

Lab Submission – 10

Arnab Mondal

20BCE1294

Program: B.Tech

Semester: Fall 2022-23

Course: CSE4001 – Parallel and Distributed Computing

Faculty: Dr. Sudha A

Date: 03-10-2022

Exercise: 10

1. Use inbuilt MPI_Bcast and MPI_Ibcast for broadcasting the message from root to all the other process
2. Use ur own function broad_cast with MPI Send and Recv

OPTIONal for practice:

Use MPI_Wtime ,MPI_Barrier if required to check the time among MPI_Bcast, MPI_Ibcast and ur own function.

bcast.c

```
#include <stdio.h>
#include <string.h>
#include <mpi.h>
int main(int argc, char **argv)
{
    char message[20];
    int i,rank,size;
    MPI_Status status;
    int root=0;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if(rank==root)
    {
```

```

        strcpy(message, "Hello World");
    }
    MPI_Bcast(message,13,MPI_CHAR, root, MPI_COMM_WORLD);
    printf("Message from process %d: %s\n", rank, message);
    MPI_Finalize();
}

```

Output:

```

arnab@arnab-VirtualBox:~/Desktop$ mpirun -np 4 ./object
Message from process 0: Hello World
Message from process 2: Hello World
Message from process 1: Hello World
Message from process 3: Hello World

```

ibcast.c

```

#include <stdio.h>
#include <string.h>
#include <mpi.h>
int main(int argc, char **argv)
{
    char message[20];
    int i,rank,size;
    MPI_Status status;
    MPI_Request request = MPI_REQUEST_NULL;
    int root=0;MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if(rank==root)
    {
        strcpy(message, "Hello World");
    }
    MPI_Ibcast(message,13,MPI_CHAR, root, MPI_COMM_WORLD, &request);
    MPI_Wait(&request, &status);
    if(rank==root)
    {
        strcpy(message, "What will happen?");
    }
    printf("Message from process %d: %s\n", rank, message);
    MPI_Finalize();
}

```

Output:

```
arnab@arnab-VirtualBox:~/Desktop$ gedit ibcast.c
arnab@arnab-VirtualBox:~/Desktop$ mpicc ibcast.c -o object
arnab@arnab-VirtualBox:~/Desktop$ mpirun -np 4 ./object
Message from process 3: Hello World
Message from process 0: What will happen?
Message from process 1: Hello World
Message from process 2: Hello World
```

cast2.c

```
#include<stdio.h>
#include<stdlib.h>
#include<mpi.h>
void my_bcast(void* data, int count, MPI_Datatype datatype, int root, MPI_Comm
communicator)
{
    int world_rank;
    MPI_Comm_rank(communicator, &world_rank);
    int world_size;
    MPI_Comm_size(communicator, &world_size);
    if(world_rank == root)
    {    // If we are the root process, send our data to everyone
        int i;
        for(i = 0; i < world_size; i++)
        {
            if(i != world_rank)
            {
                MPI_Send(data, count, datatype, i, 0, communicator);
            }
        }
    }
    else
    {    // If we are a receiver process, receive the data from the root
        MPI_Recv(data, count, datatype, root, 0, communicator,
MPI_STATUS_IGNORE);
    }
}
int main(int argc, char** argv)
{
    MPI_Init(NULL, NULL);
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    int data;
    if(world_rank == 0)
    {
        data = 100;
        printf("Process 0 broadcasting data %d\n", data);
```

```
        my_bcast(&data, 1, MPI_INT, 0, MPI_COMM_WORLD);
    }
    else
    {
        my_bcast(&data, 1, MPI_INT, 0, MPI_COMM_WORLD);
        printf("Process %d received data %d from root process\n", world_rank,
data);
    }
    MPI_Finalize();
}
```

Output:

```
arnab@arnab-VirtualBox:~/Desktop$ gedit cas2.c
arnab@arnab-VirtualBox:~/Desktop$ mpicc cas2.c -o object
arnab@arnab-VirtualBox:~/Desktop$ mpirun -np 4 ./object
Process 0 broadcasting data 100
Process 2 received data 100 from root process
Process 3 received data 100 from root process
Process 1 received data 100 from root process
```