

Dated :
Assessment No. : 10

Collective Communication Using MPI_Scatter

Scenario – 1

Study the given C program that takes an array of elements and distributes the elements in the order of process rank. The first element (in red) goes to process zero, the second element (in green) goes to process one, and so on. Although the root process (process zero) contains the entire array of data, MPI_Scatter will copy the appropriate element into the receiving buffer of the process. Analyse its key factors in terms of network application system.

Source Code:

MPI Code: Collective Communications

```
*****/
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
#define SIZE 4

int main (int argc, char *argv[])
{
    int numtasks, rank, sendcount, recvcount, source;
    float sendbuf[SIZE][SIZE] = {
        { 1.0, 2.0, 3.0, 4.0},
        { 5.0, 6.0, 7.0, 8.0},
        { 9.0, 10.0, 11.0, 12.0},
        { 13.0, 14.0, 15.0, 16.0} };
    float recvbuf[SIZE];
    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
    if (numtasks == SIZE) {
        source = 1;
        sendcount = SIZE;
        recvcount = SIZE;
        MPI_Scatter(sendbuf,sendcount,MPI_FLOAT,recvbuf,recvcount,
            MPI_FLOAT,source,MPI_COMM_WORLD);
        printf("rank= %d Results: %f %f %f %f\n",rank,recvbuf[0],
            recvbuf[1],recvbuf[2],recvbuf[3]);
    }
    else
        printf("Must specify %d processors. Terminating.\n",SIZE);
    MPI_Finalize();
}
```

Dated :
Assessment No. : 10

Collective Communication Using MPI_Scatter

Conceptual Discussion:

Execution:

Results: