**Parallel Programming Exercise 8 – 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 that implements matrix-vector multiplication based on a checkboard block decomposition of the matrix. the program will read the matrix and vector from an input file and print out the result.

Concept:

First create cartesian communicator, then processor 0 read the matrix and vector, assign certain part to each processor. Each processor compute their own sub vector, and use MPI\_Reduce to collect the final result.

# Theoretical Analysis Model

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

Computation complexity: Θ(n^2/p)

Communication complexity: Θ()

Total time complexity: Θ(n^2/p+)

Iso-efficiency: n>C)

T(n,1)= Θ(n^2), To(n,p)=pk=Θ()

M(n)=n^2, M(Cplogp)/p=C^2(log)^2

# Performance Benchmark(m=n=10)

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

Estimated execution time: 0.0001p+0.0004+0.00025+2λ(logp)

Time:

1. create communicator

2. function **read\_checkboard\_matrix**

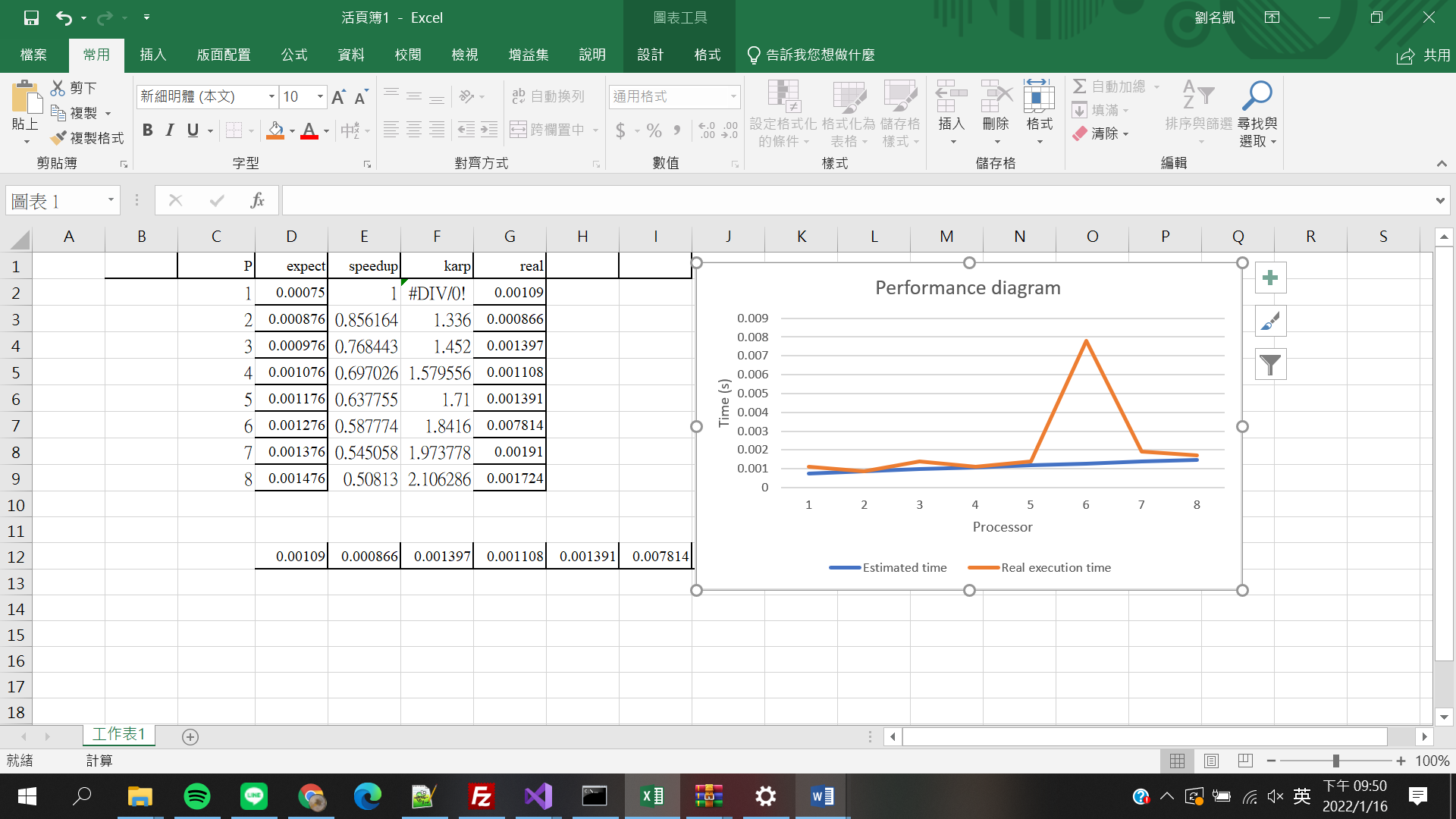
3. function **read\_block\_vector**

4. MPI\_Reduce and MPI\_Boardcast

λ=0.000013.

Table . The execution time

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Processors | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Real execution time | 0.00109 | 0.000866 | 0.001397 | 0.001108 | 0.001391 | 0.007814 | 0.001910 | 0.001724 |
| Estimate execution time | 0.00075 | 0.000876 | 0.000976 | 0.001076 | 0.001176 | 0.001276 | 0.001376 | 0.001476 |
| Speedup | 1 | 0.856 | 0.768 | 0.697 | 0.637 | 0.587 | 0.545 | 0.508 |
| Karp-flatt metrics | x | 1.336 | 1.452 | 1.579556 | 1.71 | 1.8416 | 1.973778 | 2.106286 |



# Conclusion and Discussion

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

: As number of processor increase, speedup will decread. Because computation time is relatively small compared to communication time. Therefore, the more processor, the slower the program.

1. How can you improve your program further more

: Right now my program can only read double, which means the range of elements in matrix is limited. this is a huge problem, since parallel programming are mostly used in simulation which often involve large number. So if I can make the program able to read any kind of data, it would definitely be better.

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 that make the program slower when increasing number of processors.

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

: e is increasing when number of processor increase. So we can know that overhead is the main reason why this program have limited speedup. And the scalability function says this program has pretty good scalability.

**Appendix(optional):**