**Parallel Programming Exercise 9 – 10**

|  |  |
| --- | --- |
| **Author:** | 李明軒 ([b08505012@ntu.edu.tw](mailto:b08505012@ntu.edu.tw)) |
| **Student ID** | B08505012 |
| **Department** | Engineering Science and Ocean Engineering |

# Problem and Proposed Approach

(Brief your problem, and give your idea or concept of how you design your program.)

Problem:

Write a parallel program to find first eight perfect number.

Concept:

Using Manager/worker method. Manager will assign new work to worker once they finish their current job. Processor 0 will be the manager, other processors will be worker. So the number of processor must >=2.

# Theoretical Analysis Model

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

time complexity: Θ(1) (since only find 8 perfect number)

# Performance Benchmark(m=n=10)

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

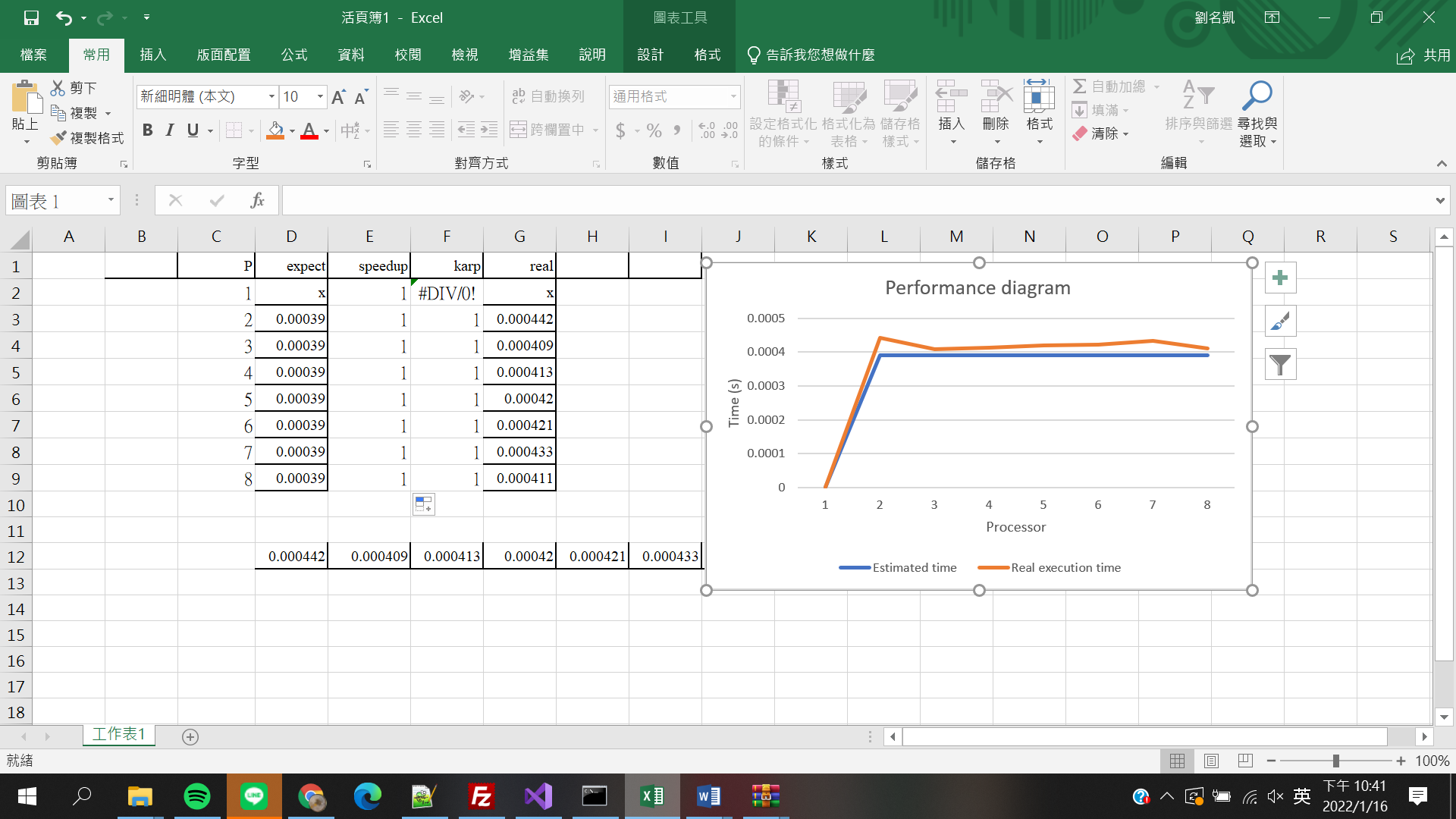
Estimated execution time ≒ while loop time in manager()+ sorting time.

While loop time ≒ 0.0003888607 (I don’t know why it’s independent of number of processors …)

Sorting time ≒ 0.000000953674316

Table . The execution time

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Processors | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Real execution time | x | 0.000442 | 0.000409 | 0.000413 | 0.000420 | 0.000421 | 0.000433 | 0.000411 |
| Estimate execution time | x | 0.00039 | 0.00039 | 0.00039 | 0.00039 | 0.00039 | 0.00039 | 0.00039 |
| Speedup | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Karp-flatt metrics | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



# Conclusion and Discussion

1. What is the speedup respect to the number of processors used?

: The benchmarking result is pretty weird, because the manager/worker program speedup should be increasing when we increase number of processors. But the result shows that the execution time remain the same no matter how many processors are, so the speedup is always 1.

1. How can you improve your program further more

: In my program, workers’ job is to test whether the number they received is a perfect number, and they only test one number at a time. If I can assign more job to worker at a time, there will be less communication, increase the efficiency of the program.

1. How does the communication and cache affect the performance of your program?

: In this program, communication time dominates the total execution time, so the communication is the main reason why my program is slow.

1. How does the Karp-Flatt metrics and Iso-efficiency metrics reveal?

: Like I said in discussion 1, my benchmarking result is weird, speedup are all the same, so e are all the same too.

**Appendix(optional):**

**p=10**

