CBC Data Therapy

High-Performance Computing Basics

Xanadu Cluster

Development of models begins at small scale.

Working on your laptop is convenient, simple.

Actual analysis, however is slow.

"Scaling up" typically means a small server or fast multi-core desktop.

Speed exists, but for very large models, not significant.

Single machines don't scale up forever.

For the larger problems/models, a different approach is required

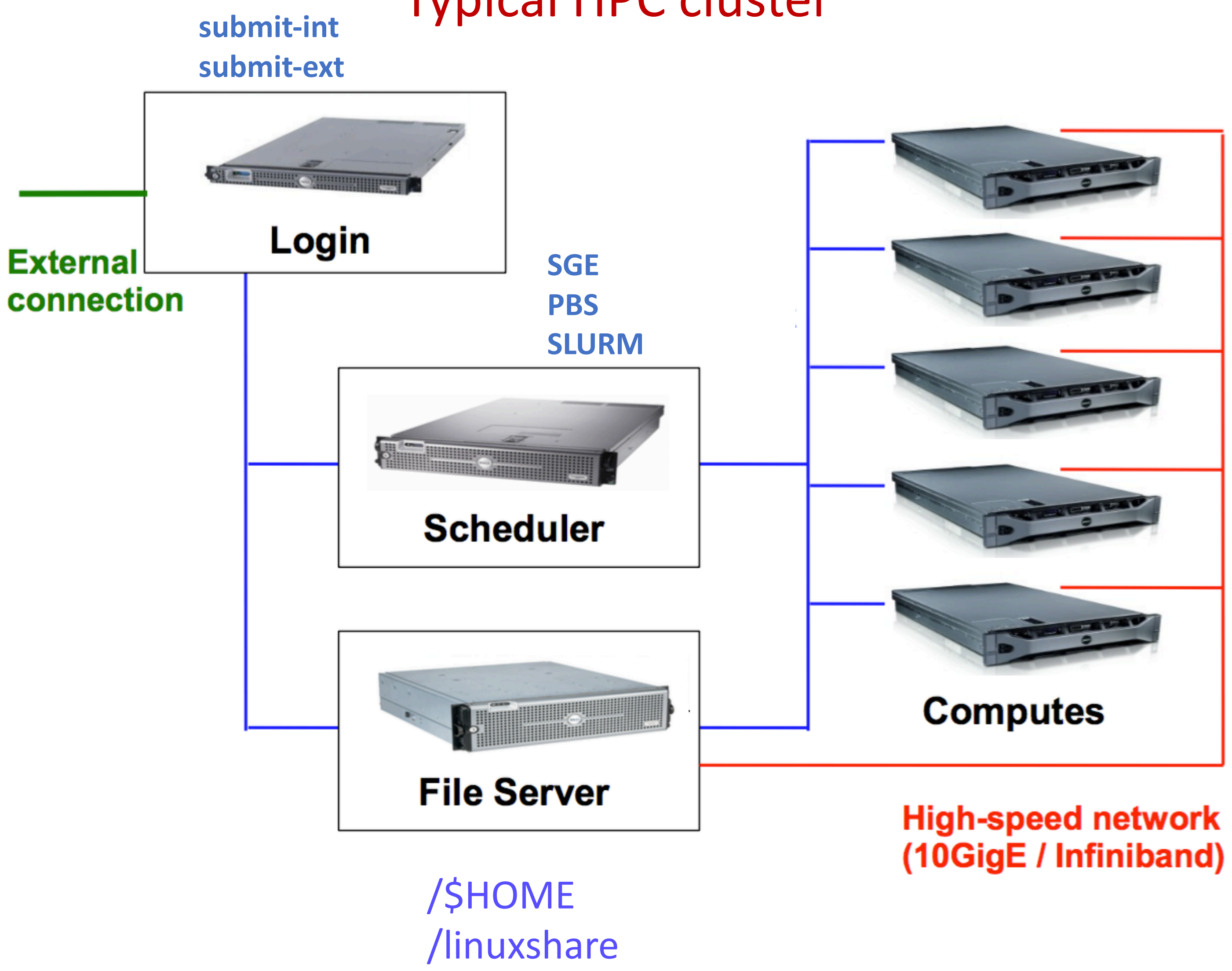
High-performance computing (HPC)

High-Performance computing involves mainly distinct computer processors working together on the same problem/calculation.

Large problem/calculations are divided into smaller parts and distributed among the many computers.

HPC is a cluster of quasi-independent computers which are coordinated by a central scheduler.

Typical HPC cluster

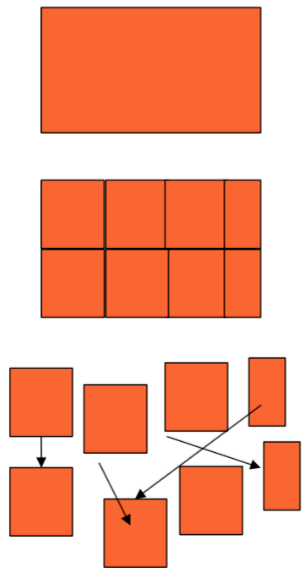


SGE:
Sun Grid Engine
PBS:
Portable Batch System
SLURM:
Simple Linux Utility for Resource Management

Performance comes at a price: **Complexity**

- Applications must be written specifically to take advantage of distributed computing

- Debugging becomes more of challenge



Applications must be written specifically to take advantage of distributed computing.

- Explicitly split your problem into smaller "chunks"
- "Message passing" between processes

In Summary

- Why HPC

– A huge number of computational and memory requirements

– Cannot be afforded by a PC efficiently

– Speeds and feeds are the keywords

- Who uses High-Performance Computing

– Research institutes, universities and government labs

    - Weather and climate research, bioscience, energy, military etc.

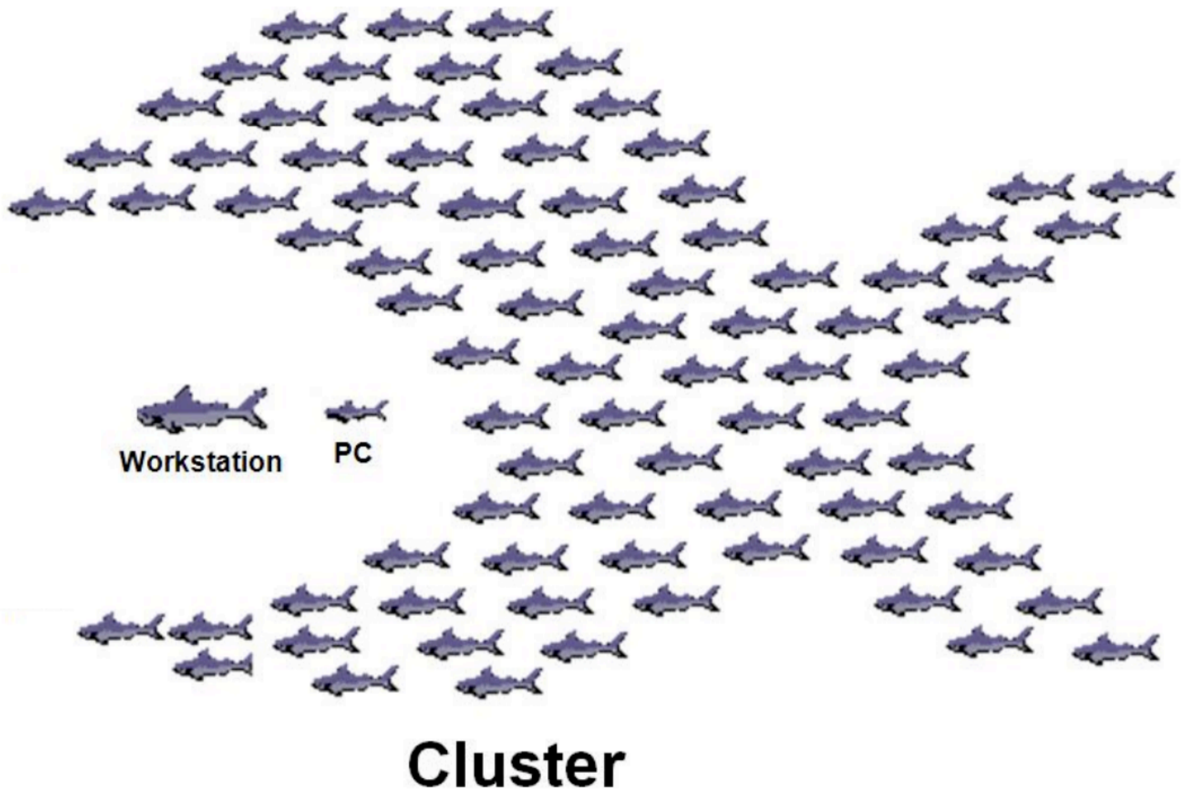
– Engineering design: more or less every product we use

    - Automotive, aerospace, oil and gas explorations, digital media, financial simulation
    - Mechanical simulation, package designs, silicon manufacturing etc.

- Similar concepts

– Parallel computing: computing on parallel computers

– Super computing: computing on world 500 fastest supercomputers



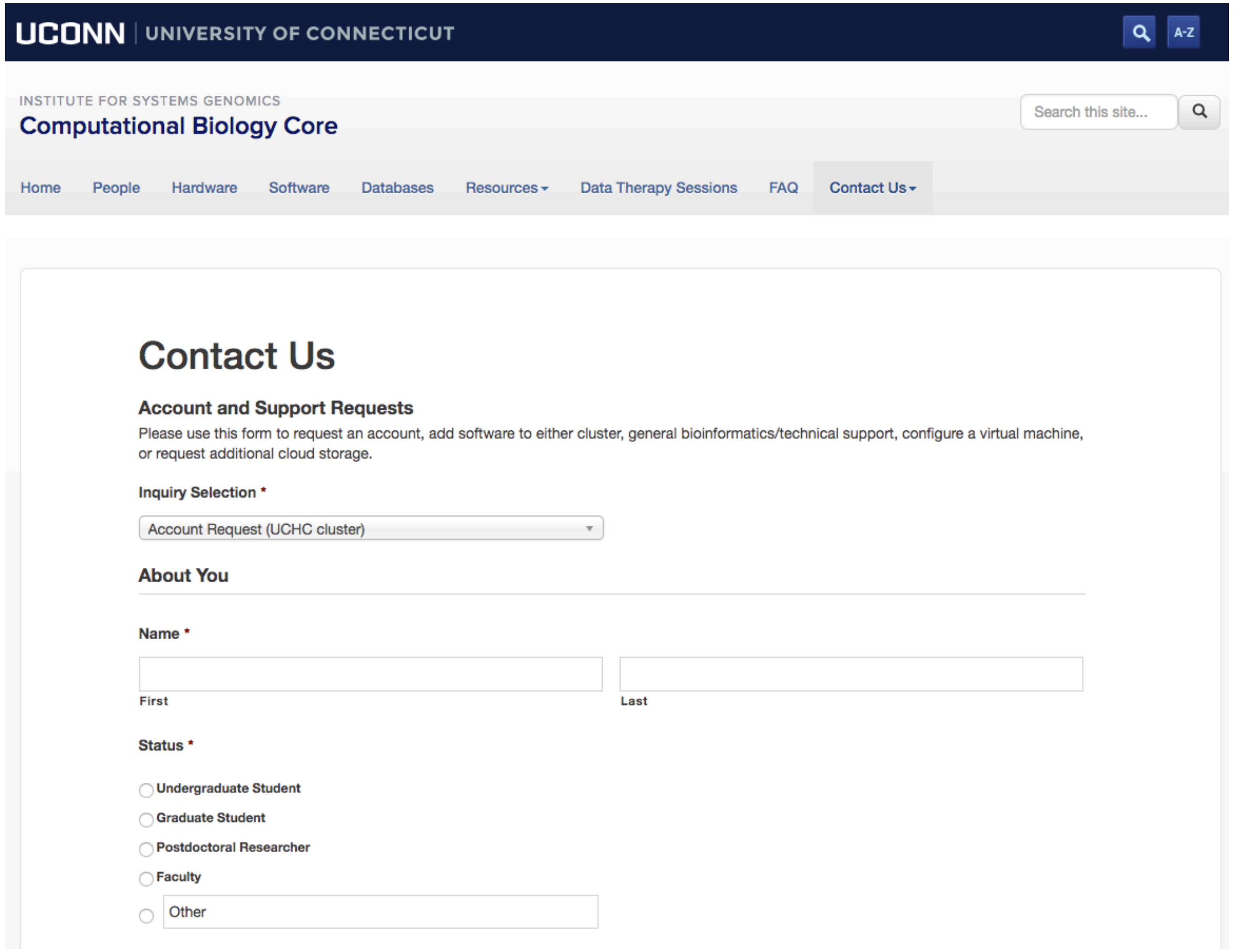
Parallel Computing on a Large Number of Servers is More Efficient than using Specialized Systems

HPC at UCONN

BBC (Storrs): SGE - Research and Teaching

HPC1 (UCH) : PBS - Advanced Research

Xanadu (UCH): SLURM - Advanced research



Contact Us

Account and Support Requests

Please use this form to request an account, add software to either cluster, general bioinformatics/technical support, configure a virtual machine, or request additional cloud storage.

Inquiry Selection *

Account Request (UCHC cluster)

About You

Name *

First

Last

Status *

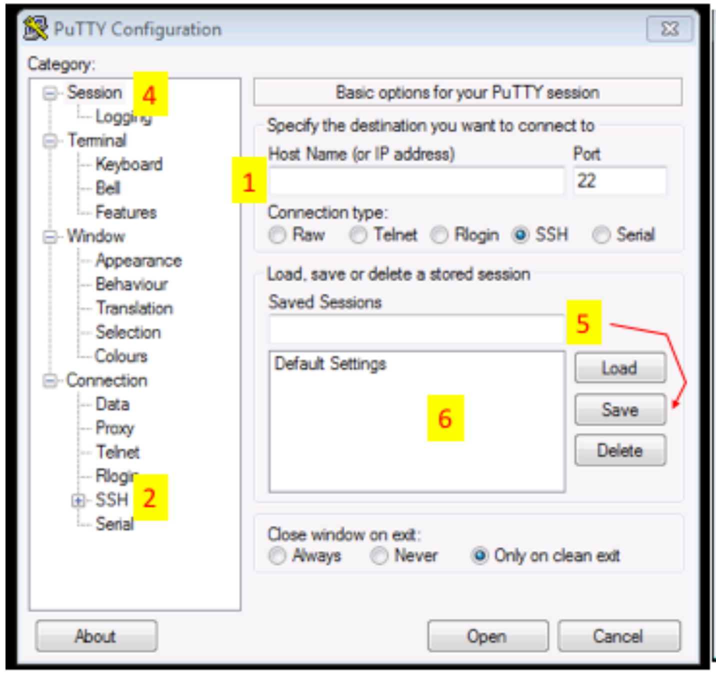
- Undergraduate Student
- Graduate Student
- Postdoctoral Researcher
- Faculty
- Other

Connecting to Xanadu

Mac : Terminal : ssh username@xanadu-submit-ext.cam.uchc.edu

Windows : Putty

Window 1



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

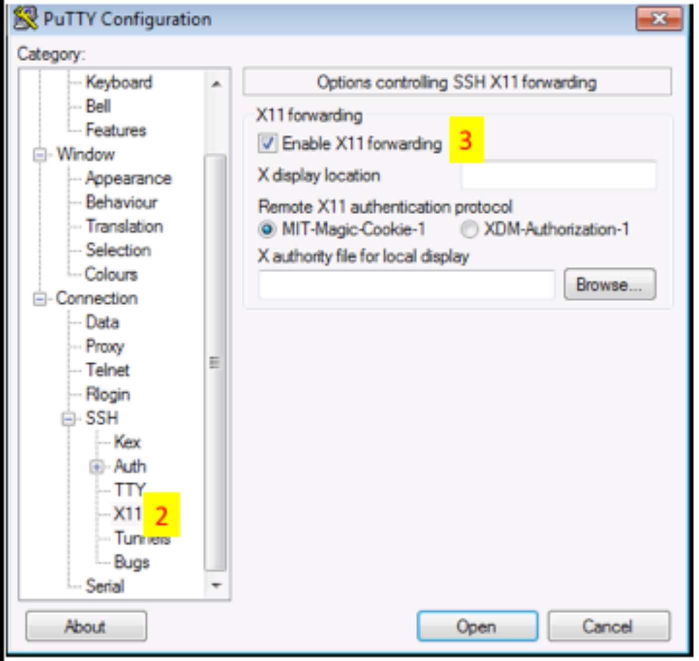
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

Window 2



Options controlling SSH X11 forwarding

X11 forwarding

[✓ Enable X11 forwarding]

X display location

Remote X11 authentication protocol: [✓ MIT-Magic-Cookie-1] XDM-Authorization-1

X authority file for local display

Open Putty it will open window1.

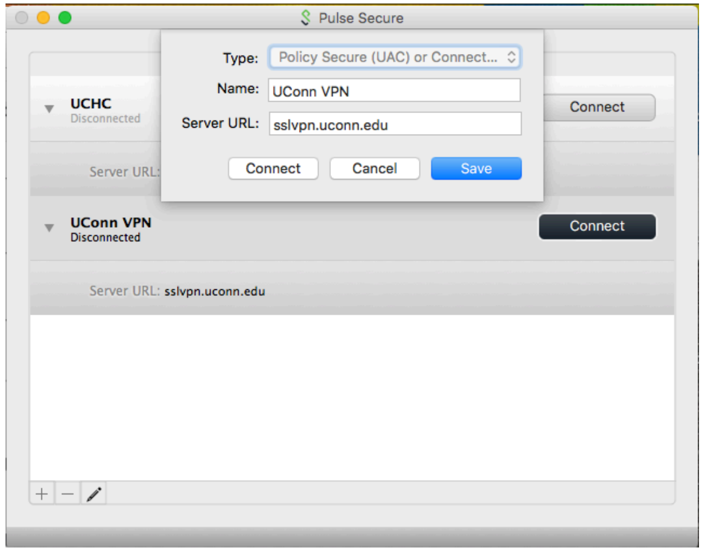
1. Provide host name e.g. username@xanadu-submit-ext.cam.uchc.edu
2. Expand SSH tab and select X11 (shown in window2)
3. Enable X11 forwarding by selecting it. (window2)
4. Scroll up the left panel and select Session.(window1)
5. Name your session e.g. BBC_cluster and click save tab to save.
6. Your session name should appear in saved sessions.

Double click on your session name to connect to server with SSH session.

Login: From outside the network

Use VPN (Open Pulse secure)

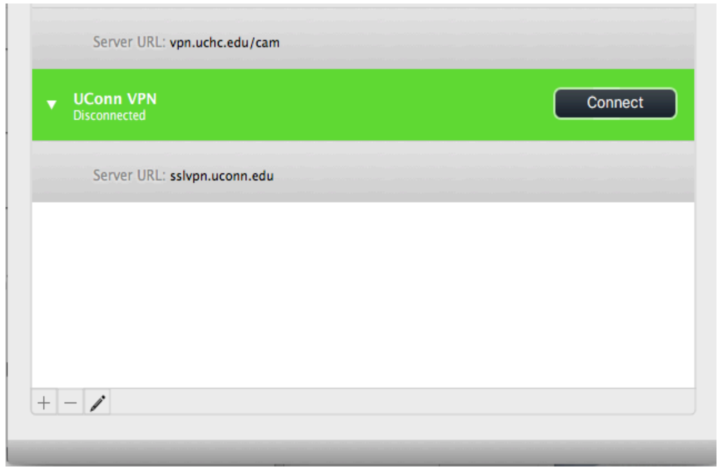
1. Open Pulse secure
2. Add new connection
3. Set Server URL to : sslvpn.uconn.edu
4. Save
5. Connect and login with NetID and Passwd



Type: Policy Secure (UAC) or Connect...
Name: UConn VPN
Server URL: sslvpn.uconn.edu

UCHC
Disconnected
Server URL:

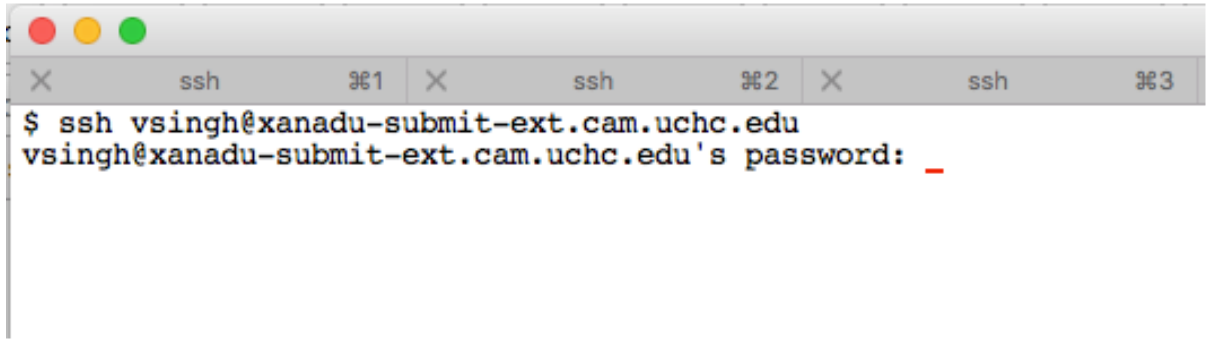
UConn VPN
Disconnected
Server URL: sslvpn.uconn.edu



Server URL: vpn.uchc.edu/cam

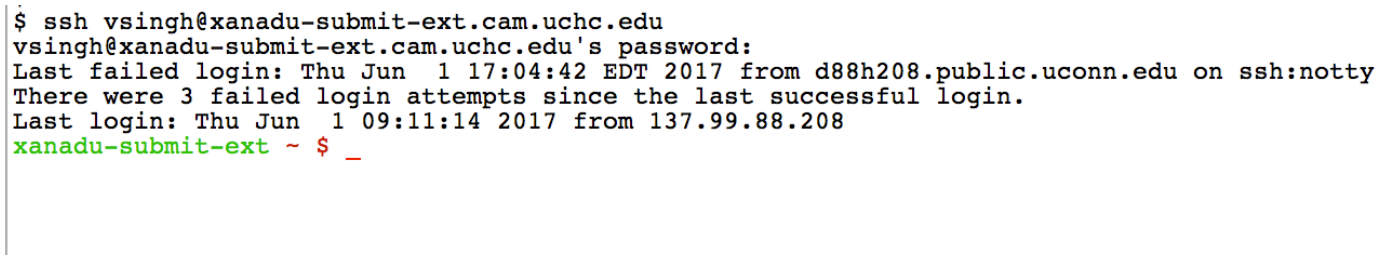
UConn VPN
Disconnected
Server URL: sslvpn.uconn.edu

Login: (using terminal on mac)



```
$ ssh vsingh@xanadu-submit-ext.cam.uchc.edu
vsingh@xanadu-submit-ext.cam.uchc.edu's password: _
```

Logged on ext-submit node:



```
$ ssh vsingh@xanadu-submit-ext.cam.uchc.edu
vsingh@xanadu-submit-ext.cam.uchc.edu's password:
Last failed login: Thu Jun  1 17:04:42 EDT 2017 from d88h208.public.uconn.edu on ssh:notty
There were 3 failed login attempts since the last successful login.
Last login: Thu Jun  1 09:11:14 2017 from 137.99.88.208
xanadu-submit-ext ~ $ _
```

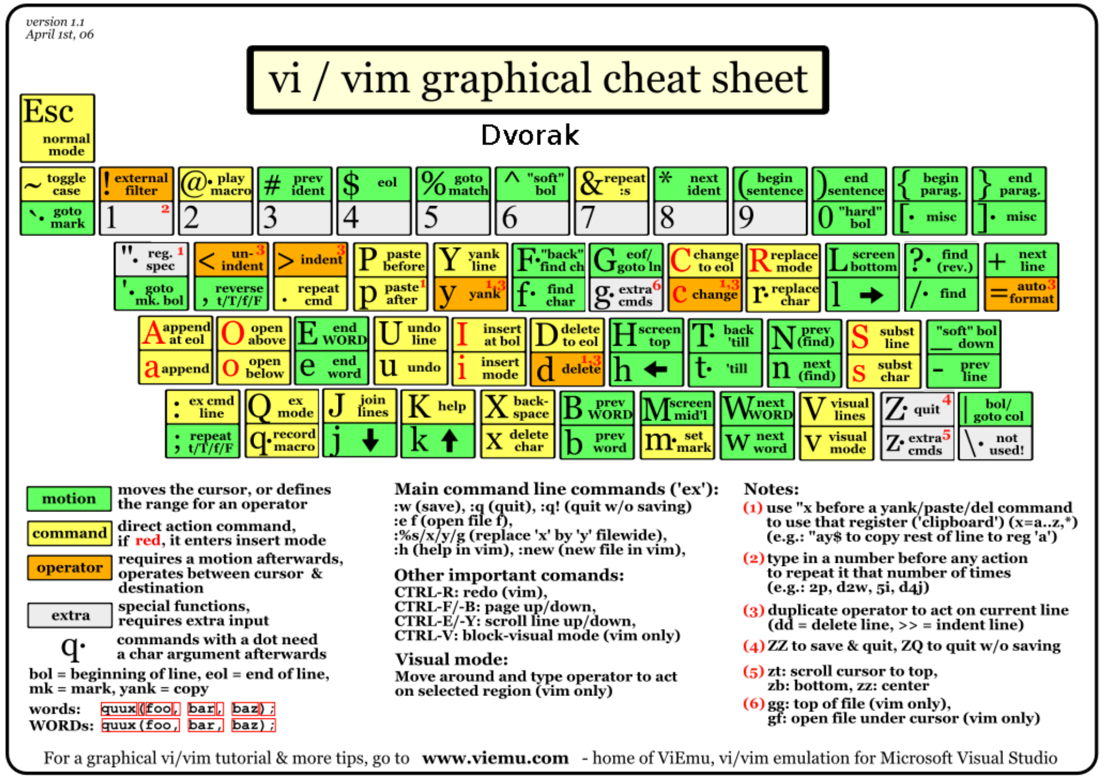
Common Linux commands

`pwd`	:	Present Working directory
`cd destination`	:	Change directory to destination
`cd`	:	Change directory to $HOME
`ls`	:	List contents of directory
`cp source/file destination/file`	:	Copy file from source in destination folder
`mv source/file destination/file`	:	Move file from source to destination folder
`mv name name2`	:	Rename file from name to name2
`touch filename`	:	Create an empty file with name filename
`mkdir directory`	:	Make directory
`rm file`	:	delete file
`rm –r directory`	:	delete file with its content
`~`	:	Home directory
`cat`	:	Read contents of file
`less`	:	Contents of file, scroll, q to quit it
`head -10 file`	:	first 10 lines of file
`tail -10 file`	:	Bottom 10 lines of file

Resources: http://linuxcommand.org/writing_shell_scripts.php

Common Linux commands to edit file

```
vi filename        : Open file in vim to edit
esc i              : Insert or edit file
esc q !            : quit file without saving changes
esc w q !          : Save and quit file
esc dd
```



motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
q·	commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: quux(foo, bar, baz);
WORDs: quux(foo, bar, baz);

Main command line commands ('ex'):
:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important comands:
CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:
(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay$ to copy rest of line to reg 'a')
(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)
(3) duplicate operator to act on current line (dd = delete line, >> = indent line)
(4) ZZ to save & quit, ZQ to quit w/o saving
(5) zt: scroll cursor to top, zb: bottom, zz: center
(6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to **www.viemu.com** - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Software/tool/packages on cluster

Environment Modules:

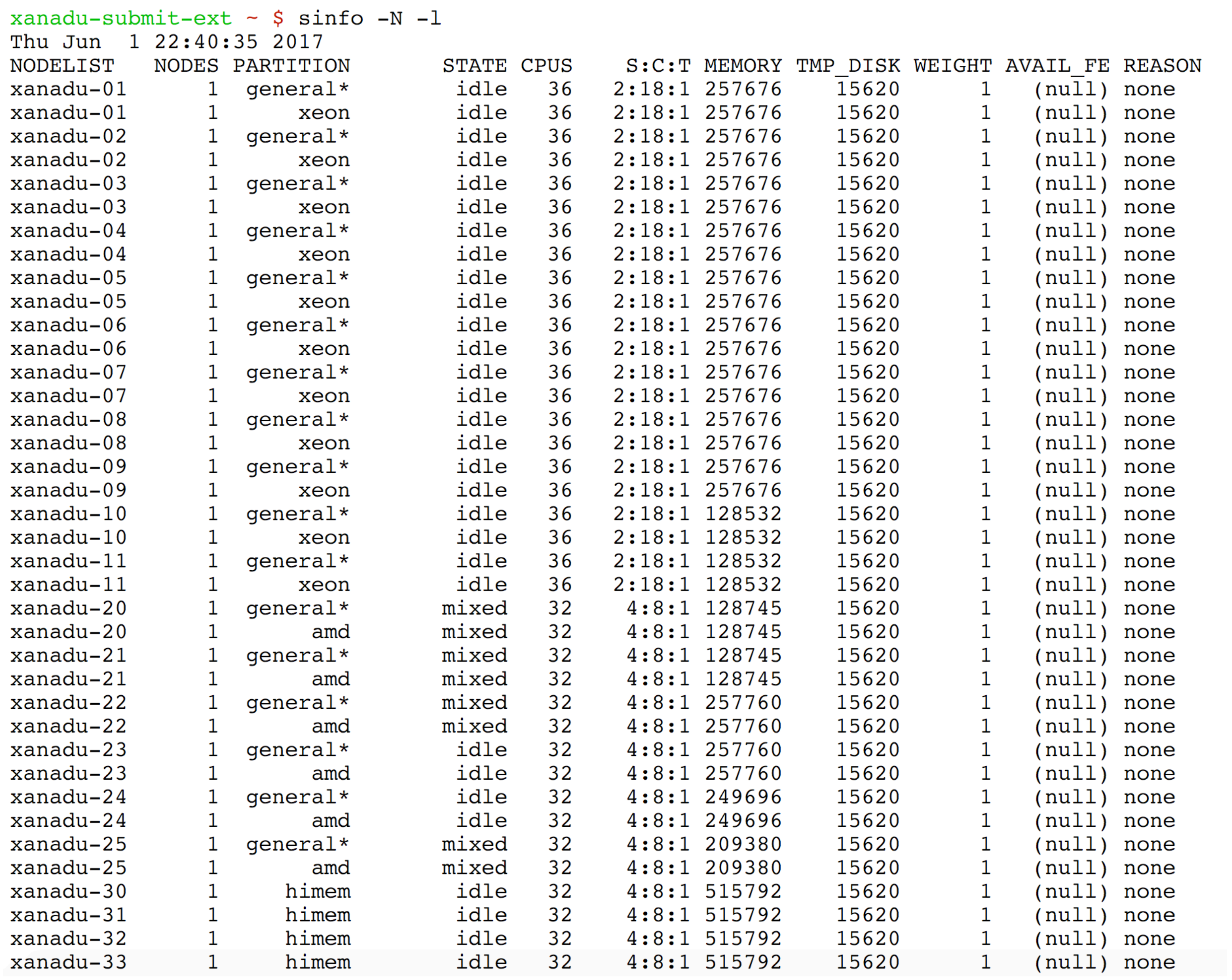
The Environment Modules package provides for the dynamic modification of a user's environment via modulefiles.

```
module avail                  : List modules that are available
module load modulefile        : Loads the module to user environment
module list                   : List modules that are loaded
module unload modulefile      : unloads module from user environment
module display modulefile     : Displays information on module
swap [modulefile1] modulefile2 :Switch loaded modulefile1 with modulefile2.
```

Xanadu Resources

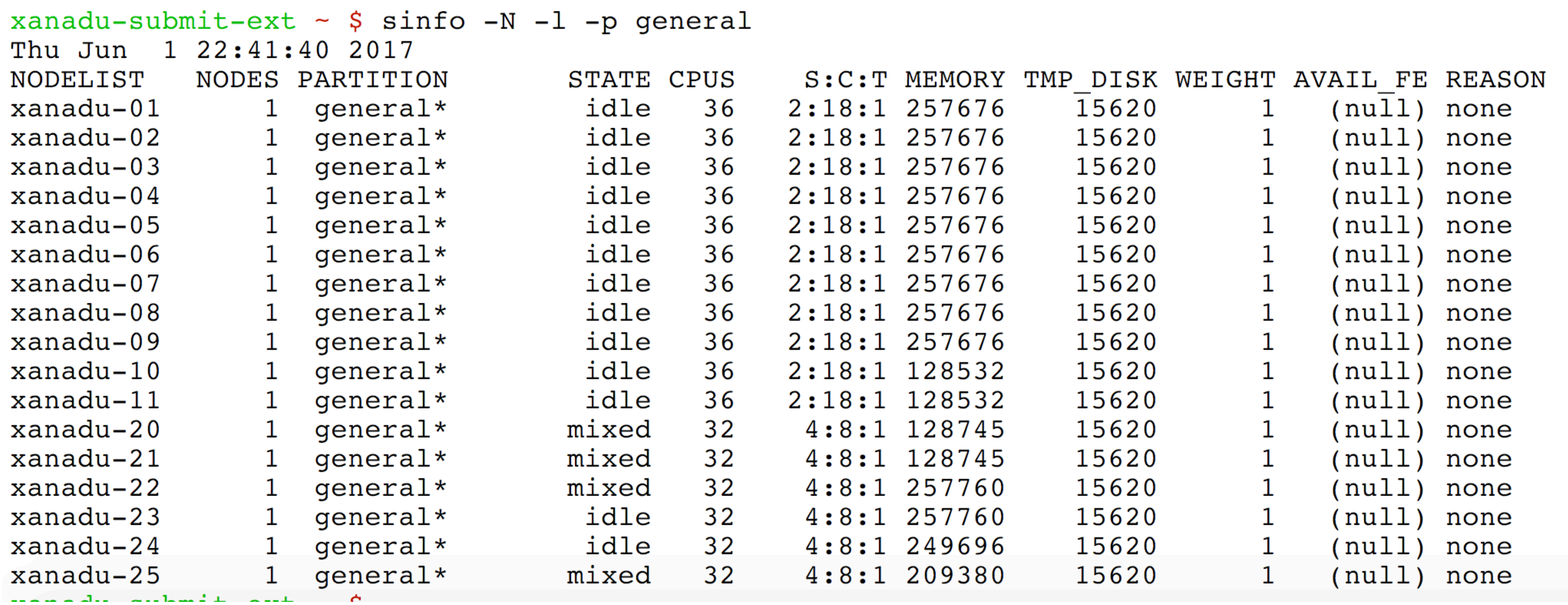
Partitions

```
xanadu-submit-ext ~ $ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*     up   infinite      4    mix xanadu-[20-22,25]
general*     up   infinite     13   idle xanadu-[01-11,23-24]
xeon         up   infinite     11   idle xanadu-[01-11]
amd          up   infinite      4    mix xanadu-[20-22,25]
amd          up   infinite      2   idle xanadu-[23-24]
himem        up   infinite      4   idle xanadu-[30-33]
xanadu-submit-ext ~ $
```



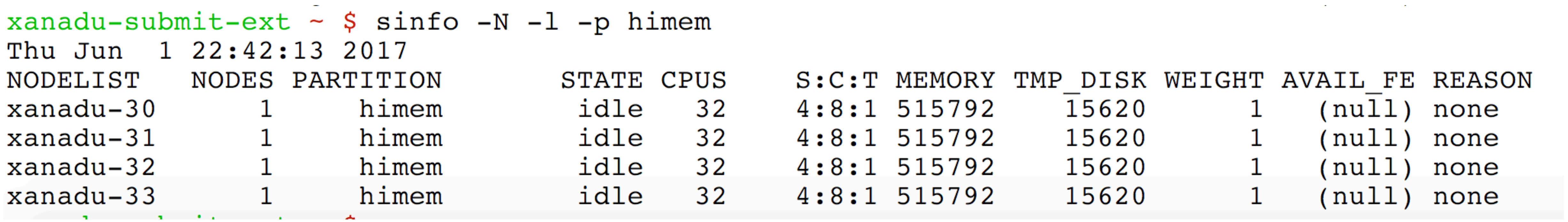
```
xanadu-submit-ext ~ $ sinfo -N -l
Thu Jun  1 22:40:35 2017
NODELIST    NODES PARTITION       STATE CPUS    S:C:T MEMORY TMP_DISK WEIGHT AVAIL_FE REASON
xanadu-01       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-01       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-02       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-02       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-03       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-03       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-04       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-04       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-05       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-05       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-06       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-06       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-07       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-07       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-08       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-08       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-09       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-09       1      xeon        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-10       1  general*        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-10       1      xeon        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-11       1  general*        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-11       1      xeon        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-20       1  general*       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-20       1       amd       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-21       1  general*       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-21       1       amd       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-22       1  general*       mixed   32    4:8:1 257760    15620      1   (null) none
xanadu-22       1       amd       mixed   32    4:8:1 257760    15620      1   (null) none
xanadu-23       1  general*        idle   32    4:8:1 257760    15620      1   (null) none
xanadu-23       1       amd        idle   32    4:8:1 257760    15620      1   (null) none
xanadu-24       1  general*        idle   32    4:8:1 249696    15620      1   (null) none
xanadu-24       1       amd        idle   32    4:8:1 249696    15620      1   (null) none
xanadu-25       1  general*       mixed   32    4:8:1 209380    15620      1   (null) none
xanadu-25       1       amd       mixed   32    4:8:1 209380    15620      1   (null) none
xanadu-30       1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-31       1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-32       1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-33       1     himem        idle   32    4:8:1 515792    15620      1   (null) none
```

Partition: general



```
xanadu-submit-ext ~ $ sinfo -N -l -p general
Thu Jun  1 22:41:40 2017
NODELIST    NODES PARTITION       STATE CPUS    S:C:T MEMORY TMP_DISK WEIGHT AVAIL_FE REASON
xanadu-01       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-02       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-03       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-04       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-05       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-06       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-07       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-08       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-09       1  general*        idle   36   2:18:1 257676    15620      1   (null) none
xanadu-10       1  general*        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-11       1  general*        idle   36   2:18:1 128532    15620      1   (null) none
xanadu-20       1  general*       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-21       1  general*       mixed   32    4:8:1 128745    15620      1   (null) none
xanadu-22       1  general*       mixed   32    4:8:1 257760    15620      1   (null) none
xanadu-23       1  general*        idle   32    4:8:1 257760    15620      1   (null) none
xanadu-24       1  general*        idle   32    4:8:1 249696    15620      1   (null) none
xanadu-25       1  general*       mixed   32    4:8:1 209380    15620      1   (null) none
```

Partition: himem



```
xanadu-submit-ext ~ $ sinfo -N -l -p himem
Thu Jun  1 22:42:13 2017
NODELIST   NODES PARTITION       STATE CPUS    S:C:T MEMORY TMP_DISK WEIGHT AVAIL_FE REASON
xanadu-30      1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-31      1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-32      1     himem        idle   32    4:8:1 515792    15620      1   (null) none
xanadu-33      1     himem        idle   32    4:8:1 515792    15620      1   (null) none
```

Composing a script for cluster

Script:

1. Resource request
   - number of CPUs,
   - computing expected duration,
   - amounts of RAM or disk space, etc
2. Job commands
   - describe **tasks** that must be done, software which must be run

Resource request:

```
#!/bin/bash
#SBATCH --job-name=myscript
#SBATCH -n 1
#SBATCH -N 1
#SBATCH --partition=general
#SBATCH --mail-type=END
#SBATCH --mail-user=first.last@uconn.edu
#SBATCH -o myscript.out
#SBATCH -e myscript.err
```

`#SBATCH --job-name=myscript` Is the name of your script

`#SBATCH -n 1` Request number of cores for your job

`#SBATCH -N 1` This line requests that the cores are all on node. Only change this to >1 if you know your code uses a message passing protocol like MPI. SLURM makes no assumptions on this parameter -- if you request more than one core (-n > 1) and your forget this parameter, your job may be scheduled across nodes ; and unless your job is MPI (multinode) aware, your job will run slowly, as it is oversubscribed on the master node and wasting resources on the other(s).

`#SBATCH --partition=general` This line specifies the SLURM partition (in this instance it will be the general partition) under which the script will be run

`#SBATCH --mail-user=first.last@uconn.edu` Email which the notification should be sent to

`#SBATCH --mail-type=END` Mailing options to indicate the state of the job. In this instance it will send a notification at the end

`#SBATCH -o myscript.out` Specifies the file to which the standard output will be appended

`#SBATCH -e myscript.err` Specifies the file to which standard error will be appended

more on Resource request:

```
#!/bin/bash
#SBATCH --time=10-01:00:00 # days-hh:mm:ss
#SBATCH --job-name=masurca_KG
#SBATCH --mail-user=user@uconn.edu
#SBATCH --mail-type=ALL
#SBATCH --comment=dataset_wuith_jump_libraries
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=10240 # 10GB
or #SBATCH –mem=100G
#SBATCH -o filterGTF-%j.output
#SBATCH -e filterGTF-%j.error
```

Job commands:

They are regular linux/module commands

```
echo "Hello World"
```

Final script:

```
#!/bin/bash
#SBATCH --job-name=myscript
#SBATCH -n 1
#SBATCH -N 1
#SBATCH --partition=general
#SBATCH --mail-type=END
#SBATCH --mail-user=first.last@uconn.edu
#SBATCH -o myscript.out
#SBATCH -e myscript.err

echo "Hello World"
```

save the script as `myscript.sh`

Script submission and other commands

```
sbatch myscript.sh           : Sumit script for execution
squeue                       : Status of Jobs currently running on cluster (all users)
squeue –j jobIDNUMBER        : Status of job with jobIDNumber
squeue -u UserID             : Status of all the jobs submitted by user
scancel jobID_number         : Delete job with jobID_number
scancel –u UserID            : Delete all the jobs of a user
```

Script for Array jobs

```
#!/bin/bash
#SBATCH —mail-user=user@uconn.edu
#SBATCH --mail-type=ALL
#SBATCH --ntasks=1
#SBATCH --time=00:15:00
#SBATCH --mem=1G
#SBATCH --array=1-1002%100
#SBATCH --output=fastqc_%A_%a.out
```

This line will create 1002 jobs, but it instructs slurm to limit the total number of simultaneously running jobs to 100. This avoids swamping the queue, and shares bursting level with others in the group

This will create 1002 files to catch stdin, stdout and stderr for each respective job in the array. If the array job ID is 23678, we will fine 1002 files starting with fastqc_23678_1.out … fastqc_23678_1002.out

```
cd /NGSseq/data
module load fastqc/0.11.5

echo "SLURM_JOBID: " $SLURM_JOBID
echo "SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID
echo "SLURM_ARRAY_JOB_ID: " $SLURM_ARRAY_JOB_ID

arrayfile=`ls | awk -v line=$SLURM_ARRAY_TASK_ID '{if
(NR == line) print $0}'`

fastqc $arrayfile
```

Start: Slurm job ID and increase with each array job
Slurm job ID
Array job ID : 1-1002

This will list all the files from the directory (`/NGSseq/data`) and then pick up one file at a time andthen run it through `fastqc` application.

Thank you