

# HPC basics and introduction to anadu cluster

Vijender Singh

Computational Biology Core



# A non textbook introduction...





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)

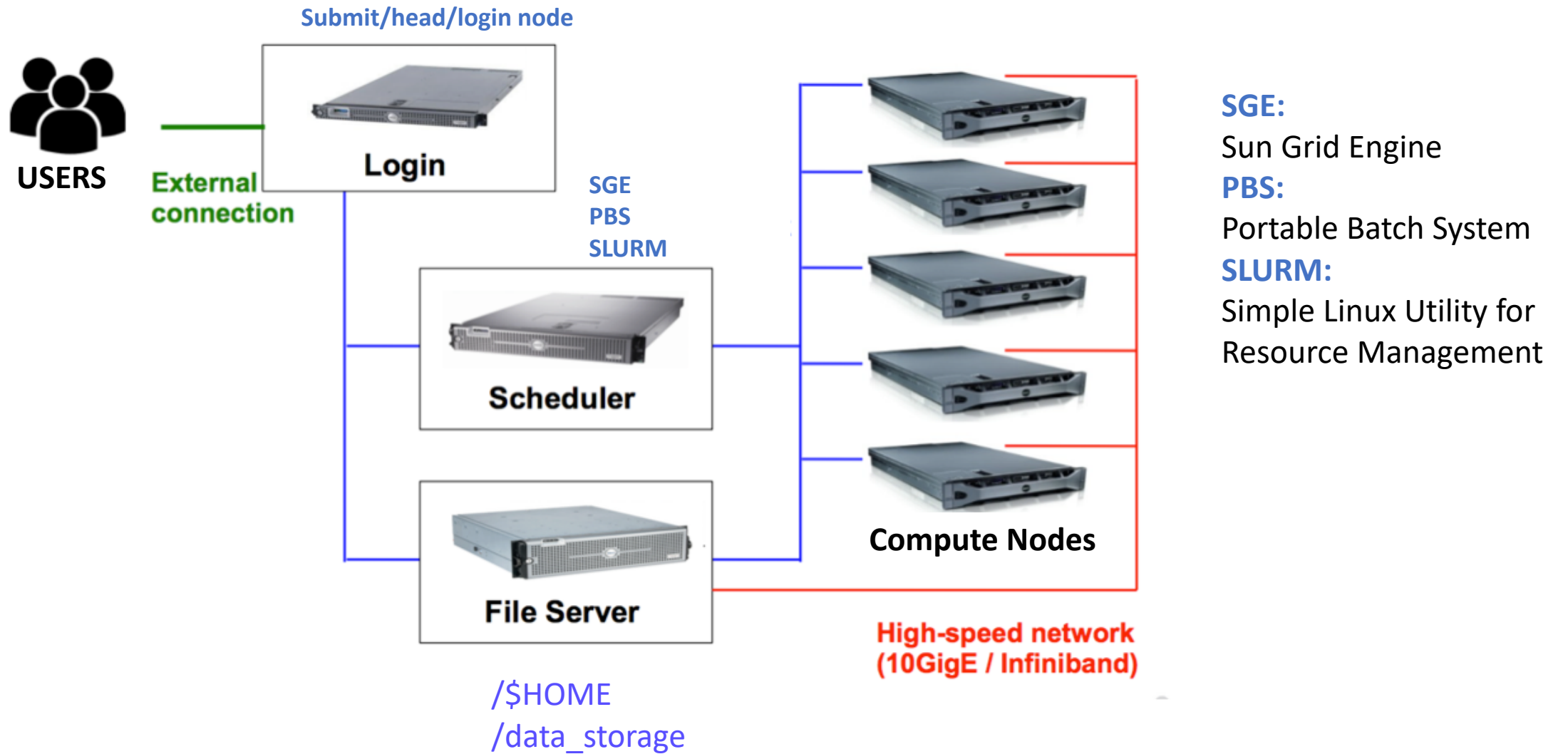
High-Performance computing involves mainly distinct computer processors working together on the same problem/calculation.

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



# HPC architecture : Xanadu (Queue/Partition info )

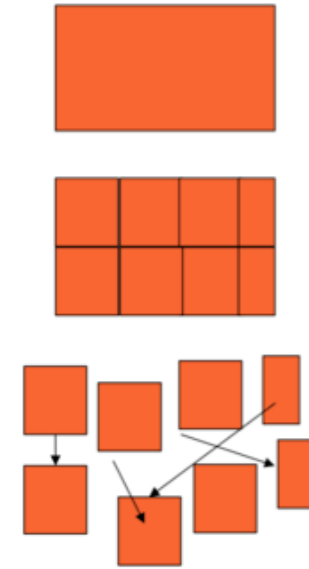


cpu : 4  
RAM: 8GB

cpu : 32 – 48  
RAM: 128 – 512 GB

## Performance comes at a price: **Complexity**

- Applications must be written specifically to take advantage of distributed computing
- Debugging becomes more of a challenge

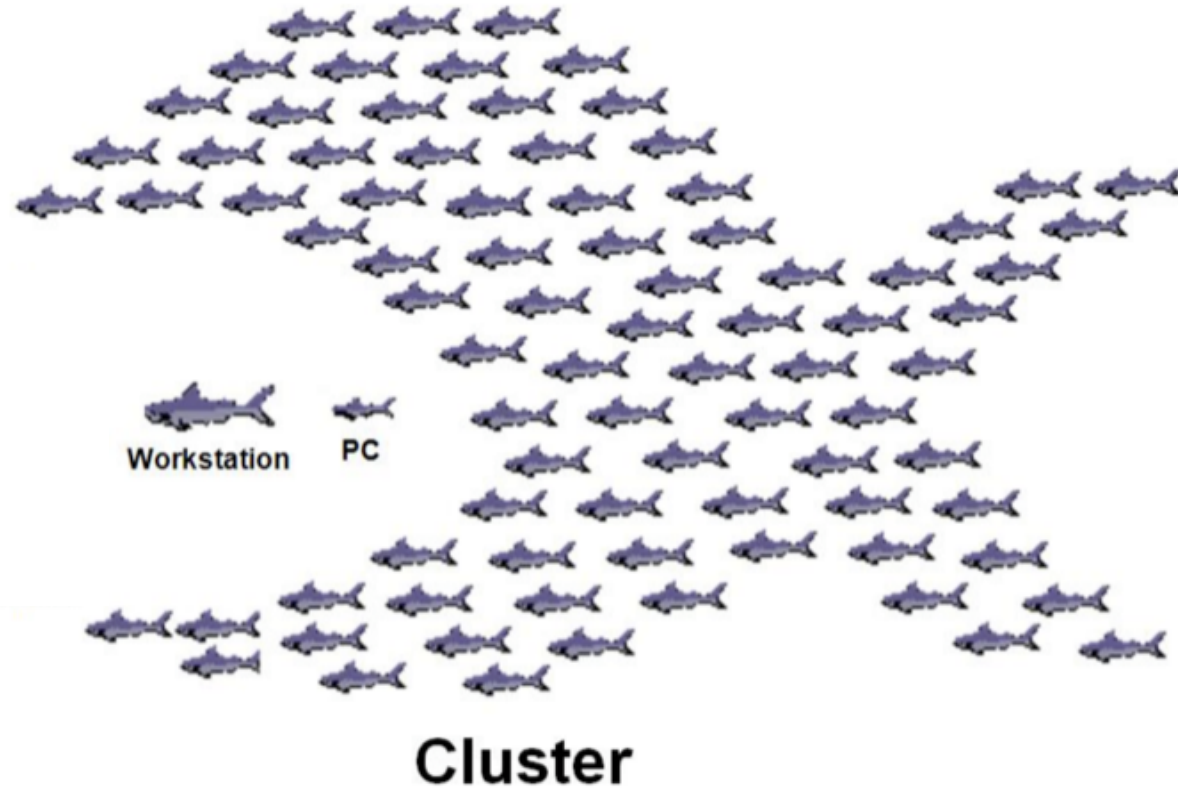


*Applications must be written specifically to take advantage of distributed computing.*

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







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



# Accessing anadu



# Get an account

<https://bioinformatics.uconn.edu/contact-us/> or google search : CBC UCONN

INSTITUTE FOR SYSTEMS GENOMICS  
**Computational Biology Core** Search this

Home People Hardware Software ▾ Databases Tutorials Resources ▾ Data Therapy Rate Plan **Contact Us ▾** FAQ

## 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 - Xanadu)- Recommended ▾

### About You

**Name \***

First

Last



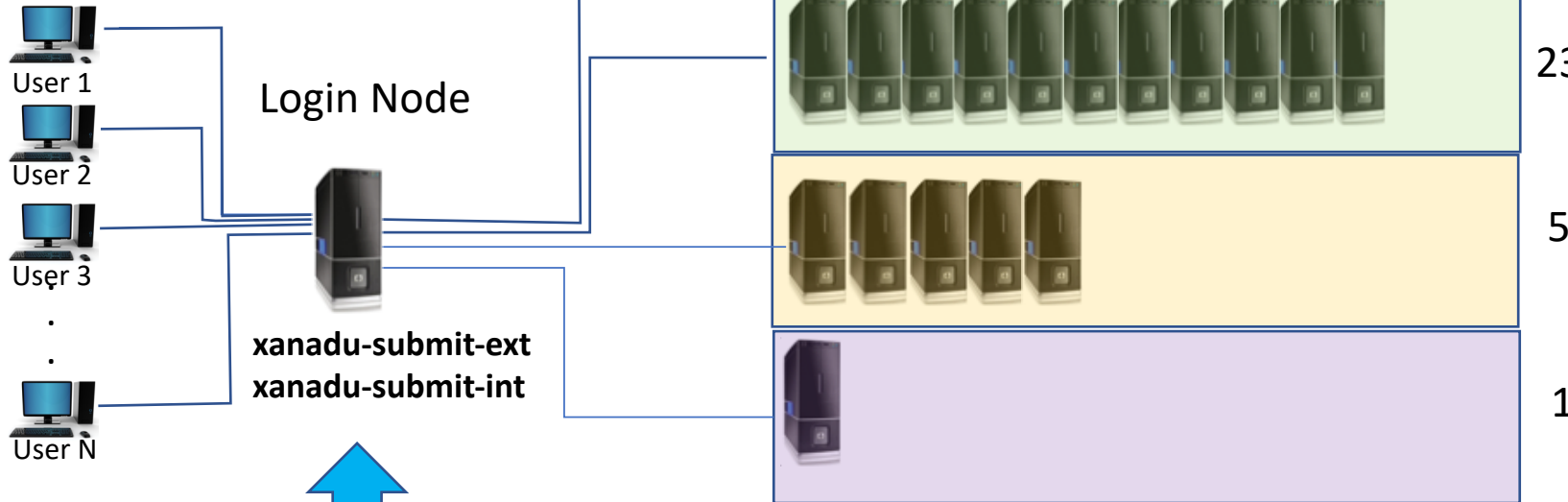
# Connecting to anadu...



# Xanadu

cpu : 4  
RAM: 8 GB

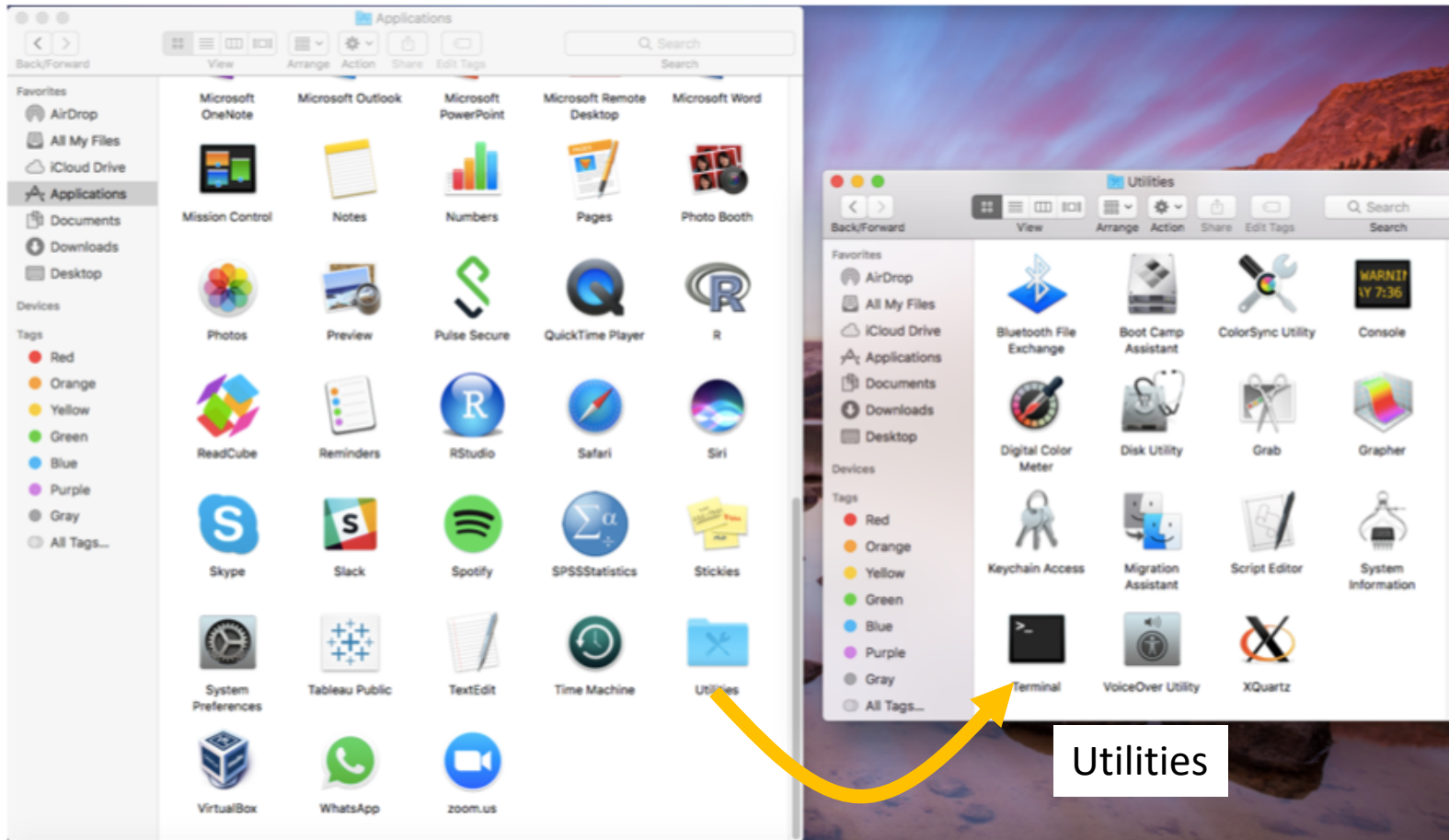
cpu : 32 – 48  
RAM: 128 – 512 GB



Queue/Partition	Nodes
xeon/general	xanadu01-11
amd/general	shangrila01-19 xanadu20-28
himem1-5	xanadu29-31
gpu /xeon/ general	xanadu01

xanadu-submit-ext.cam.uchc.edu  
xanadu-submit-int.cam.uchc.edu

# Mac



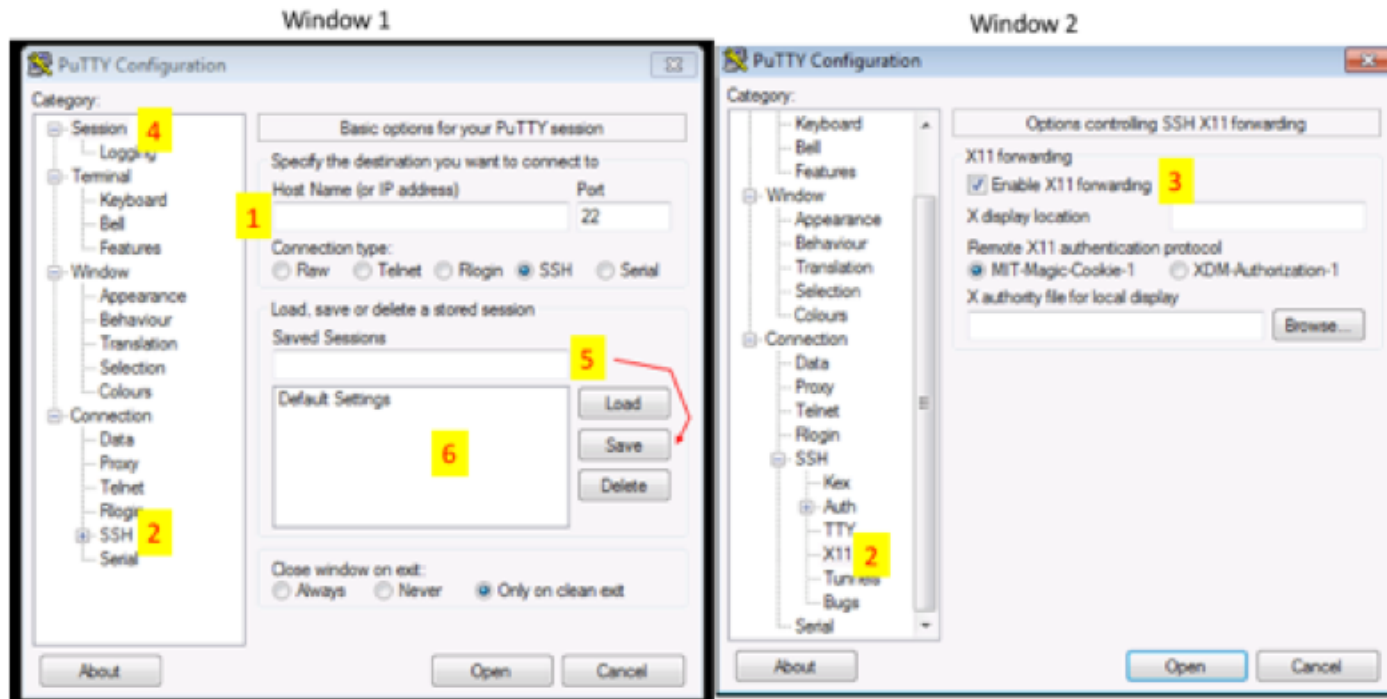
Applications

## TERMINAL



# Windows

**Putty** : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> or google search “putty”



Open Putty it will open window1.

1. Provide host name e.g. [username@xanadu-submit-ext.cam.uchc.edu](mailto: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.



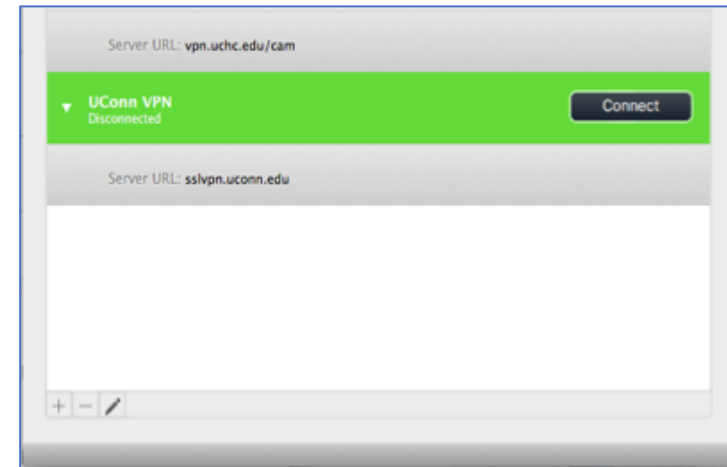
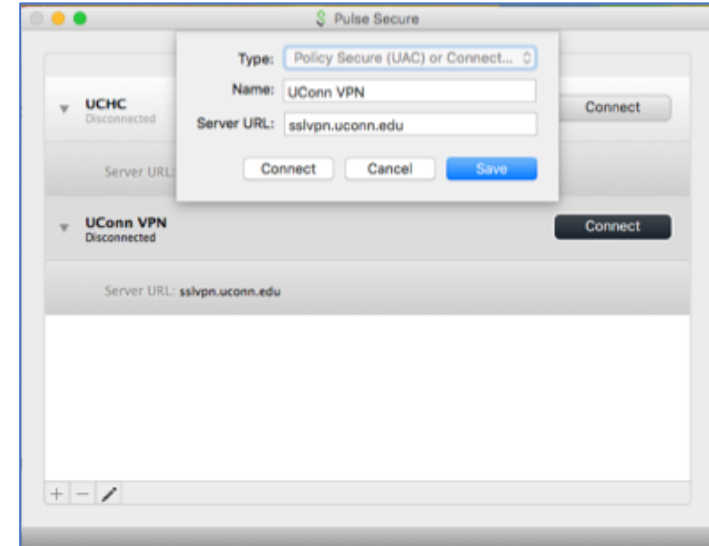


<https://remoteaccess.uconn.edu/vpn-overview/connect-via-vpn-client-2/>

Login: On the submit-int node (Storrs and Outside Health Centre)

Use VPN (Open Pulse secure)

1. Open Pulse secure
2. Add new connection
3. Set Server URL to : vpn.uchc.edu/cam
4. Save
5. Connect and login with CAM ID and Passwd



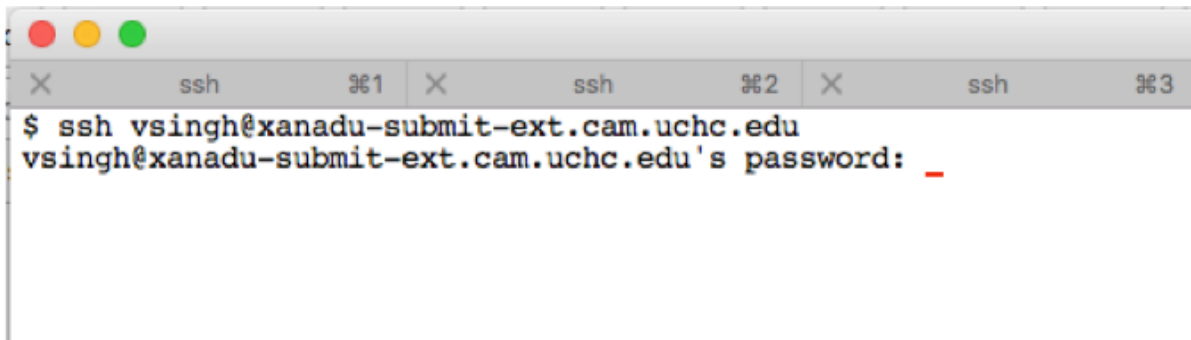


## Connecting to Xanadu

xanadu-submit-int Node: ssh <user\_name>@xanadu-submit-int.cam.uchc.edu

Xanadu-submit-ext Node: ssh <user\_name>@xanadu-submit-ext.cam.uchc.edu

Login: (using terminal on mac)



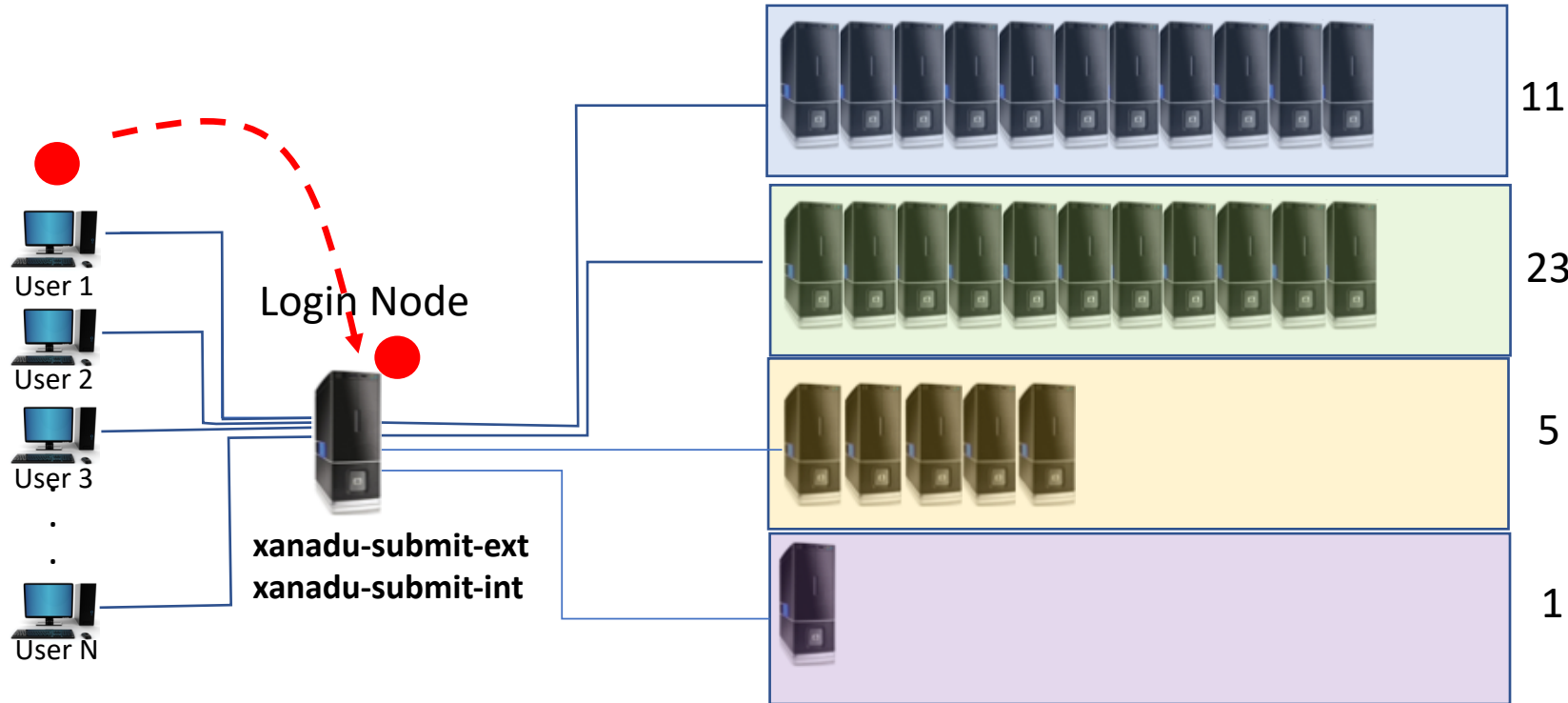
```
ssh %1 ssh %2 ssh %3
$ ssh vsingh@xanadu-submit-ext.cam.uchc.edu
vsingh@xanadu-submit-ext.cam.uchc.edu's password: _
```

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 ~ $ _
```



[ssh username@xanadu-submit-ext.cam.uchc.edu](ssh://username@xanadu-submit-ext.cam.uchc.edu)

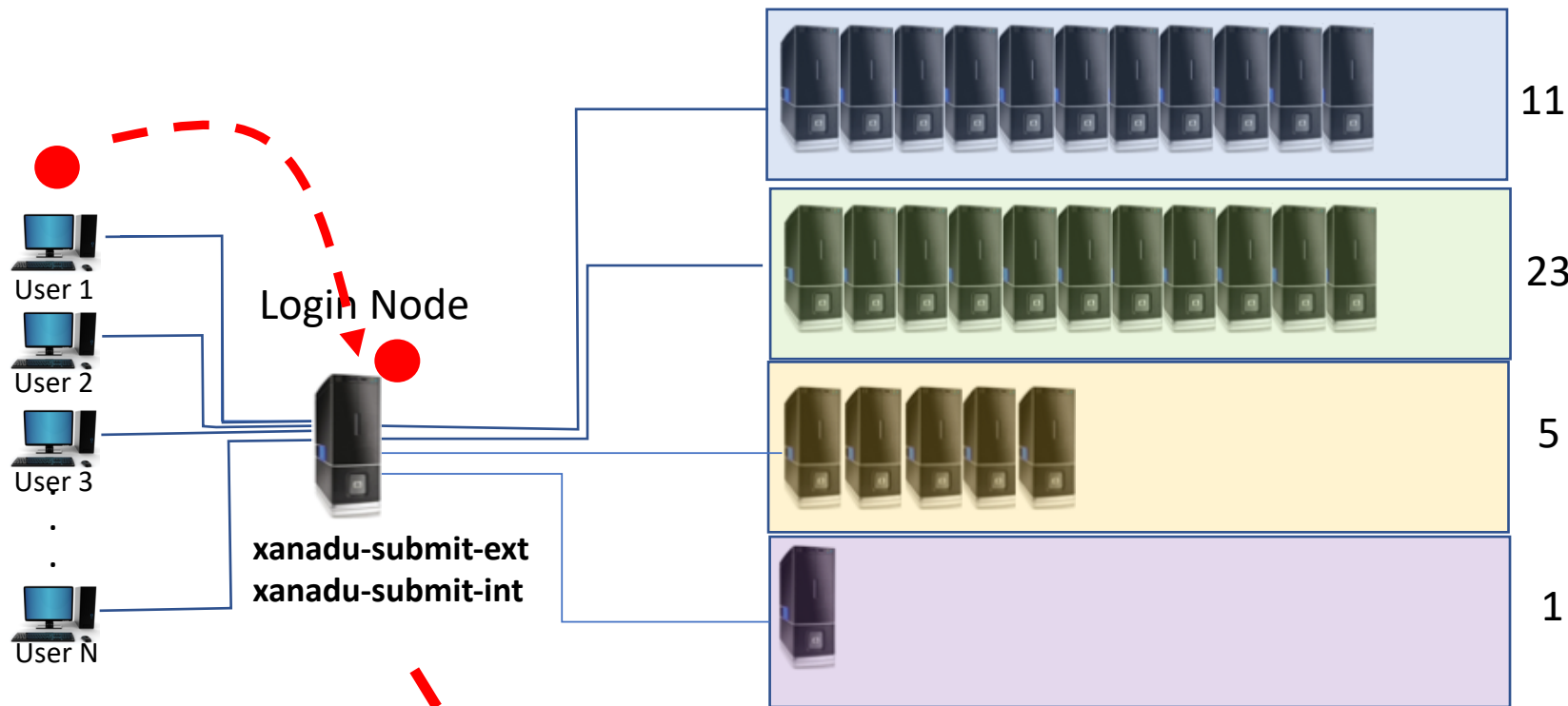


Queue/Partition	Nodes
xeon/general	xanadu01-11
amd/general	shangrila01-19 xanadu20-28
himem1-5	xanadu29-31
gpu /xeon/ general	xanadu01

cpu : 2  
RAM: 8GB

cpu : 32 – 48  
RAM: 128 – 512 GB

`ssh username@xanadu-submit-ext.cam.uchc.edu`



Queue/Partition	Nodes
xeon/general	xanadu01-11
amd/general	shangrila01-19 xanadu20-28
himem1-5	xanadu29-31
gpu /xeon/ general	xanadu01

`srun --pty bash` : Start of an interactive session  
`sbatch myscript.sh`

# Common Linux commands

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

Resources: [http://linuxcommand.org/writing\\_shell\\_scripts.php](http://linuxcommand.org/writing_shell_scripts.php)



# Text Editors



- NANO
- VIM
- EMACS

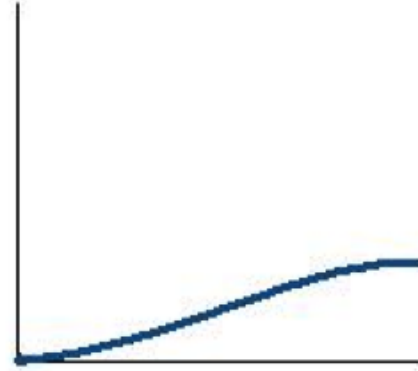
Classical learning curves for some common editors

©  
11-17-09

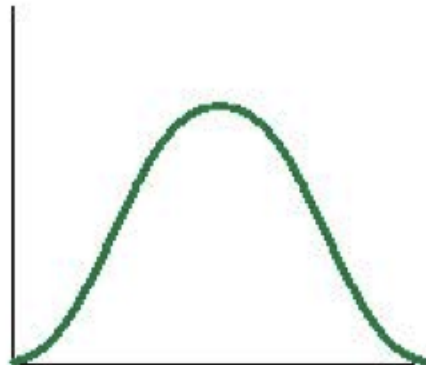
Notepad



nano



Visual Studio



vim



emacs



# Demo Nano and VIM



# Software/tool/packages on cluster

## Environment Modules:

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

```
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.
```



# Demo module





# 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 ~ $
```

mix  
idle  
allocated



```
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	15620	1	(null)		none
xanadu-02	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-03	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-04	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-05	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-06	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-07	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-08	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-09	1	general*	idle	36	2:18:1	257676	15620	15620	1	(null)		none
xanadu-10	1	general*	idle	36	2:18:1	128532	15620	15620	1	(null)		none
xanadu-11	1	general*	idle	36	2:18:1	128532	15620	15620	1	(null)		none
xanadu-20	1	general*	mixed	32	4:8:1	128745	15620	15620	1	(null)		none
xanadu-21	1	general*	mixed	32	4:8:1	128745	15620	15620	1	(null)		none
xanadu-22	1	general*	mixed	32	4:8:1	257760	15620	15620	1	(null)		none
xanadu-23	1	general*	idle	32	4:8:1	257760	15620	15620	1	(null)		none
xanadu-24	1	general*	idle	32	4:8:1	249696	15620	15620	1	(null)		none
xanadu-25	1	general*	mixed	32	4:8:1	209380	15620	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





# Summary of the nodes in the Xanadu cluster

NODELIST	PARTITION	PROCESSOR TYPE	CPUS	MEMORY
xanadu-01	general*	Xeon	36	256G
xanadu-02	general*	Xeon	36	256G
xanadu-03	general*	Xeon	36	256G
xanadu-04	general*	Xeon	36	256G
xanadu-05	general*	Xeon	36	256G
xanadu-06	general*	Xeon	36	256G
xanadu-07	general*	Xeon	36	256G
xanadu-08	general*	Xeon	36	256G
xanadu-09	general*	Xeon	36	256G
xanadu-10	general*	Xeon	36	128G
xanadu-11	general*	Xeon	36	128G
xanadu-20	general*	AMD	32	128G
xanadu-21	general*	AMD	32	128G
xanadu-22	general*	AMD	32	256G
xanadu-23	general*	AMD	32	256G
xanadu-24	general*	AMD	32	256G
xanadu-25	general*	AMD	32	200G
shangrila01	general*	AMD	48	256G
shangrila02	general*	AMD	48	256G
shangrila03	general*	AMD	48	256G
shangrila04	general*	AMD	48	256G
shangrila05	general*	AMD	48	256G

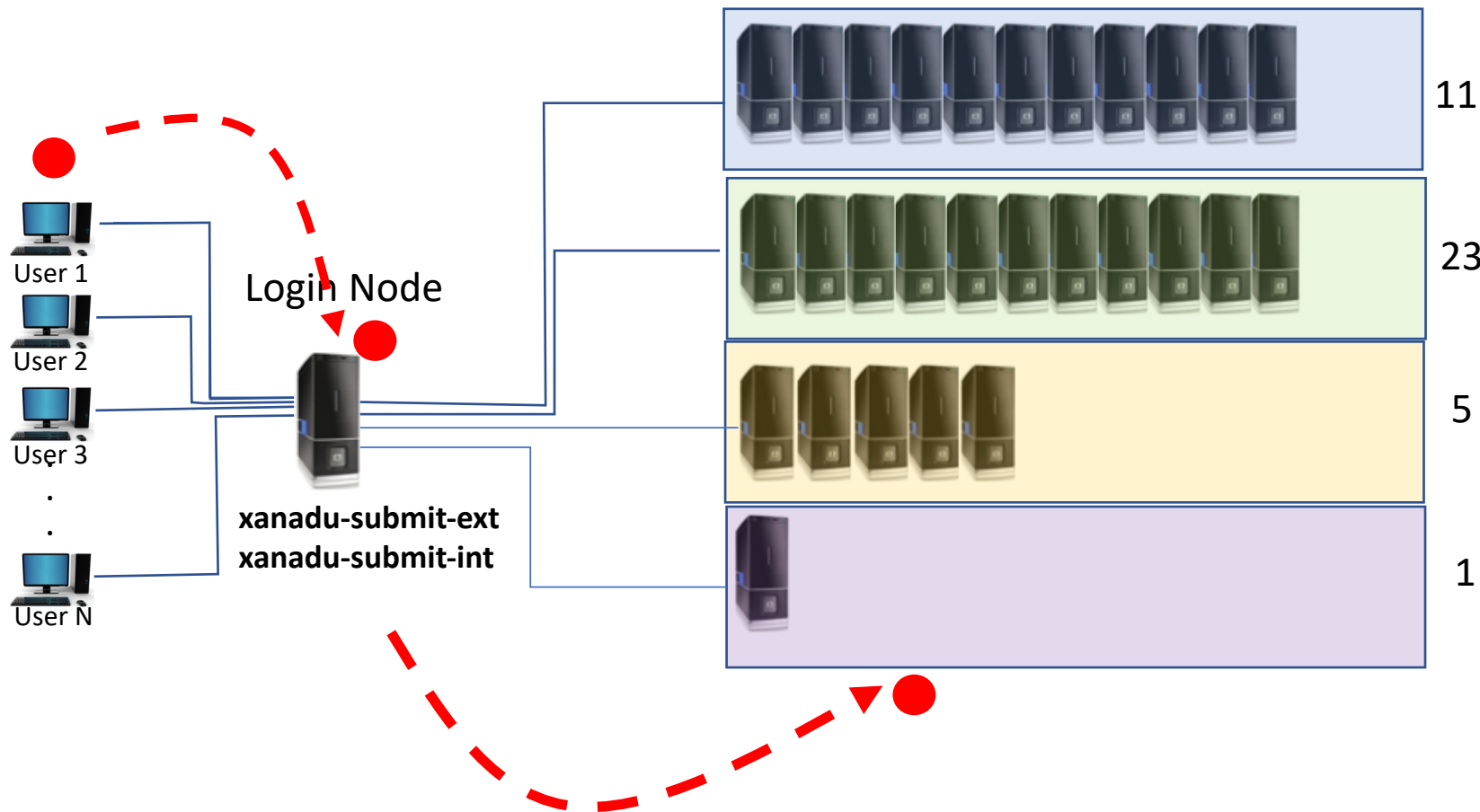
shangrila06	general*	AMD	48	256G
shangrila07	general*	AMD	48	128G
shangrila08	general*	AMD	48	128G
shangrila09	general*	AMD	48	256G
shangrila10	general*	AMD	48	256G
shangrila11	general*	AMD	32	128G
shangrila12	general*	AMD	32	128G
shangrila13	general*	AMD	32	256G
shangrila14	general*	AMD	32	256G
shangrila15	general*	AMD	32	128G
shangrila16	general*	AMD	32	128G
shangrila17	general*	AMD	32	128G
shangrila18	general*	AMD	32	128G
xanadu-30	himem1		32	514G
xanadu-31	himem2		32	514G
xanadu-32	himem3		32	514G
xanadu-33	himem4		32	514G
xanadu-29	himem5		32	514G



# Interactive session



`ssh username@xanadu-submit-ext.cam.uchc.edu`



Queue/Partition	Nodes
xeon/general	xanadu01-11
amd/general	shangrila01-19 xanadu20-28
himem1-5	xanadu29-31
gpu /xeon/ general	xanadu01

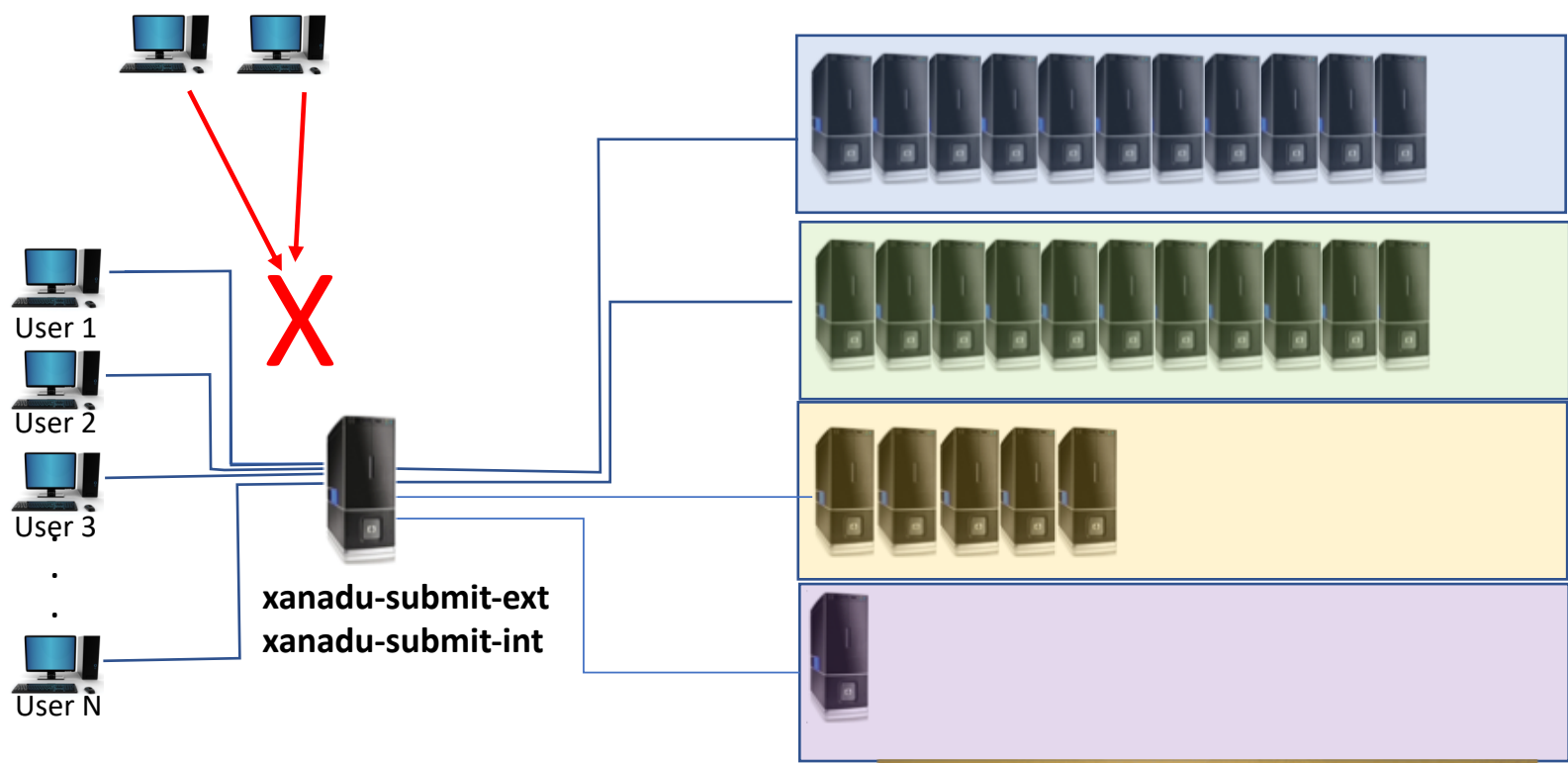
`srun --pty bash` : Start of an interactive session

# Xanadu





# Xanadu



Queue/Partition	Nodes
xeon/general	xanadu01-11
amd/general	shangrila01-19 xanadu20-28
himem1-5	xanadu29-31
gpu /xeon/ general	xanadu01



# Interactive session

```
srun --pty bash
```

```
screen -S screen_name : Start screen session
srun --pty bash        : Start interactive session
hostname               : Confirm interactive session
.....                 : Run code /commands (wget/ftp/others)
Ctrl + A + D          : Detach an active screen

screen -r NNNN         : Attach a detached screen
screen -ls             : List all screens
screen -X -S NNNN quit : Kill or quit a screen session
```



# 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 -j myscript
#SBATCH -n 1
#SBATCH -N 1
#SBATCH -c 1
#SBATCH -p general
#SBATCH --mail-type=END
#SBATCH --mail-user=first.last@uconn.edu
#SBATCH -o myscript-%j.out
#SBATCH -e myscript-%j.err
```

**#SBATCH -j myscript** Is the name of your job

**#SBATCH -n 1** Request number of tasks

**#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 -c 1** Request number of cores for your job

**#SBATCH -p 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-%j.out** Specifies the file to which the standard output will be appended

**#SBATCH -e myscript-%j.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_with_jump_libraries
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -c 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 -c 1
#SBATCH --partition=general
#SBATCH --mail-type=END
#SBATCH --mail-user=first.last@uconn.edu
#SBATCH -o myscript-%j.out
#SBATCH -e myscript-%j.err

hostname
echo "Hello World"
```

save the script as `myscript.sh`



Submit the script:

`sbatch`

```
$ sbatch myscript.sh
```

Monitor Job:

`squeue`

```
$ squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
246233	general	STO_001_	dtrujill	CG	4:48	1	xanadu-21
225302	general	file18	etulman	CG	22-11:03:24	1	xanadu-21
297908	himem	masurca_	hralicki	PD	0:00	1	(Resources)
301086	himem	ProtMasN	vsingh	PD	0:00	1	(Priority)
301089	himem	ProtMasW	vsingh	PD	0:00	1	(Priority)
301013	amd	ProtMasN	vsingh	R	21:44:59	1	xanadu-24
301008	amd	ProtMasW	vsingh	R	21:46:47	1	xanadu-20
297400	himem	bfo_111	dtrujill	R	1-23:18:04	4	xanadu-[30-33]
203251	general	blastp	cfisher	R	9-05:35:43	1	xanadu-23
203252	general	blastp	cfisher	R	9-05:35:43	1	xanadu-23



## queue -j jobIDNUMBER

```
$ clear$ queue -j 301013
```

```
JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
301013      amd ProtMasN vsingh  R   21:49:06      1 xanadu-24
```

## JOB STATE CODES

Jobs typically pass through several states in the course of their execution. The typical states are PENDING, RUNNING, SUSPENDED, COMPLETING, and COMPLETED. An explanation of each state follows.

CA	CANCELLED	Job was explicitly cancelled by the user or system administrator. The job may or may not have been initiated.
CD	COMPLETED	Job has terminated all processes on all nodes with an exit code of zero.
CF	CONFIGURING	Job has been allocated resources, become ready for use (e.g. booting).
CG	COMPLETING	
F	FAILED	Job terminated with non-zero exit code or other failure condition.
NF	NODE_FAIL	Job terminated due to failure of one or more allocated nodes.
PD	PENDING	Job is awaiting resource allocation.
PR	PREEMPTED	Job terminated due to preemption.
R	RUNNING	Job currently has an allocation.
ST	STOPPED	Job has an allocation, but execution has been stopped with SIGSTOP signal. CPUs have been retained by this job.
S	SUSPENDED	Job has an allocation, but execution has been suspended and CPUs have been released for other jobs.
TO	TIMEOUT	Job terminated upon reaching its time limit.





## squeue -u userID

```
$squeue -u vsingh
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
301086	himem	ProtMasN	vsingh	PD	0:00	1	(Priority)
301089	himem	ProtMasW	vsingh	PD	0:00	1	(Priority)
301013	amd	ProtMasN	vsingh	R	22:00:08	1	xanadu-24
301008	amd	ProtMasW	vsingh	R	22:01:56	1	xanadu-20

## squeue -u userID -t PENDING

```
$ squeue -u vsingh -t PENDING
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
301086	himem	ProtMasN	vsingh	PD	0:00	1	(Priority)
301089	himem	ProtMasW	vsingh	PD	0:00	1	(Priority)

## squeue -u userID -t RUNNING



`scontrol show jobid <jobid>`

```
$ scontrol show jobid 301086
```

```
JobId=301086 JobName=ProtMasNoSOAP  
UserId=vsingh(183147) GroupId=domain users(10000) MCS_label=N/A  
Priority=5262 Nice=0 Account=pi-wegrzyn QOS=general  
JobState=PENDING Reason=Priority Dependency=(null)  
Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0  
RunTime=00:00:00 TimeLimit=UNLIMITED TimeMin=N/A SubmitTime=2017-06-28T15:50:23  
EligibleTime=2017-06-28T15:50:23 StartTime=2018-06-27T14:06:17 EndTime=Unknown  
Deadline=N/A  
PreemptTime=None SuspendTime=None SecsPreSuspend=0  
Partition=himem AllocNode:Sid=xanadu-submit-ext:32674  
ReqNodeList=(null) ExcNodeList=(null)  
NodeList=(null)  
NumNodes=1-1 NumCPUs=30 NumTasks=30 CPUs/Task=1 ReqB:S:C:T=0:0:*:*  
TRES=cpu=30,mem=512000,node=1 Socks/Node=* NtasksPerN:B:S:C=0:0:*:1 CoreSpec=*  
MinCPUsNode=1 MinMemoryNode=500G MinTmpDiskNode=0 Features=(null) Gres=(null)  
Reservation=(null) OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)  
Command=/home/CAM/vsingh/protea_repens/scripts/assemble_protea_config_noSOAPassem  
bly.sh WorkDir=/home/CAM/vsingh/protea_repens/scripts  
StdErr=/home/CAM/vsingh/protea_repens/LogFiles/ProtMasNoSOAP-301086.error  
StdIn=/dev/null StdOut=/home/CAM/vsingh/protea_repens/LogFiles/ProtMasNoSOAP-  
301086.output Power=
```

`scontrol show jobid -dd <jobid>`



## Script submission and other commands

<code>sbatch myscript.sh</code>	: Submit script for execution
<code>squeue</code>	: Status of Jobs currently running on cluster (all users)
<code>squeue -j jobIDNUMBER</code>	: Status of job with jobIDNumber
<code>squeue -u UserID</code>	: Status of all the jobs submitted by user
<code>scancel jobID_number</code>	: Delete job with jobID_number
<code>scancel &lt;jobid&gt; &lt;index&gt;</code>	: Delete an array job
<code>scancel -u UserID</code>	: 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 --mem=1G
#SBATCH --array=1-1002%100
#SBATCH --output=fastqc_%A_%a.out
hostname
```

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 find 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 and then run it through [fastqc](#) application.



## Some important Informations:

Xanadu	
User Space	/home/CAM/username or /home/FCAM/username (default)
Lab/Group Space	/UCHC/LABS/ (Request)
Archiving data	/linuxshare/userfolder (Request)
Collaborative projects	/linuxshare/projects (Request)

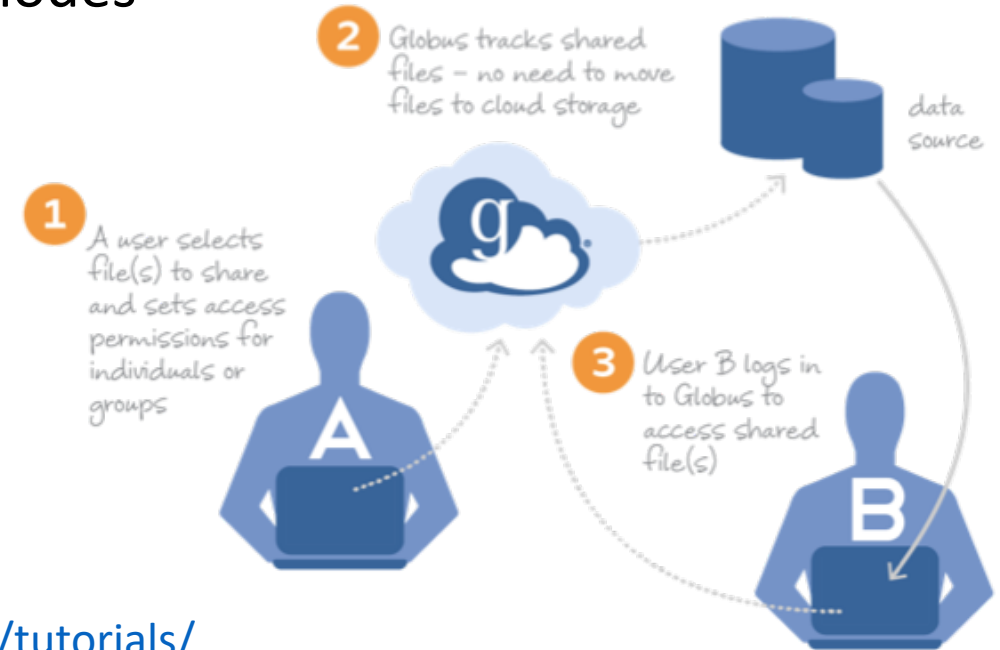
**/scratch** : Please do not use this directory as long term storage the disk gets cleaned up regularly.



# Data Transfer to/from Xanandu (wget/ftp/sftp/scp)

**DON'T** : Do not initiate transfer on Submit nodes

**DO** : Start an interactive session  
: Use **scp** for small size files  
: For large size files use Globus



Tutorial: <https://bioinformatics.uconn.edu/resources-and-events/tutorials/>

Section: Data transfer



# Reporting Issues or submitting requests

<https://bioinformatics.uconn.edu/contact-us/>

<https://bioinformatics.uconn.edu/contact-us/technical-issues/>

Email : [cbcsupport@uconn.edu](mailto:cbcsupport@uconn.edu)

The screenshot shows the 'Contact Us' page of the Institute for Systems Genomics Computational Biology Core. The page has a navigation menu with links for Home, People, Hardware, Software, Databases, Tutorials, Resources, Data Therapy, Rate Plan, Contact Us, and FAQ. A search bar is located in the top right corner. The main content area is titled 'Contact Us' and includes a sub-heading 'Account and Support Requests'. Below this, there is a paragraph of instructions: '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.' The form itself is titled 'Inquiry Selection \*' and features a dropdown menu with the following options: 'Account Request (UCHC cluster - Xanadu)- Recommended', 'Account Request (UCHC cluster - Xanadu)- Recommended', 'CAM Account Request (Geneious)', 'Software/Database Request', 'Bioinformatics/Technical Support', and 'Account for Course (BBC)'. To the right of the dropdown is a text input field labeled 'Last'. Below the dropdown is a 'Status \*' section with two radio button options: 'Undergraduate Student' and 'Graduate Student'.



# Cluster Etiquette



Do not run code on the head node.

Do not ssh directly into a node.

Do not submit a large number of jobs without testing.

Do not Hog Resources.



Do monitor your jobs periodically

Monitor your disk usage:

Do not fill up the whole disk with unnecessary output files from your runs.





## TOP: Process Status

'D' = uninterruptible sleep

'R' = running

'S' = sleeping

'T' = traced or stopped

'Z' = zombie



Thank you

