

Parallel Programming Exercise 5 – 11

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.)

題目要求計算調和級數，計算到第 1,000,000 項以及要求精度到小數下第 100 位

我的作法就是每一項取夠多位小數直接加起來，最後取 100 位小數。

考慮 A/B ， $A = B*Q + R$ 兩邊同乘 100，就變成 $100A = B*100Q + 100R$ ，原本 Q 小數下兩位就會變成整數最後兩位。所以實際上是將被除數 1 乘上 $1eK$ ，直接做整數的長除法，得到的商的整數部份就包含小數下 K 位的部份。

因為 `long long` 無法容納太大的數字，所以我改成用 `int array` 存大數，每一個 `int` 存 0 ~ 999,999,999 的數字 (base 1,000,000,000)。取 10 的次方可以避免進位之間的轉換，而 1,000,000,000 可以保證加法需要 `carry` 的時候不會 `overflow`。

最後使用 `Block decomposition` 將 1~1,000,000 分成 p 個 `block`，每個 `process` 算自己 `block` 的總和，最後用 `reduce` 算出全部的和，輸出時直接將每個 `int` 轉成字串接在一起就好了。

2 Theoretical Analysis Model

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

N : 項數

d : 小數精度

L : array 長度, $L = \left\lceil \frac{[d + \log_{10} N]}{\log_{10} (base)} \right\rceil + 1 = O(d + \log_{10} N)$

λ : process 間傳資料所需時間

χ : reduce 兩個值所需時間

Array 長度: $\log_{10}(base)$ 固定為 9，除了 d 位小數以外額外取了 $\log_{10}(N)$ 位數，以考慮 d 以後的位數進位的影響

每一項的計算複雜度固定，只和 array 長度有關。

Sequential time complexity: $O(\chi NL)$

(每次 reduce 都需要 reduce L 個值)

Computation time complexity: $O(\frac{N}{p} \chi L)$

Communication time complexity: $O(\lceil \log p \rceil (\lambda + \chi)L)$

Parallel time complexity: $O(\chi \frac{N}{p} L + \lceil \log p \rceil (\lambda + \chi)L)$

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	0.3472	0.1760	0.1188	0.0891	0.0727	0.0599	0.0514	0.0449
Estimate execution time	0.3562	0.1781	0.1188	0.0891	0.0713	0.0594	0.0510	0.0446
Speedup	x	2.01	2.93	3.94	4.63	5.83	6.78	7.74
Karp-flatt metrics	x	-0.0027	0.0112	0.0053	0.0200	0.0059	0.0055	0.0047

Execution time vs. Number of processor

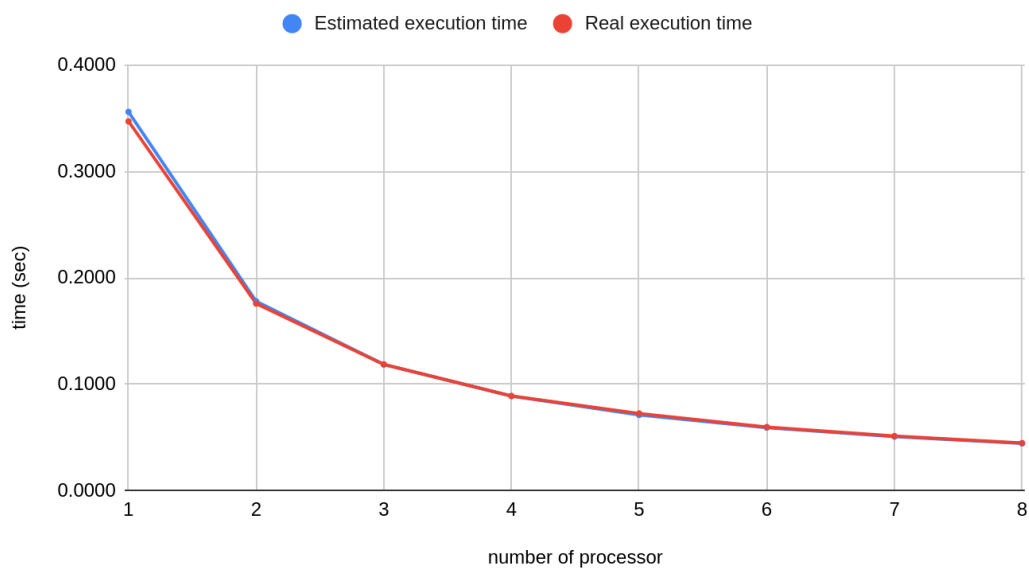


Figure 1. The performance of diagram

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?

)

1. speedup 隨著 p 增大時增加
2. 現在是每次兩個大數加起來的時候馬上做 carry，如果 base 取小一點，就可以在同一格內放比較大的數字而不 overflow，可以變成加好幾個大數再做一次 carry，這部份應該可以再調整。
3. 根據 model 算出來的 computation time 和 communication time，主要花的時間都在 computation，communication 影響不大。
4. Karp-Flatt metrics 大致隨著 p 而下降，代表 sequential 部份佔了很大一部分。

Appendix(optional):

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