Assignment 3: One-to-Many Communication in MPI

Assignment 3: One-to-Many Communication in MPI

Task 1: Basic Understanding of Broadcast

Write an MPI program where:

- Process 0 initializes an integer variable.

- Use MPI_Bcast to broadcast this variable to all other processes.

- Every process should print the received value.

Questions:

1. What happens if a non-root process changes the value before the broadcast?

2. What constraints must be followed when calling MPI_Bcast?

3. How does MPI_Bcast differ from point-to-point send/receive operations?

Task 2: Data Scattering and Gathering (Intermediate)

Write a program that:

- Initializes an array of 16 integers in process 0.

- Use MPI_Scatter to send 4 integers to each of 4 processes.

- Each process multiplies its chunk by 2.

- Use MPI_Gather to collect the updated data back to process 0.

- Process 0 should print the final array.

Instructions:

- Assume total number of processes is 4.

- Use correct datatypes and counts for MPI_Scatter and MPI_Gather.

Follow-up Questions:

a. What will happen if the number of processes is not evenly divisible by the data size?

b. Modify the program to handle uneven distribution using dynamic offsets (optional).

Task 3: Distributed Reduction and All-Gather (Advanced)

Write a program using 6 processes where:

- Each process generates a random number between 1 and 100.

- Use MPI_Allgather to collect all numbers into an array on every process.

- Use MPI_Reduce to compute the maximum value among all numbers, with process 0 printing it.

- Then, use MPI_Allreduce to compute the average value and display it from all processes.

Enhancements:

- Time the MPI_Allgather and MPI_Reduce phases using MPI_Wtime.

- Print both raw timing and number of operations performed.

Questions:

a. Why is MPI_Allgather more expensive than MPI_Gather?

b. Can MPI_Allreduce be used as a substitute for Reduce+Broadcast? Explain.

c. What challenges arise if the data is large (e.g., arrays of size 1 million)?

Note:

- Use only MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Allgather, MPI_Reduce, and MPI_Allreduce in this assignment.

- All programs must be generalizable to different process sizes (where feasible).