COMP5112/MSBD5009 - MPI Manual

Please read the ***README.pdf*** first

MPI environment setup

Please run `which mpicc` to check the MPI environment on the CS lab 2 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/bin"' >> ~/.cshrc_user
  ```

- After that, re-login (`logout` and `login`) to active the new environment.
- You can use the command `which mpicc` to check your configuration.

Compile and run MPI application

In this step, we take the `mpi_hello.c` as an example.

Run MPI application on a single node

compile and run:

```
$ mpicc -std=c99 -o mpi_hello mpi_hello.c
$ mpiexec -n <number of processes> ./mpi_hello
```

e.g. `mpiexec -n 2 ./mpi_hello`

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

e.g. `mpiexec --oversubscribe -n 8 ./mpi_hello`

An example output:

```
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!
```

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.

For example, if we want to run our application on nodes: `csl2wk10`, `csl2wk11` and `csl2wk12`, the hostfile should be:

```
csl2wk10 slots=4
csl2wk11 slots=4
```

```
csl2wk12 slots=4
```

Each node will have 4 avaliable slots (12 slots in total).

The number of processes in the `mpiexec` command should less than or equal the total number of avaiable 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 cannnot find `id_rsa` and `id_rsa.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`:

  ```
  $ touch ~/.ssh/authorized_keys
  $ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
  ```

- add target nodes to `known_hosts` (take `csl2wk10`, `csl2wk11` and `csl2wk12` as an example):

  ```
  $ 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=c99 -o mpi_hello mpi_hello.c
$ mpiexec --hostfile hostfile -n <number of processes> ./mpi_hello
```

e.g. `mpiexec --hostfile hostfile -n 12 ./mpi_hello`

An exmaple output:

```
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!
```