Parallel Programming Exercise Chpater 9 – 9.10

Author:	張瀚文(b07505027@ntu.edu.tw)
Student ID	B07505027
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.)

Manager-worker model

群舉 n，如果 2^n – 1 是質數，則可以找到一個完美數。直到找到 8 個完美數為止。數字由 manager 分配給 worker，因為沒有辦法預期每個數計算所需的時間，也沒辦法預期找到第 8 個完美數 n 會到多大，所以採用此模型希望能夠可以平衡負載

2 Theoretical Analysis Model

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

判斷質數 + 傳輸 n 次

$$\sqrt{2^n-1}\frac{n}{p}\chi + n(\lambda + \frac{4}{\beta})$$

3 Performance Benchmark

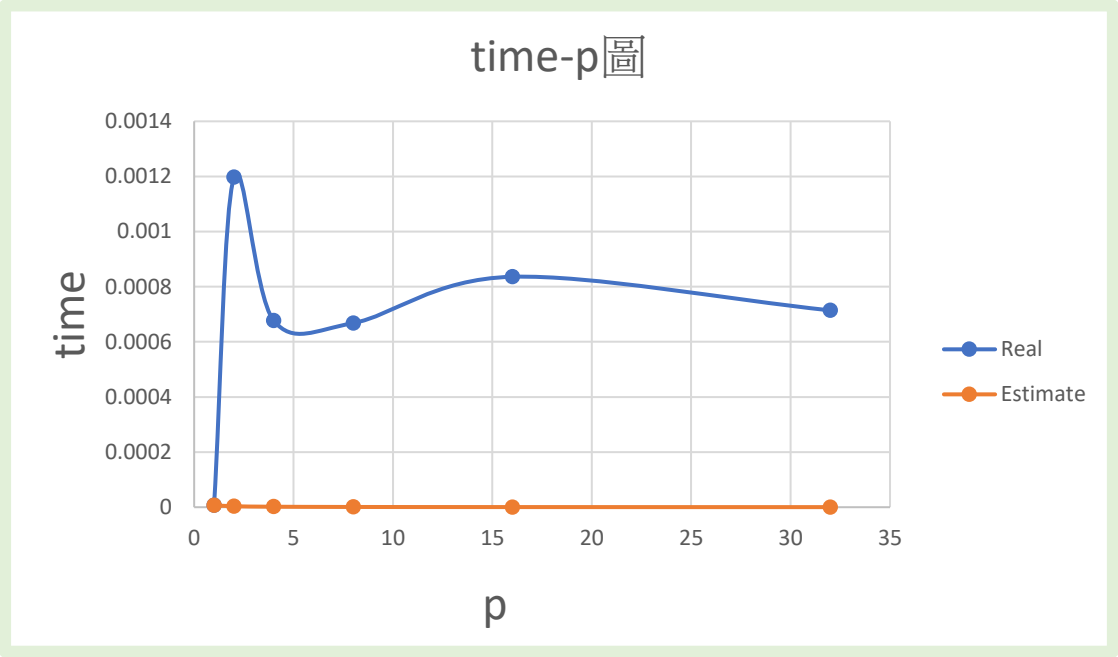
(Give your idea or concept of how you design your program.)

Table 1. The execution time

Processors	1	2	4	8	16	32	64
Real execution time	0.0000	0.001197	0.000677	0.000668	0.000836	0.000714	2.850854
Estimate execution time	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Speedup	1.0000	0.0058	0.0103	0.0105	0.0084	0.0098	0.0000
Karp-flatt	#DIV/0!	0.0020	0.0015	0.0011	0.0012	0.0017	0.0204

metrics							

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. <1，在 process 數量是 1 的時候最快。在 2<=p<=8 時還算正常。後來出現異常
2. 這個問題採用這種方式似乎不太妥當，因為實際上算到第 8 個完美數的時候，n 才到 31 而已，每個 process 只會被分到個位數次，而主要時間都畫在判斷質數。就算算到第 10 個完美數，n 也只到 89，而且因為第 9 個完美數的位數已經達到 37 位，用 c++內建 long long 算也會溢位，算不出來...。因此應該要設計一個好的判斷質數方法，或是做大數運算，才對目前比較有幫助。
3. 其實只有一個 process 的話最快，可以馬上算出來.....因為 n 只會到 31，很快就結束了。當 process 數量很大的時候，雖然可以在更短的時間內驗證更多的 n，但似乎因為 process 太多，讓算好完美數的 process 沒辦法趕快回傳完美數，造成多算了很多不用算的 n (而且其實在 n 很小的時候就會溢位，造成程式異常.....)。
4. 雖然有點不太可採信，但 e 在 p<64 時似乎是常數

5 Appendix(optional):

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

p = 4 時的結果，看起來比較正常（但比 p = 1 時慢 XD）

```
----------------------------------------
n = 2, the 1 perfect number is: 6
n = 3, the 2 perfect number is: 28
n = 5, the 3 perfect number is: 496
n = 7, the 4 perfect number is: 8128
n = 13, the 5 perfect number is: 33550336
n = 17, the 6 perfect number is: 8589869056
n = 19, the 7 perfect number is: 137438691328
n = 31, the 8 perfect number is: 2305843008139952128
Verified n range: 1 ~ 486
0) Execution time 0.000677
Process number: 4   , Max time: 0.000677
----------------------------------------
```

p = 64 下的結果

```
----------------------------------------
n = 2, the 1 perfect number is: 6
n = 7, the 2 perfect number is: 8128
n = 13, the 3 perfect number is: 33550336
n = 3, the 4 perfect number is: 28
n = 17, the 5 perfect number is: 8589869056
n = 5, the 6 perfect number is: 496
n = 19, the 7 perfect number is: 137438691328
n = 31, the 8 perfect number is: 2305843008139952128
Verified n range: 1 ~ 2364705
0) Execution time 2.850854
Process number: 64  , Max time: 2.850854
----------------------------------------
```

可以看到 n 算到了 2364705，多算了很多次導致驗證過的完美數沒有馬上被回報（或溢位產生奇怪的錯誤）