CL-9
BE-9 P9
Roll No. 43104

Assignment 2:

Problem Statement : Design a distributed application using MPI for computation where root process has an array of elements equal to the size of processors which is divided to the worker processes which calculates the reciprocal and resultant array will be displayed at root

Code :

```
import mpi.MPI;

   public class ScatterGather {
    public static void main(String args[]){
   //Initialize MPI execution environment
    MPI.Init(args);
   //Get the id of the process
    int rank = MPI.COMM_WORLD.Rank();
   //total number of processes is stored in size
    int size = MPI.COMM_WORLD.Size();
    int root=0;
   //array which will be filled with data by root process
    float sendbuf[]=null;

    sendbuf= new float[size];

   //creates data to be scattered
    if(rank==root){
        sendbuf[0] = 10;
        sendbuf[1] = 20;
        sendbuf[2] = 30;
        sendbuf[3] = 40;

       //print current process number
        System.out.print("Processor "+rank+" has data: ");
        for(int i = 0; i < size; i++){
            System.out.print(sendbuf[i]+" ");
        }
        System.out.println();
    }
```

```
    //collect data in recvbuf
     float recvbuf[] = new float[1];

    //following are the args of Scatter method
    //send, offset, chunk_count, chunk_data_type, recv, offset, chunk_count,
chunk_data_type, root_process_id
     MPI.COMM_WORLD.Scatter(sendbuf, 0, 1, MPI.FLOAT, recvbuf, 0, 1,
MPI.FLOAT, root);
     System.out.println("Processor "+rank+" has data: "+recvbuf[0]);
     System.out.println("Processor "+rank+" is reciprocating the data");
     recvbuf[0]= 1/recvbuf[0];
    //following are the args of Gather method
    //Object sendbuf, int sendoffset, int sendcount, Datatype sendtype,
//Object recvbuf, int recvoffset, int recvcount, Datatype recvtype,
//int root)
     MPI.COMM_WORLD.Gather(recvbuf, 0, 1, MPI.FLOAT, sendbuf, 0, 1,
MPI.FLOAT, root);
    //display the gathered result
     if(rank==root){
        System.out.println("Process 0 has data: ");
         for(int i=0;i<4;i++){
             System.out.print(sendbuf[i]+ " ");
         }
     }
    //Terminate MPI execution environment
     MPI.Finalize();
    }
}
```

Output :

Ajinkya@Tikhe:-/Desktop/MPI$ javac -cp $MPJ HOME/lib/mpj.jar ScatterGather.java

Ajinkya@Tikhe:-/Desktop/MPI$ $MPJ HOME/bin/mpjrun.sh - np 4 ScatterGather MPJ Express (0.44) is started in the multicore configuration
Processor 0 has data: 10.0 20.0 30.0 40.0

Processor 2 has data: 30.0

Processor 2 is reciprocating the data

Processor 3 has data: 40.0

Processor 3 is reciprocating the data

Processor 0 has data: 10.0

Processor 1 has data: 20.0

Processor 1 is reciprocating the data

Processor 0 is reciprocating the data

Process has data: 0.1 0.05 0.033333335 0.025