Lab Session Week 9

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Student Email** |
| 32844700 | Teh Jia Xuan |  |

Task 1 – Code

Task 2 – Code (basic)

{Answers/Code}

{Screenshots}

Task 2 – Code (Extended Task)

{Answers/Code}

{Screenshots}

Task 2 – Observations, results and explanation

Basic code

A screen shot of a computer

Description automatically generated

Arrangement

A screen shot of a computer program

Description automatically generated

Extended code

A screen shot of a computer

Description automatically generated

Arrangement

A screen shot of a computer program

Description automatically generated

Here are the reasons why my extended code is faster:

1. We can observe that extended code is faster than basic code as I have optimised the code using MPI\_Isend and MPI\_Irecv. By using non blocking communication it allows processes to continue computation while messages are being sent or received. Hence, reducing idle times.
2. Reduced communication overhead, in my code, Instead of sending a single prime number and waiting for a response in every iteration. I have generated all 500 prime numbers and store it in an array. After that, I sent the array across its adjacent processors at once. So I just need to send once instead of sending 500 times. This minimises the communication overhead caused by repeatedly sending and receiving messages. This makes the code more efficient.
3. Improved message size efficiency, sending a single large message is more efficient than sending 500 smaller messages due to it has fixed communication cost attached with each message.

Task 4 – Code (basic)

Time

A screenshot of a computer program

Description automatically generated

Arrangement

A screen shot of a computer program

Description automatically generated

Task 4 – Code (Extended Task)

A screen shot of a computer

Description automatically generated

Arrangement

A screen shot of a computer

Description automatically generated

Task 4 – Observations, results and explanation

We can observe that the refine version is running faster than the basic one. It improves from 2.981s to 0.447s. The reasons that it is running faster are:

1. Packing multiple primes in a buffer, by packing 500 prime numbers into a single buffer and sending that buffer in one communication, it reduces the total number of messages exchanged. Each communication in MPI involves some latency and overhead, so reducing the number of sends and receives leads to significant performance improvements.
2. Non blocking communication with MPI\_Isend and MPI\_Irecv, non blocking communication allows processes to overlap computation and communication. Instead of waiting for each communication to complete before proceeding, the program continues its computation while messages are being sent and received. Hence, it spent less idle time waiting for communication to finish.