Bach Khoa University Of Technology
Faculty of Computer Science & Engineering

Tutorial: Configure The Cluster for MPI Program

Minh Thanh Chung

September 10, 2017

Goal: introduce how to access and configure a cluster to run MPI programs. Each student has 2 machines to configure the cluster by yourself.

Contents

1 Introduction

1.1 The list of IP addresses on each machine for student

Table 1: IP addresses for compute nodes

MSSV	**Name**	**Account**	**IP node1**	**IP node2**
1510180	Nguyen Quoc Bao	MSSV	10.1.6.1	10.1.6.32
1410274	L Quang Binh	–	10.1.6.2	10.1.6.33
1410483	Vo Hung Cuong	–	10.1.6.3	10.1.6.34
1410662	Hoang Tuan Dung	–	10.1.6.4	10.1.6.35
1410579	Nguyen Duc Duy	–	10.1.6.5	10.1.6.36
1411631	Tran Duc Hung	–	10.1.6.6	10.1.6.37
1411500	Ta Manh Huy	–	10.1.6.7	10.1.6.38
1411764	Nguyen Cao Minh Khanh	–	10.1.6.8	10.1.6.39
1412844	Pham Tien Phat	–	10.1.6.9	10.1.6.40
1412961	Nguyen Hua Gia Phuc	–	10.1.6.10	10.1.6.41
1413171	Ho Bao Quoc	–	10.1.6.11	10.1.6.42
1413765	Dang An Thinh	–	10.1.6.12	10.1.6.43
1513293	Mai Le Thong	–	10.1.6.13	10.1.6.44
1413843	Tran Minh Thong	–	10.1.6.14	10.1.6.45
1414216	Nguyen Minh Tri	–	10.1.6.15	10.1.6.46
1414272	Duong Viet Trung	–	10.1.6.16	10.1.6.47
1414768	Tran Quang Vu	–	10.1.6.17	10.1.6.48
1410806	Nguyen Manh Dat	–	10.1.6.18	10.1.6.49

1.2 How to access the server

In this course, each student has an account to access the server and do examples or exercises on it. However, if you want to do the Lab on your laptop, you can set up your own environment with Virtual Machine. Because all of Labs and Assignment will be tested on Linux OS.

For Windows: a simple tool that you can use is Putty (`https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html`). Information of gateway to access from outside networks:

- Hostname: **hpcc.hcmut.edu.vn**
- Port: 22

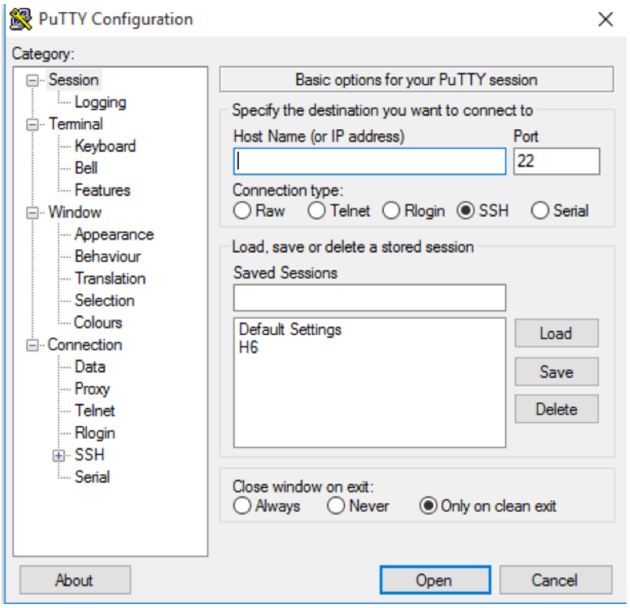
After that, click open. Then, you need an account and password to access the gateway (this is a public account):

- Username: **hpcc**
- Password: **bkhpcc**

At this step, you are just in the gateway, then, you need to access your machine by ***ssh*** protocol.

```
$ ssh username@IP_address

// For example:
// IP_address: 10.1.6.1 for student 1510180
```



Host Name (or IP address)

Port 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions

Default Settings
H6

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 1: GUI of Putty

```
// IP_address: 10.1.6.32 for student 1510180
// username: [ID_of_student] - password: [ID_of_student]
```

For Linux, you do not need to use other tools, you can access by ***ssh*** protocol.

```
$ ssh hpcc@hpcc.hcmut.edu.vn

// Then, you continuously ssh to the machine
// For example:
// IP_address: 10.1.6.1 for student 1510180
// IP_address: 10.1.6.32 for student 1510180
// username: [ID_of_student] - password: [ID_of_student]
```

1.3 Configure a cluster with 2 nodes

Each student has 2 nodes to configure the cluster to run MPI programs. The first thing you need to take care that is the connection between 2 nodes. However, the network of these 2 nodes has already configured for this. Therefore, you just need to configure how you can access from node1 to node2 without authentication requirement.

1. Log into node1

2. Create a ssh-key for your account:

```
$ ssh-keygen // then, just enter. Recommend: do not set the pass-phrase.
```

3. Copy the public key to node2 and itself:

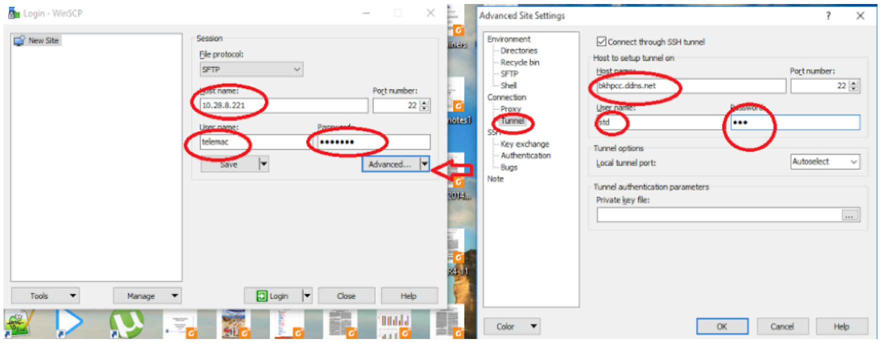
```
$ ssh-copy-id IP_address_of_node2 // then, enter password of your
    account on node2
```

```
        $ ssh-copy-id IP_address_of_node1 (itself) // then, enter password of
            your account on node1
```

4. Open a new terminal and log into node2 and do step 1 to 3 again on node2

1.4 How to transfer data between host and server

For Windows, you also use a tool to copy data between server and host, WinSCP (https://winscp.net/eng/download.php). Information for data transfer (with the window on the left hand side):



Login - WinSCP

Session

File protocol: SFTP

Host name: 10.28.8.221 Port number: 22

User name: telemac Password: •••••••

Advanced Site Settings

☑ Connect through SSH tunnel

Host to setup tunnel on

Host name: bkhpcc.ddns.net Port number: 22

User name: std Password: •••

Tunnel options

Local tunnel port: Autoselect

Tunnel authentication parameters

Private key file:

Figure 2: Data transfer between host and server by WinSCP

- Hostname: **IP_address** (of the machine)
- Username: **username**
- Password: **password**

However, to transfer the data, you need pass over the gateway, therefore, in the tab Advanced you need to fill some info such as (note, choose Tunnel at Connection Tab):

- Hostname: **hpcc.hcmut.edu.vn** (address of gateway)
- Username: **hpcc**
- Password: **bkhpcc**

For Linux, you also can copy data by command line:

```
// For example with student 1510180
// Step 1: Establish SSH tunnel. Pick a temporary port between 1024 and 32768 (1234 in
     this example). Port 22 will be used by scp. For example:
// $ ssh -L 1234:<address of R known to G>:22 <user at G>@<address of G> cat -
$ ssh -L 1234:10.1.6.1:22 hpcc@hpcc.hcmut.edu.vn cat -

// Step 2: Open another terminal for next step. Run scp against port 1234 pretending
    127.0.0.1 (localhost) is the remote machine R, and the command will be sent to R.
    For example:
// $ scp -P 1234 <user at R>@127.0.0.1:/path/to/file file-name-to-be-copied
$ scp -P 1234 1510180@127.0.0.1:/home/1510180/a.txt /home/
```

2 Practice

Test the cluster with a MPI program

With a MPI program requires data or something, you need to create the same path on both compute nodes. For example:

```
// On node1, you have a program in:
$ ls /home/1510180/
$ prog1.c prog hostfile

// If so, on node2, you also need to create or copy files/data to the same path
$ ls /home/1510180/
$ prog1.c prog hostfile
```

Note: usually, on a cluster, they deploy a solution for this called Distributed File System, such as NFS (https://help.ubuntu.com/community/SettingUpNFSHowTo).

Examples:

```
/* Name: mpi_hello_bsend.c */
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
#define  MASTER         0

int main(int argc, char *argv[]){
        int numtasks, taskid, len, partner, message;
        char hostname[MPI_MAX_PROCESSOR_NAME];
        MPI_Status status;

        MPI_Init(&argc, &argv);
        MPI_Comm_rank(MPI_COMM_WORLD, &taskid);
        MPI_Comm_size(MPI_COMM_WORLD, &numtasks);

        if (numtasks % 2 != 0){
                if (taskid == MASTER)
                        printf("Quitting. Need an even number of tasks: numtasks = %d\n", numtasks);
        }
        else{
                if (taskid == MASTER)
                        printf("MASTER: number of MPI tasks is %d\n", numtasks);

                MPI_Get_processor_name(hostname, &len);
                printf("Hello from task %d on %s!\n", taskid, hostname);

                if (taskid < numtasks/2){
                        partner = numtasks/2 + taskid;
                        MPI_Send(&taskid, 1, MPI_INT, partner, 1, MPI_COMM_WORLD);
                        MPI_Recv(&message, 1, MPI_INT, partner, 1, MPI_COMM_WORLD, &status);
                }
                else if (taskid >= numtasks/2){
                        partner = taskid - numtasks/2;
                        MPI_Recv(&message, 1, MPI_INT, partner, 1, MPI_COMM_WORLD, &status);
                        MPI_Send(&taskid, 1, MPI_INT, partner, 1, MPI_COMM_WORLD);
                }
                printf("Task %d is partner with %d\n", taskid, message);
        }

        MPI_Finalize();
}
```

Compile the program:

```
$ mpicc mpi_hello_bsend.c -o mpi_hello_bsend
```

Create a hostfile to define the addresses of compute nodes in the cluster

```
$ vim hostfile // any name that you want

// The content of file: hostfile
10.1.6.1  // node1
10.1.6.32 // node2
```

Run the MPI program on 2 nodes

```
$ mpirun -np 8 -npernode 4 -hostfile hostfile mpi_hello_bsend
// tag: -npernode means that you want to run 8 mpi processes, and divide 4 processes
    for each node. So, 4 processes will run on node1 and 4 other processes with run on
    node2.
```

3 Questions

1. Why we need to configure ssh without password for the authentication on each node?
2. If we do not set the number of processes per node (*-npernode*), how is the number of processes running on each node?
3. When we increase the number of compute nodes in a cluster and run the program with more processes in parallel, how is the performance? Why?