

Parallel Programming Exercise 6 – 8

| | |
|-------------------|---|
| Author: | 李子筠(b06901145@ntu.edu.tw) |
| Student ID | b06901145 |
| Department | Electrical 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.)

The program implements ping pong test, sending message back and forth between two processes to measure average message passing time.

The outer for loop iterate through different `mesg_len`, measuring t with respect to different n .

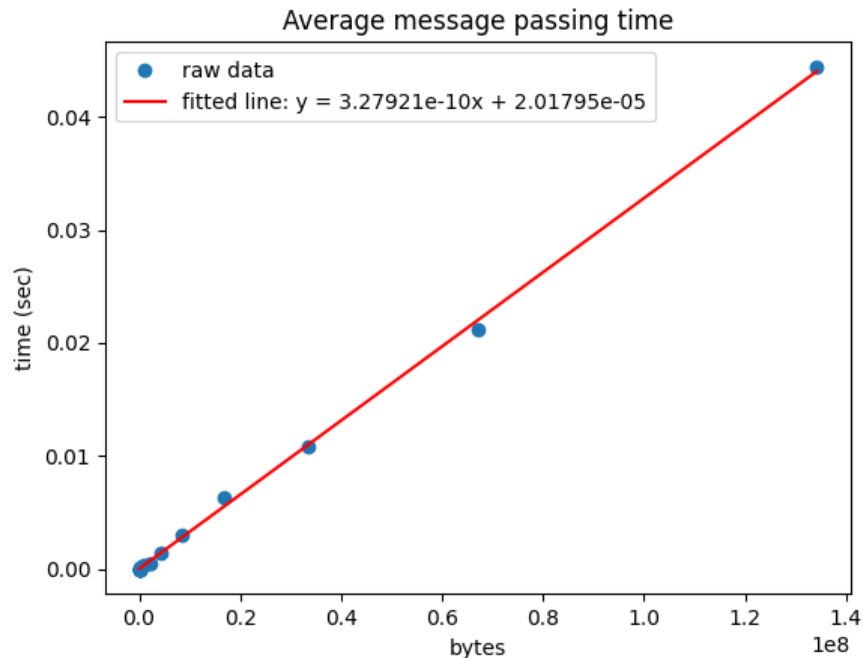
The inner for loop measures the same point multiple times. By taking the average of multiple measurement, we can reduce the deviation of measured time.

Message passing time and the size of message is related by $t = \lambda + n/\beta$

```
for (int mesg_len = MSG_START; mesg_len <= BUFFER_SIZE; mesg_len *= 2) {
    double elapsed_time = 0;
    for (int sample = 0; sample < SAMPLE_SIZE; ++sample) {
        MPI_Barrier(MPI_COMM_WORLD);
        if (rank == 0) {
            elapsed_time += -MPI_Wtime();
            MPI_Send(buf, mesg_len, MPI_CHAR, 1, 0, MPI_COMM_WORLD);
            MPI_Recv(recv_buf, mesg_len, MPI_CHAR, 1, 1, MPI_COMM_WORLD, &status);
            elapsed_time += MPI_Wtime();
        } else {
            MPI_Recv(buf, mesg_len, MPI_CHAR, 0, 0, MPI_COMM_WORLD, &status);
            MPI_Send(buf, mesg_len, MPI_CHAR, 0, 1, MPI_COMM_WORLD);
        }
    }
    if (rank == 0) {
        printf("%d, %f\n", mesg_len, elapsed_time / 2 / SAMPLE_SIZE);
    }
}
```

2 Theoretical Analysis Model

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



The estimated parameters are: $\lambda = 2.017 \cdot 10^{-5}, \beta = 3.049 \cdot 10^9$

3 Performance Benchmark

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

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?

)

Appendix(optional):

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