Building an HPC Cluster

What are Clusters?

A computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer. Clusters are generally connected by a fast Local Area Network. Parallel programs that run on one of the nodes uses the processing power of all the nodes and produces the result.

Generally clusters are tightly coupled ie. All the motherboards will be stacked into a single cabinet and connected using some interconnection network. They'll share RAM, Hard Disk and other peripherals. Operating System runs on one of the nodes and controls the activities of other nodes. For more on Clusters, refer the Wiki Page.

What is a Beowulf Cluster?

Beowulf Clusters are cheap clusters created using off the shelf components. You can create a Beowulf cluster with just a few crap computers and an ethernet segment in your backyard. Although they don't give you top-notch performance, their performance is many-fold better than a single computer. A variant of Beowulf Clusters allows OS to run on every node and still allow parallel processing. And this is what exactly we're going to do here.

Kick Start Your Cluster

Prerequisites

Atleast Two Computers with a Linux Distribution installed in it. Make sure that your system has GCC installed in it. A network connection between them. If you have just two computers, you can connect them using an ethernet wire. Make sure that IP addresses are assigned to them. If you dont have a router to assign IP, you can statically assign them IP addresses. Click Here to know how to assign static IP addresses.

Rest of the document will assume that we are having two computers having host names node0 and node1. Let node0 be the master node.

The following steps are to be done for every node

Add the nodes to the /etc/hosts file. Open this file using your favourite text editor and add your node's IP address followed by its host name. Give one node information per line. For example,

```
node0 10.1.1.1

node1 10.1.1.2
```

Create a new user in both the nodes. Let us call this new user as mpiuser. You can create a new user through GUI by going to System->Administration->Users and Groups and click "Add User". Create a new user called mpiuser and give it a password. Give administrative privileges to that user.

Make sure that you create the same user on all nodes. Although same password on all the nodes is not necessary, it is recommended that you do so because it'll eliminate the need to remember passwords for every node.

Now download and install ssh-server in every node. Execute the command sudo apt--get install openssh-server in every machine.

Now logout from your session and log in as mpiuser.

Open terminal and type the following ssh-keygen -t dsa. This command will generate a new ssh key. On executing this command, it'll ask for a paraphrase. Leave it blank as we want to create a passwordless ssh (Assuming that you've a trusted LAN with no security issues).

A folder called .ssh will be created in your home directory. Its a hidden folder. This folder will contain a file id_dsa.pub that contains your public key. Now copy this key to another file called authorized_keys in the same directory. Execute the command in the terminal cd /home/mpiuser/.ssh; cat id_dsa.pub >> authorized_keys;.

Now download MPICH from the following website(MPICH1). Please download the MPICH 1.xx version from the website. Do not download MPICH 2 version. I was unable to get MPICH 2 to work in the cluster.

Untar the archive and navigate into the directory in the terminal. Execute the following commands:

```
mkdir /home/mpiuser/mpich1

./configure --prefix=/home/mpiuser/mpich1

make

make install
```

Open the file .bashrc in your home directory. If file does not exist, create one. Copy the following code into that file

```
export PATH=/home/mpiuser/mpich1/bin:$PATH

export PATH

LD_LIBRARY_PATH="/home/mpiuser/mpich1/lib:$LD_LIBRARY_PATH"

export LD_LIBRARY_PATH
```

Now we'll define the path to MPICH for SSH. Run the following command: sudo echo /home/mpiuser/mpich1/bin >> /etc/environment

Now logout and login back into the user mpiuser.

In the folder mpich1, within the sub-directory share or util/machines/ a file called machines.LINUX will be found. Open that file and add the hostnames of all nodes except the home node ie. If you're editing the machines.LINUX file of node0, then that file will contain host names of all nodes except node0. By default MPICH executes a copy of the program in the home node. The machines.LINUX file for the machine node0 is as follows

```
node1 : 2
```

The number after : indicates number of cores available in each of the nodes.

Cluster is Ready!!

Your cluster is ready!!! You can test run your programs by compiling the code available in the examples directory within mpich1 directory. Since example files have a MakeFile associated with them, you can compile the code by simply typing make command in the terminal after navigating to the corresponding directory.

To execute your code, make sure that the executable is at the same path in all nodes ie. If "foo" is your executable present in the path /home/mpiuser/mpich1/example/foo in node0, then that executable must be present in the same path in all other nodes.

To execute the code foo, type the following command in terminal after navigating to the location of the executable: mpirun -np 2 foo. Its enough to run the command in any one of the nodes, but make sure that the executable file is present in the same path in all the nodes. Here mpirun is the command that will run our program in all the nodes specified in the machines. LINUX file. "-np 2" flag indicates the number of processes to be spawned. Here we spawn two processes. By default one process will be spawned in the home node. Since here we used "-np 2", two processes will be spawned, one in the host machine and other in the node listed in the machines.LINUX file. If the machines.LINUX file has 10 nodes listed and "-np 2" flag is used, only the node represented by the first entry in the file is attached to the cluster.