

Lab 5: Collective Communication

Name: Waris Damkham

ID: 6388014

Sec: 1

Save your file to "lab5_62xxxxxx.pdf" and upload it to MyCourses website.

Lab: 1. Pi Calculation

Result from running with number of intervals (n) = __15__ and number of processes (np) = __5__:

```
Enter the number of intervals: (0 quits) 15
pi is approximately 3.1419630237914191, Error is 0.0003703702016260
```

Result from running with number of intervals (n) = __20__ and number of processes (np) = __8__:

```
Enter the number of intervals: (0 quits) 20
pi is approximately 3.1418009868930934, Error is 0.0002083333033003
```

Result from running with number of intervals (n) = __30__ and number of processes (np) = __11__:

```
Enter the number of intervals: (0 quits) 30
pi is approximately 3.1416852461797493, Error is 0.0000925925899562
```

Lab: 2. Modified integersum.c

Source Code:

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[])
{
    int rank, size;
    MPI_Status status;
    int interval;
    int number, start, end, sum, GrandTotal;
    int proc;
    int left, right;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    if (rank == 0)
    {
        GrandTotal = 0;
        printf("Enter the number for LEFT interval: ");
        fflush(stdout);
        scanf("%d", &left);
        printf("Enter the number for RIGHT interval: ");
        fflush(stdout);
        scanf("%d", &right);
    }

    MPI_Bcast(&left, 1, MPI_INT, 0, MPI_COMM_WORLD);
    MPI_Bcast(&right, 1, MPI_INT, 0, MPI_COMM_WORLD);
    if (right > left)
    {
        interval = (right - left + 1) / (size - 1);
        start = (rank - 1) * interval + left;
        end = start + interval - 1;
        if (rank == (size - 1))
        { /* for last block */
            end = right;
        }
        sum = 0; /*Sum locally on each proc*/
        for (number = start; number <= end; number++)
            sum = sum + number;

        /*send local sum to Master process*/
        printf("Rank %d: start %d, end %d, local sum %d \n", rank, start, end, sum);
        // MPI_Send(&sum, 1, MPI_INT, 0, 123, MPI_COMM_WORLD);
        MPI_Reduce(&sum, &GrandTotal, 1, MPI_INT, MPI_SUM, 0, MPI_COMM_WORLD);
        if (rank == 0)
        {
            printf("Grand total = %d \n", GrandTotal);
        }
    }
    MPI_Finalize();
}
```

Result:

```
[u6388014@cluster ~]$ mpicc -o integersum integersum.c
[u6388014@cluster ~]$ mpirun -np 5 ./integersum
Enter the number for LEFT interval: 1
Enter the number for RIGHT interval: 100
Rank 1: start 1, end 25, local sum 325
Rank 0: start -24, end 0, local sum -300
Rank 3: start 51, end 75, local sum 1575
Rank 4: start 76, end 100, local sum 2200
Rank 2: start 26, end 50, local sum 950
Grand total = 4750
```