

MSBD5009 Environment Setup

February 13, 2025

1 Login CSE Lab2 Machines

Please activate your Unix account at UG domain first following this guide:

<https://cssystem.cse.ust.hk/UGuides/activation.html>

Then use ssh to login the machines:

```
$ ssh <msbd5009stuXX>@csl2wkXX.cse.ust.hk (where XX=01..50)
```

for example

```
$ ssh hyubc@csl2wk01.cse.ust.hk
```

The machines are accessible from outside the campus. If the connection is slow, please use the VPN provided by ITSC: <https://itsc.ust.hk/services/cyber-security/vpn>.

If you are a Windows user, please use a SSH client (e.g., PuTTY) to login the machine.

2 MPI Environment Setup

Please run which mpicc to check the MPI environment on the CS lab2 workstations.

If you get outputs like:

```
$ mpicc: command not found.
```

Then you need to setup the MPI environment first:

- Add the OpenMPI installation path to your shell environment:

```
$ echo 'setenv PATH "${PATH}:/usr/local/software/openmpi-5/bin"' >> ~/.cshrc_user
```

- Re-login (logout and login) to active the new environment.
- Use the command which mpicc to check your configuration.

3 Compile and Run MPI Application

We take the `mpi_hello.c` as an example.

3.1 Run MPI Application on a Single Node

Compile and run:

```
$ mpicc -std=c++11 -o mpi_hello mpi_hello.c
```

```
$ mpiexec -n 2 ./mpi_hello
```

outputs

```
Greetings from process 0 of 2!
```

```
Greetings from process 1 of 2!
```

If you need to run more processes than the physical cores of CPU, you need to add `--oversubscribe` to the `mpiexec` command:

```
$ mpiexec --oversubscribe -n 8 ./mpi_hello
# outputs
Greetings from process 0 of 8!
Greetings from process 1 of 8!
Greetings from process 2 of 8!
Greetings from process 3 of 8!
Greetings from process 4 of 8!
Greetings from process 5 of 8!
Greetings from process 6 of 8!
Greetings from process 7 of 8!
```

3.2 Run MPI Application on Multiple Nodes

- Prepare hostfile: For running an MPI application on multiple nodes, we usually need to decide which nodes we will use to run the application. We can put the hostnames or IPs into a hostfile:

```
# New a file named hostfile, add following information.
# For example, run application on nodes: csl2wk10, csl2wk11, csl2wk12,
# each node will have 4 available slots
csl2wk10 slots=4
csl2wk11 slots=4
csl2wk12 slots=4
```

The number of processes in the mpiexec command should less than or equal the total number of available slots (here is 12).

- Prepare SSH:
 - Check if you already have a RSA pri/pub key pair:


```
# ls ~/.ssh
authorized_keys id_rsa id_rsa.pub known_hosts
```
 - If you cannot find idrsa and idrsa.pub, you need to generate your RSA keys first:


```
$ ssh-keygen -t rsa -b 4096
# repeatedly press <enter> until finish
```
 - Add your RSA key to authorized_keys file:


```
$ touch ~/.ssh/authorized_keys
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```
 - Repeatedly ssh to csl2wk10 csl2wk11 and csl2wk12 from current machine(csl2wk01) to add target nodes to known_hosts file:


```
$ ssh csl2wk10
# type <yes> in you terminal
# you should login csl2wk10 successfully,
# please type <exit> in you terminal to back to your working workstation
$ ssh csl2wk11
# type <yes> in you terminal
# you should login csl2wk11 successfully,
# please type <exit> in you terminal to back to your working workstation
$ ssh csl2wk12
# type <yes> in you terminal
# you should login csl2wk12 successfully,
# please type <exit> in you terminal to back to your working workstation
# repeat above steps if you want to add more nodes to the `known_hosts`
```
 - Compile and Run your MPI application:


```
$ mpicc -std=c++11 -o mpi_hello mpi_hello.c
$ mpiexec --hostfile hostfile -n 12 ./mpi_hello
```

```
# outputs
Greetings from process 0 of 12!
Greetings from process 1 of 12!
Greetings from process 2 of 12!
Greetings from process 3 of 12!
Greetings from process 4 of 12!
Greetings from process 5 of 12!
Greetings from process 6 of 12!
Greetings from process 7 of 12!
Greetings from process 8 of 12!
Greetings from process 9 of 12!
Greetings from process 10 of 12!
Greetings from process 11 of 12!
```