HPC/LINUX

Some Commands
that were discussed during the workshop at Kashmir University

internal ip of KUHPC: 172.16.139.201
external ip of KUHPC: 45.249.235.183

```
ssh -X $USER@172.16.139.201
```

or

```
ssh -X $USER@45.249.235.183
```

[Replace $USER by your username to login to KUHPC as above]

22-24 May 2024

GIT

- To download a git repository into your local machine (laptop, desktop etc)
- Type in your terminal of your laptop:

```
term> git clone $LINK_TO_REPOSITORY
```

- Example: Aseem's git repo where codes etc developed during the workshop are publicly available:

```
term> git clone https://github.com/a-paranjape/hpctutorial.git
```

- To update the repository in your laptop, go to the directory that you got in the previous step and pull as:

```
term> cd hpctutorial
term> git pull --rebase
```

- Also, one can directly access the repository by typing the link (after git clone in the example above) in the web browser

SCP

- It's like "cp" command that you would use to copy files from one location to another within your local machine [be careful about the spaces etc]

```
term> cp $FROM $TO
```

- Google search for standard linux/unix commands, such as, `ls, cp, cd, mkdir, ls -ltr, rsync, rm, rm -r, rm -rf, pwd, cp -r, scp, scp -r`

- To copy files/folders from one machine to another machine, which could be the cluster or your second laptop/desktop, one can use "cp" over "ssh", i.e., "scp".

- Basic format is still same as "cp":

```
term> scp $FROM_OTHER_OR_THIS $TO_HERE_OR_THERE
```

- Example: copy a file "/home/user/test.txt" from your laptop to iucaa1 user at hpc:

```
term> scp /home/user/test.txt iucaa1@172.16.139.201:/home/iucaa1/.
```

- From hpc to your laptop:

```
term> scp iucaa1@172.16.139.201:/home/iucaa1/test.txt /home/user/.
```

- To transfer/copy folders, use "scp -r" , i.e., recursively.

Load modules

- In the cluster, type in the terminal:

```
term> module avail
```

(lists various packages, e.g, anaconda/python etc, that are available on the hpc cluster, but are not in your path by default)

- To use these packages, type, e.g.,:

```
term> module load anaconda3
```

(this loads it only for that particular session, so one has to repeat above command every time one logs in. To avoid this, add this to your .bashrc file)

- Open .bashrc in your home folder (e.g., /home/iucaa1 for iucaa1 user) using your favorite editor (e.g., vi/vim/emacs/nano etc):

```
term> vi ~/.bashrc
```

this open the file in which add a line (press "i" on your keyboard to insert text etc in the file opened by "vi" ; "~/" in the front allows you to open this file from anywhere in hpc):
module load anaconda3

- You can load as many modules as possible (available from step 1 above) by simply adding "module load module1 module2" etc in .bashrc

qsub

- Now, recall our discussions in the class
- Never run your codes on login/master node
- Get in the compute node interactively by typing:

```
term> qsub -I
```

gives you by default one core in a compute node. You must type "exit" explicitly after you are done with your jobs. Better to specify walltime, ncpus, etc as:

```
term> qsub -I -l select=1:ncpus=4 -l walltime=00:05:00
```

gives you a session with four processors on a compute node for 5 minutes, after which the session will be automatically terminated by the PBS job scheduler that is installed in the hpc

- Aliases: google to know more about adding aliases to your .bashrc file; examples

```
alias js = "qstat -u iucaa1"
alias hpcin = "ssh -X iucaa1@172.16.139.201
```

- `term> source ~/.bashrc`

PBS job script

- Open a file using editor, say, vi as: `vi PBS_JOB_SUBMISSION_SCRIPT`
- Call this file anything you like, and add text such as following in this file

```
#!/bin/bash

#PBS -l walltime=00:05:00
#PBS -l select=1:ncpus=4
#PBS -N test
#PBS -k oe
cd $PBS_O_WORKDIR

### THIS LINE IS A COMMENT
### LOAD MODULES ETC AS BELOW, BUT THIS MAY BE OPTIONAL
module load gcc openmpi-3.0.0 anaconda3

python $YOUR_PYTHON_CODE

### example: python pool.py

### play with such script for, say, your fortran, c code
### google for sample PBS job submission scripts etc
```

- Submit this from the login node itself as:

```
term> sub PBS_JOB_SUBMISSION_SCRIPT
```