Parallel Programming Exercise　9 – 10

Author:	邱亮茗 (r12942159@ntu.edu.tw)
Student ID	R12942159
Department	Graduate Institute of Communication 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.)

A perfect number is a positive integer equal to the sum of all its divisors excluding itself. Euclid provided a formula to generate perfect numbers: If 2**n−1 is prime, then (2 ** n − 1) * 2 ** (n − 1) is a perfect number. The task is to find the first eight perfect numbers using parallel programming.

Step1: Focus on identifying Mersenne primes (2**n−1), as finding them efficiently leads to discovering perfect numbers.

Step2: Divide the search range for n among multiple processors or threads.

Step3: For each n where 2**n−1 is prime, calculate the corresponding perfect number (2**n−1)·2**(n−1).

2 Theoretical Analysis Model

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

Vector size: N

Number of processes: P, each handles approximately N/P candidate values of n , where N is the total range of n values.

Communication: log(P).

⇨ T(N,1) = O(N)

⇨ T(N,P) = O(P * log(P))

⇨ N >= C * P * log(P)

⇨ iso-efficiency metrics**:** M(C * P * log(P)) / P = C * log (P).

⇨ i.e Good scalable algorithm.

3 Performance Benchmark

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

Table 1. The execution time with 10000 iterations

Processors	1	2	3	4	5	6	7	8
Real execution time	5.535017	0.008505	0.093468	0.004461	0.058289	0.002844	0.010583	0.004645
Estimate execution time	5. 535017	2.76751	1. 84501	1.38375	1.10700	0.92250	0.790717	0.691877
Speedup	1	325.39	19.73	310.18	18.99	324.36	74.71	148.95
Karp-flatt metrics		-0.99	-0.47	-0.33	-0.23	-0.19	-0.16	-0.14

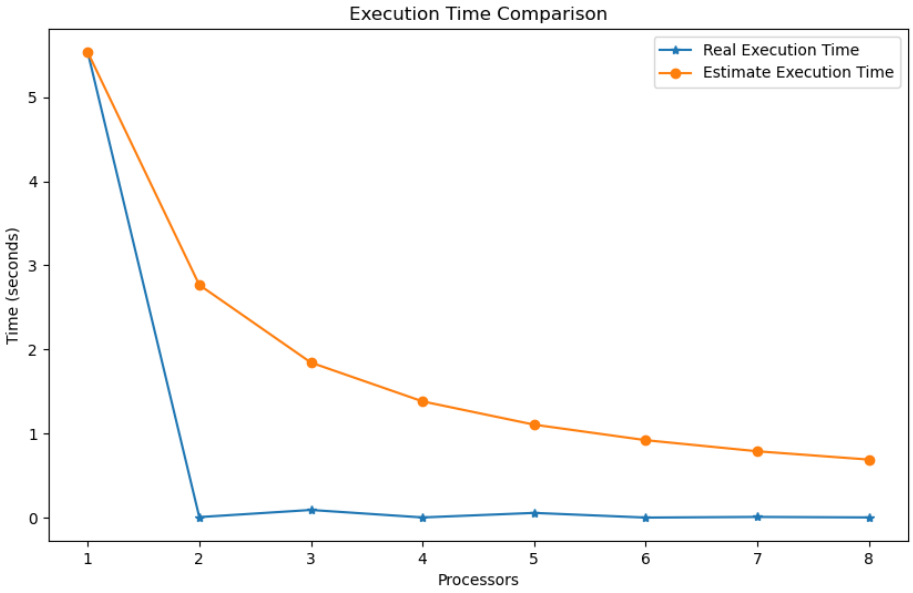

Figure 1. The performance of diagram

4 Conclusion and Discussion

(Discuss the following issues of your program

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

   The speedup is inconsistent and does not scale linearly with the number of processors. Some configurations (e.g., 3, 5 processors) show lower speedup values (19.73, 18.99), suggesting that load imbalance may limit performance improvements. In this case, the program demonstrates some impressive speedup in certain cases, inefficiencies in scaling are evident, particularly as the number of processors increases. This indicates a need for better optimization in workload distribution, communication, and memory usage to achieve consistent scalability.

2. How can you improve your program further more?

   The Manager/Worker method can improve the program by:
   (1) Reducing Load Imbalance: Assign tasks dynamically based on worker availability to ensure all processors are utilized effectively.
   (2) Minimizing Idle Time: The manager assigns smaller chunks of work at a time to balance the workload among workers.
   (3) Reducing Communication Overhead: Use asynchronous communication to overlap computation and communication, ensuring workers do not wait for instructions.

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

   Communication Overhead: Increased processors require more communication to synchronize tasks and using efficient communication protocols and reducing the frequency of synchronization can mitigate this..
   Cache Efficiency: If the working set exceeds the processor's cache size, there will be frequent memory access, leading to cache misses.

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

   The iso-efficiency metric measures how well the parallel system scales as the problem size increases. If the iso-efficiency metric is high, it means the system requires a significant increase in workload to maintain efficiency as more processors are added. But in this case, the inconsistent speedup values and Karp-Flatt metrics suggest that the program has room for optimization in terms of balancing workload.

)

Appendix(optional):

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