

## Lab #7: MPI Basics

PCSE 2015

### Lab assignment

Starting from scratch, construct your own basic MPI program. Your program should incorporate the following MPI functions: `MPI_Init`, `MPI_Abort`, `MPI_Comm_size`, `MPI_Comm_rank`, `MPI_Get_processor_name`, `MPI_Wtime`, and `MPI_Finalize`. Your program should:

1. properly initialize and finalize the MPI environment;
2. should check to see if the initialization was successful, and, if not, then issue an error statement and abort;
3. ascertain the number of MPI tasks available within the program;
4. have each MPI task print its rank and processor/hostname;
5. call the `sleep(int seconds)` function (for C folks, use header file `unistd.h`) where the number of seconds of sleep is some function of rank;
6. call `MPI_Wtime` function before and after the call to `sleep` and store the results in two different double variables;
7. for each rank, print the rank and time elapsed between `MPI_Wtime` calls.

Now that you have a program, try compiling with the default compiler and MPI stack, that is, make sure the modules `intel/13.0.2.144` and `mvapich2/1.9a2` are loaded. Use `mpicc` or `mpif90` to then compile your program.

What happens when you use `icc` or `ifort` instead?

If you have an interactive development session running, end it. From there, use the syntax `idev -N 1 -n ``num_MPI_tasks``` and substitute ```num_MPI_tasks``` with 1, for now.

Once in the interactive session, run your program as just

`./a.out`

i.e. no `ibrun` or `mpirun`. What is your output?

Now, use the syntax

```
ibrun ./a.out
```

What's the output?

Now, quit your interactive development session. Start a new one with `num_MPI_tasks` set to 2. Repeat the runs

```
./a.out
```

```
ibrun ./a.out
```

What's changing in the output?

Quit your interactive development session again. Start a session with just `idev` and repeat the runs

```
./a.out
```

```
ibrun ./a.out
```

What's the output now? Got it?

Lastly, try asking for two nodes instead of one `idev -N 2 -n num_MPI_tasks` varying `num_MPI_tasks` with a value of 2, 32, and 64. Explain the differences in behavior/output between the three. What happened with 64 tasks?