**Parallel Programming Exercise 9 – 10**

|  |  |
| --- | --- |
| **Author:** | 蕭彧 ([r11922138@ntu.edu.tw](mailto:r11922138@ntu.edu.tw)) |
| **Student ID** | R11922138 |
| **Department** | Computer Science |

# Problem and Proposed Approach

This programming exercise is to find the first six perfect numbers, a positive integer whose value is equal to the sum of all its positive factors. There’s a hint for this problem. The Greek mathematician Euclid (c. 300 BCE) showed that if 2n-1 is prime, then (2n-1)2n-1 is a perfect number. I design a manger worker program to solve this problem. Manager will dispatch n (n > 1) to worker, and worker check whether 2n-1 is prime. If it’s a prime number, worker will return (2n-1)2n-1 to the manager. Manager itself owns a vector that store the perfect number, if the vector size is six, the while loop will terminate. The key of this program is how to efficiently do a prime number test. Below is my is\_prime function.

bool is\_prime(uint64\_t n) {

    if (n <= 1) return false;

    if (n <= 3) return true;

    if (n % 2 == 0) return false; // Exclude even numbers except 2

    // Check divisibility for odd numbers starting from 3 up to sqrt(n)

    for (int i = 3; i <= sqrt(n); i += 2) {

        if (n % i == 0) {

            return false; // If divisible, it's not a prime number

        }

    }

    return true;

}

# Theoretical Analysis Model

Assume we need do prime test on n number.

χ is time for is\_prime function

λ is the message latency

β is network bandwidth

Sequential execution time: n\*χ

Data Transfer time: 2n(λ + 8/β)

Expected execution time: χ ⎡*n*/*p*⎤ + 2n(λ + 8/β)

# Performance Benchmark

The message latency is λ = 0.001521 sec, and the network bandwidth is β = 1658 MB/sec

Table . The execution time

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Processors | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| Real execution time | 0.00061 | 0.0007 | 0.00012 | 0.00012 | 0.00205 | 0.0005 | 0.0077 | 0.00674 |
| Estimate execution time | 0.00065 | 0.00074 | 0.00022 | 0.00018 | 0.00245 | 0.00068 | 0.0089 | 0.0074 |
| Speedup | 1 | 0.87 | 5.08 | 5.08 | 0.3 | 1.22 | 0.079 | 0.09 |
| Karp-flatt metrics | - | 1.3 | 0.07 | 0.082 | 3.49 | 0.81 | 12.84 | 11.19 |

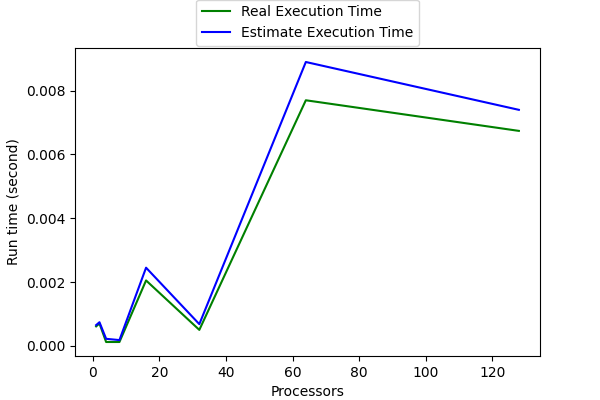


Figure . The performance of diagram

# Conclusion and Discussion

Figure 1 shows that this program is heavily dominate by the communication time. The reason is it only let us to find first 6 perfect number. If the program is finding 60 perfect numbers, I believe the communication cost will be negligible. The prime testing function also determine the run time. A smart prime testing can significantly reduce the run time. We only need to check whether a number is its factor to square root of prime tester.