

```

1  //-----
2  // topology 2-dimensional
3  //-----
4  //
5  //          0 -> 4 -> 8 -> 12      Sum
6  //          |   |   |   |
7  //          1 -> 5 -> 9 -> 13      24
8  //          |   |   |   |
9  //          2 -> 6 -> 10 -> 14     28
10 //          |   |   |   |
11 //          3 -> 7 -> 11 -> 15     32
12 //
13 // Sum   6    22   38    54
14 //
15 #if (defined __linux__ ) || (defined _AIX) || (defined __APPLE__)
16 #include <sys/types.h>
17 #include <sys/stat.h>
18 #include <unistd.h>
19 #elif (defined _WIN32) || (defined _WIN64)
20 #include <conio.h>
21 #include <direct.h>
22 #endif
23
24 #include<mpi.h>
25 #include<stdlib.h>
26 #include<stdio.h>
27 #include <iostream>
28 using namespace std;
29
30 int main(int argc, char** argv)
31 {
32     int i, rank, size, src, dest, direction = 0, mess1 = 0, mess2 = 0, mysumm = 0;
33     int dims[2], periods[2];
34     double time;
35     MPI_Request request;
36     MPI_Status status;
37     MPI_Comm mycomm_cart;
38
39     MPI_Init(&argc, &argv);
40     MPI_Barrier(MPI_COMM_WORLD);
41     time = -MPI_Wtime();
42
43     MPI_Comm_size(MPI_COMM_WORLD, &size);
44     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
45     if (rank == 0) {
46         printf("\nThere are %d processes.", size);
47         fflush(stdout);
48     }
49
50     if (size != 16) {
51         if (rank == 0)
52             cout << "\n\nPlease, set the number of processes to 16!\n"; // size = 16
53         goto end;
54     }
55
56     dims[0] = size / 4;
57     dims[1] = 4;
58     periods[0] = 1;
59     periods[1] = 1;
60
61     MPI_Cart_create(MPI_COMM_WORLD, 2, dims, periods, true, &mycomm_cart);
62     mess1 = rank;
63
64     if (rank == 0) {
65         cout << "\n-----";
66         cout << "\nSelect a direction:\n 0. horizontal\n";
67         cout << " 1. vertical\n\n direction = ";
68         cin >> direction;
69         cout << "-----\n";
70         fflush(stdout);
71     }
72     MPI_Bcast(&direction, 1, MPI_INT, 0, MPI_COMM_WORLD);

```

```
73
74     for (i = 0; i < size / 4; i++) {
75         MPI_Cart_shift(mycomm_cart, direction, 1, &src, &dest);
76         printf("%d -> %d", src, rank);
77         fflush(stdout);
78         MPI_Issend(&mess1, 1, MPI_INT, dest, 0, mycomm_cart, &request);
79         MPI_Recv(&mess2, 1, MPI_INT, src, 0, mycomm_cart, &status);
80         MPI_Wait(&request, &status);
81         mysumm += mess2;
82         mess1 = mess2;
83     }
84
85     printf("process %d: summ of messages = %d", rank, mysumm);
86
87 end:
88     time += MPI_Wtime();
89     if (rank == 0)
90         printf("\nTime = %10.6f s\n", time);
91
92     fflush(stdout);
93     MPI_Finalize();
94
95     return 0;
96 }
97
```