

High Performance Computing (CS6230)

This tutorial is written to help you setup your CADE lab machines to run a hybrid program using MPI and OpenMP.

Setting up your CADE machines to run MPICH-2:

1. ssh into your CADE account.
2. Create a file **“.mpd.conf ”** in your **“home/username/”** directory
(Please note that the file above is **“DOT mpd DOT conf”**)
3. In that file enter just a single line,
MPD_SECRETWORD=anything
4. Then type in **“chmod 600 ~/.mpd.conf”**
5. Now you are done with creating a mpd file which spawns processes. The next few steps are similar to how you had created the RSA key in Raven cluster. They are done so that, you can ssh into different nodes for running your jobs in parallel using MPI.
6. Type in **“ssh -keygen -t rsa”** in the terminal
7. For **Enter file in which to save the key (/home/user/.ssh/id_rsa):**, just press enter choosing the default value.
8. For **Enter passphrase (empty for no passphrase):** press enter so you don't need a password to connect to the other cluster nodes.
9. For **Enter same passphrase again:** press enter so you don't need a password to connect to the other cluster nodes.
10. At the prompt, type **mv ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys** which makes it so the public/private key pair is used (eliminating the need for a password when using ssh).
11. Now, MPI can ssh to different nodes without you having to type in password to access each node. But, if the nodes which you are going to connect to are not present in the **“~/.ssh/known_hosts”** file, then you need to type in **“yes”** (RSA fingerprint) for the first time you are going to connect to a node via MPI, as described later. That will be painful. Hence, we can make a handful of nodes as known_hosts so that, everything happens without bothering you.
12. In order to do that, run the script.sh given to you by typing **“sh script.sh”**. If you open the script, there are ssh commands to a bunch of 25 nodes such as,
“ssh username@lab3-5.eng.utah.edu”
Instead of lab3-x, you can use lab2-x or lab1-x
In order to know how many nodes are present in lab 1-x or 2-x or 3-x you can look up in the following link.
http://www.cade.utah.edu/index.php?module=faq&FAQ_op=view&FAQ_id=457

If you plan to use 30 nodes for your MPI programming, you need to have 30 nodes in the script so that you can make all the 30 nodes as part of `known_hosts`.

13. Once you execute that script, you will have to type **“yes”** to log into the first node specified in the `script.sh` file. Once you log in, type **“exit”** to get out.
This process continues until you have logged into all the nodes that were given in the `script.sh` file and have exited out of all the nodes.
14. Once this gets over, a minimum of 30 nodes will now be present in your **“~/.ssh/known_hosts”** file. It means that, if you do a **“mpdboot”** which we are going to look next, you don't have to worry about typing in passwords to those nodes.
15. Next step would be to create a file called `mpd.hosts` in the directory in which you have your programs.
16. A sample `mpd.hosts` has been provided. It consists of the nodes that you want to connect to, when you use MPI. It is to be noted here that, the nodes you have in the `mpd.hosts` file should be present in the **“~/.ssh/known_hosts”** file. If you plan to use 30 nodes, type in 30 nodes in there.
17. At this point, your CADE machine is setup to use MPI.
18. Whenever you first log into your CADE account, in order to run your programs in parallel, you need to do a **“mpdboot”** as given below.

“mpdboot -f mpd.hosts -n 6”

Here `mpd.hosts` is the file we created in the current directory
6 is equal to the number of nodes that we have specified in `mpd.hosts`

19. If you want to increase the number of nodes that you want to use, or when you are going to log out of your account, type **“mpdallexit”** in the terminal. This will remove all the nodes that you had requested to be connected. Now, once again you need to do a **mpdboot** as specified above, probably with more nodes if you want to increase the number of nodes. This is a 2 step process.
 - A. Increase the number of nodes in the `mpd.hosts` file
 - B. Specify that many nodes when you use `mpdboot` command.
20. Once you are done till this step, you can compile the program `hybrid.cc` that has been given to you by the following command.

“mpicxx -m32 -fopenmp hybrid.cc -o hybrid”

Now, you can execute the exe by,
“mpiexec -n 6 ./hybrid”

Here again, 6 is the number of nodes that we need to connect to.

22. For this program, I had used 10 threads by using
“**setenv OMP_NUM_THREADS 10**” in the terminal

Explanation for the code hybrid.cc:

In the given code, a 2D array `a[5][10]` is initialized in the Master, with numbers corresponding to the cell number from 0 to 49. Then the master sends each row to a Slave. The slave receives the row, and finds the number present in each location in that row. Then, it increments the location by the number that was present in each location. For example,

`a[0][1]` contains 1, hence, the value is incremented 1 time and hence the final value is 2.
`a[1][0]` contains 10, hence it is incremented 10 times and hence the final value is 20.

The slaves perform this operation and send the row back to the master. The master updates the rows appropriately.

OpenMP comes into play at the place where we increment the array locations in the slave. Since, there are 10 elements in each row, they can be run by 10 different threads. Hence I had set the `OMP_NUM_THREADS` to 10 initially.

Files attached with this tutorial: `script.sh`, `mpd.hosts`, `hybrid.cc`