
PARALLEL AND DISTRIBUTED COMPUTING

ASSIGNMENT 1

SUBMITTED BY:

SANA ALI KHAN

18I-0439

CS-D

SUBMITTED TO:

SIR EHTESHAM ZAHOOR

Step-wise Implementation

For this assignment, I used primarily MPI calls with some openMP functionality. From MPI, I used synchronous send and recv, and from openMP I used the barrier and critical clauses. The main purpose of the assignment is to speed up the searching process; to that end, there are a number of processes created, with each being assigned a fixed, equal amount of data to search through.

The master process initializes the data, distributes it, and receives updates from slave processes. Upon a slave finding the number, it informs master and master informs all other slaves to abort search and terminate. Following is the sequence of steps and their screenshots:

- Master process (of rank 0) initializes and fills an array of numbers, the size being (constant number * no of slave processes.) The constant number is the amount of data each process will be given, and is fixed in the program to ensure equal distribution. The array values are generated randomly, range 0-100.
- Iterating through the list, the master process sends a chunk of the array to each slave process. Alongside the array chunk, it also sends the number-to-be-searched. (Done using `MPI_Send`)

```
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop$ mptexec -n 10 ./a1
There are 9 slave processes, each will have 3 numbers
Process of rank 0 has input data: 70 94 26 58 2 82 19 81 77 78 32 25 34 26 36 8 40 99 9 10 71 39 80 40 83 41 37
Enter number to search for: 81

Process of rank 2 has input data: 58 2 82
Process of rank 8 has input data: 39 80 40
Process of rank 5 has input data: 34 26 36
Process of rank 9 has input data: 83 41 37
Process of rank 3 has input data: 19 81 77
Process of rank 1 has input data: 70 94 26
Process of rank 7 has input data: 9 10 71
Process of rank 6 has input data: 8 40 99
Process of rank 4 has input data: 78 32 25
```

Figure 1

- As shown in Figure 1, the array of 27 ($3 * 9$) numbers is divided equally, with every slave process receiving 3 numbers. (Done using `MPI_Recv`)

- Every slave process has the same code. They each iterate through their own array; for every iteration, `array[i]` is compared to the number-to-be-searched – whether they are same or not, the slave informs the master. Different tags allow the master to know whether the slave is reporting that it has found the number (`tag = 2`), or has not found the number yet (`tag = 3`), or has not found the number at all (`tag = 4`)
- During this, the master is constantly receiving updates from every slave process as to whether they have found the number or not. Iterating through the processes, it first sends a signal to every slave (`tag = 0` indicates to slave that they may continue their search.) Then master waits for every slave to report back on the result of their one iteration. As the `MPI_Recv` call in master is set to receive message from any source, therefore there is no restriction that slave processes must report in according to rank – whichever finishes one iteration first will send message to master first.
- If the message master has received has `tag = 2`, this means a slave has found the number. The master will immediately send an empty message to all slaves (with `tag = 1`, indicating that slave must abort search).

```

sana@sanakhan: ~/Desktop/PDC A1
sana@sanakhan:~/Desktop/PDC A1$ mpiCC -fopenmp a1.cpp -o a1
sana@sanakhan:~/Desktop/PDC A1$ mpiexec -n 3 ./a1
There are 2  slave processes, each will have 3 numbers

Process of rank 0 has input data: 9 45 42 75 82 62
Enter number to search for: 75

Process of rank 1 has input data: 9 45 42
Process of rank 2 has input data: 75 82 62

Process 2: Number found at index 0 —————→ 1
Master: Process 2 has found the number. —————→ 2
Master: Signaling all processes to abort! —————→ 3

Process 1: Aborting search!
Process 2: Aborting search!
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ |

```

Figure 2

- As shown in Figure 2, once a slave has found the number (1) and reported to master (2), the master process signals all slaves to abort search and terminate (3).

- Slaves receive an empty message from master at the start of every iteration. The tag indicates whether to continue search or abort search (tag = 0 for continue and tag = 1 for abort.)
- If the slaves are still searching, upon receiving the abort message from master, they will quit searching, print abort message and terminate naturally (shown in Figure 3.)

```
sana@sanakhan:~/Desktop/PDC A1$ mplexec -n 7 ./a1
There are 6 slave processes, each will have 3 numbers

Process of rank 0 has input data: 77 48 91 97 10 39 5 2 51 68 19 43 86 20 91 39 59 48
Enter number to search for: 91

Process of rank 1 has input data: 77 48 91
Process of rank 2 has input data: 97 10 39
Process of rank 6 has input data: 39 59 48
Process of rank 3 has input data: 5 2 51
Process of rank 4 has input data: 68 19 43
Process of rank 5 has input data: 86 20 91

Process 1: Number found at index 2
Master: Process 1 has found the number.
Master: Signaling all processes to abort!

Process 5: Aborting search!
Process 4: Aborting search!
Process 3: Aborting search!
Process 2: Aborting search!
Process 6: Aborting search!
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ |
```

Figure 3

- In the case that a slave has gone through its array and not found the number it will report to master (tag = 4). If all slaves report this message, then master concludes that the number is not in the array, prints a message for user and exits (shown in Figure 4.)

```
sana@sanakhan:~/Desktop/PDC A1$ mpicc -fopenmp a1.cpp -o a1
sana@sanakhan:~/Desktop/PDC A1$ mplexec -n 6 ./a1
There are 5 slave processes, each will have 3 numbers

Process of rank 0 has input data: 44 89 43 18 80 35 38 18 81 31 70 90 96 26 88
Enter number to search for: 0

Process of rank 2 has input data: 18 80 35
Process of rank 4 has input data: 31 70 90
Process of rank 3 has input data: 38 18 81
Process of rank 5 has input data: 96 26 88
Process of rank 1 has input data: 44 89 43
Process 1: Finished search, number not found.
Process 2: Finished search, number not found.
Process 3: Finished search, number not found.
Process 4: Finished search, number not found.
Process 5: Finished search, number not found.

Master: All processes have finished searching and have not found the given number in the array.
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ |
```

Figure 4

Outputs

Following are screenshots of outputs with different number of processes:

1. 7 processes (6 slaves, each with 3 numbers) :

```
sana@sanakhan: ~/Desktop/PDC A1
There are 7 slave processes, each will have 3 numbers

Process of rank 0 has input data: 83 57 32 33 65 69 60 40 30 69 17 17 0 29 71 89 51 36 47 41 82
Enter number to search for: 0

Process of rank 7 has input data: 47 41 82
Process of rank 5 has input data: 0 29 71
Process of rank 2 has input data: 33 65 69
Process of rank 1 has input data: 83 57 32
Process of rank 6 has input data: 89 51 36
Process of rank 3 has input data: 60 40 30
Process of rank 4 has input data: 69 17 17

Process 5: Number found at index 0

Master: Process 5 has found the number.
Master: Signaling all processes to abort!

Process 6: Aborting search!
Process 2: Aborting search!
Process 3: Aborting search!
Process 4: Aborting search!
Process 7: Aborting search!
Process 1: Aborting search!
Process 5: Aborting search!
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$
```

2. 5 processes (4 slaves, each with 4 numbers)

```
sana@sanakhan: ~/Desktop/PDC A1
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ mpiCC -fopenmp a1.cpp -o a1
sana@sanakhan:~/Desktop/PDC A1$ mplexec -n 5 ./a1
There are 4 slave processes, each will have 4 numbers

Process of rank 0 has input data: 47 38 10 25 70 21 58 19 90 91 78 22 60 49 32 8
Enter number to search for: 70

Process of rank 2 has input data: 70 21 58 19
Process of rank 3 has input data: 90 91 78 22
Process of rank 4 has input data: 60 49 32 8
Process of rank 1 has input data: 47 38 10 25

Process 2: Number found at index 0

Master: Process 2 has found the number.
Master: Signaling all processes to abort!

Process 4: Aborting search!
Process 3: Aborting search!
Process 2: Aborting search!
Process 1: Aborting search!
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$
```

3. 20 processes (19 slaves, each with 3 numbers)

```
sana@sanakhan: ~/Desktop/PDC A1
There are 20 slave processes, each with 3 numbers.

Process of rank 0 has input data: 91 96 52 95 98 13 20 31 20 9 28 2 26 85 94 64 99 39 78 8 26 83 33 10 77 20
19 81 52 85 83 44 33 36 39 31 1 59 15 74 69 95 76 47 80 70 11 80 61 89 88 88 24 73 50 53 45
Enter number to search for: 1

Process of rank 1 has input data: 91 96 52
Process of rank 4 has input data: 9 28 2
Process of rank 5 has input data: 26 85 94
Process of rank 13 has input data: 1 59 15
Process of rank 15 has input data: 76 47 80
Process of rank 17 has input data: 61 89 88
Process of rank 19 has input data: 50 53 45
Process of rank 2 has input data: 95 98 13
Process of rank 3 has input data: 20 31 20
Process of rank 6 has input data: 64 99 39
Process of rank 7 has input data: 78 8 26
Process of rank 8 has input data: 83 33 10
Process of rank 9 has input data: 77 20 19
Process of rank 10 has input data: 81 52 85
Process of rank 11 has input data: 83 44 33
Process of rank 12 has input data: 36 39 31
Process of rank 14 has input data: 74 69 95
Process of rank 16 has input data: 70 11 80
Process of rank 18 has input data: 88 24 73

Process 13: Number found at index 0

Master: Process 13 has found the number.
Master: Signaling all processes to abort!

-----
Sana Ali Khan
18i-0439
Number found at index " << i << endl;
-----
Process 2: Aborting search!
Process 3: Aborting search!
Process 4: Aborting search!
Process 7: Aborting search!
Process 8: Aborting search!
Process 9: Aborting search!
Process 11: Aborting search!
Process 13: Aborting search!
Process 14: Aborting search!
Process 16: Aborting search!
Process 17: Aborting search!
Process 19: Aborting search!
Process 1: Aborting search!
Process 5: Aborting search!
Process 6: Aborting search!
```

4. 7 processes (6 slaves, each with 4 numbers)

```
sana@sanakhan: ~/Desktop/PDC A1
sana@sanakhan:~/Desktop/PDC A1$ mpiexec -n 7 ./a1
There are 6 slave processes, each will have 4 numbers

Process of rank 0 has input data: 13 56 96 14 90 65 79 73 99 79 83 67 45 16 99 32 80 23 51 77 41 87 63 75
Enter number to search for: -1

Process of rank 5 has input data: 80 23 51 77
Process of rank 2 has input data: 90 65 79 73
Process of rank 4 has input data: 45 16 99 32
Process of rank 6 has input data: 41 87 63 75
Process of rank 1 has input data: 13 56 96 14
Process of rank 3 has input data: 99 79 83 67
Process 1: Finished search, number not found.
Process 2: Finished search, number not found.
Process 3: Finished search, number not found.
Process 5: Finished search, number not found.
Process 4: Finished search, number not found.
Process 6: Finished search, number not found.

Master: All processes have finished searching and have not found the given number in the array.
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ |
```

5. 10 processes (9 slaves, each with 4 numbers)

```
sana@sanakhan: ~/Desktop/PDC A1
Process of rank 0 has input data: 51 51 53 22 93 5 71 56 77 66 62 39 78 74 92 47 91 74 8 41 23 77 70 43 40 31 9
8 87 3 99 88 6 2 94 28 95
Enter number to search for: 3

Process of rank 6 has input data: 23 77 70 43
Process of rank 3 has input data: 77 66 62 39
Process of rank 8 has input data: 3 99 88 6
Process of rank 1 has input data: 51 51 53 22
Process of rank 9 has input data: 2 94 28 95
Process of rank 5 has input data: 91 74 8 41
Process of rank 7 has input data: 40 31 98 87
Process of rank 2 has input data: 93 5 71 56
Process of rank 4 has input data: 78 74 92 47

Process 8: Number found at index 0

Master: Process 8 has found the number.
Master: Signaling all processes to abort!

Process 9: Aborting search!
Process 1: Aborting search!
Process 3: Aborting search!
Process 2: Aborting search!
Process 4: Aborting search!
Process 7: Aborting search!
Process 8: Aborting search!
Process 5: Aborting search!
Process 6: Aborting search!
-----
Sana Ali Khan
18i-0439
-----
sana@sanakhan:~/Desktop/PDC A1$ |
```


6. 10 processes (9 slaves, each with 10 numbers)

```
[1]+ Stopped (ault) << mpirun -n 10 ./a1 rank << " has input data: "
sana@sanakhan:~/Desktop/PDC: A1$ mpirun -n 10 ./a1
There are 9 slave processes, each will have 10 numbers

Process of rank 0 has input data: 57 55 13 19 56 44 48 54 14 18 76 51 74 51 44 70 8 92 67 31 79 3 49 76 74 56 54 33 66 52 83 76 8 48 95 16 93 95 70 7 65 98 10 39 49 6 61 9 98 28 92 77 83 93 6 9 1 12 94 68
65 77 96 25 78 43 41 71 38 11 30 55 61 40 94 62 47 7 24 97 87 68 27 22 62 33 31 63 45 26
Enter number to search for: 22

Process of rank 2 has input data: 76 51 74 51 44 70 8 92 67 31
Process of rank 1 has input data: 57 55 13 19 56 44 48 54 14 18
Process of rank 7 has input data: 65 77 96 25 78 43 41 71 38 11
Process of rank 3 has input data: 79 3 49 76 74 56 54 33 66 52
Process of rank 4 has input data: 83 76 8 48 95 16 93 95 70 7
Process of rank 8 has input data: 30 55 61 40 94 62 47 7 24 97
Process of rank 9 has input data: 87 68 27 22 62 33 31 63 45 26
Process of rank 5 has input data: 65 98 10 39 49 6 61 9 98 28
Process of rank 6 has input data: 92 77 83 93 6 9 1 12 94 68

// Abort search
Process 9: Number found at index 3 TAG == 0) {
    if (for_recv[i] == to_find) {
Master: Process 9 has found the number.
Master: Signaling all processes to abort!
// indicates that this slave did manage to find the number.
-----
Sana Ali Khan
18i-0439
// signal master that number has been found
-----, 1, MPI_INT, 0, 2, MPI_COMM_WORLD));
Process 3: Aborting search!
Process 4: Aborting search!
Process 6: Aborting search!
Process 1: Aborting search! // signal master that number has not been found (yet)
Process 2: Aborting search! MPI_Send(&buf, 1, MPI_INT, 0, 3, MPI_COMM_WORLD);
Process 5: Aborting search!
Process 7: Aborting search!
Process 8: Aborting search!
Process 9: Aborting search! to terminate search and abort.
sana@sanakhan:~/Desktop/PDC: A1$ |== MPI_TAG == 11 f
```