

Code:

1. ScatterGather.java

J ScatterGather.java 9+ ●

J ScatterGather.java > ScatterGather > main(String[])

```
1  import mpi.MPI;
2
3  import java.util.Scanner;
4
5  import mpi.*;
6
7  public class ScatterGather {
    Run | Debug
8      public static void main(String[] args) throws Exception{
9          MPI.Init(args);
10         // Get rank of each process and size of communicator
11         int rank = MPI.COMM_WORLD.Rank();
12         int size = MPI.COMM_WORLD.Size();
13
14         int unitsize = 5;
15         int root = 0;
16         int send_buffer[] = null;
17         // 1 process is expected to handle 4 elements
18         send_buffer = new int [unitsize * size];
19         int recieve_buffer[] = new int [unitsize];
20         int new_recieve_buffer[] = new int [size];
21
22         // Set data for distribution
23         if(rank == root) {
24             int total_elements = unitsize * size;
25             System.out.println("Enter " + total_elements + " elements");
26             for(int i = 0; i < total_elements; i++) {
27                 System.out.println("Element " + i + "\t = " + i);
28                 send_buffer[i] = i;
29             }
30         }
31     }
```

J ScatterGather.java 9+ ×

J ScatterGather.java > ScatterGather > main(String[])

```
32 // Scatter data to processes
33 MPI.COMM_WORLD.Scatter(
34     send_buffer,
35     0,
36     unitsize,
37     MPI.INT,
38     recieve_buffer,
39     0,
40     unitsize,
41     MPI.INT,
42     root
43 );
44
45 // Calculate sum at non root processes
46 // Store result in first index of array
47 for(int i = 1; i < unitsize; i++) {
48     recieve_buffer[0] += recieve_buffer[i];
49 }
50 System.out.println(
51     "Intermediate sum at process " + rank + " is " + recieve_buffer[0]
52 );
53
54
55 // Gather data from processes
56 MPI.COMM_WORLD.Gather(
57     recieve_buffer,
58     0,
59     1,
60     MPI.INT,
61     new_recieve_buffer,
62     0,
```

```
63     1,
64     MPI.INT,
65     root
66 );
67
68 // Aggregate output from all non root processes
69 if(rank == root) {
70     int total_sum = 0;
71     for(int i = 0; i < size; i++) {
72         total_sum += new_recieve_buffer[i];
73     }
74     System.out.println("Final sum : " + total_sum);
75 }
76
77 MPI.Finalize();
78 }
79 }
```

Output:

1. Compilation and Execution

```
varadmash@varadmash-G3-3590: ~/LP5_lab/Assignment3
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment3$ javac -cp $MPJ_HOME/lib/mpj.jar ScatterGather.java
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment3$ $MPJ_HOME/bin/mpjrun.sh -np 4 ScatterGather
MPJ Express (0.44) is started in the multicore configuration
Enter 20 elements
Element 0      = 0
Element 1      = 1
Element 2      = 2
Element 3      = 3
Element 4      = 4
Element 5      = 5
Element 6      = 6
Element 7      = 7
Element 8      = 8
Element 9      = 9
Element 10     = 10
Element 11     = 11
Element 12     = 12
Element 13     = 13
Element 14     = 14
Element 15     = 15
Element 16     = 16
Element 17     = 17
Element 18     = 18
Element 19     = 19
Intermediate sum at process 0 is 10
Intermediate sum at process 1 is 35
Intermediate sum at process 3 is 85
Intermediate sum at process 2 is 60
Final sum : 190
```