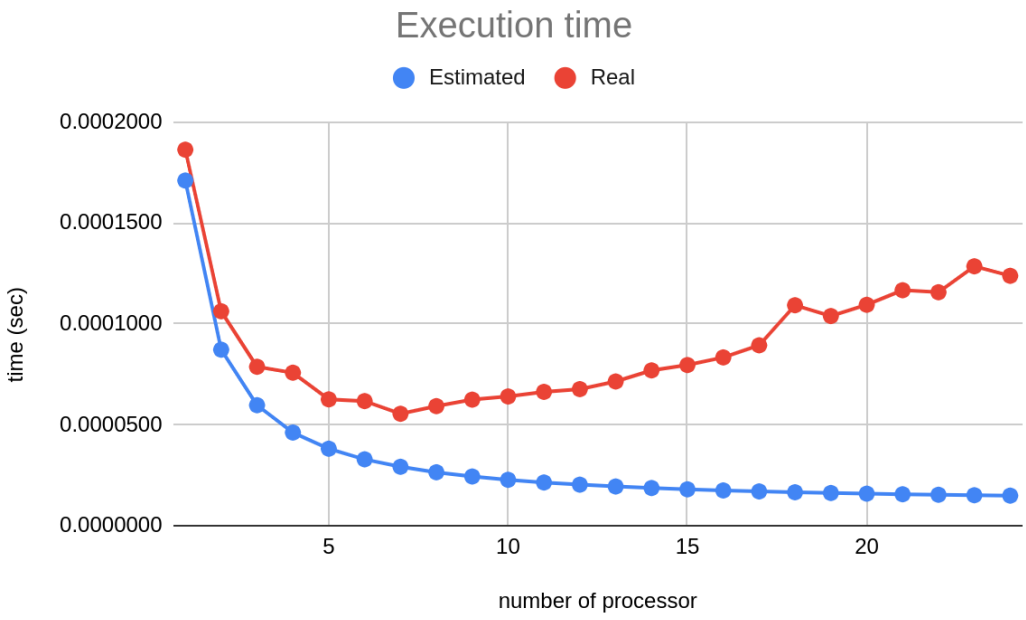

Figure 2. The performance diagram (~24 processors)

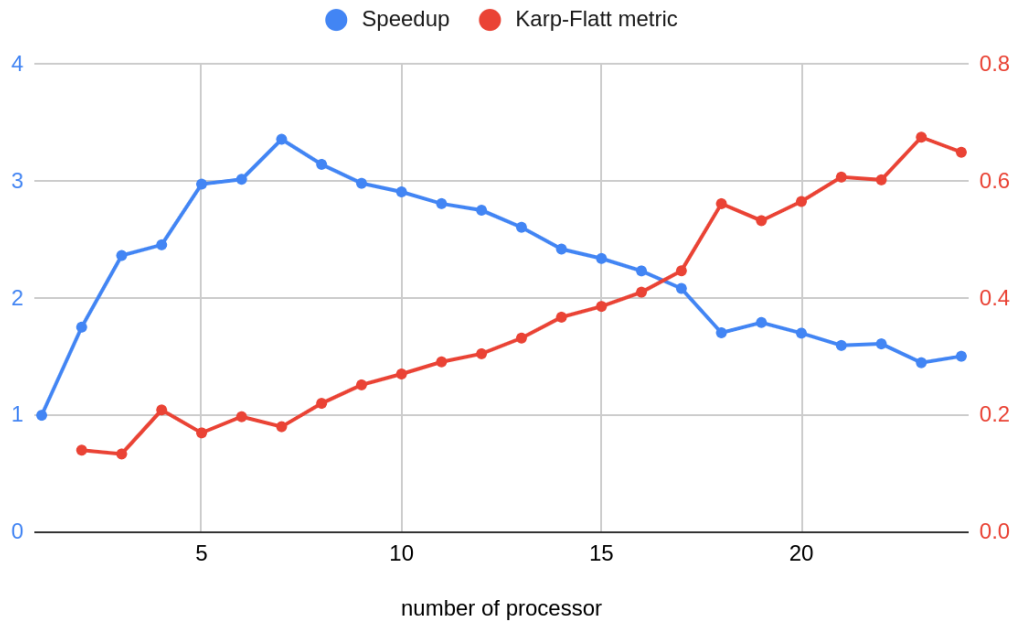

Figure 3. The performance diagram (~24 processors)

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. execution time 在 1~8 個 process 的情況下，估計大致上跟實際測量到得很接近，speedup 持續上升。後來則是 speedup 持續下降，但都維持在 1 以上。
2. 目前的實作是 process 內部每次算一個 term，就把 index 加上 2*(processor 數量)，可能可以換成每個 process 分到連續的一塊，index 每次只需要 +1。或是把 2i-1 和 2i 的係數直接先乘好 (16.0, 8.0)。
3. 主要的 communication 是 reduction 的部份，雖然只有 reduce p 個值，但 speedup 還是會受其影響越來越小。因為每個 process 只有用到一個 double 的記憶體，所以基本上 cache 影響不大。
4. Karp-Flatt metric 持續上升，代表主要是平行 overhead 影響，導致 speedup 無法再提高

Appendix(optional):

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