Parallel Programming Exercise 4 – 12

Author:	林子傑 (r10525069@ntu.edu.tw)
Student ID	R10525069
Department	Engineering Science and Ocean 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.)

Problem:利用 Simpson's Rule 計算 PI。

Proposed Approach:每個 processor 用迴圈將 f 函數的第 in/p 到 (i+1)n/p-1 項的結果加總，再用 reduction 將所有 processor 的和在進行加總。為了測試平行運算的效能，將 n 設為 1000000。

2 Theoretical Analysis Model

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

Sequential algorithm complexity：$\Theta(n)$

Parallel computational complexity：$\Theta(n/p)$

Parallel communication complexity：$\Theta(\log p)$

Parallel overhead：$T_o(n, p) = \Theta(p \log p)$

Iso-efficiency relation：$p >= Cp \log p$

$M(n)=1$

$M(Cp \log p)/p = C \log p / p$

3 Performance Benchmark

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

時間分析：

The time to perform add：χ

Sequential execution time：$(n-1)\chi$

Parallel：

The computation time for each process: $\chi(\lceil n/p \rceil -1)$

A reduction of p values distributed among p tasks can be preformed in $\lceil \log p \rceil$ communication steps.

Each reduction stop requires time：$\lambda + \chi$

Parallel execution time：$(\lceil n/p \rceil -1)\chi + \lceil \log p \rceil (\lambda + \chi)$

Table 1. The execution time

Processors	1	2	3	4	5	6	7	8
Real execution time	0.017284	0.008892	0.005917	0.004464	0.003593	0.003019	0.002566	0.002324
Estimate execution time	0.015	0.0075	0.005	0.00375	0.003	0.0025	0.002143	0.001875
Speedup		1.94377	2.921075	3.871864	4.810465	5.725075	6.735776	7.437177
Karp-flatt metrics		0.028928	0.01351	0.011031	0.00985	0.009604	0.006538	0.010811

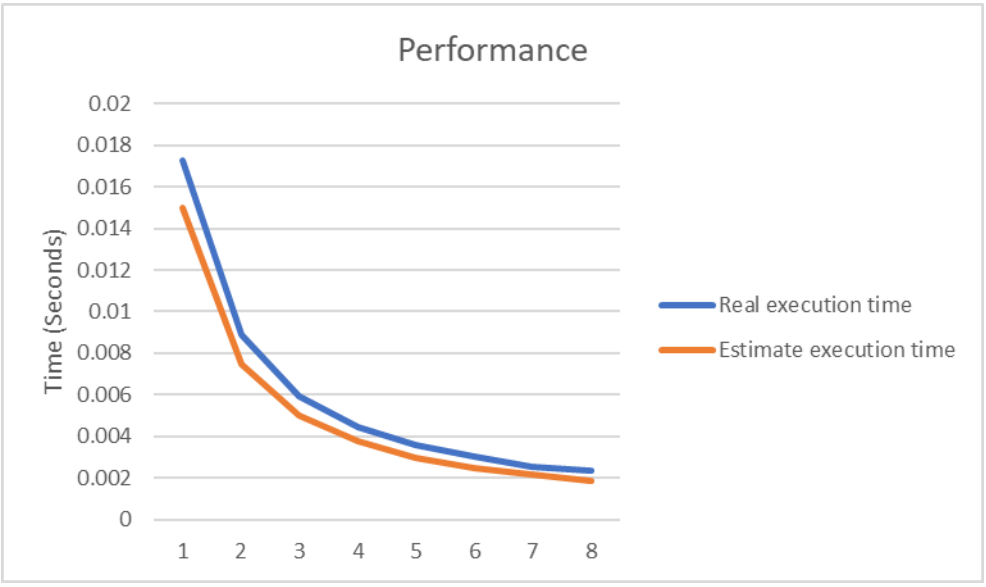

Figure 1. The performance of diagram

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?

)

從 speedup 的數據來看，當 processor 增加，speedup 的數據也會增加，本問題適合用平行計算。從 Iso-efficiency metrics 顯示出這個程式有很好的 Scalability。

Appendix(optional):

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