Getting started with parallel programming models

Horia Burca and Rodrigo Arias

Task 3.1

Check the loaded modules in your environment.

```
% module list
Currently Loaded Modules:
  1) intel/2017.4   2) impi/2017.4   3) mkl/2017.4   4) bsc/1.0
```

Task 3.2

Create, compile and run a Hello World program with MPI.

```
% cat mpi_helloworld.c
#include <mpi.h>
#include <stdio.h>

int main (int argc, char **argv)
{
    int size, rank;
    MPI_Init (&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    printf ( "I am %d of %d\n", rank, size );
    MPI_Finalize();

    return 0;
}

% mpicc mpi_helloworld.c -o mpi_helloworld

% mpirun ./mpi_helloworld
I am 0 of 48
I am 4 of 48
I am 15 of 48
I am 27 of 48
I am 28 of 48
I am 37 of 48
I am 38 of 48
I am 40 of 48
I am 41 of 48
I am 42 of 48
I am 46 of 48
I am 1 of 48
I am 2 of 48
...
I am 36 of 48
I am 20 of 48
I am 24 of 48
```

Task 3.3

Submit your "Hello World" program.

The job script:

```
% cat job
#!/bin/bash
#SBATCH --job-name="MPI_HelloWorld"
#SBATCH --workdir=.
#SBATCH --output=output_%J.out
#SBATCH --error=output_%J.err
#SBATCH --ntasks=16
#SBATCH --tasks-per-node=8
#SBATCH --time=00:01:00
#SBATCH --exclusive
#SBATCH --qos=debug

mpirun ./mpi_helloworld
```

When it's submitted:

```
% sbatch job

% squeue
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       2500123      main MPI_Hell sam14015 PD       0:00      2 (Priority)
% squeue
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       2500123      main MPI_Hell sam14015  R       0:05      2 s05r2b[64,66]
% squeue
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       2500123      main MPI_Hell sam14015 CG       0:10      2 s05r2b[64,66]
% squeue
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       2500123      main MPI_Hell sam14015 CG       0:10      1 s05r2b66
% squeue
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

Check the output of the execution. What happened with the order of outputs?

```
% cat output*.out
I am 1 of 16
I am 2 of 16
I am 5 of 16
I am 6 of 16
I am 7 of 16
I am 0 of 16
I am 3 of 16
I am 4 of 16
```

```
I am 10 of 16
I am 9 of 16
I am 8 of 16
I am 15 of 16
I am 12 of 16
I am 11 of 16
I am 14 of 16
I am 13 of 16
```

It's not sorted.

Task 3.4

Modify your solution that just prints a line of output from each process so that the output is printed in process rank order: process 0 output first, then process 1, and so on.

```
% cat mpi_helloworld.c
#include <mpi.h>
#include <stdio.h>

int main (int argc, char **argv)
{
    int size, rank, dummy=0;
    MPI_Init (&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);

    if (rank != 0)
    {
        MPI_Recv(&dummy, 1, MPI_INT, rank-1,
            0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    }

    printf("I am %d of %d\n", rank, size);

    if (rank < size-1)
    {
        MPI_Send(&dummy, 1, MPI_INT, rank+1,
            0, MPI_COMM_WORLD);
    }

    MPI_Finalize();

    return 0;
}

% mpicc mpi_helloworld.c -o mpi_helloworld

% mpirun ./mpi_helloworld
I am 0 of 48
I am 1 of 48
I am 2 of 48
I am 3 of 48
I am 4 of 48
```

```
...
I am 42 of 48
I am 43 of 48
I am 44 of 48
I am 45 of 48
I am 46 of 48
I am 47 of 48
```

Now is sorted.