High Performance Computing (HPC) Club Training Session

Xinsheng (Shawn) Qin
Jan 13 2017

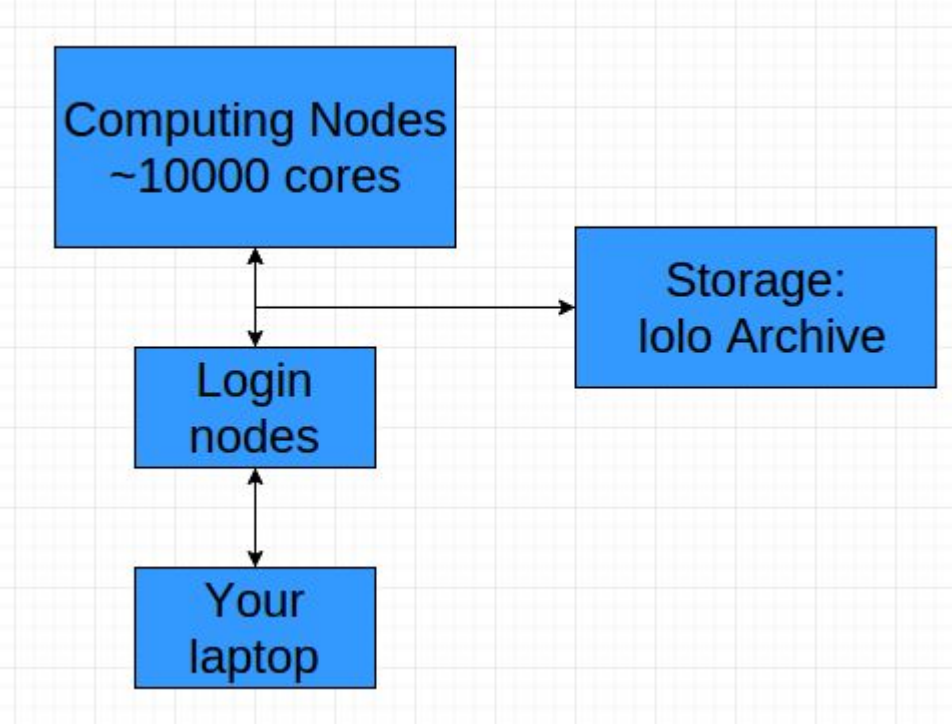
UNIVERSITY of WASHINGTON

Outline

- The Hyak Supercomputer
- Log in to Hyak
- Transfer files between your PC and Hyak
- Submitting your first job
- Practice

The Hyak Supercomputer

- A typical **node** has 16 processor **cores** and 128GB of memory
- All the nodes run CentOS 6 Linux
- ~10000 cores in total
- Club members have access to 1028 cores
- Node types
  - Computing nodes
  - Login nodes
    - Submitting jobs
    - Transferring files



Log in to Hyak

Remotely Accessing a Linux System via SSH protocol

- Use a SSH client program
  - Linux & Mac: just open a terminal
    - Mac: application->utilities->terminal
    - Ubuntu: Search for “terminal” in your applications

usage:ssh [options] user@IP-address

e.g

ssh xsqin@hyak.washington.edu

ssh xsqin@americano.amath.washington.edu

ssh shawn@128.95.220.141

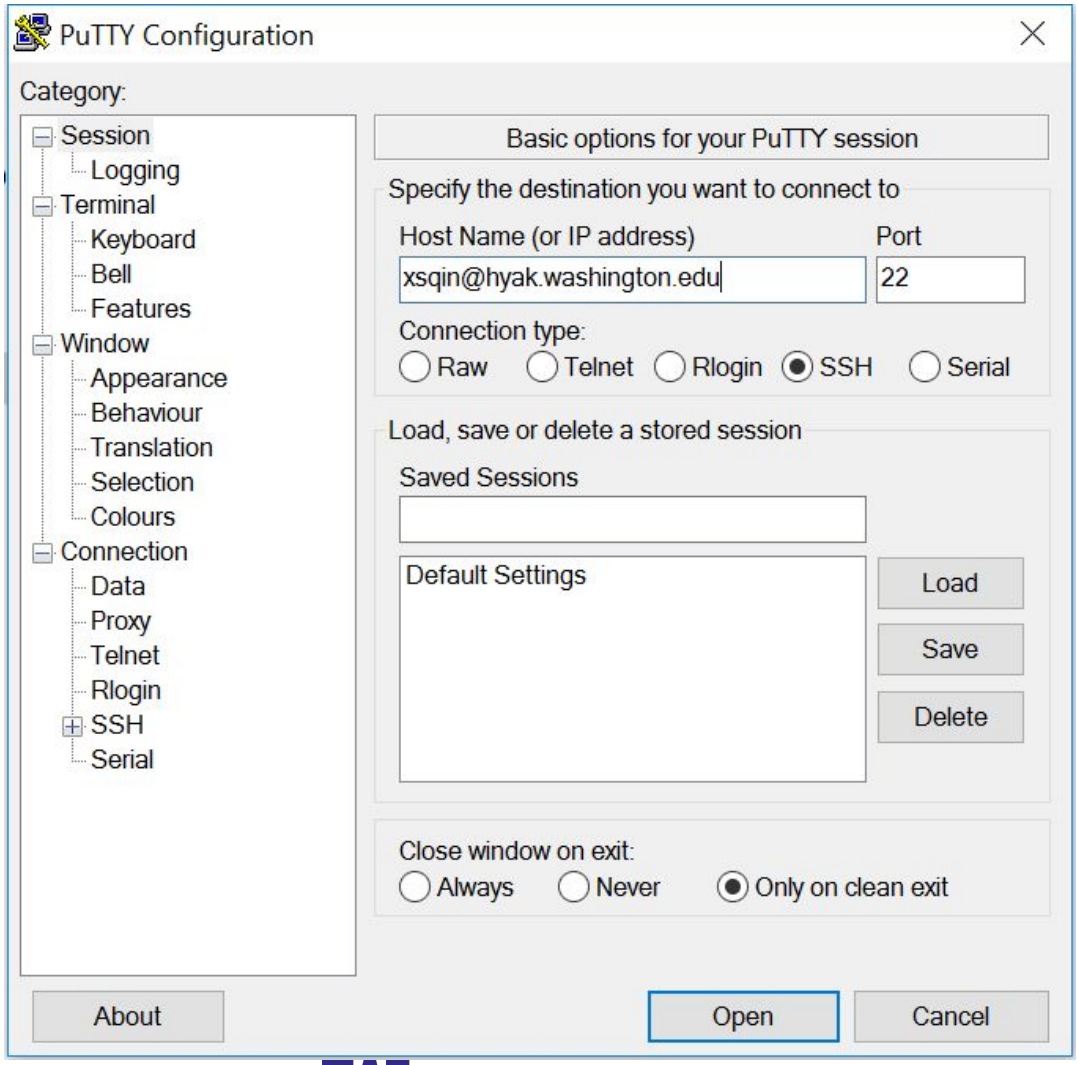
Then enter your password and PRN from your token

  - Windows: putty
    - http://www.putty.org/

Log in to Hyak

- putty
- More info here:

http://wiki.cac.washington.edu/display/hyakusers/Logging+In



Linux command basis

- Show current directory: pwd
- $HOME directory
- Show contents in current directory: ls [options]
- change current directory: cd + [path]
  - cd absolute_path
  - cd relative_path
- Create a new folder: mkdir folder_name
- Remove a file/folder: rm [options]
- Rename a file/folder: mv old_name new_name

Managing your files

- Transfer files from your local desktop to the server
    - Linux & Mac: SCP
        - scp [options] <source directory> <target directory>
    - Windows: WinSCP
        - https://winscp.net/eng/download.php
- Where should I put my job outputs?
    - Gscratch space: fast, temporary
        - /gscratch/<your group>/<your netid>
        - e.g. /gscratch/stf/xsqin
    - Lolo: slow, long-term storage
        - /lolo/archive/hyak/<your group>/<netid>
        - /lolo/archive/hyak/stf/xsqin

Submitting your first job

- Node types:
  - logging node: used to submit jobs. Don't execute heavy tasks on it
  - computing node: where your jobs are running
- Using qsub
  - qsub [options] command-for-running-your-job
  - e.g. qsub -l walltime=2:00:00 -I -V matlab
- PBS (Portable Batch System) Jobscripts
  - instructions for the scheduler
  - setting up the work environment
  - executing your production program
- http://wiki.cac.washington.edu/display/hyakusers/Hyak+Job+Scheduler#HyakJobScheduler-ImportantInformation

Submitting your first job

qsub run.sh

```
#!/bin/bash

##============ instructions for the scheduler ==============##

## name of the job
#PBS -N case17

## specify resources requested
#PBS -l walltime=1:00:00:00,nodes=3:ppn=16,mem=180gb,feature=16core

#PBS -d /civil/shared/motley/xsqin/tsunami_inundation/modified-3D/case17_force
#PBS -o /civil/shared/motley/xsqin/tsunami_inundation/modified-3D/case17_force

## Send an email to your UW email when the job begins, finishes or fails.
#PBS -m abe

#PBS -W group_list=hyak-motley
##PBS -W group_list=hyak-lehman
##PBS -W group_list=hyak-stf

##=================== working environment ====================##
module load icc_14.0-impi_4.1.1




##============== executing your production program ============##
source /gscratch/motley/shared/OpenFOAM231/OpenFOAM-2.3.1/etc/bashrc



#decomposePar -force
#mpirun -np 64 interFoam -parallel
#echo $MPI_ROOT
mpirun -np 64 interFoam -parallel >> run.log 2>&1
```

Submitting your first job

Practice:

Create a directory on Hyak for your code, e.g. $HOME/hello_world

go to:

https://github.com/xinshengqin/HPCC_training_session.git

copy and paste

- the code in ./hello_world/test.c
- the code in ./hello_world/run.sh

into two different text files on your laptop.

Edit run.sh:

change working directory and output directory (e.g. $HOME/hello_world)

Submitting your first job

Practice:

Upload the two text files to Hyak:

e.g. $HOME/hello_world

Compile the c code:

gcc test.c -o test

Test your code:

./test

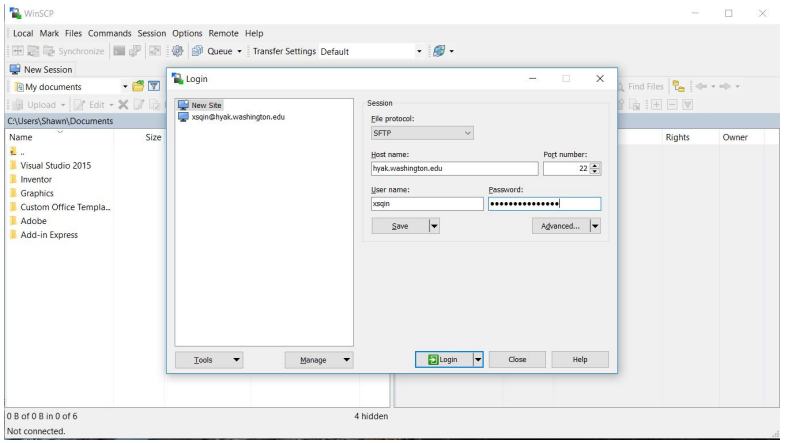
For windows user:

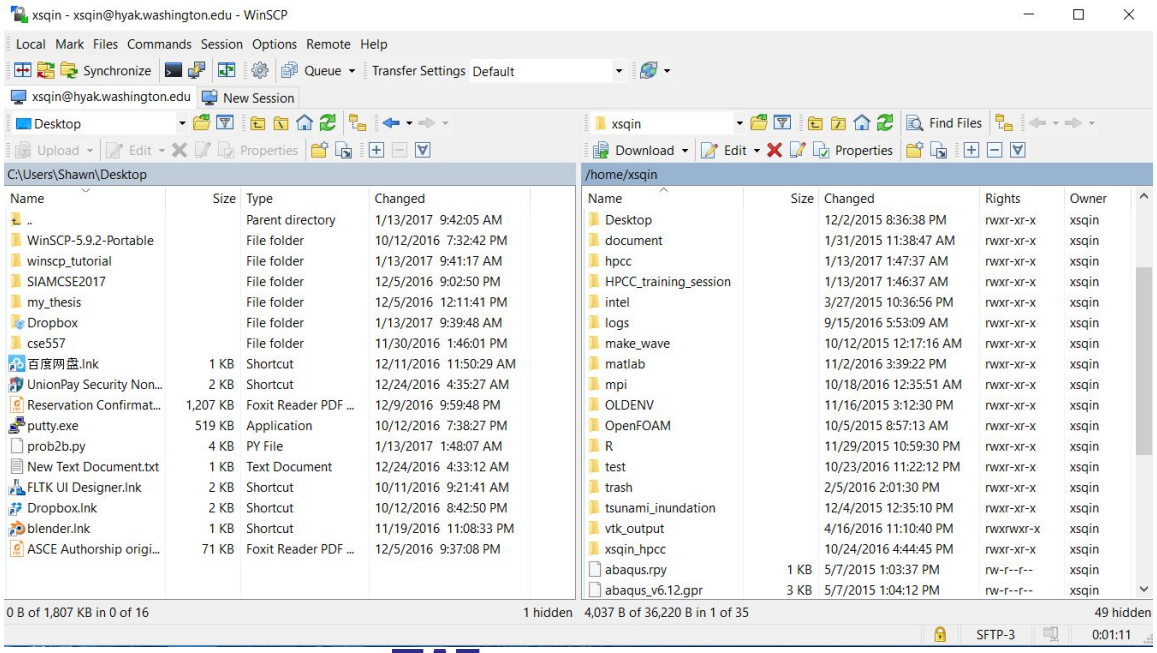
convert run.sh to UNIX

format with command:

dos2unix run.sh

Submit the job: qsub run.sh





Managing Jobs in Your Queues

- Check the status of your job
  checkjob <job_id>
  qstat -f <job_id>
- Queue Status
  - showq -w qos=<your_group>
  - showq -w user=<your_netid>
  - e.g. showq -w qos=stf

- To cancel a job
  mjobctl -c <job_id>
  qdel <job_id>

Two types of jobs

- Background job
  - what we just submitted:
  - qsub some_script.sh
- Interactive jobs
  - qsub -I -V -W group_list=hyak-stf -l walltime=2:00:00