Using MPI

MPI (“Message Passing Interface”) is a library of functions and definitions that allow processes on multiple processors to communicate with each other. The MPI API has been standardized, although there are several implementations of MPI; The version we have installed is called Open MPI; https://www.open-mpi.org.

MPI allows a user to create a single executable that can be distributed and executed simultaneously on a group of machines (sometimes called a “cluster”). The API provides functions that allow a node to uniquely identify itself within the cluster and to send and receive messages from other nodes. It also provides routines used to manage the system.

Initial MPI setup

The following section describes the setup necessary to use MPI on the CS MPI machines. This setup should only need to be done once. We will be using 6 Linux servers. Their names are cs-mpix.cs.uidaho.edu, where x ranges from 1 to 6. All the machines are identical however.

1. Log into any of the MPI servers with your University netID and password. If you are off campus, you will first need to connect to the University VPN then ssh into a server.
2. One file needs to be copied from the directory /var/opt/openmpi on any of the servers to your home directory

   ```
   cp /var/opt/openmpi/.mpi_hostfile $HOME
   ```

3. Due to Open MPI not being installed in the traditional path and a required library needed, users will need to update their .bashrc files or copy the sample .bashrc file into their home directory.

   Check for the existing .bashrc in home directory:

   ```
   less ~/.bashrc
   ```

   If it exists:

   Copy contents of /var/opt/openmpi/.bashrc to existing bashrc

   If does not exist:

   ```
   cp /var/opt/openmpi/.bashrc $HOME
   ```

   Optionally, you can also copy the example file cpi.c over to your home directory from the directory /var/opt/openmpi/examples on any of the servers:

   ```
   cp /var/opt/openmpi/examples/cpi.c $HOME
   ```

You should now be set up to run MPI programs!

Running an MPI Program

Each time you wish to run MPI, you will need to log in and do the following:

Compile and run your program. To aid in this process, our MPI implementation comes with a couple of scripts that serve as “wrappers” to the regular gcc/g++ compilers called mpicc and mpic++ - these add the necessary steps required to compile MPI programs.

To test it out, try running the test program cpi.c . The “-np 6” parameter is not required as the servers are single core only, np will declare total processes to run across all nodes. Do the following:

```
mpicc -o cpi cpi.c

mpiexec -np 6 --hostfile .mpi_hostfile ./cpi
```

You should get no errors during compilation. The output should look like the following:

```
cs-mpix:~$ mpiexec -np 6 --hostfile .mpi_hostfile ./cpi

        Process 0 on cs-mpi1

        Process 1 on cs-mpi2

        Process 4 on cs-mpi3

        Process 2 on cs-mpi4

        Process 5 on cs-mpi5

        Process 3 on cs-mpi6

cs-mpix:~$ pi is approximately 3.1416009869231249, Error is 0.0000083333333318 wall clock
time = 0.305170
```

Note that each message comes from a different machine, and they are not in any particular order.

If receiving any errors make sure that the .bashrc file is correctly loaded and that the .mpi hostfile only includes the correct server names, in the format cs-mpix.ad.uidaho.edu, where x is an integer from 1 to 6.