Python for Machine Learning and Data Analysis

Dr. Handan Liu

h.liu@northeastern.edu

Northeastern University

Spring 2019

Discovery Cluster and Linux Fundamentals

Lecture 6

Contents

- Connect to Discovery Cluster
- Logging in
- How to use Discovery (Linux Fundamentals)
- Overview of Discovery
- Software on Discovery
- SLURM
- Manage jobs on the cluster

Secure SHell (SSH)

- Secure SHell (SSH) is a software standard to support encrypted data transfer between two computers. It can be used to support secure logins, file transfers or general purpose connects.
- SSH or VPN?
  - Both SSH and the VPN provide secure data transmission
  - SSH connections encrypt data transfer even between devices on the same network.
  - VPN connections only encrypt between your computer and the destination network.
  - VPN may be preferable when all activity between your machine and a secure network needs to be protected, or to access restricted services on the destination network.

Connect to Discovery Cluster

- Connect to Discovery Cluster from Windows Clients
  - Recommend: MobaXterm.
    - Download MobaXterm: http://mobaxterm.mobatek.net/download.html
  - Putty
  - Oracle Virtual Box: needs a lot of memory, 50GB+ disk to be useful
  - Cygwin: needs cygwin, xming
- Connect to Discovery Cluster from terminal of MacOS or Ubuntu
  - Terminal in Ubuntu
  - Terminal in Mac or iTerm (additionally installed)
- Graphical User Interface (GUI), optional
  - Mac: Xquartz
  - Windows: X11

Logging in

$ ssh -X username@domain_name

$ ssh -Y username@domain_name

-X or -Y enable X11 connection forwarding

-X treats X11 clients as untrusted

-Y treast X11 clients as trusted

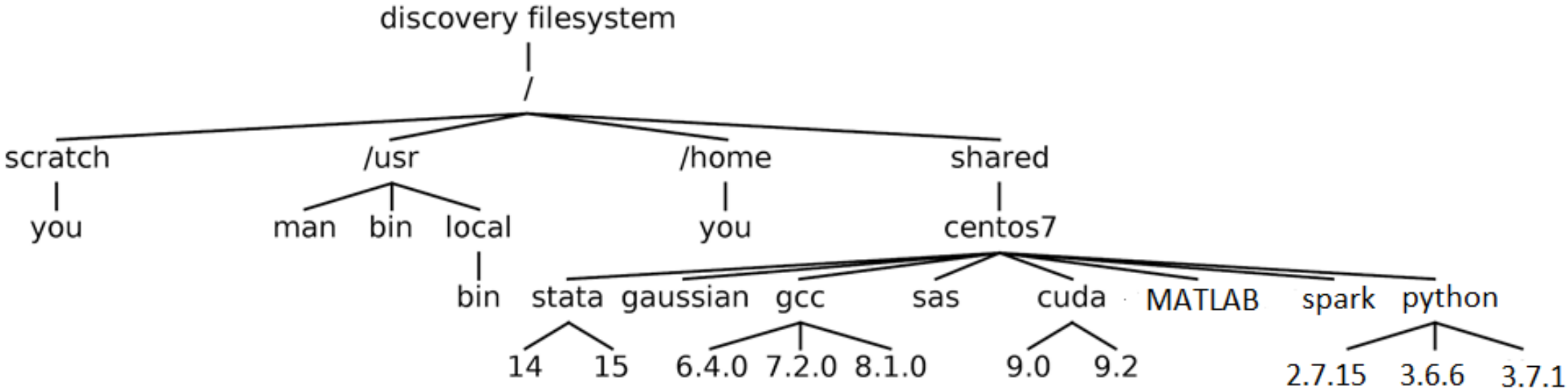
- X11 forwarding provides one way for an SSH connection to access graphical applications running on the SSH server.

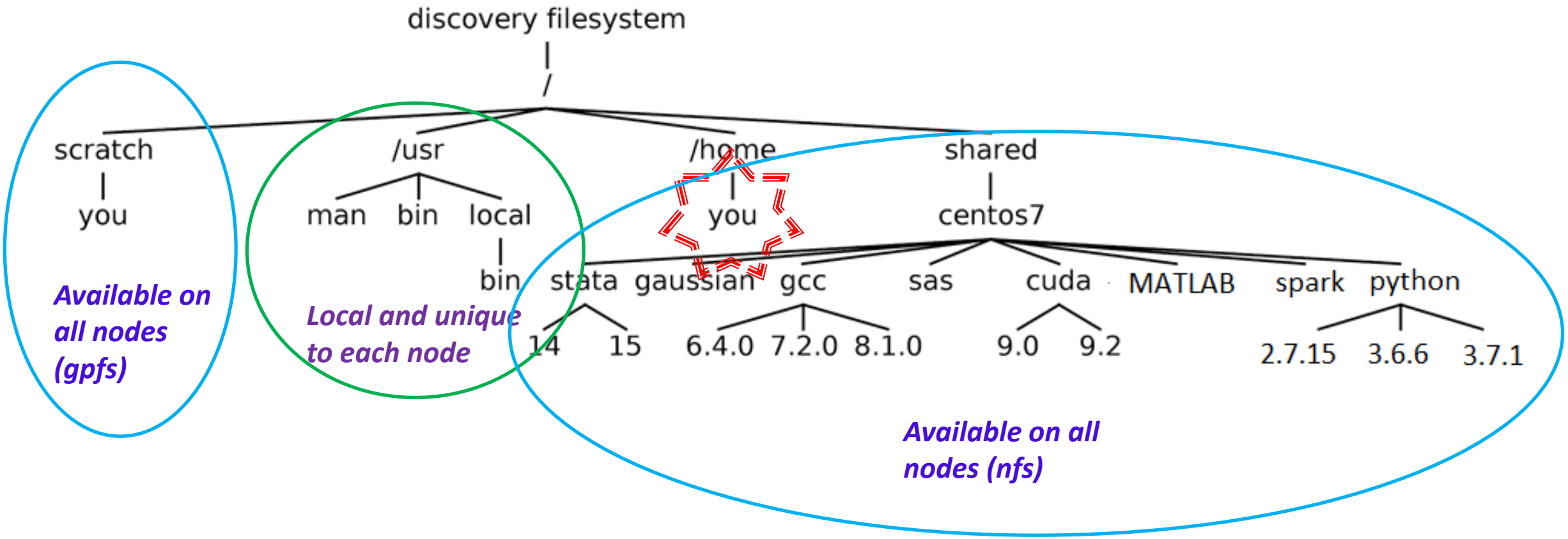
For example:

$ ssh -X hliu@login.discovery.neu.edu

Navigation: It's a tree!

- Everything in Linux is a tree
  - Filesystem
  - Process table
- Discovery filesystem




discovery filesystem
/
scratch
you
/usr
man
bin
local
bin
/home
you
shared
centos7
stata
14
15
gaussian
6.4.0
7.2.0
8.1.0
gcc
sas
cuda
9.0
9.2
MATLAB
spark
2.7.15
python
3.6.6
3.7.1
Available on all nodes (gpfs)
Local and unique to each node
Available on all nodes (nfs)

Filesystem navigation: viewing

- ls -- list files
  - Options: `ls -[lart1d]` most common options
    - `-l:` long listing
    - `-a`: all files (including hidden files, (ones that begin with a .))
    - `-r:` show in reverse order (default sorting order is alphanumeric)
    - `-t`: sort by time modified
    - `-1` show all in 1 column
    - `-d` show directory names but not their content

Filesystem navigation commands

- cd : change directory
- pwd : print working directory
- mkdir newdir (make directory “newdir”)
- mkdir -p newdir/subdir1/subdir2/subdir3 (make path newdir/subdir1/subdir2/subdir3)

File and directory manipulation

copy : `cp [-rpiv]`

- files: `cp file1 file2`
- Directories: `cp -r dir1 dir2`

move/rename: `mv oldpath/oldname newpath/newname`

Remove: `rm -[irf] / rmdir -[pmv]`

- Files: `rm file1 file2 file3 …. filek filen`
- Directories: `rmdir dir1 dir2 dir3 … dirk dirn`

Viewing file content

`cat`: print content of file to the stdout (more on streams later)

`more`: print content of file to the screen but pause between pages

`less`: like more but with more options

`vim`: open file in an editor

More file utilities: heads & tails

Show the first N lines of a file:

```
head -N filename
```

Show the last N lines of a file:

```
tail -N filename
```

Follow a file as it is being written to:

```
tail -f filename
```

grep is a very powerful utility and has many options

`grep --help` will show all options

Most common are:

`-l just show list of files that match a pattern`

`-i ignore case`

`-n show line numbers where the pattern is found`

`-h do not show filename in front of the line where the pattern was found`

`-v pattern :`

Permissions and ownership

Long listing (ls -l) shows ownership and permissions on a file or a directory

```
drwxr-xr-x 4 hliu users  367 Feb 11 14:32 python
```

Type, user perms, group perms, others' perms, owner, group, size, date modified

```
-rwxr-xr-x 1 hliu users 66 Mar 27 2018 hello.py
```

Permissions

permissions/owners

Change permissions: chmod [-R] mode file

- Mode: permissions: r-read, w-write, x-execute for owner (user) group, others
- chmod ugo+rwx file: makes file readable, writable and executable by owner, their, group and everyone else
- chmod u=rwx,g=rx,o=x makes the file or directory readable, writable, executable by user, readable & executable by group and executable only by others.

Cont'd

Octal method: x := 1, w:=2, r:=4

```
777 => rwxrwxrwx

755 => rwxr-xr-x

644 => rw-r--r--

750 => rwxr-x---
```

archives

Sometimes it may be necessary to pack up an entire directory into a single file, for archiving, reducing the number of files in the filesystem, or easy transfer

Create an archive from directory foo:

```
tar cf foo.tar ./foo
```

- `c: create, f: name of file to crate`

Unpack an archive:

```
tar xvf foo.tar
```

- `x: xtract, v: verbose, f: filename of the archive`

Cont'd

Often tar archives are compressed to form what is known in the uinx/linux world as a "tarball"

To create or unpack a tarball, use the z option

```
tar czf foo.tgz ./foo
```

```
tar zxf foo.tgz
```

- `zip creates archives compatible with windows archives`

compression

Compressing is not limited to tar files. Any file can be compressed using several compression utilities. Difference is in the compression algorithm

- `gzip`
- `bzip2`
- `zip`

What modern supercomputers look like

The IBM Blue Gene/P supercomputer "Intrepid" at Argonne National Laboratory runs 164,000 processor cores using normal data center air conditioning, grouped in 40 racks/cabinets connected by a high-speed 3-D torus network.

See details: https://en.wikipedia.org/wiki/Supercomputer

MGHPCC

- The MGHPCC is operated by the Massachusetts Green High Performance Computing Center Inc. whose members are the five founding institutions: Boston University, Harvard University, the Massachusetts Institute of Technology, Northeastern University and the University of Massachusetts.
- http://www.mghpcc.org


MGHPCC

Overview of Discovery

- Login nodes: login-00, login-01
- Node for transferring data: xfer → support scp and rsync
- More than 1,000 compute nodes,
- 8 public partitions, several private partitions
- Operation system: CentOS 7.5
- 24 hours time limit per user
- Discovery Cluster runs SLURM as the workload manager
  - SLURM: Simple Linux Utility Resource Manager
- Storage locations: /home and /scratch
- Software platform and modules

Computing Resources (updated to 02/2019)

Partitions	Usage	# of nodes	Node Description	Features
general	CPU	316	Public	24 hours limit
gpu	1 GPU per node	48	Public	24 hours limit
test	CPU	2	Public	1 hours limit
interactive	CPU	2	Public	8 hours limit
phi	CPU (now)	8	Public	24 hours limit
fullnode	CPU	408	Specifically Apply	24 hours limit
multigpu	8 GPU per node	20	Specifically Apply	24 hours limit
infiniband	Infiniband backplane	64	Specifically Apply	24 hours limit
private		~ 240	Owned by the labs of Faculty Professor	no time limit

- Total nodes are ~ 1100; the number of CPU is over 22,000 physically.

Storage Resources

Name	Intent	File System	Total Size	Maximum Usage	Data Lifespan
/home	Long-term storage of files	NFS	9 TB	100GB hard limit per user	Unlimited
/scratch	User data during computational runs	GPFS	1.1 PB	1TB soft limit per user	30 days
/shared	Software modules			Only Read	

Data Transfer

- Command line: scp, rsync
- Graphically: MobaXterm, FileZilla
- Run the commands on your local machine:

    $ scp (-r for folder) source destination

    $ rsync -auv local remote

Remote address:

username@xfer.discovery.neu.edu:/path

sftp://xfer.discovery.neu.edu

Data transfer – FileZilla or MobaXterm

- Download FileZilla
- Host: sftp://xfer.discovery.neu.edu
- Username: your username
- Password: your password
- Port: 22, by default
- Supports remote editing

Software Platform

- Operation System: CentOS 7.5
- Compilers: GNU gcc 4.8.5 as default: gcc, g++, gfortran
- Libraries: BLAS, LAPACK, ScaLapck
- MPI: OpenMPI 3.1.1 / 3.1.2 / and previous versions
  - mpicc, mpic++/mpicxx, mpif77, mpif90, mpirun, mpiexec
- CUDA: cuda 9.0 and 9.2
- Python: 2.7.15 and 3.x.x, latest is 3.7.1

Software and Packages on Discovery

- Molecular Dynamics:
  - GROMACS, AMBER, NAMD/VMD, CHARMM, NWChem, LAMMPS
- Computational Chemistry, Molecular Modeling, Bioinformatics, Materials Science, Nanostructures and related areas:
  - Gaussian, NWchem, GAMESS, GROMACS, NBO, MOPAC, Quantum ESPRESSO, ORCA, BerkeleyGW, Avogadro, OpenBabel
- Atomistic, Particle Physics: OVITO, ROOT
- Statistical and data analysis: SAS, R, STATA
- Computational Fluid Dynamics: Ansys Fluent, OpenFOAM
- Scientific Computing: MATLAB
- Machine Learning (Deep Learning) Packages: scikit-learn, TensorFlow, Theano, Keras, etc..
- ……..

Modules

- Softwares are organized into modules

```
# check all the available modules
    $ module avail
# check all the available modules whose name starts with abc
    $ module avail abc
# show detailed description and pre-requisites of module abc
    $ module whatis abc
# load module abc so that you can use it
    $ module load abc
# list currently loaded modules
    $ module list
```

.bashrc

- .bashrc is a shell script that Bash runs whenever it is started interactively. It initializes an interactive shell session.
- You can put commands here to set up the shell for use in your particular environment, or to customize things to your preferences.
- $ ls -la → to show the hide files and see .bashrc
- vi .bashrc
- Exit and login again, the new .bashrc becomes effective.
- This is just for advanced using. If you don't know how to modify it, don't do anything in .bashrc.

SLURM = SIMPLE LINUX UTILITY FOR RESOURCE MANAGEMENT

http://slurm.schedmd.com

- Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.
- Slurm requires no kernel modifications for its operation and is relatively self-contained.
- As a cluster workload manager, Slurm has three key functions.
  - First, it allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work.
  - Second, it provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes.
  - Finally, it arbitrates contention for resources by managing a queue of pending work.

SLURM Commands

- sinfo -- Report system status (nodes, queues, etc.)
- sbatch – Submit script for later execution (batch mode)
- srun – Create a job allocation and launch a job step (interactive mode, and typically an MPI job)
- scancel -- Cancel a running or pending job or step
- squeue -- Report job and job step status

Running jobs - Interactive mode

- Run a parallel job on cluster managed by Slurm.
- If necessary, srun will first create a resource allocation in which to run the parallel job.
- For interactive mode, run the following command to get a compute node on Discovery for parallel jobs:

  $ srun -N #of nodes -n #of tasks -p partition

  --mem=memory_to_use command to run

- https://slurm.schedmd.com/srun.html to see more.

For example:

$ srun -n 8 --mem=4gb -p general --x11 --pty /bin/bash

Explanation of commands:

- srun - Run a slurm job.
- -n 2 - Specify that two tasks will be run, so specify at least 2 CPU cores.
- --mem=4gb - Specify that a minimum of 4GB of memory should be allocated to the job.
- -p general - Request nodes from the general partition.
- --x11 - Enable graphical interfaces in the compute nodes through the SSH connection using the X Window System.
- --pty /bin/bash After provisioning the job, start a pseudo terminal and then open the standard Bash Unix shell.

Running jobs - Batch mode

- Submit to the cluster for executing in the background
- $ sbatch configfile
- Configuration file
  - ✓ Parameters, what resource you want
  - ✓ Commands, instructions to be executed
- A little more effort
- For longer and bigger jobs
  - Don’t suggest to submit batch file for the exercises in this course.

Sample batch script

```
#!/bin/bash - shebang, tells the executor

## Both in single-letter and whole-word formats, e.g. -N 1 and --nodes=1
## Normal configurations
#SBATCH --job-name=test
#SBATCH --output=test.out
#SBATCH --error=test.err
#SBATCH --time=01:00:00

## Parallel configurations
#SBATCH -p gpu
#SBATCH -N 2
#SBATCH -n 4 OR --ntasks-per-node=4
#SBATCH --cpus-per-task=4
#SBATCH --gres=gpu:1

## for example: run a mpi parallelized C code compiled with mpicc
mpirun  -n 32 helloworld
```

Be Sure to Keep in Mind

- Don’t run any jobs (e.g. python) on login nodes.
- Otherwise your account will be banned for a period of time, depending on the extent of your abuse.
- Always using the below to run the interactive works on compute node:
  - $ srun -p general --pty /bin/bash

Special tips – Install Python package

- Use the anaconda on cluster

  $ module load anaconda/xxx

  $ conda install packagename

- Install it on your own (download the source code)

  >>> import sys

  >>> sys.path.append(path_to_the_package)

- Install your own anaconda/virtual env/…

User-Contributed Tutorials

In addition to this guide, other members of the NU community have kindly volunteered to create additional tutorials. We encourage everyone to consult these other resources, since they complement the current document.

- *Introduction to Discovery for Windows Users* by Cuneyt Eroglu (video) : https://www.youtube.com/watch?v=y81iaRXzO9o
- *Step-by-step Examples and Demos for Discovery* from Kane Group (wiki) : https://github.com/NEU-ABLE-LAB/NEU_RC_HELLO

The End!

Assignment:

1) Login to Discovery and practice the commands I ran in this class.
2) Remember: **Don’t run any jobs (python) on login nodes!!**