Linux HPC Workshop

S. T. Balan,

R. Rollins, P. Owen

Department of Physics and Astronomy
University College London

October 2014

How to access training materials?

url

https://github.com/Astrophysics-UCL/HPCInfo/tree/master/training/linux_hpc_workshop_oct_2014

What will you learn?

- Accessing Astrophysics group machines
- Using linux console for your research
- Running your programs in HPC machines

Accessing machines from outside

You will need a *username* and *password*

steps

```
# step 1 login to zuserver
ssh -YC username@zuserver.star.ucl.ac.uk

# step 2 login to other machines from
# zuserver

ssh -YC username@splinter.star.ucl.ac.uk
```

command structure

structure

```
# [command] -[option[s]] -[argument]
```

Example

```
ls -la
makdir hello_wrold
cp hello.cpp new_hello.cpp
```

Linux console cheat sheat I

navigation and help

```
ls -lah dir_name
cd dir_name
cd ..
cd -
man command_name
pwd
exit
```

copy or move

```
cp src dest
cp -r src dest
mv src dest
ln -s src targ
```

create or delete

```
touch file.txt
mkdir dir_name
mkdir -p prt/dir
rm -i file.txt
rm -rf dir_name
```

find or search

```
locate file
whereis file
grep "bla" file
awk ’pattern’ file
```

Linux console cheat sheat II

file contents

```
cat file
more file
less file
head file
tail file
nm object_file
readelf shared_obj_file
ldd executable
```

process management

```
ps -e
kill
killall
top
```

ssh

```
ssh usr@host
ssh -YC user@host
scp usr@host:file dest
```

system info

```
uname -a
who
whoami
whois
which
finger
ping
echo $VAR_NAME
```

Linux console cheat sheat III

& ; | ¿ ¡

```
& # backgraound
; # combine
\ # next line
| # combine
* # wildcard
> # output
< # input
```

Text editors

```
emacs
vi
gedit
```

web

```
firefox
google-chrome
wget
curl
```

publishing

```
latex
pdflatex
bibtex
```

Linux console cheat sheat IV

compressed files

```
gzip
gunzip
tar xvzf
tar cvzf
tar xvjf
tar xvJf
```

images

```
eog
xfig
gimp
gthumb
convert
```

development

```
make
cmake
python
gcc
g++
gfortran
```

scientific

```
gnuplot
R
matlab
IDL
```

Exercises I

1. In your home directory create a dirctory called `linux_hpc_workshop`
2. Change directory to `linux_hpc_workshop`
3. What is the present working directory
4. Make a directory `level_1/level_2`, and move to `level_1/level_2` in one command
5. Move back to previous directory
6. Remove the directory (and its contents) `level_1`
7. Make a symbolic link to `usr/lib` in the current directory called `my_sybolic_link`
8. Create a file called `bla.txt` contents “this file has a word called bla”
9. Add another line in `bla.txt` called “this is the second line”
10. Check if it worked
11. Search for the phrase *bla* in `bla.txt`

Exercises II

1. Find the location of your python installation
2. Find the installtion location(s) of `liblapack.a`
3. Find whether an object `daxpy` is in `liblapack.a`
4. Find the value the environment variable `PATH` and `LD_LIBRARY_PATH`
5. Set the environment variable `MY_LINUX_HPC_VAR` to the absolute path to `linux_hpc_workshop`
6. Add, i.e append the absolute path to `linux_hpc_workshop` to the `PATH`
7. Use the source command do the last two steps from source file.
8. Use man command to find the option of `ls` that shows the output in Kilobyte,Megabyte

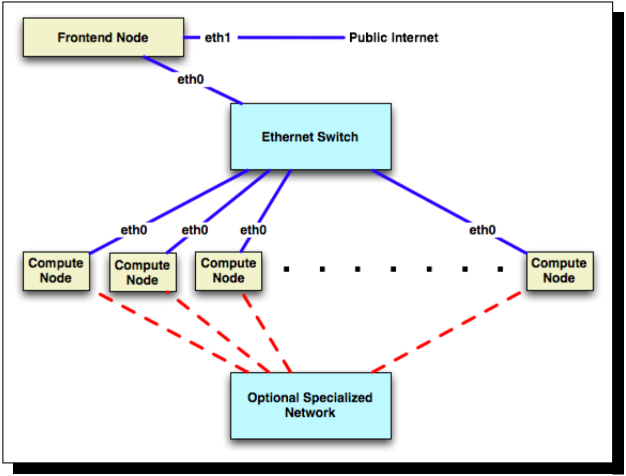
Exercises III

1. Find hostname,processor type,operating system version and write these info inot a text file called `info.txt`
2. Find the list of people who are loged into the system
3. Find the process that is taking most of the CPU at the moment
4. Find ids of the processes that you are running
5. Make a directory called `to_be_compressed`. Add the files `hello.cpp` and `hello.py` in this dir Now compress this directory using tar and zip
6. Delete the dirctory `to_be_compressed` and extract the files from `to_be_compressed.tar.gz`
7. Use wget to download files from `ftp://heasarc.gsfc.nasa.gov/software/fitsio/c/cfitsio3370.tar.gz`
8. What is the size of the item you just downloaded in MB
9. Find the number of occurrences of the phrase `table` is easy in all the files with extension `.h`
10. Remove all the fiels with extension `.h`
11. Copy the files with extension `.c` into a new directory `c_files`

HPC Facilities

machine	type	cores	memory
SPLINTER-1	distributed	96	48GB
SPLINTER-2	shared	40	1TB
PHALANX	shared	32	512GB

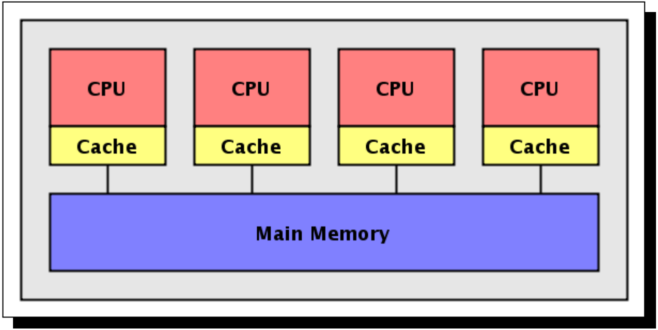
SPLINTER distributed



[1]

[1] http://www.rocksclusters.org/

SPLINTER shared



[2]

[2] http://www.cs.rit.edu/

Best practices I

- Choose the machines that are suited for your problem
- Read the User Guide
- Do not run your programs in the login node
- Do not install common software locally
- Request optimum resouces
- Minimise data transfer between nodes,
- Backup! Backup! Backup!

Submitting jobs

commands

```
qsub jon_script
qsub -I
checkjob job_id
qstat
showq
qdel
```

Example

```
#!/bin/bash
#PBS -N hello_world_program
#PBS -l nodes=1:ppn=4
#PBS -l mem=2gb
#PBS -j oe
#PBS -V

# source the required scripts
# this sets the PATH
source /home/sbalan/binpaths.sh
# this sets the LD_LIBRARY_PATHS
source /home/sbalan/libpaths.sh

# run my program
/home/sbalan/hello.exe
```

Exercises III

1. Login to your HCP machine and find the path to your `HOME` directory and your quota
2. Find the processor type and the version of your operating system
3. Request an interative queue and run the `hello_world.exe`
4. Sumbit `hello_world.exe` using a job script, find its `jobid`, check the output log.
5. Compile `big_mem_example`, submit it using a `job-script` and see how much menory it uses
6. Compile `time_pause_example`, submit it using a `job-script` and kill this job using its `jobid`.
7. In the previous example see what happens when you play with the time requested.

More information

ap-wiki

`http://www.ucl.ac.uk/star/GroupAWiki`

UCL Research Computing Platforms

`https://wiki.rc.ucl.ac.uk/wiki/Main_Page`

DiRAC

`http://www.dirac.ac.uk/`