Parallel Programming Exercise 5 – 11

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: 計算 1/1+1/2+1/3+…+1/n。

Proposed Approach:每個 processor 負責 in/p 到 (i+1)n/p-1 的範圍，如果邊界是偶數，就往外 +1 或 -1。每個 processor 維護一個變數 sum，加總區間內所有 I 的倒數，最後用 reduction 將所有 processor 的 sum 加總。

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：$To(n, p) = \Theta(p \log p)$

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

$M(n)=n$

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

3 Performance Benchmark

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

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	0.607344	0.301069	0.208903	0.149901	0.125805	0.104973	0.090546	0.079043
Estimate execution time	0.6	0.3	0.2	0.15	0.12	0.1	0.085714	0.075
Speedup		2.017292	2.907301	4.051634	4.827662	5.785716	6.707574	7.683716
Karp-flatt metrics		-0.008572	0.015942	-0.004248	0.008925	0.007407	0.007266	0.00588

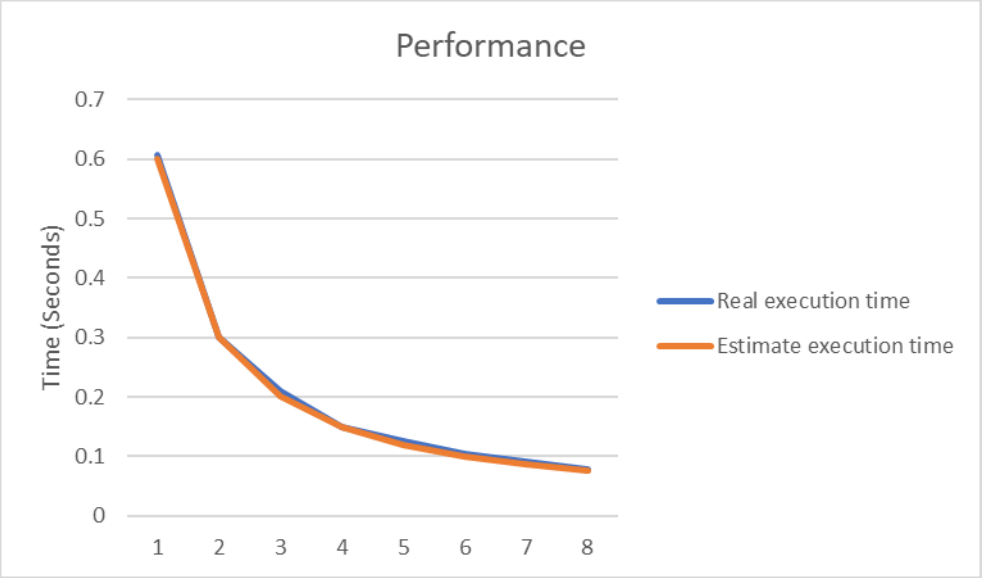

Figure 1. The performance of diagram

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)