A very short introduction to using Trinity Centre for High Performance Computing resources

Erwan Moreau

moreaue@cs.tcd.ie

CLG meeting, August 2011
Why using TCHPC resources?

- computer intensive tasks
- more computing power available
  - lonsdale: Opteron 2.3GHz 64bits, 154 nodes, each node has 16G RAM and 8 cores (1232 cores total)
- spare your local machine resources; no need to wait for other jobs to end on CLG machines

Constraints:
- some “administrative” preliminaries
- a bit more complicated (depending on what you want to do)
- preferably for parallel jobs (TCHPC recommendation)
First step: create your account

1. Read TCHPC policy/rules:
   http://www.tchpc.tcd.ie/resources/policies
2. Submit application for an account:
   https://www.tchpc.tcd.ie/support/apply
3. A copy of your ID card is required → go to TCHPC (2nd floor in the Lloyd Institute)
Apply for a project

- The “organizing unit” on the cluster is the “project”:
  - you are given access to some amount of space and computing time according to your project specification
  - Example: the smallest possible project in the form is 100,000 CPU hours/year

- Apply on
  [https://www.tchpc.tcd.ie/resource_application](https://www.tchpc.tcd.ie/resource_application)

- Once your project is accepted, you are given a “project code” that you must provide each time you submit a job

- **Warning**: in the application form you must confirm that:
  - *This research does not require ethics approval or has received ethics approval*
  - *There are no Data Protection or other legal issues related to the provision of this service*
Preparation: copy your code and data

- connect: ssh moreaue@lonsdale.tchpc.tcd.ie
- copy your code
  - remark: cvs, svn, hg, git available
- configure your workspace:
  - install any external libraries you need (in your home directory)
    - (ask me for details about installing Perl modules)
  - configure everything:
  - your code/libraries will run from another machine
  - for example, set environment variables like $PATH, $PERL5LIB, $CLASSPATH in your ~/.bashrc or ~/.bash_profile file.
- Huge data: use your project directory
  - e.g. in /projects/pi-vogel/HPC_11_00205/, see https://www.tchpc.tcd.ie/resources/datapolicy
Example 1: single process

Warning: TCHPC recommends not to run such single process jobs (it wastes 7 cores computing time)

#!/bin/bash
#SBATCH -n 1  # using 1 core
#SBATCH -t 00:30:00  # max time 30mn
#SBATCH -p debug  # "debug" partition (test purpose, otherwise "compute")
#SBATCH -U HPC_11_00205  # project code
#SBATCH -J simple_test1  # a meaningful name (for yourself)

source $HOME/.bash_profile  # contains "export PATH=...." etc.

name=test1
projectDir=/projects/pi-vogel/HPC_11_00205/
dataDir=$projectDir/clean-selection

similarity-ranker.pl [...] > $projectDir/$name.out 2>$projectDir/$name.err
Example 2 : multiple process

main script :

#!/bin/bash
#SBATCH -n 8 # using 8 core
#SBATCH -t 00:30:00 # max time 30mn
#SBATCH -p debug # "debug" partition (test purpose, otherwise "compute")
#SBATCH -U HPC_11_00205 # project code
#SBATCH -J simple_test2 # a meaningful name (for yourself)
srun --multi-prog test2.conf

test2.conf srun config file :
# srun multiple program configuration file for test2
0 bash /projects/pi-vogel/HPC_11_00205//test2.0.sh
1 bash /projects/pi-vogel/HPC_11_00205//test2.1.sh
2 bash /projects/pi-vogel/HPC_11_00205//test2.2.sh
[......]
7 bash /projects/pi-vogel/HPC_11_00205//test2.7.sh

test2.?.sh individual process script :
source /home/users/moreaue/.bash_profile
similarity-ranker.pl -m 150-200 [....] >/projects/pi-vogel/HPC_11_00205//test2...
Example 3: multiple process, automatic generation

```
[......]
nb=8
step=50
startPos=0
multiProgConfigFile="$projectDir/$name.conf"
jobScriptPrefix="$projectDir/$name"

echo "# srun multiple program configuration file for $name" > $multiProgConfigFile
echo >> $multiProgConfigFile
for i in $(seq 0 $(( nb - 1 ))); do
    endPos=$(( startPos + step ))
    jobScript="$jobScriptPrefix.$i.sh"
    echo "# job script automatically generated by $scriptLocation on $theDate" > $jobScript
    echo "source $HOME/.bash_profile" >> $jobScript
    echo "similarity-ranker.pl -m $startPos-$endPos -s 1 -o $projectDir/$name.$i.scores >$projectDir/$name.$i.out.2 2>$projectDir/$name.$i.err.2" >> $jobScript
    echo "$i bash $jobScript" >> $multiProgConfigFile
    startPos=$endPos
done
sr
```
Useful commands

Slurm commands:

- submit a job: `sbatch test2.sh`
- Display queue/partition names, available nodes etc: `sinfo`
- Display info about a job: `scontrol show jobid 108`
- Display jobs in the queue: `squeue [-p <partition>] [-u <user>]`

Other commands:

- Display my disk usage: `myquota`
- Display my CPU hours: `sbank balance statement`

Further details: [http://www.tchpc.tcd.ie/node/129](http://www.tchpc.tcd.ie/node/129)
Useful links

https://www.tchpc.tcd.ie/node/78
https://www.tchpc.tcd.ie/support/resource_allocation
https://www.tchpc.tcd.ie/resources/datapolicy
https://www.tchpc.tcd.ie/resources/clusterschedulepolicy
https://www.tchpc.tcd.ie/node/531
http://www.tchpc.tcd.ie/node/129
http://www.tchpc.tcd.ie/resources/acknowledgementpolicy

Useful commands to work with a remote connection:

- screen: persistent (text mode) window manager
  - connect once and have multiple windows
  - deconnect and re-connect to your session from any computer
  - don’t lose your session if the connection fails
- ssh-keygen: not to type your password every time
- ssh ControlMaster option: to connect several times using the same connection (faster)