

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF MATHEMATICS AND
INFORMATION SCIENCE

FIRST SEMESTER 2021/2022
HIGH PERFORMANCE COMPUTING (HPC)

PROJECT 2

BY:

ANDRA UMORU (324334)

SUBMITTED TO:

OKULICKA-DLUZEWSKA FELICJA

TABLE OF CONTENT

TABLE OF CONTENT	2
PROJECT TWO	3
1.1 PROJECT DESCRIPTION.....	3
1.2 OVERVIEW OF VIRTUAL TOPOLOGY	3
1.3 MPI CARTESIAN TOPOLOGY	3
1.3.1 MPI Cartesian Rank.....	3
1.3.2 MPI Cartesian Shift.....	4
1.4 THE SOURCE CODE	4
1.5 RESULT (OUTPUT).....	6
1.6 REFERENCE	7
FIGURE 1B: THE RESULT	6
FIGURE 2A: THE RESULT	6
FIGURE 3C: THE RESULT	7

PROJECT TWO

1.1 Project description

Create the virtual topology with at least 8 nodes and for each process find its neighbor's:

Student 12: Cartesian Size 4 x 4 non-periodic.

1.2 Overview of Virtual Topology

Virtual topology can allow MPI to optimize communications by creating scheme fitting the communication pattern. there may be no relation between the physical structure of the parallel machine and the process topology. The creating a topology produces a new communicator with new interior ranks. Topology types are graph topologies and Cartesian topologies.

1.3 MPI Cartesian Topology

Each process is 'connected' to its neighbors in a virtual grid. Boundaries can be cyclic , or not. Processes are identified by the Cartesian coordinates.

To create a cartesian topology, the following is the function that does that:

```
int MPI_Cart_create (MPI_Comm comm_old,  
int ndims,  
int *dims,  
int *period,  
int reorder,  
MPI_Comm *comm_cart)
```

For my project two, ndims = 2

dims = [4, 4]

per [0, 0] – non – periodic

1.3.1 MPI Cartesian Rank

Mapping process grid coordinates to ranks, it has the following syntax:

```
int MPI_Cart_rank (MPI_Comm comm,  
int *coords,  
int *rank)
```

1.3.2 MPI Cartesian Shift

This cartesian functionality is used in computing ranks of neighboring processes. It has the following syntax in C programming language:

```
int MPI_Cart_shift (MPI_Comm comm,  
int direction,  
int disp,  
int *rank_source,  
int *rank_dest)
```

1.4 The Source Code

The source code was written in C programming language. Below is the source code for project two:

```
#include <stdio.h>    //including the C standard library  
#include <mpi.h>      //Including the MPI library  
  
// Cartesian topology 4 x 4  
// 0 - 1 - 2 - 3  
// | | | |  
// 4 - 5 - 6 - 7  
// | | | |  
// 8 - 9 - 10 - 11  
// | | | |  
// 12 - 13 - 14 - 15  
  
// C Language main function  
int main(int argc, char **argv)  
{  
  
    //Declaring the size and the rank variables  
    int size,rank;  
  
    //Initializing the MPI Environment  
    MPI_Init(&argc,&argv);  
  
    // Getting number of process  
    MPI_Comm_size(MPI_COMM_WORLD,&size);  
  
    // Getting rank of the process  
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);  
  
    //Getting the number of dimensions
```

```

int dim [2]={4,4};

//Getting the period --> [0, 0] means non - periodic
int per [2]={0,0};
MPI_Comm com;

// Creating the cartesian topology
MPI_Cart_create(MPI_COMM_WORLD,2,dim,per,1,&com);

//Mapping the neighbours in the cartesian topology
int cord [2]={0,0};
int crank;
MPI_Cart_coords(com,rank,2,cord);

// Printing the cartesian topology together with the mappings
printf("I am %d and my coordinates are (%d,%d)\n", rank, cord[0],cord[1]);
printf(" my neighbours are:\n");
int c,d,neighbour;
for(c=0;c<2;c++)
{
for(d=-1;d<2;d++)
{ // shift left and right by 1 step (d=-1,d=1)
if (d!=0)
{
MPI_Cart_shift(com,c,d,&rank,&neighbour);
if (neighbour>=0)
printf(" neighbour [%d,%d] has rank %d\n",c,d,neighbour);
}
}
}

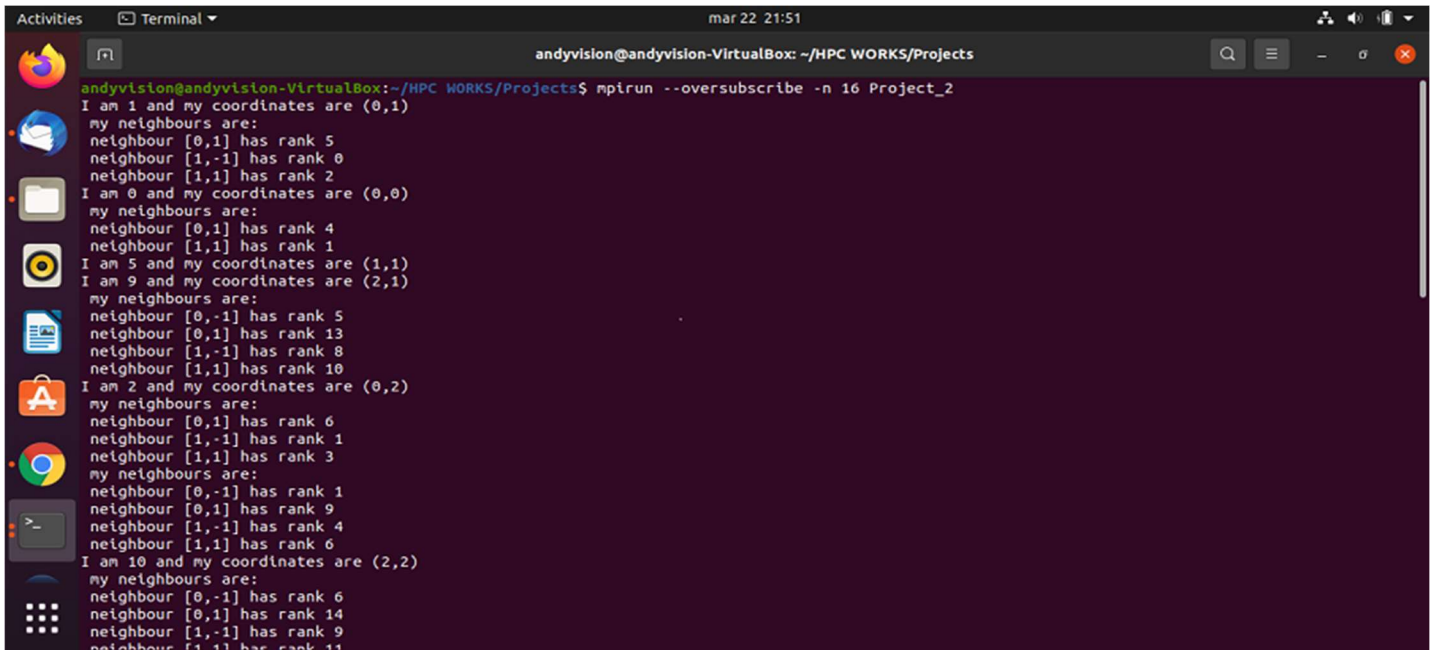
// Exiting or Terminating the cartesian MPI routine
MPI_Finalize();

return 0;
}

```

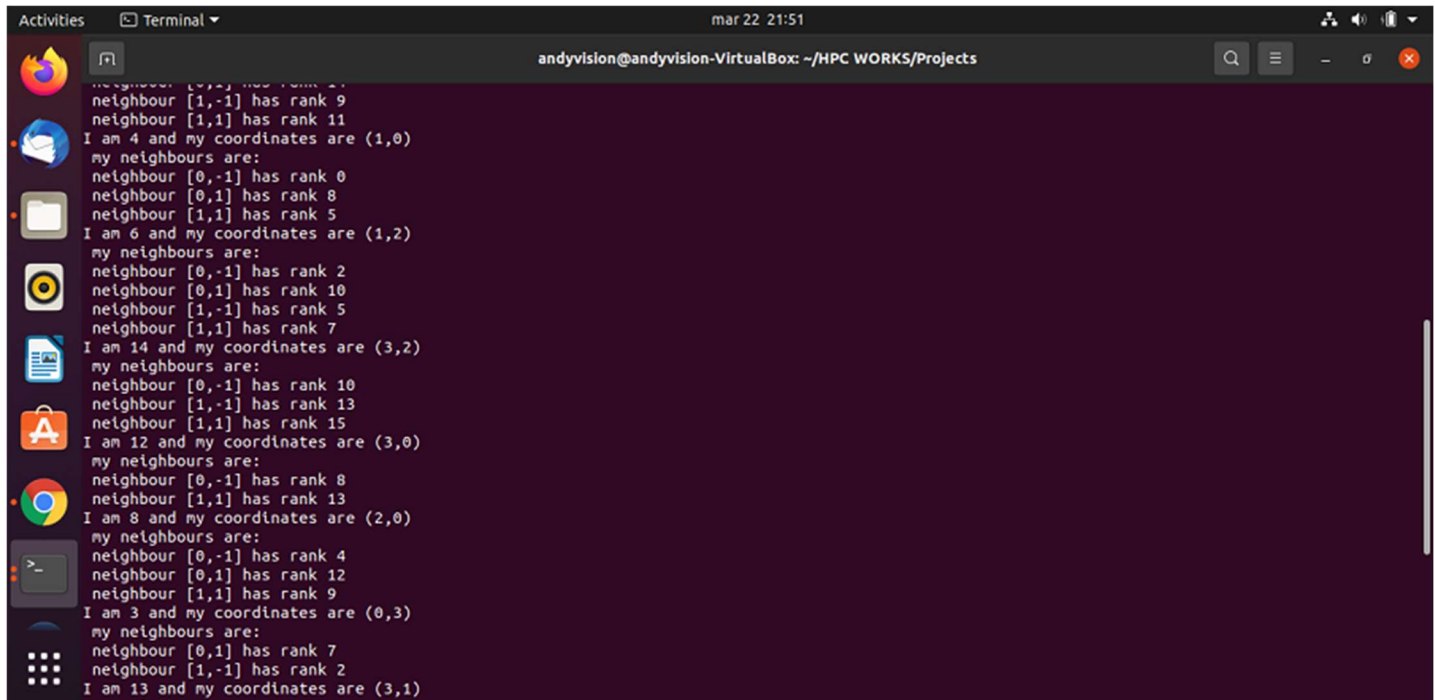
1.5 Result (Output)

After compiling and running the MPI cartesian program, the figures below displays is the output:



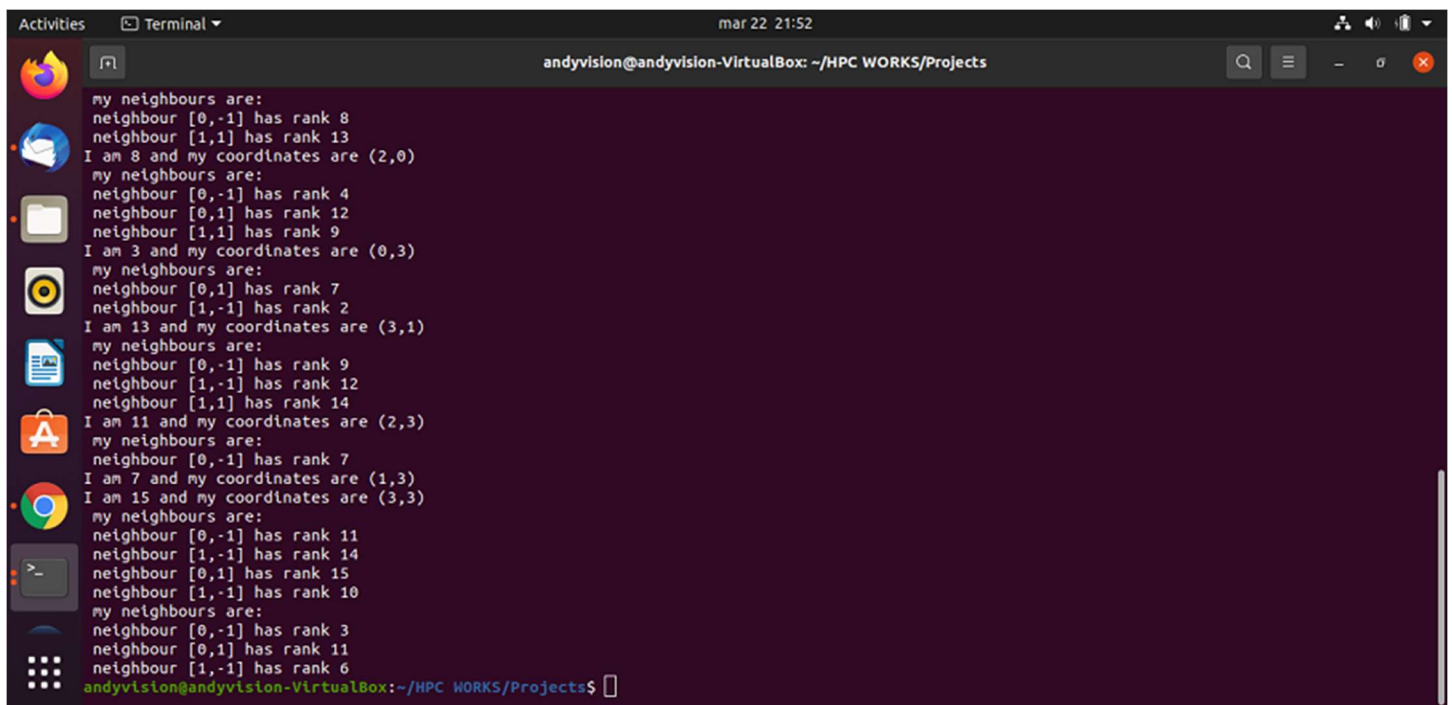
```
andyvision@andyvision-VirtualBox: ~/HPC WORKS/Projects
andyvision@andyvision-VirtualBox:~/HPC WORKS/Projects$ mpirun --oversubscribe -n 16 Project_2
I am 1 and my coordinates are (0,1)
my neighbours are:
neighbour [0,1] has rank 5
neighbour [1,-1] has rank 0
neighbour [1,1] has rank 2
I am 0 and my coordinates are (0,0)
my neighbours are:
neighbour [0,1] has rank 4
neighbour [1,1] has rank 1
I am 5 and my coordinates are (1,1)
I am 9 and my coordinates are (2,1)
my neighbours are:
neighbour [0,-1] has rank 5
neighbour [0,1] has rank 13
neighbour [1,-1] has rank 8
neighbour [1,1] has rank 10
I am 2 and my coordinates are (0,2)
my neighbours are:
neighbour [0,1] has rank 6
neighbour [1,-1] has rank 1
neighbour [1,1] has rank 3
my neighbours are:
neighbour [0,-1] has rank 1
neighbour [0,1] has rank 9
neighbour [1,-1] has rank 4
neighbour [1,1] has rank 6
I am 10 and my coordinates are (2,2)
my neighbours are:
neighbour [0,-1] has rank 6
neighbour [0,1] has rank 14
neighbour [1,-1] has rank 9
neighbour [1,1] has rank 11
```

Figure 2a: The result



```
neighbour [1,-1] has rank 9
neighbour [1,1] has rank 11
I am 4 and my coordinates are (1,0)
my neighbours are:
neighbour [0,-1] has rank 0
neighbour [0,1] has rank 8
neighbour [1,1] has rank 5
I am 6 and my coordinates are (1,2)
my neighbours are:
neighbour [0,-1] has rank 2
neighbour [0,1] has rank 10
neighbour [1,-1] has rank 5
neighbour [1,1] has rank 7
I am 14 and my coordinates are (3,2)
my neighbours are:
neighbour [0,-1] has rank 10
neighbour [1,-1] has rank 13
neighbour [1,1] has rank 15
I am 12 and my coordinates are (3,0)
my neighbours are:
neighbour [0,-1] has rank 8
neighbour [1,1] has rank 13
I am 8 and my coordinates are (2,0)
my neighbours are:
neighbour [0,-1] has rank 4
neighbour [0,1] has rank 12
neighbour [1,1] has rank 9
I am 3 and my coordinates are (0,3)
my neighbours are:
neighbour [0,1] has rank 7
neighbour [1,-1] has rank 2
I am 13 and my coordinates are (3,1)
```

Figure 1b: The result



The screenshot shows a terminal window titled "andyvision@andyvision-VirtualBox: ~/HPC WORKS/Projects". The terminal output displays a sequence of messages from an agent, likely representing a robot in a grid world. The messages are as follows:

```
my neighbours are:
neighbour [0,-1] has rank 8
neighbour [1,1] has rank 13
I am 8 and my coordinates are (2,0)
my neighbours are:
neighbour [0,-1] has rank 4
neighbour [0,1] has rank 12
neighbour [1,1] has rank 9
I am 3 and my coordinates are (0,3)
my neighbours are:
neighbour [0,1] has rank 7
neighbour [1,-1] has rank 2
I am 13 and my coordinates are (3,1)
my neighbours are:
neighbour [0,-1] has rank 9
neighbour [1,-1] has rank 12
neighbour [1,1] has rank 14
I am 11 and my coordinates are (2,3)
my neighbours are:
neighbour [0,-1] has rank 7
I am 7 and my coordinates are (1,3)
I am 15 and my coordinates are (3,3)
my neighbours are:
neighbour [0,-1] has rank 11
neighbour [1,-1] has rank 14
neighbour [0,1] has rank 15
neighbour [1,-1] has rank 10
my neighbours are:
neighbour [0,-1] has rank 3
neighbour [0,1] has rank 11
neighbour [1,-1] has rank 6
andyvision@andyvision-VirtualBox:~/HPC WORKS/Projects$
```

Figure 3c: The result

1.6 Reference

- HPC Lecture Notes