CPSC 479 Introduction To High Performance Computing

Group Fly Project 1 - Report

Group Member

Dhyey Desai | CWID (885451609) | dhyeydesai@csu.fullerton.edu
Lency Lakhani | CWID (885196055) | lencylakhani@csu.fullerton.edu
Darshit Desai | CWID (885516997) | darshdesai05@csu.fullerton.edu

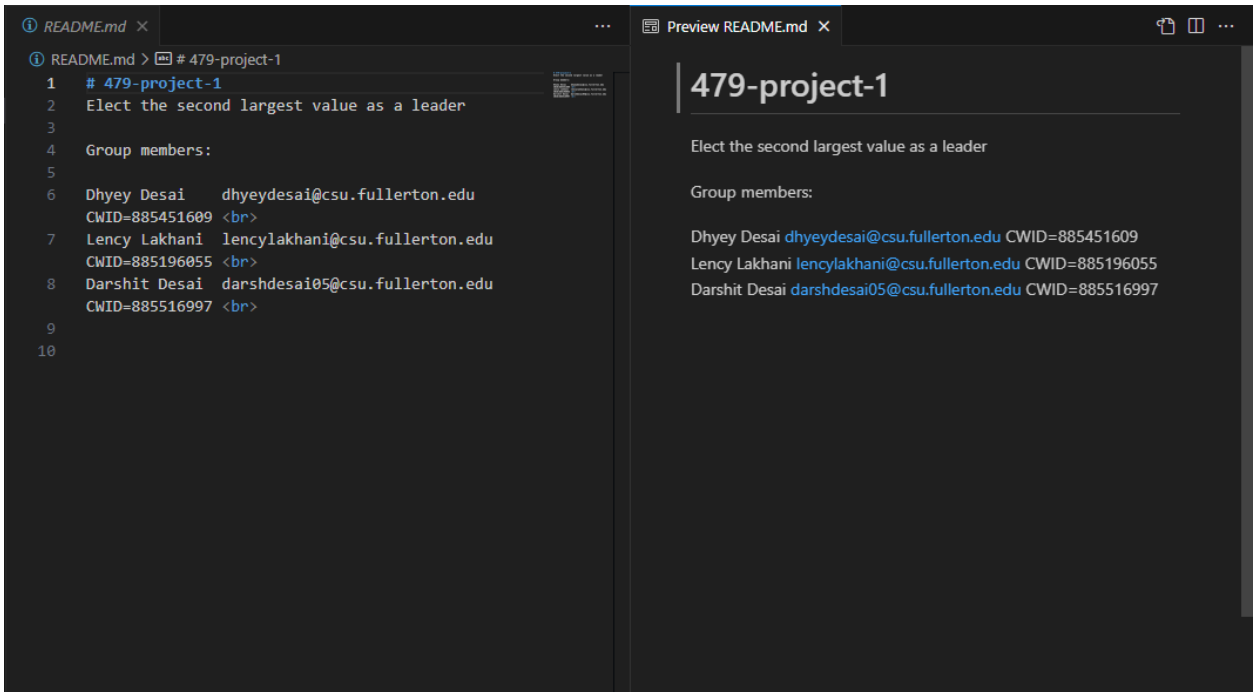
Professor Doina Bein

Department of Computer Science
California State University, Fullerton

Introduction:

In this project you will design and implement one algorithm related to leader election on a ring topology. You will design one algorithm, describe the algorithm using clear pseudocode and implement your algorithm using MPI primitives, compile, test it, and submit BOTH the report (as a PDF file) and the program. Let N be the size of the distributed system, i.e. run your program with N identical copies. The value of N must be greater than 5. You can choose the value of N to be between 6 and 20. **But each run of a program must terminate within 60 minutes if N=10**.

Team Members:



PseudoCode:

```
Initialize size, rank, leader_number
Initialize MPI with argc & argv
Get the current rank of the process to variable “rank”
Get the total number of processes to variable “size”

If size < 5 || size > 20:
    Finalize MPI
    Print an error message
    Exit

//generate random number for all processors
Create an array 'random' of size 'size'
If rank is 0
    For i from 0 to size-1
        Generate a random number between 10 and 99 and store it in random[i]

//distribute this array to all processor
Broadcast the 'random' array to all processes from rank 0

int 'd' =  random[rank] * 10 + (random[rank] % 2)
Int number =  d * 100 + rank
Print "Random number = number by <rank> rank"

Create an integer array 'A' of size 2, initialized with {0, 0}

If rank is 0
```

Set A[0] = ‘number'

Set A[1] = 0

MPI_send(A, 2, MPI_INT, 0, MPI_COMM_WORLD);

Print "Rank 0 sent the value = A[0] A[1] to rank 1"

Receive A[2] from (rank-1) % size

Print "Rank 0 received value = A[0] A[1]"

Else (for other ranks)

Receive array A from (rank-1) % size

Print "Rank rank has received the value = A[0] A[1] from rank (rank-1) % size"

//adjust highest two value into array A

If 'number' is greater than A[0]

Set A[1] to A[0]

Set A[0] to 'number'

Else if 'number' is greater than A[1] and less than A[0]

Set A[1] to 'number'

Send array A to (rank+1) % size

Print "Rank rank sent the value = A[0] A[1] to rank (rank+1) % size"

//second ring communication for identify leader

For the second communication

If rank is 0

Set 'leader_number' = A[1]

```
    If 'leader_number' is equal to 'number'
        Print "Rank 0 is the leader with number = leader_number"

    Send 'leader_number' to rank+1

    Receive 'leader_number' from (rank-1) % size
Else (for other ranks)
    Receive 'leader_number' from (rank-1) % size

   //check if current process is leader or not
    If 'leader_number' is equal to 'number'
        Print "Rank rank is the leader with number = leader_number"

    Send 'leader_number' to (rank+1) % size

Finalize MPI
Exit
```

Output:

1) N = 16

```
dhyeydesai@titanv1:~/project1$ mpirun -n 16 project1
Random number= 12008 by rank 8
Random number= 67109 by rank 9
Random number= 26010 by rank 10
Random number= 81100 by rank 0
Rank 0 sent the value= 81100 0 to rank 1.
Random number= 93102 by rank 2
Random number= 73104 by rank 4
Random number= 10005 by rank 5
Random number= 14006 by rank 6
Random number= 40007 by rank 7
Random number= 10012 by rank 12
Random number= 84013 by rank 13
Random number= 16014 by rank 14
Random number= 11115 by rank 15
Random number= 52001 by rank 1
Rank1 has received the value= 81100 0 from rank 0
Rank 1 sent the value= 81100 52001 to rank 2
Random number= 64003 by rank 3
Rank3 has received the value= 93102 81100 from rank 2
Rank 3 sent the value= 93102 81100 to rank 4
Rank2 has received the value= 81100 52001 from rank 1
Rank 2 sent the value= 93102 81100 to rank 3
Random number= 35111 by rank 11
Rank4 has received the value= 93102 81100 from rank 3
Rank 4 sent the value= 93102 81100 to rank 5
Rank5 has received the value= 93102 81100 from rank 4
Rank 5 sent the value= 93102 81100 to rank 6
Rank6 has received the value= 93102 81100 from rank 5
Rank 6 sent the value= 93102 81100 to rank 7
Rank7 has received the value= 93102 81100 from rank 6
Rank 7 sent the value= 93102 81100 to rank 8
Rank8 has received the value= 93102 81100 from rank 7
Rank 8 sent the value= 93102 81100 to rank 9
Rank9 has received the value= 93102 81100 from rank 8
Rank 9 sent the value= 93102 81100 to rank 10
Rank11 has received the value= 93102 81100 from rank 10
Rank10 has received the value= 93102 81100 from rank 9
Rank 10 sent the value= 93102 81100 to rank 11
Rank 11 sent the value= 93102 81100 to rank 12
Rank12 has received the value= 93102 81100 from rank 11
Rank 12 sent the value= 93102 81100 to rank 13
Rank13 has received the value= 93102 81100 from rank 12
Rank 13 sent the value= 93102 84013 to rank 14
Rank14 has received the value= 93102 84013 from rank 13
Rank 14 sent the value= 93102 84013 to rank 15
Rank15 has received the value= 93102 84013 from rank 14
```

```
Rank1 has received the value= 81100 0 from rank 0
Rank 1 sent the value= 81100 52001 to rank 2
Random number= 64003 by rank 3
Rank3 has received the value= 93102 81100 from rank 2
Rank 3 sent the value= 93102 81100 to rank 4
Rank2 has received the value= 81100 52001 from rank 1
Rank 2 sent the value= 93102 81100 to rank 3
Random number= 35111 by rank 11
Rank4 has received the value= 93102 81100 from rank 3
Rank 4 sent the value= 93102 81100 to rank 5
Rank5 has received the value= 93102 81100 from rank 4
Rank 5 sent the value= 93102 81100 to rank 6
Rank6 has received the value= 93102 81100 from rank 5
Rank 6 sent the value= 93102 81100 to rank 7
Rank7 has received the value= 93102 81100 from rank 6
Rank 7 sent the value= 93102 81100 to rank 8
Rank8 has received the value= 93102 81100 from rank 7
Rank 8 sent the value= 93102 81100 to rank 9
Rank9 has received the value= 93102 81100 from rank 8
Rank 9 sent the value= 93102 81100 to rank 10
Rank11 has received the value= 93102 81100 from rank 10
Rank10 has received the value= 93102 81100 from rank 9
Rank 10 sent the value= 93102 81100 to rank 11
Rank 11 sent the value= 93102 81100 to rank 12
Rank12 has received the value= 93102 81100 from rank 11
Rank 12 sent the value= 93102 81100 to rank 13
Rank13 has received the value= 93102 81100 from rank 12
Rank 13 sent the value= 93102 84013 to rank 14
Rank14 has received the value= 93102 84013 from rank 13
Rank 14 sent the value= 93102 84013 to rank 15
Rank15 has received the value= 93102 84013 from rank 14
Rank 15 sent the value= 93102 84013 to rank 0

Rank 0 received value= 93102 84013

Rank 13 is the leader with number= 84013
dhyeydesai@titanv1:~/project1$
```

2) N = 6

```
dhyeydesai@titanv1:~/project1$ mpic++ project1.cc -o project1
dhyeydesai@titanv1:~/project1$ mpirun -n 6 project1
Random number= 21100 by rank 0
Rank 0 sent the value= 21100 0 to rank 1.
Random number= 50001 by rank 1
Rank1 has received the value= 21100 0 from rank 0
Rank 1 sent the value= 50001 21100 to rank 2
Random number= 40002 by rank 2
Rank2 has received the value= 50001 21100 from rank 1
Rank 2 sent the value= 50001 40002 to rank 3
Random number= 54003 by rank 3
Rank3 has received the value= 50001 40002 from rank 2
Rank 3 sent the value= 54003 50001 to rank 4
Random number= 53104 by rank 4
Rank4 has received the value= 54003 50001 from rank 3
Rank 4 sent the value= 54003 53104 to rank 5
Random number= 17105 by rank 5
Rank5 has received the value= 54003 53104 from rank 4
Rank 5 sent the value= 54003 53104 to rank 0

Rank 0 received value= 54003 53104

Rank 4 is the leader with number= 53104
dhyeydesai@titanv1:~/project1$
```

How To Run Project

1. Download project.cc file from github
2. In terminal, write ***mpic++ project1.cc -o project1***
3. Then to execute code, ***mpirun -n 10***<*no of processor*> ***project1***