RUNNING A MPI CLUSTER WITHIN A LAN
PART 2. CONFIGURE A REMOTE ACCESS

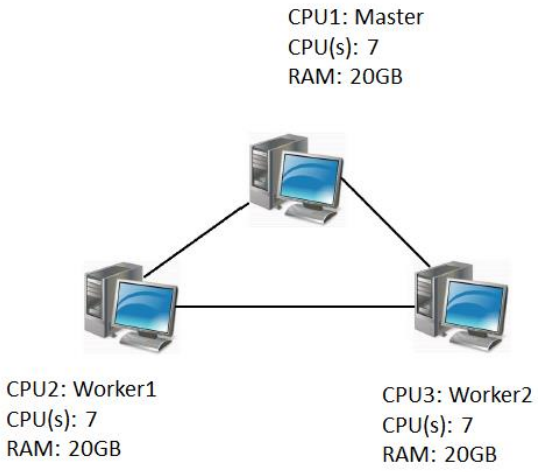

Figure 1. MPI Cluster with 3PCs.

MASTER:

1. Configure a virtual LAN using Hamachi. **It is only necessary if the computers are not into a LAN.**

   a. Download Hamachi from https://www.vpn.net/linux
      $ sudo dpkg -i logmein-hamachi_2.1.0.203-1_amd64.deb

   b. Install Graphical User Interface GUI HAGUICHI
      $ sudo add-apt-repository -y ppa:webupd8team/haguichi
      $ sudo apt update
      $ sudo apt install -y haguichi

   c. Open Haguichi, generate a new network and share the name and password with the workers.

Figure 2. Haguichi new network.

   d. Consult your IP ADD and share it with the workers.
      $ ifconfig

   e. If ifconfig command does not work, you might require install net-tools package
      $ sudo apt-get install net-tools

f. Test the communication between the computers using the ping command. Example: We assume 20.10.0.6 as the worker's IP.
   $ ping 20.10.0.6

2. Configure your host's file: You can name the workers in the network to avoid typing the IP addresses for each node.

   a. Modify the hosts' file to include each worker
      $ sudo nano /etc/hosts

   b. Append a new line to include the master & worker's name and IP. If you are into a virtual LAN, use the Haguichi's address. For example:
      127.0.0.1   localhost
      20.10.0.4   master
      20.10.0.6   worker

   c. Test the communication with the workers using the ping command. Example:
      $ ping worker

3. Create a new user: We use a custom account to keep our configurations simple.

   a. Let us create the new user mpiuser
      $ sudo adduser mpiuser

   b. Assign a name and password, keep they in mind to future process.

4. Setting up SSH: The machines will communicate over the network via SSH protocol. The instruction d,e and f must be executed in the worker computer.

   a. Install SSH server
      $ sudo apt-get install openssh-server

   b. Test SSH connection
      $ ssh mpiuser@master

   c. Return to the original season
      $ exit

   d. Test the connection from each worker. The follow instruction requires that the worker had configured the LAN and modified the hosts' file. **In the worker type:**
      $ ssh mpiuser@master

   e. Probe that the worker access to the master by displaying the CPU features. **In the worker** (loging in the Master by SSH) type.
      $ lscpu

   f. Return to the original worker season. **In the worker type:**
      $ exit

5. Passwordless connection: The master and workers must be continually sharing information. To enable more easier login, we generate keys and copy them to other machines' list of authorized_keys. **The following instructions require that the workers had set the mpiuser account.**
   a. Login with mpi account.
      $ su – mpiuser

   b. Generate RSA keys. Skip with enter all the questions.
      $ ssh-keygen -t rsa

   c. Copy the key into the workers. The next commands should be ejected per each worker.
      $ ssh mpiuser@worker mkdir -p .ssh
      $ cat .ssh/id_rsa.pub | ssh mpiuser@worker 'cat >> .ssh/authorized_keys'

   d. Test SSH passwordless connection.
      $ ssh mpiuser@worker

   a. Probe that it is the master and exit.
      $ lscpu
      $ exit

Worker:

2. Configure a virtual LAN using Hamachi. **It is only necessary if the computers are not into a LAN.**

   a. Download Hamachi from https://www.vpn.net/linux
      $ sudo dpkg -i logmein-hamachi_2.1.0.203-1_amd64.deb

   b. Install Graphical User Interface GUI HAGUICHI.
      $ sudo add-apt-repository -y ppa:webupd8team/haguichi
      $ sudo apt update
      $ sudo apt install -y haguichi

   c. Open Haguichi and join to the master network.

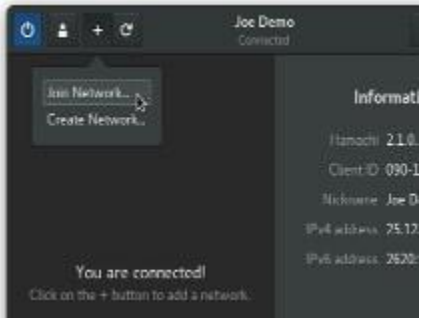

Figure 2. Hamachi join to the master network.

   d. Consult your IP ADD and share it with the Master.
      $ ifconfig

e. If ifconfig command does not work, you might require install net tools package
   $ sudo apt-get install net-tools

f. Test the communication between the computers using the ping command. Example: We assume 20.10.0.4 as the Master's IP.
   $ ping 20.10.0.4

3. Configure your host's file: You can give a name to the master in the network to avoid typing the IP address.

   a. Modify the hosts file to include the master IP
      $ sudo nano /etc/hosts

   b. Append a new line with the master & workers´s name and IP. If you are into a virtual LAN, use the HAGUICHI's address. For example:
      127.0.0.1   localhost
      20.10.0.4   master
      20.10.0.6   worker

   c. Test the communication with the master using the ping command. Example:
      $ ping master

4. Create a new user: We use a custom account to keep our configurations simple.

   a. Let us create the new user mpiuser.
      $ sudo adduser mpiuser

   b. Assign a name and password, keep they in mind to future process.

5. Setting up SSH: The machines are going be communicating over the network via SSH protocol. The instruction d,e and f must be executed in the master computer.

   a. Install SSH server
      $ sudo apt-get install openssh-server

   b. Test SSH connection
      $ ssh mpiuser@worker

   c. Return to the original season
      $ exit

   d. Test the connection from the master. The follow instruction requires that the master had configured the LAN and modified the hosts' file. **In the master type:**
      $ ssh mpiuser@worker

e. Probe that the master can access to the worker by displaying the CPU features. **In the master** (loging in the worker by SSH) type.
   $ lscpu

f. Return to the original master season. **In the master type:**
   $ exit

6. Passwordless connection: The master and workers must be continually sharing information. To enable more easier login, we generate keys and copy them to other machines' list of authorized_keys. **The following instructions require that the master had set the mpiuser account.**

   a. Login with mpi account
      $ su – mpiuser

   b. Generate RSA keys. Skip with enter all the questions
      $ ssh-keygen -t rsa

   c. Copy the key into the master.
      $ ssh mpiuser@master mkdir -p .ssh
      $ $ cat .ssh/id_rsa.pub | ssh mpiuser@master 'cat >> .ssh/authorized_keys'

   d. Test SSH passwordless connection
      $ ssh mpiuser@master

   e. Probe that it is the master and exit.
      $ lscpu
      $ exit