OpenMPI User Guide @ IST RNL

Parallel and Distributed Computing (PDC)

2017/2018

This document provides a jump-start for running MPI applications at RNL. The installed version of MPI is OpenMPI 1.10.3.

There are two ways in which you can run MPI applications at RNL:

- using a cluster of machines dedicated to PDC students where you have full control of MPI, but limited to 9 machines (each with 4 cores) shared by all students.
- a cluster over all machines in all RNL labs that is managed by condor.

1 Dedicated PDC Machines

1.1 Machine Access

The cluster of machines dedicated to PDC is made of 9 machines named `cpd-1` through `cpd-9`. These machines are only accessible from the `cluster` machine, to which you can login from anywhere, eg:

```
$ ssh istXXXXX@cluster.rnl.tecnico.ulisboa.pt
Password:
istXXXXX@borg:~$ ssh cpd00@cpd-1
Password:
Have a lot of fun...
cpd00@cpd-1:~>
```

where `XXXXX` is your Fenix id.

Each group will have its own account, be sure you get the password for it. You will need to copy any file you may need to this account (eg, using the `scp` command). You can login to any of the 9 machines and you will see the same contents in your account in all of them.

1.2 Compiling

The compile command for a program in the C language is `mpicc`. `mpicc` accepts all `gcc` flags. For example:

```
$ mpicc -g -o hello_world hello_world.c
```

compiles file `hello_world.c` with debugging information, and creates an executable named `hello_world`.

Note that you cannot import the binary, you will need to compile the source code in one of the cluster machines.

1.3 Executing an MPI Application

An MPI application is launched using the command `mpirun`. The switch `-np <p>` can be used to specify the number of processes to be created, which, by default, are all created on the local machine. For example,

```
$ mpirun -np 4 hello_world
```

will create four processes on the local machine, all running `hello_world`.

If the application requires arguments, they should appear after the program name.

The switch `-host` followed by a list of machines separated by commas executes the application on the specified machines. The command:

```
cpd00@cpd-1:~> mpirun --host cpd-4,cpd-7 hello_world
```

will create two processes, one on machine `cpd-4` and another on `cpd-7`, both running the program `hello_world`.

It is also possible to create a file with the list of machines, one per line, where the application should run. For example, if the contents of file `my-nodes.txt` is:

```
cpd-4 slots=1
cpd-7 slots=1
```

then the command:

```
cpd00@cpd-1:~> mpirun --hostfile my-nodes.txt hello_world
```

will have the same behavior as described above.

2 Running the Application on RNL's Cluster

All machines in the RNL's labs are set up to work as a cluster. To prevent jobs from interfering with other people using the machines, the jobs in the cluster are managed by Condor.

Jobs need to be submitted from the `cluster` machine. To submit a job to Condor, the following command is available:

```
$ condor_mpi -n <number of nodes> <full path to executable> <executable args>
```

(to learn more about job submissions see: `https://rnl.tecnico.ulisboa.pt/servicos/cluster/#mpi`)

Note that you cannot compile your program in the `cluster` machine. Either use one of the lab machines, or one of the PDC cluster machines, as described above in Section 1.2.

MPI expects the executable to be in the same location in all machines. For this reason, you will need to use the "cirrus" filesystem. Each student (of this PDC course) has a personal directory (`/mnt/cirrus/users/Y/Z/istXXXYZ`). Similarly, each PDC group has its own directory: `/mnt/cirrus/cpd/cpdXX`. The executable MPI file should be put under in this filesystem.

The output of jobs submitted to Condor is placed in three files in the local directory: `out`, `err` and `log` (caution: previous versions of these files will be overwritten). You can check the status of your jobs using `condor_q`.