Configuring and Running MPI Programs

This document describes the procedure of bringing up MPI ring on a network of systems.

The following steps need to be done just once in order to configure the systems so that they may be able to connect to each other via MPI.

1. All the systems should have the same username through which the MPI codes will be run. Eg. all the machines which are meant to be used as MPI cluster should have a common username say **user_mpi** ***(user_mpi)***. This account will be used to run the MPI codes.
2. On the above mentioned account of each system there should be a file named **.mpd.conf** in the home directory i.e. /home/user_mpi . This file should have just 1 line written in it, which is **MPD_SECRETWORD=mpipassword** . The "**mpipassowrd**" can be replaced by any string literal password which you want. But ensure that ALL the systems have the same password. When the systems connect to each other in the MPI ring, they communicate through this password, and this must be same in all of them. Also the permissions of this file (.mpd.conf) should be made 644 by issuing **chmod 644 .mpd.conf**
3. IF all the systems have a hostname then to make life easier one can edit the **/etc/hosts** file of all the systems, which are going to be a part of the cluster, to include the hostnames of all the other systems in the cluster. This will enable the users to access the other machines via ***ssh*** by only issuing the hostname. eg. if a system with IP 10.3.3.25 has a hostname mangal then , if I am at a system with IP 10.3.3.22 and its /etc/hosts has the hostname to IP address mapping of mangal then I can ssh into mangal by issuing **ssh user_mpi@mangal** or **scp file.c mangal:** So one can ssh and scp the files very easily. Notice that in scp we did not mention the username i.e. because when we do not enter the username is ssh/scp it takes the username of the current system as default. So in this case it is equivalent to user_mpi@mangal, since we are issuing scp from user_mpi account of some other system.
4. There is a standard technique wherein one can ssh/scp into other systems without having to enter the password again and again. Since while using MPI we will need to ssh and scp many times it is advisable to configure all the systems such that any one of them can access any other of them via ssh/scp without having to enter the password. This will again be very helpful in the long run.
5. In the username **user_mpi** one will have to add the path of MPI's **bin** directory (the place where all the executables are stored) to the system's **PATH** variable. This can be done by editing the **.bashrc** file of **user_mpi** account and adding the following line in it, **export PATH=$PATH:/opt/mpi-install/bin** if the mpi is installed in **/opt/** directory or else replace this path with the ***bin*** directory of the current installer.

Following the above procedure will configure the MPI on a system i.e. when followed the above steps for all the systems in the cluster, the MPI programs are now ready to execute on the cluster.

The following steps now show the steps that should be taken in order to execute MPI programs once the above steps have been followed correctly.

1. Firstly, we need to bring up the **MPI Ring,** the MPI ring means that out of all the available systems for use we choose a fixed number of systems which will be active while we run the program. The ring can be made of >1 systems. Bringing up the ring implies that the systems are now connected to each other via MPI and are ready to share the workload. Please notice that it is not necessary to use ALL the systems in the ring when we execute the code, we can use a subset of the systems in the ring. How to do it will be shown later.
2. To bring up the ring there are two methods.
    a. Using the **mpd.hosts** file. Create a file named **mpd.hosts** in the home directory of all the systems on **user_mpi** account and write all the hostnames of the systems in the cluster in it. These hostnames will be the same as hostnames in the **/etc/hosts** file. The MPI will read this file to choose the hostnames to enter in the ring and the **/etc/hosts** file to resolve the hostnames. Once you have created the **mpd.hosts** file in the home directory of **user_mpi** you can now bring up the ring with just one command i.e. **mpdboot –n <a_positive_number > &** where <a_positive_number> should be replaced by a number >1. When this command is issued, the MPI will read the mpd.hosts file in the home directory and bring up n systems (n is the number given). The systems will be brought up in order they are listed in the mpd.hosts file. It is very possible that this command fails and all the systems do not enter the ring, try to change the order of hostnames in the mpd.hosts file and try again, else resort to second method. To check whether the systems are in the ring or not just issue the command **mpdtrace** This command will list all the systems currently in the ring so adjust accordingly.
    b. The second method is bruteforce method i.e. one has to ssh into each system and connect it to the existing MPI ring. Firstly, choose any 1 system and issue the command **mpd &.** This will bring up an MPI ring of only 1 system. Then issue the command **mpdtrace –l** this command will give an output of the form **shukra(10.3.3.21)_43213** This output means that the hostname of system is **shukra** and its IP address is **10.3.3.21** and MPI ring is active on this system on port number **43213** . The port number is of importance to us since we will use this port number for other systems to connect to this system. MPI basically connects via a port to all the systems and now we will connect all the other systems to this system using this port number. To add another system into the ring, login to that system with the same username, **user_mpi,** and issue the command **mpd –h shukra –p 43213 &** This command connects the current system to the host (-h) **shukra** at port number **43213** The hostname is that of the system to which you want to connect this system to and the port number is the same which was the output from **mpdtrace –l** command. In this way you can connect multiple systems to this system and all of them will form a ring. SSH into the systems and issue the same command to get that system into the ring. To check whether the systems are in the ring or not just issue the command

mpdtrace This command will list all the systems currently in the ring so adjust accordingly.

3. After bringing up the MPI ring with any of the above two methods you are now ready to run MPI programs from any of those systems onto any number of systems in the ring. To run the program follow the steps:
    a. Compile the program **mpicc <source_filename>.c –o <executable_filename>. eg. mpicc matrixMultiplication.c –o matrixMul.** This executable should be stored in the Home Directory of the system ONLY.
    b. (**Very Important**) SCP the <executable_filename> (in above example **matrixMul**) to ALL the systems on which you want the code to run in the **Home Directory.**
    c. Issue the command to run the command : **mpirun –np <n> ./<executable_filename> <n>** should be replaced by a number which is <= the number of systems currently in the ring and ALL of them should have a copy of the executable SCP'd to them in their home directory.

The above steps need to be followed in order to execute the MPI programs correctly. Please notice that MPI is not a very stable API so the systems may be disconnected at times. So keep checking the status of the ring by issuing the command **mpdtrace** this will list all the systems currently in the ring. Also if a particular system is unable to join the ring via one host/port, try to connect it to some other system already in the ring. Keep trying certain combinations or try with a different system. **SCP'ing the executables is very important failing which the code will not run on multiple systems and give an error message.**

For any further queries onto configuring the system and running the programs feel free to mail me.

Avi Dullu
avi.dullu@gmail.com