

INTRO TO PALMETTO

TUE VU – CCIT/CITI



INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



1. Introduction to Clemson super-computer: the Palmetto



clemson palmetto

clemson palmetto

clemson palmetto **fellows**

clemson palmetto **tree**

clemson palmetto **ballroom**

clemson palmetto **pact scholarship**

clemson palmetto **login**

Clemson University Palmetto Cluster

<https://www.palmetto.clemson.edu/palmetto/> ▾

Palmetto Office Hours; Email ithelp; Research support. Information for New Users. Obtaining an account; Training; On-boarding sessions. Acknowledging ...

Basic Usage

Basic Usage. Logging in. Two-Factor Authentication (2FA ...

About Palmetto Cluster

About Palmetto Cluster. Overview; Compute Hardware. Myrinet ...

Clemson University Palmetto ...

The Palmetto Cluster is Clemson University's primary high ...

[More results from clemson.edu »](#)

Software on Palmetto

Many site-licensed software packages are available on ...

COMSOL

COMSOL is an application for solving Multiphysics problems ...

Compilers

Compilers available on Palmetto. Available Compilers; Compiling ...

1. Introduction to Clemson super-computer: the Palmetto

Palmetto Cluster Documentation

General Information ▾

- General Information
- About Palmetto Cluster

FAQ

User's Guide ▾

Software Guide ▾

Programmer's Guide ▾

JupyterHub ▾

Owner's Guide ▾

About Palmetto Cluster

Table of Contents

- Overview
- Compute Hardware
 - Myrinet phases
 - Infiniband phases
 - GPUs
 - Intel Xeon Phi accelerators
 - Big-memory nodes
- Storage
- Job scheduling
- Condominium model

Overview



Palmetto is Clemson University's primary high-performance computing (HPC) resource; heavily utilized by researchers, students, faculty, and staff from a broad range of disciplines.

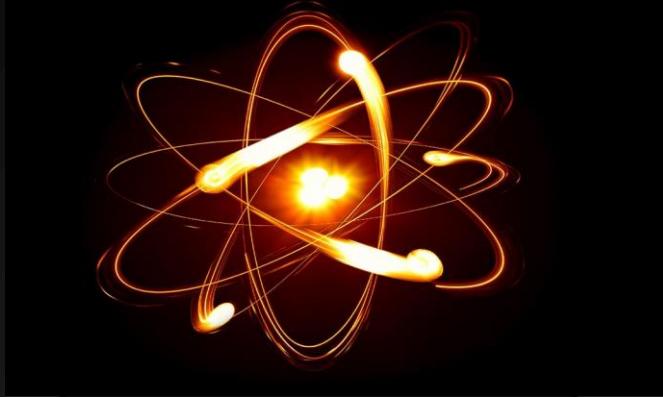
Currently, Palmetto is comprised of 2021 compute nodes (totalling 23072 CPU cores), and features:

- More than 2000 compute nodes
- More than 700 GPU nodes
- More than 23000 CPU cores
- 6 large memory nodes (up to 2TB)
- 10 Gbps Ethernet, 10 Gbps Myrinet and 56Gbps Infiniband networks
- maximum run time for a single task limited to 72 hours (Infiniband nodes) or 168 hours (Myrinet nodes)

2014:

- Top 4 fastest supercomputer among US public universities.
- Rank #155 fastest supercomputer in TOP500

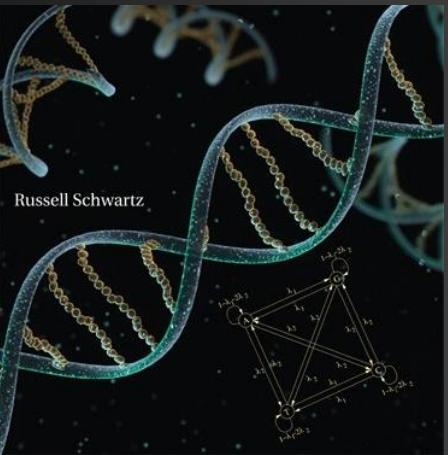
1. Introduction to Clemson super-computer: the Palmetto



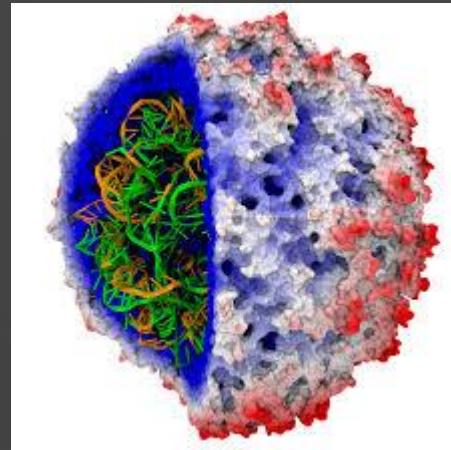
Quantum mechanics



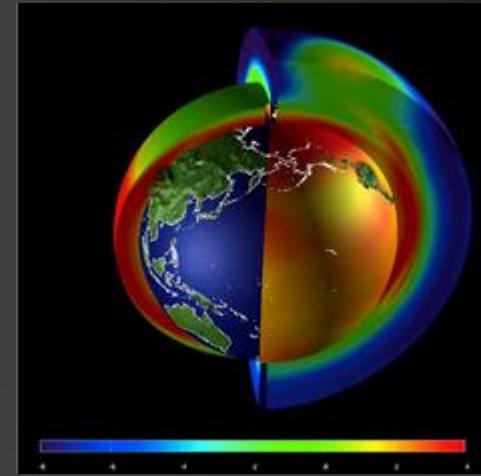
Weather forecast



Biological modeling



Molecular modeling



Climate research

1. Introduction to Clemson super-computer: the Palmetto

Register for Palmetto account:

<https://citi.sites.clemson.edu/new-account/>

1. Introduction to Clemson super-computer: the Palmetto

For user:

- Free for all CLEMSON research staffs, postdoc researchers, research students
- For Palmetto research account: the account will be deactivated upon terminating the contract with CU
- For Palmetto educational account: the account will be deactivated upon the semester ends
- 100gb personal space (backed up)
- Unlimited scratch storage (for model running and temporary storage): deleted after 30 days if untouched



Storage

Various options for storing data (on temporary and permanent basis) are available to researchers using Palmetto:

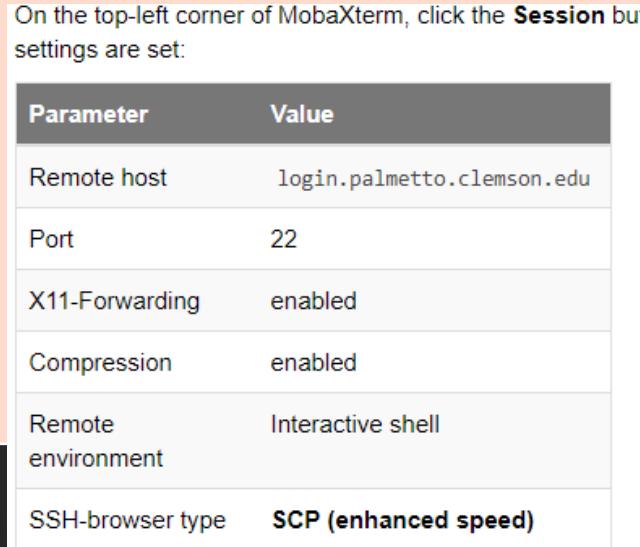
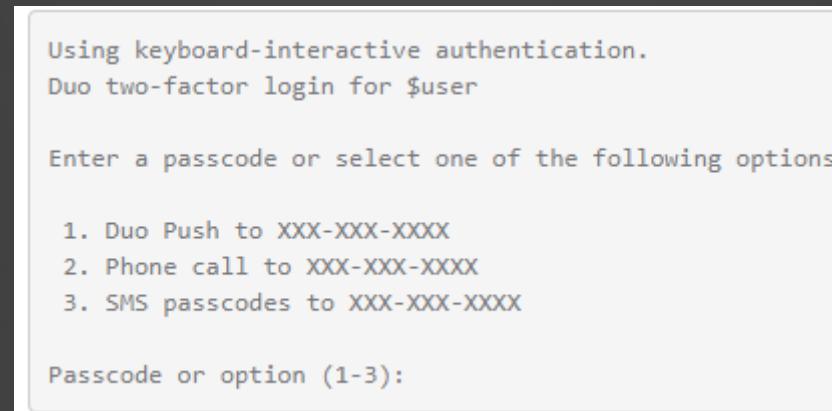
Location	Available space	Notes
/home	100 GB per user	Backed-up nightly, permanent storage space accessible from all nodes
/scratch1	233 TB shared by all users	Not backed up, temporary work space accessible from all nodes, OrangeFS Parallel File System
/scratch2	160 TB shared by all users	Not backed up, temporary work space accessible from all nodes, XFS
/scratch3	129 TB shared by all users	Not backed up, temporary work space accessible from all nodes, ZFS
/local_scratch	Varies between nodes (99GB-800GB)	Per-node temporary work space, accessible only for the lifetime of job

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



2. Assessing the Palmetto cluster

Window Moba-XTerm	Mac OS														
<p>On the top-left corner of MobaXterm, click the Session button. settings are set:</p> <table border="1"><thead><tr><th>Parameter</th><th>Value</th></tr></thead><tbody><tr><td>Remote host</td><td>login.palmetto.clemson.edu</td></tr><tr><td>Port</td><td>22</td></tr><tr><td>X11-Forwarding</td><td>enabled</td></tr><tr><td>Compression</td><td>enabled</td></tr><tr><td>Remote environment</td><td>Interactive shell</td></tr><tr><td>SSH-browser type</td><td>SCP (enhanced speed)</td></tr></tbody></table>  <p>A screenshot of the MobaXterm application window. It shows a sidebar on the left with icons for file operations, a terminal window in the center displaying command history, and a configuration panel on the right. The configuration panel has tabs for 'Session' (selected), 'Appearance', 'Network', 'Keyboard', 'Mouse', 'Clipboard', 'File Transfer', 'X11', 'Advanced', and 'Help'. Under the 'Session' tab, there are sections for 'Local Hostname', 'Local Port', 'Remote Hostname', 'Remote Port', 'X11 Forwarding', 'Compression', 'Remote Environment', 'SSH Browser Type', and 'Session Timeout'. The 'Remote Hostname' field is set to 'login.palmetto.clemson.edu'.</p>	Parameter	Value	Remote host	login.palmetto.clemson.edu	Port	22	X11-Forwarding	enabled	Compression	enabled	Remote environment	Interactive shell	SSH-browser type	SCP (enhanced speed)	<p>\$ ssh username@login.palmetto.clemson.edu</p> <p>\$ ssh -X username@login.palmetto.clemson.edu</p>  <p>A screenshot of a terminal window with a light gray background. It displays the following text: Using keyboard-interactive authentication. Duo two-factor login for \$user Enter a passcode or select one of the following options: 1. Duo Push to XXX-XXX-XXXX 2. Phone call to XXX-XXX-XXXX 3. SMS passcodes to XXX-XXX-XXXX Passcode or option (1-3):</p>
Parameter	Value														
Remote host	login.palmetto.clemson.edu														
Port	22														
X11-Forwarding	enabled														
Compression	enabled														
Remote environment	Interactive shell														
SSH-browser type	SCP (enhanced speed)														

2. Assessing the Palmetto cluster

When logged in, you are presented with a welcome message and the following “prompt”:

```
[username@login001 ~]$
```

```
• MobaXterm 11.0 •
(SSH client, X-server and networking tools)

> SSH session to tuev@user.palmetto.clemson.edu
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)
  • DISPLAY : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

Success. Logging you in...
last login: Wed Nov 28 15:36:58 2018 from 198.21.191.34

-----
Welcome to the PALMETTO CLUSTER at CLEMSON UNIVERSITY

* Email ithelp@clemson.edu with questions or to report problems.

* Palmetto "office hours" are every Wednesday 8am-11am in 412 Cooper Library.

* Quarterly maintenance periods: May (followed by Top 500 benchmark), August, November and Feb. Email will be sent before each period with details of cluster availability.

  User guide: http://www.palmetto.clemson.edu/palmetto
  Sample programs: https://github.com/clemsonciti/palmetto-examples
  JupyterHub: https://www.palmetto.clemson.edu/jupyterhub

Useful commands:
  module avail           - list available software packages
  qstat -xf jobid       - check status of your job
  qstat -Qf queuename   - check status of a queue
  checkquota            - check your disk quota
  checkqueuecfg         - check general workq max running limits
  cat /etc/hardware-table - list node hardware: ram,cores,chip,etc.
  qpeek                 - look at a running job's stdout or stderr
  whatsfree             - see what nodes are free right now

Please do not use /home as your PBS working directory. Jobs with /home as working directory may be killed as performance deteriorates.

DO NOT RUN JOBS/PROGRAMS/TESTS/PRE-OR-POST PROCESSING ON THE LOGIN NODE.
They will be terminated without notice. No exceptions.

----- This file is: /etc/motd ----- Last Updated: 21-FEB-2018 -----
```

Terminology:

Login001: head node

Node0123: compute node

/home/username/: home directory

/scratch1/username: working directory

- **Do not use /home as your working directory**
- **Do not run jobs on the login node**

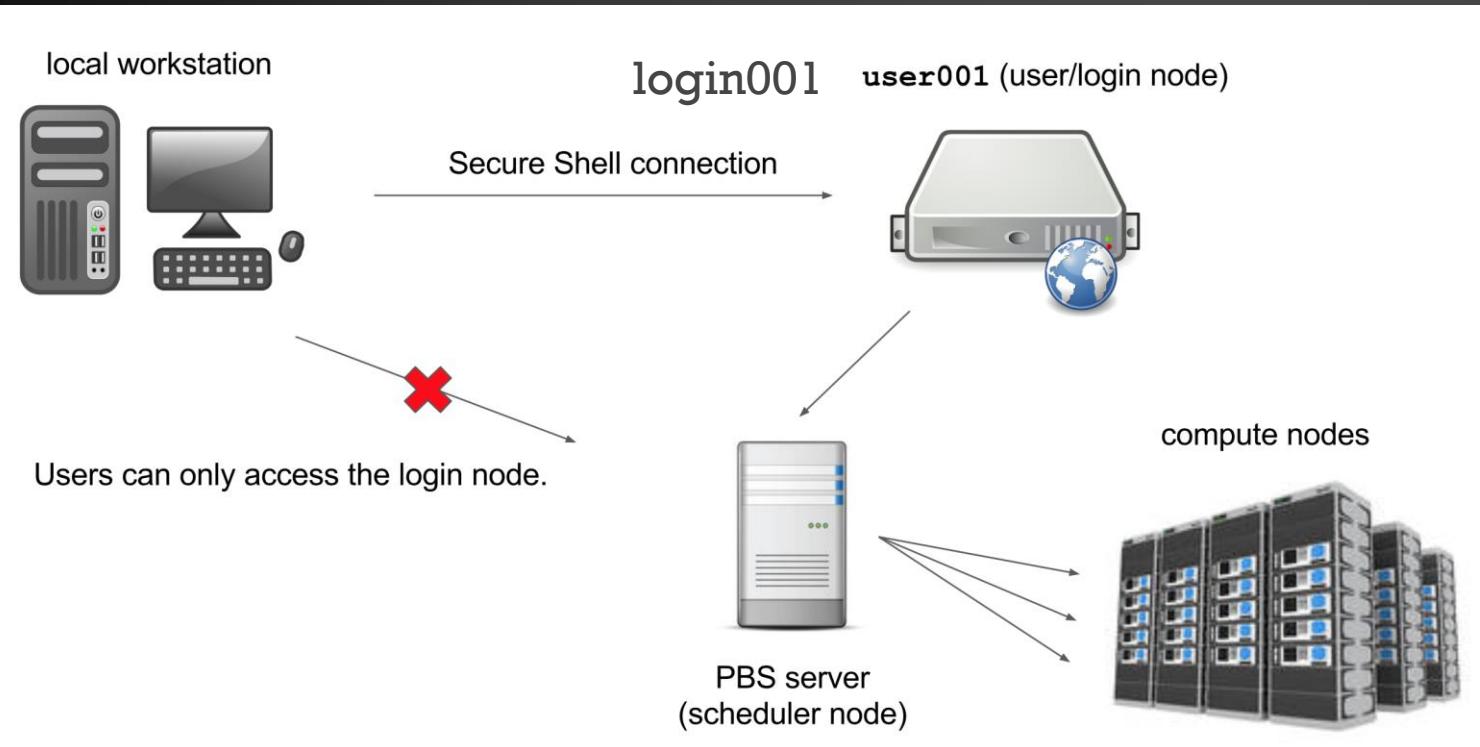


2. Assessing the Palmetto cluster

When logged in, you are presented with a welcome message and the following “prompt”:

```
[username@login001 ~]$
```

The Palmetto cluster has several “compute” nodes that can perform fast calculations on large amounts of data.



2. Assessing the Palmetto cluster

Useful commands:

\$ whatsfree

TOTAL NODES: 2030 NODES FREE: 490 NODES OFFLINE: 92 NODES RESERVED: 22			
PHASE 0	TOTAL = 10	FREE = 5	OFFLINE = 0
PHASE 1	TOTAL = 75	FREE = 0	OFFLINE = 0
PHASE 2a	TOTAL = 158	FREE = 0	OFFLINE = 36
PHASE 2b	TOTAL = 84	FREE = 0	OFFLINE = 1
PHASE 3	TOTAL = 225	FREE = 128	OFFLINE = 0
PHASE 4	TOTAL = 326	FREE = 1	OFFLINE = 0
PHASE 5a	TOTAL = 321	FREE = 226	OFFLINE = 1
PHASE 5b	TOTAL = 9	FREE = 9	OFFLINE = 0
PHASE 6	TOTAL = 67	FREE = 0	OFFLINE = 0
PHASE 7a	TOTAL = 42	FREE = 0	OFFLINE = 0
PHASE 7b	TOTAL = 12	FREE = 0	OFFLINE = 0
PHASE 8a	TOTAL = 71	FREE = 14	OFFLINE = 0
PHASE 8b	TOTAL = 57	FREE = 0	OFFLINE = 0
PHASE 8c	TOTAL = 88	FREE = 73	OFFLINE = 0
PHASE 9	TOTAL = 72	FREE = 0	OFFLINE = 0
PHASE 10	TOTAL = 80	FREE = 4	OFFLINE = 0
PHASE 11a	TOTAL = 40	FREE = 5	OFFLINE = 0
PHASE 11b	TOTAL = 4	FREE = 0	OFFLINE = 0
PHASE 12	TOTAL = 30	FREE = 0	OFFLINE = 0
PHASE 13	TOTAL = 24	FREE = 0	OFFLINE = 0
PHASE 14	TOTAL = 12	FREE = 0	OFFLINE = 0
PHASE 15	TOTAL = 32	FREE = 25	OFFLINE = 0
PHASE 16	TOTAL = 40	FREE = 0	OFFLINE = 0
PHASE 17	TOTAL = 20	FREE = 0	OFFLINE = 0
PHASE 18a	TOTAL = 2	FREE = 0	OFFLINE = 0
PHASE 18b	TOTAL = 65	FREE = 0	OFFLINE = 0
PHASE 18c	TOTAL = 10	FREE = 0	OFFLINE = 0
			BIGMEM nodes: (5) 24cores/500GB, (2) 40cores/1.5TB, (1) 64cores/2TB
			TYPE = Dell PE1950 Intel Xeon E5345, 8 cores, 11GB, mx
			TYPE = Dell PE1950 Intel Xeon E5410, 8 cores, 11GB, mx
			TYPE = Dell PE1950 Intel Xeon E5410, 8 cores, 15GB, mx
			TYPE = Sun X2200 AMD Opteron 2356, 8 cores, 15GB, mx
			TYPE = IBM DX340 Intel Xeon E5410, 8 cores, 15GB, mx
			TYPE = Sun X6250 Intel Xeon L5420, 8 cores, 30GB, mx
			TYPE = Sun X4150 Intel Xeon E5410, 8 cores, 15GB, mx
			TYPE = HP DL165 AMD Opteron 6176, 24 cores, 46GB, mx
			TYPE = HP SL230 Intel Xeon E5-2665, 16 cores, 62GB, FDR
			TYPE = HP SL250s Intel Xeon E5-2665, 16 cores, 62GB, FDR, M2075
			TYPE = HP SL250s Intel Xeon E5-2665, 16 cores, 62GB, FDR, K20, SSD
			TYPE = HP SL250s Intel Xeon E5-2665, 16 cores, 62GB, FDR, K20
			TYPE = Dell PEC6220 Intel Xeon E5-2665, 16 cores, 62GB, 10ge
			TYPE = HP SL250s Intel Xeon E5-2665, 16 cores, 125GB, FDR, K20, 10ge
			TYPE = HP SL250s Intel Xeon E5-2670v2, 20 cores, 125GB, FDR, K20, 10ge
			TYPE = HP SL250s Intel Xeon E5-2670v2, 20 cores, 125GB, FDR, K40, 10ge
			TYPE = HP SL250s Intel Xeon E5-2670v2, 20 cores, 125GB, FDR, Phi, 10ge
			TYPE = Lenovo MX360M5 Intel Xeon E5-2680v3, 24 cores, 125GB, FDR, K40, 10ge
			TYPE = Dell C4130 Intel Xeon E5-2680v3, 24 cores, 125GB, FDR, K40, 10ge
			TYPE = HP XL190r Intel Xeon E5-2680v3, 24 cores, 125GB, FDR, K40, 10ge
			TYPE = Dell C4130 Intel Xeon E5-2680v3, 24 cores, 125GB, FDR, K40, 10ge
			TYPE = Dell C4130 Intel Xeon E5-2680v4, 28 cores, 125GB, FDR, P100, 10ge
			TYPE = Dell C4130 Intel Xeon E5-2680v4, 28 cores, 125GB, FDR, P100, 10ge
			TYPE = Dell C4140 Intel Xeon 6148G, 40 cores, 372GB, FDR, V100nv, 25ge
			TYPE = Dell R740 Intel Xeon 6148G, 40 cores, 372GB, FDR, V100, 25ge
			TYPE = Dell R740 Intel Xeon 6148G, 40 cores, 748GB, FDR, V100, 25ge



2. Assessing the Palmetto cluster

Useful commands:

\$ cat /etc/hardware-table

PALMETTO HARDWARE TABLE					Last updated: July 23 2018								
PHASE	COUNT	MAKE	MODEL	CHIP(0)	CORES	RAM(1)	/local_scratch	Interconnect	GPUs	PHIs	SSD		
0	6	HP	DL580	Intel Xeon	7542	24	505 GB(2)	99 GB	lg, 10g, mx	0	0	0	
0	1	HP	DL980	Intel Xeon	7560	64	2 TB(2)	99 GB	lg, 10g, mx	0	0	0	
0	1	HP	DL560	Intel Xeon	E5-4627v4	40	1.5 TB(2)	881 GB	lg, 56g, fdr, 10ge	0	0	0	
0	1	Dell	R830	Intel Xeon	E5-4627v4	40	1.0 TB(2)	880 GB	lg, 56g, fdr, 10ge	0	0	0	
0	1	HP	DL560	Intel Xeon	6138G	80	1.5 TB(2)	3.6 TB	lg, 10ge	0	0	0	
1	92	Dell	PE1950	Intel Xeon	E5345	8	12 GB	37 GB	lg, 10g, mx	0	0	0	
2a	170	Dell	PE1950	Intel Xeon	E5410	8	12 GB	37 GB	lg, 10g, mx	0	0	0	
2b	73	Dell	PE1950	Intel Xeon	E5410	8	16 GB	37 GB	lg, 10g, mx	0	0	0	
3	226	Sun	X2200	AMD Opteron	2356	8	16 GB	193 GB	lg, 10g, mx	0	0	0	
4	326	IBM	DX340	Intel Xeon	E5410	8	16 GB	111 GB	lg, 10g, mx	0	0	0	
5a	327	Sun	X6250	Intel Xeon	L5420	8	32 GB	31 GB	lg, 10g, mx	0	0	0	
5b	9	Sun	X4150	Intel Xeon	E5410	8	32 GB	99 GB	lg, 10g, mx	0	0	0	
6	67	HP	DL165	AMD Opteron	6176	24	48 GB	193 GB	lg, 10g, mx	0	0	0	
7a	42	HP	SL230	Intel Xeon	E5-2665	16	64 GB	240 GB	lg, 56g, fdr	0	0	0	
7b	12	HP	SL250s	Intel Xeon	E5-2665	16	64 GB	240 GB	lg, 56g, fdr	2(3)	0	0	
8a	71	HP	SL250s	Intel Xeon	E5-2665	16	64 GB	900 GB	lg, 56g, fdr	2(4)	0	300 GB(7)	
8b	57	HP	SL250s	Intel Xeon	E5-2665	16	64 GB	420 GB	lg, 56g, fdr	2(4)	0	0	
8c	88	Dell	PEC6220	Intel Xeon	E5-2665	16	64 GB	350 GB	lg, 10ge	0	0	0	
9	72	HP	SL250s	Intel Xeon	E5-2665	16	128 GB	420 GB	lg, 56g, fdr, 10ge	2(4)	0	0	
10	80	HP	SL250s	Intel Xeon	E5-2670v2	20	128 GB	800 GB	lg, 56g, fdr, 10ge	2(4)	0	0	
11a	40	HP	SL250s	Intel Xeon	E5-2670v2	20	128 GB	800 GB	lg, 56g, fdr, 10ge	2(6)	0	0	
11b	4	HP	SL250s	Intel Xeon	E5-2670v2	20	128 GB	800 GB	lg, 56g, fdr, 10ge	0	2(8)	0	
12	30	Lenovo	NX360M5	Intel Xeon	E5-2680v3	24	128 GB	800 GB	lg, 56g, fdr, 10ge	2(6)	0	0	
13	24	Dell	C4130	Intel Xeon	E5-2680v3	24	128 GB	1.8 TB	lg, 56g, fdr, 10ge	2(6)	0	0	
14	12	HP	XL1X0R	Intel Xeon	E5-2680v3	24	128 GB	880 GB	lg, 56g, fdr, 10ge	2(6)	0	0	
15	32	Dell	C4130	Intel Xeon	E5-2680v3	24	128 GB	1.8 TB	lg, 56g, fdr, 10ge	2(6)	0	0	
16	40	Dell	C4130	Intel Xeon	E5-2680v4	28	128 GB	1.8 TB	lg, 56g, fdr, 10ge	2(9)	0	0	
17	20	Dell	C4130	Intel Xeon	E5-2680v4	28	128 GB	1.8 TB	lg, 56g, fdr, 10ge	2(9)	0	0	
18a	2	Dell	C4140	Intel Xeon	6148G	40	372 GB	1.9 TB(12)	lg, 56g, fdr, 40ge	4(10)	0	0	
18b	65	Dell	R740	Intel Xeon	6148G	40	372 GB	1.8 TB	lg, 56g, fdr, 25ge	2(11)	0	0	
18c	10	Dell	R740	Intel Xeon	6148G	40	748 GB	1.8 TB	lg, 56g, fdr, 25ge	2(11)	0	0	

*** PBS resource requests are always lowercase ***

- (0) CHIP has 3 resources: chip_manufacturer, chip_model, chip_type
- (1) Leave 2 or 3GB for the operating system when requesting memory in PBS jobs
- (2) Specify queue "bigmem" to access the large memory machines, only ncpus and mem are valid PBS resource requests
- (3) 2 NVIDIA Tesla M2075 cards per node, use resource request "ngpus=[1|2]" and "gpu_model=m2075"
- (4) 2 NVIDIA Tesla K20m cards per node, use resource request "ngpus=[1|2]" and "gpu_model=k20"
- (5) 2 NVIDIA Tesla M2070-Q cards per node, use resource request "ngpus=[1|2]" and "gpu_model=m2070q"
- (6) 2 NVIDIA Tesla K40m cards per node, use resource request "ngpus=[1|2]" and "gpu_model=k40"
- (7) Use resource request "ssd=true" to request a chunk with SSD in location /ssd1, /ssd2, and /ssd3 (100GB max each)
- (8) Use resource request "nphis=[1|2]" to request phi nodes, the model is Xeon 7120p
- (9) 2 NVIDIA Tesla P100 cards per node, use resource request "ngpus=[1|2]" and "gpu_model=p100"
- (10) 4 NVIDIA Tesla V100 cards per node with NVLINK2, use resource request "ngpus=[1|2|3|4]" and "gpu_model=v100nv"
- (11) 2 NVIDIA Tesla V100 cards per node, use resource request "ngpus=[1|2]" and "gpu_model=v100"
- (12) Phase18a nodes contain only NVMe storage for local_scratch.



2. Assessing the Palmetto cluster

Useful commands:

\$ checkqueuecfg

QUEUE TYPE	min_cores_per_job	max_cores_per_job	max_mem_per_queue	max_jobs_per_queue	max_walltime
MX QUEUES					
cl_solo	1	1	2000gb	1000	168:00:00
cl_single	2	24	24000gb	200	168:00:00
cl_tiny	25	128	51200gb	50	168:00:00
cl_small	129	512	20480gb	5	168:00:00
cl_medium	513	2048	65536gb	4	168:00:00
cl_large	2049	4096	0gb	0	168:00:00
IB QUEUES					
c2_single	1	24	600gb	5	72:00:00
c2_tiny	25	128	4096gb	2	72:00:00
c2_small	129	512	6144gb	1	72:00:00
c2_medium	513	2048	16384gb	1	72:00:00
c2_large	2049	4096	0gb	0	72:00:00
GPU QUEUES					
gpu_small	1	4	1	96	3840gb
gpu_medium	5	16	1	256	6144gb
gpu_large	17	256	1	2048	20480gb
SMP QUEUE					
bigmem	1	64	3	72:00:00	

'max_mem' is the maximum amount of memory all your jobs in this queue can consume at any one time. For example, if the max_mem for the solo queue is 4000gb, and your solo jobs each need 10gb, then you can run a maximum number of $4000/10 = 400$ jobs in the solo queue, even though the current max_jobs setting for the solo queue may be set higher than 400.

2. Assessing the Palmetto cluster

Useful commands:

```
$ checkquota
```

```
$ du -hs
```

```
$ pwd
```

```
$ ls
```

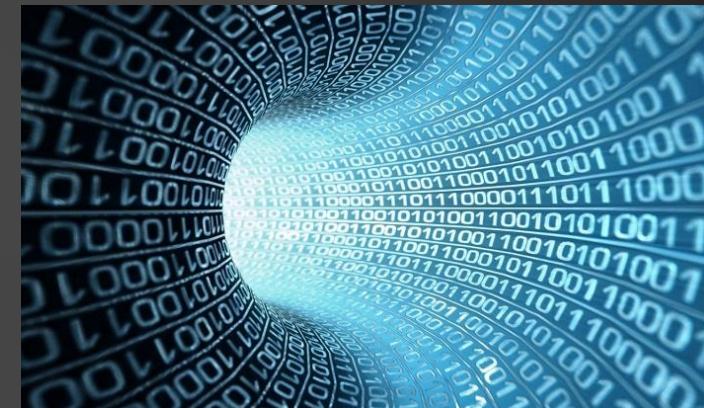
```
$ cd /scratch1/username
```

```
$ cd /scratch2/username
```

Note: Unlimited scratch storage (for model running and temporary storage): deleted after 30 days if untouched

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



3. Transferring files to and from the Palmetto

3.1 Using scp (Mac OS X)

Do not login to Palmetto. Only open your terminal.

Copy from Desktop/PC to Palmetto:

```
$ scp myfile.txt username@xfer01-ext.palmetto.clemson.edu:/home/username
```

Copy from Palmetto to Desktop/PC:

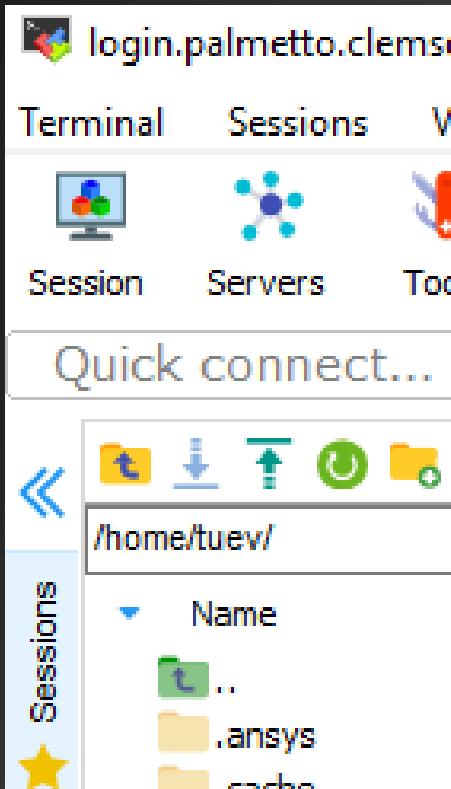
```
$ scp username@xfer01-ext.palmetto.clemson.edu:/home/username/myfile.txt .
```

For folder, using “-r”

```
$ scp -r myfolder username@xfer01-ext.palmetto.clemson.edu:/home/username
```

3. Transferring files to and from the Palmetto

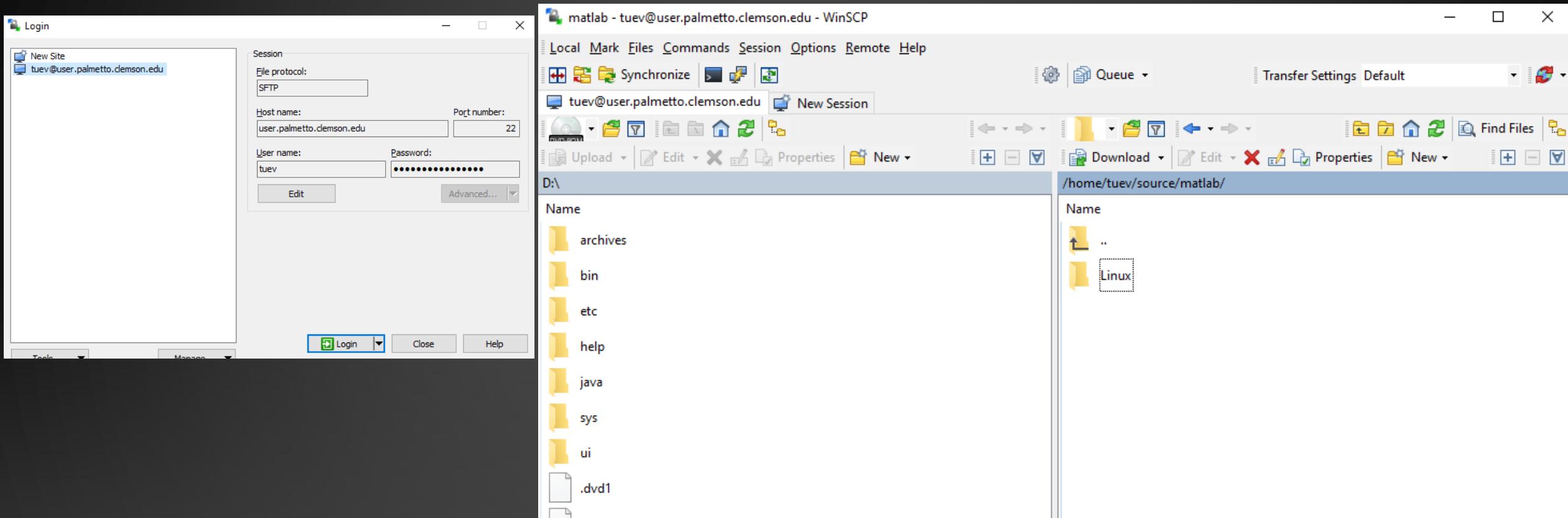
3.2 MobaXterm (For Window)



3. Transferring files to and from the Palmetto

3.2 For Window: use WinSCP

3.3 For Mac: use CyberDuck



3. Transferring files to and from the Palmetto

3.4 Using Globus for large file

Transferring larger files (more than a few megabytes)

For larger files, we recommend using the [Globus](https://www.globus.org/) file transfer application. Here, we demonstrate how to use Globus Online to transfer files between Palmetto and a local machine (laptop). However, Globus can be used for file transfers to/from other locations as well.

1. You will need to have a Globus account set up to begin. Visit <https://www.globus.org/> and set up a Globus account.
2. To begin transferring files, navigate to the Globus Online transfer utility here: <https://www.globus.org/app/transfer>.
3. The transfer utility allows you to transfer files between “endpoints”. You will need to set your local machine as a Globus Connect Personal Endpoint for the file transfer. As a part of this step, you must install the Globus Connect application (see here: <https://www.globus.org/app/endpoints/create-gcp>). After installing and running. You should then be able to set your local machine as one endpoint. In the figure below, the first endpoint is set to My Personal Mac.
4. As the second endpoint, choose clemson#xfer01-ext.palmetto.clemson.edu.
5. You can now transfer files between any locations on your local machine and the Palmetto cluster.

The screenshot shows the 'Transfer Files' interface. It has two main sections for endpoints:

- Endpoint 1:** Set to "My Personal Mac". The path is "/~/workspace/tmp/". The list contains four items:
 - planets (Folder)
 - rsub (Folder)
 - submit.py (File, 2.27 KB)
 - test.pbs (File, 78 B)
- Endpoint 2:** Set to "clemson#xfer01-ext.palmetto.clemson.edu". The path is "/scratch2/atrikut/". The list contains three files:
 - #pio_tmp_1.h5 (264.28 KB)
 - #pio_tmp_1.mpio (262.14 KB)
 - #pio_tmp_1 posix (262.14 KB)

At the top right, there is a "RECENT ACTIVITY" section with icons for files and folders. The bottom right corner features the Clemson University logo.

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



4. Running an Interactive session

Log into a compute node Interactively:

```
$ qsub -I
```

By default, qsub submits a request for 1 core on 1 compute node with 1 GB of RAM for 30 minutes.

4. Running an Interactive session

You can request more (or less) of each of these resources by modifying the qsub command. For instance, to use 2 CPU cores with 8 gb of ram for 10 minutes:

```
$ qsub -I -l select=1:ncpus=2:mem=8gb,walltime=00:10:00
```

```
select=1    # specify number of "chunk" of hardware  
ncpus=2    # specify number of cpu core  
mem=8gb    # number of memory for chunk  
walltime=00:10:00 # request 10 minutes of walltime for the job
```

```
$ qstat -u username
```

You might be assigned to node with MX connection with max walltime of 168 hour

```
|pbs02:
```

Job ID	Username	Queue	Jobname	SessID	NDS	Req'd TSK	Req'd Memory	Req'd Time	S Elap Time
4902375.pbs02	tuev	cl_singl	STDIN	25007	1	2	8gb	00:10	R 00:00

4. Running an Interactive session

To choose different connection speed: “infinity band”

```
$ qsub -I -l select=1:ncpus=2:mem=8gb:interconnect=fdr,walltime=00:10:00
```

```
$ qstat -u username
```

You will be assigned to Infinity Band queue with faster speed but less walltime

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Memory	Time	S	Time
4902383.pbs02	tuev	c2_singl	STDIN	10737	1	2	8gb	00:10	R	00:00

```
|IB QUEUES      min_cores_per_job  max_cores_per_job  max_mem_per_queue  max_jobs_per_queue  max_walltime
|c2_single       1                  24                600gb            5                  72:00:00
```

4. Running an Interactive session

Request more than 1 chunk

```
$ qsub -I -l select=2:ncpus=3:mem=8gb,walltime=00:10:00
```

```
$ qstat -u username
```

pbs02:									
Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S Time
4902417.pbs02	tuev	cl_singl	STDIN	26659	2	6	16gb	00:10 R	00:00
tuev@node0600	1+■								

Working with module

```
$ module avail  
$ module add  
$ module list  
$ module rm  
$ module purge
```

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



5. Running a first job

- Julia: high-level dynamic programming language that was originally designed to address the needs of high-performance numerical analysis and computational science
- The Gurobi Optimizer is a commercial optimization solver for linear programming (LP), quadratic programming (QP), etc.

In this first job I want to use Julia and Gurobi solver to solve a linear math problem using Palmetto HPC



Problem?
Maximize $x + y$

s.t. $50x + 24y \leq 2400$
 $30x + 33y \leq 2100$
 $x \geq 5, y \geq 45$



5. Running a first job



Julia for Mathematical Optimization

Request an Interactive node:

```
$ qsub -I -l select=1:ncpus=8:mem=16gb,walltime=01:00:00
```

Go to working folder:

```
$ cd /scratch1/username  
$ mkdir Julia  
$ cd Julia  
$ nano jump_gurobi.jl  
$ ls
```

Problem?

Maximize $x + y$

s.t. $50x + 24y \leq 2400$
 $30x + 33y \leq 2100$
 $x \geq 5, y \geq 45$

using JuMP, Gurobi

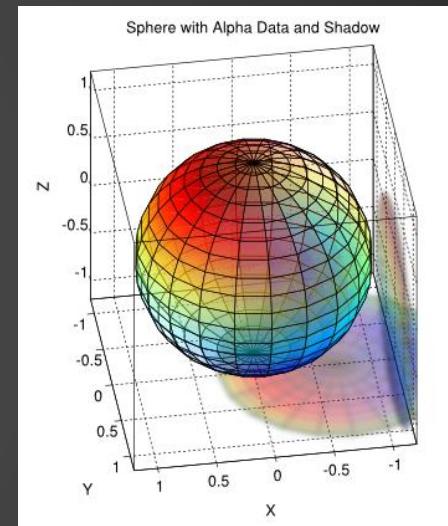
```
m = Model(solver=GurobiSolver(Presolve=0))  
  
@variable(m, x >= 5)  
@variable(m, y >= 45)  
  
@objective(m, Max, x + y)  
@constraint(m, 50x + 24y <= 2400)  
@constraint(m, 30x + 33y <= 2100)  
  
status = solve(m)  
println(" x = ", getvalue(x), " y = ", getvalue(y))
```

5. Running a first job

```
$ module add julia/0.6.2 gurobi/7.0.2
$ module list
$ julia
julia > Pkg.add("JuMP")
julia > Pkg.add("Gurobi")
Julia > quit()
$ julia jump_gurobi.jl
```

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



6. Running a job with graphical display

Mac OS X users will need to install XQuartz: <https://www.xquartz.org/>.

After installing Xquartz, do not open/launch it.

Open a terminal, and enter the following command **without logging in to Palmetto**:

```
$ defaults write org.macosforge.xquartz.X11 enable_iglx -bool true
```

```
ssh -Y username@login.palmetto.clemson.edu
```

6. Running a job with graphical display

```
$ qsub -I -X  
$ glxgears
```

```
$ module add matlab/2017a  
$ matlab
```

Mac user:

```
ssh -Y username@login.palmetto.clemson.edu
```

6. Running a job with graphical display

Palmetto Login Virtual Machines
(LoginVM)

- Virtual Machine that emulates in software the login node on the Palmetto Cluster
- Each VM can be configured with its own hardware specifications like the number of CPU cores, memory size, and hard drive volume size.

2 major features:

- Remote Desktop Access: alternative way to run X server on the local desktop
- Exclusive use of VM so user can use VM like their own workstation

6. Running a job with graphical display

Palmetto Login Virtual Machines (LoginVM)

Request your VM

Step 1. In a browser, go to the URL: <https://www.palmetto.clemson.edu/loginvm/>.

Step 2. Enter your Clemson Username and Password on the Clemson login page.

Step 3. On the Palmetto Login VM page, click the link Request VM.

Step 4. On the Request VM page, choose an appropriate template, click the button Submit. This page does not show up if you already have an assigned VM.

Step 5. If everything works correctly, a Response page will show the IP address and other information of the VM assigned to you. You can use this IP Address to access your VM.

6. Running a job with graphical display

Palmetto Login Virtual Machines
(LoginVM)

Access VM

Remote Desktop	Secure Shell
Windows: Remote Desktop Connections	Use the same approach to ssh to Palmetto. The remote host is changed to the IP address
MacOS: Microsoft/Apple Remote Desktop	

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



7. How to install your software and configure your .bashrc

In this example, I'm gonna show you how to download installer source and install that in your home directory: the GNU hello program

First, download the installer:

```
$ qsub -I  
$ mkdir source  
$ mkdir Applications  
$ cd source  
$ wget http://ftp.gnu.org/gnu/hello/hello-2.10.tar.gz  
  
$ tar -xvf hello-2.10.tar.gz  
$ ls  
$ cd hello-2.10/  
$ ls
```

7. How to install your software and configure your .bashrc

Second: see the instruction for install

```
$ cat INSTALL
```

```
$ ./configure --prefix=/home/username/Applications  
$ make  
$ make install  
$ which hello
```

7. How to install your software and configure your .bashrc

Third: edit the .bashrc

```
$ nano ~/.bashrc
```

```
export PATH=$PATH:$HOME/Applications/bin  
alias chao = '("~/Applications/bin/hello'"
```

```
$ source ~/.bashrc  
$ cd  
$ hello  
$ chao
```

Available software installer in Palmetto

Palmetto Cluster Documentation

General Information ▾

User's Guide ▾

Software Guide ▾

- Software on Palmetto
- ABAQUS
- Amber
- ANSYS
- COMSOL
- GROMACS
- HOOMD
- Java
- LAMMPS**
- MATLAB
- MrBayes
- Paraview
- Rclone
- Singularity
- Tassel
- Tensorflow
- Trinity

Programmer's Guide ▾

JupyterHub ▾

Owner's Guide ▾

LAMMPS

Table of Contents

- [Installing LAMMPS on Palmetto cluster](#)
- [Running LAMMPS - an example](#)

There are a few different versions of LAMMPS available on the cluster, but users are encouraged to install their own version of LAMMPS in case newer versions different configurations are desired.

```
$ module avail lammps
lammps/10Jan15-dp    lammps/17Dec13-dp    lammps/17Dec13-dp-k20 lammps/29Aug14-sp-k20
```

Installing LAMMPS on Palmetto cluster

In this example, we will demonstrate installing LAMMPS (version 22Aug18).

1. After logging in, ask for an interactive session (with GPU):

```
$ qsub -I -l select=1:ncpus=8:mpiprocs=8:ngpus=1:mem=32gb:gpu_model=k40,walltime=2:00:00
```

2. Load the required modules. Specifically, note that we are loading the CUDA-enabled module `openmpi/1.10.3`:

```
$ module load gcc/5.4.0
$ module load openmpi/1.10.3
$ module load fftw/3.3.4-g481
$ module load cuda-toolkit/8.0.44
$ module load cmake/3.10.0
```

3. Download the LAMMPS source code from <http://lammps.sandia.gov/download.html>. The detailed instruction on usage and compilation options are available at <http://lammps.sandia.gov/doc/Manual.html>.
4. Unpack the source code and enter the package directory:

```
$ tar -xvf lammps-stable.tar.gz
$ cd lammps-22Aug18
```

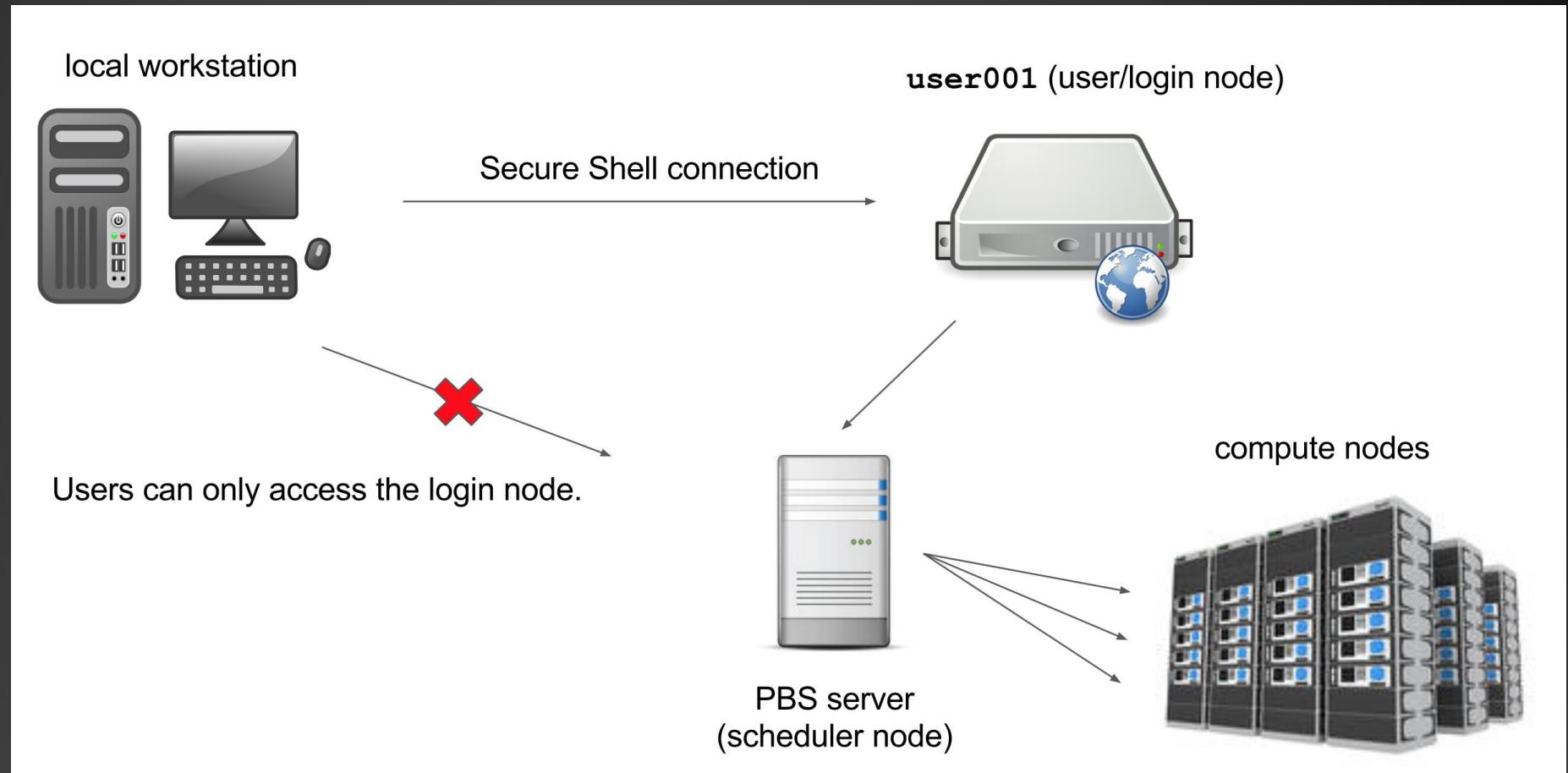
INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



8. Submit batch job

Using PBS scheduler



8. Submit batch job

The first batch job

```
$ cd /scratch1/username/  
$ mkdir bash1  
$ cd bash1  
$ nano myfirst.pbs
```

```
#PBS -N my_first_batch_job  
#PBS -l select=1:ncpus=4:mem=8gb:interconnect=fdr  
#PBS -l walltime=00:30:00  
#PBS -j oe  
#PBS -m abe
```

```
echo Starting to sleep for 100s
```

```
sleep 100
```

PBS job options

The following switches can be used either with `qsub` on the command line, or with a `#PBS` directive in a batch script.

Parameter	Purpose	Example
<code>-N</code>	Job name (7 characters)	<code>-N maxrun1</code>
<code>-l</code>	Job limits (lowercase L), hardware & other requirements for job.	<code>-l select=1:ncpus=8:mem=1gb</code>
<code>-q</code>	Queue to direct this job to (<code>workq</code> is the default, <code>supabad</code> is an example of specific research group's job queue)	<code>-q supabad</code>
<code>-o</code>	Path to stdout file for this job (environment variables are not accepted here)	<code>-o stdout.txt</code>
<code>-e</code>	Path to stderr file for this job (environment variables are not accepted here)	<code>-e stderr.txt</code>
<code>-m</code>	mail event: Email from the PBS server with flag <code>abort\begin\end\or no</code> mail for job's notification.	<code>-m abe</code>
<code>-M</code>	Specify list of user to whom mail about the job is sent. The user list argument is of the form: <code>[user[@host],user[@host],...]</code> . If <code>-M</code> is not used and <code>-m</code> is specified, PBS will send email to <code>userid@clemson.edu</code>	<code>-M user1@domain1.com,user2@domain2.com</code>
<code>-j oe</code>	Join the output and error streams and write to a single file	<code>-j oe</code>
<code>-r n</code>	Ask PBS not to restart the job if it's failed	<code>-r n</code>

8. Submit batch job

```
$ qsub myfirst.pbs  
  
$ qstat  
  
$ qstat -u username  
  
#Debug output and error file
```

Job submission and control on Palmetto

The Palmetto cluster uses the Portable Batch Scheduling system (PBS) to manage jobs. Here are some basic PBS commands for submitting, querying and deleting jobs:

Command	Action
qsub -I	Submit an interactive job (reserves 1 core, 1gb RAM, 30 minutes walltime)
qsub xyz.pbs	Submit the job script xyz.pbs
qstat <job id>	Check the status of the job with given job ID
qstat -u <username>	Check the status of all jobs submitted by given username
qstat -xf <job id>	Check detailed information for job with given job ID
qsub -q <queuename> xyz.pbs	Submit to queue queuename
qdel <job id>	Delete the job (queued or running) with given job ID
qpeek <job id>	“Peek” at the standard output from a running job
qdel -Wforce <job id>	Use when job not responding to just qdel

8. Submit batch job

The second batch job

```
$ cd /scratch1/username/Julia  
$ nano run_julia.pbs
```

```
#PBS -N run_julia  
#PBS -l select=1:ncpus=4:mem=8gb:interconnect=fdr  
#PBS -l walltime=00:30:00  
#PBS -j oe  
#PBS -m abe  
  
cd $PBS_O_WORKDIR  
module add julia/0.6.2 gurobi/7.0.2  
julia jump_gurobi.jl
```

8. Submit batch job

<https://github.com/clemsonciti/palmetto-examples>

Example PBS scripts

A list of example PBS scripts for submitting jobs to the Palmetto cluster can be found [here](#).

This branch is 33 commits ahead of shwina:master.	
 shwina	Update ABAQUS script
 ABAQUS	Update ABAQUS script
 ANSYS	Update copies to local_scratch
 Amber	formatting fixes
 COMSOL	Add ANSYS and COMSOL examples
 CUDA	Adding ABAQUS example
 GNU-Parallel	Adding ABAQUS example
 Gaussian-parallel	pull everything up one directory
 Gaussian	pull everything up one directory
 Installation	pull everything up one directory
 Job-arrays	Update README.md
 LAMMPS	Revert to old job script for LAMMPS
 MATLAB	Adding ABAQUS example
 MPI	Add MPI hello world
 Mathematica	formatting fixes
 NAMD	formatting fixes
 OpenMP	formatting fix

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



9. Parallelism

Often, you may need to perform *high-throughput* computations, i.e., run many independent tasks concurrently, with each task operating on a different piece of data. There are primarily two ways of performing such computations on Palmetto:

1. Using job arrays
2. Using GNU parallel

Example: Create 8 new folders. Go to each folder, create a text file

Hint:

- **mkdir:** to make new folder
- **cd:** to change directory
- **touch:** to create a new file

9. Parallelism

Using job arrays

```
$ cd /scratch1/username  
$ mkdir Parallel  
$ cd Parallel  
$ nano parallel_array.pbs
```

```
#PBS -N Test_Array_Job  
#PBS -l select=1:ncpus=1:mem=1gb  
#PBS -l walltime=00:02:00  
#PBS -j oe  
#PBS -J 1-8  
  
cd $PBS_O_WORKDIR  
  
mkdir dir_PBS_${PBS_ARRAY_INDEX}  
cd dir_PBS_${PBS_ARRAY_INDEX}  
touch file${PBS_ARRAY_INDEX}.txt
```

9. Parallelism

Using GNU parallel

```
$ nano parallel_gnu.pbs
```

```
#PBS -N gnu-parallel-example
#PBS -l select=1:ncpus=8:mem=1gb,walltime=00:05:00
#PBS -j oe

module add gnu-parallel

cd $PBS_O_WORKDIR

seq 1 8 | parallel -j8 'mkdir GNU_{ }; cd GNU_{ }; touch file_{ }'
```

INTRODUCTION TO RESEARCH COMPUTING ON THE PALMETTO CLUSTER

1. Introduction to Clemson super-computer: the Palmetto
2. Assessing the Palmetto cluster
3. Transferring files to and from the Palmetto
4. Running an Interactive session
5. Running a first job
6. Running a job with graphical display
7. How to install your software and configure your .bashrc
8. Submitting batch jobs
9. Parallelism
10. JupyterHub



10. JupyterHub

JupyterHub allows users to interact with the Palmetto cluster from their web browsers using the [Jupyter Notebook](#) interface, and to prototype and develop code in **Python**, **R**, **MATLAB** and several other languages.

Logging In

<https://www.palmetto.clemson.edu/jupyterhub>.

Common problems/issues

Program crashes on login node with message `Killed`

When running commands or editing files on the login node, users may notice that their processes end abruptly with the error message `Killed`. Processes with names such as `a.out`, `matlab`, etc., are automatically killed on the login node because they may consume excessive computational resources. Unfortunately, this also means that benign processes, such as editing a file with the word `matlab` as part of its name could also be killed.

Solution: Request an interactive session on a compute node (`qsub -I`), and then run the application/command.

Home or scratch directories are sluggish or unresponsive

The `/home` and `/scratch` directories can become slow/unresponsive when a user (or several users) read/write large amounts of data to these directories. When this happens, all users are affected as these filesystems are shared by all nodes of the cluster.

To avoid this issue, keep in mind the following:

1. **Never** use the `/home` directory as the working directory for jobs that read/write data. If too many jobs read/write data to the `/home` directory, it can render the cluster unusable by all users. Copy any input data to one of the `/scratch` directories and use that `/scratch` directory as the working directory for jobs. Periodically move important data back to the `/home` directory.
2. Try to use `/local_scratch` whenever possible. Unlike `/home` or the `/scratch` directories, which are shared by all nodes, each node has its own `/local_scratch` directory. It is much faster to read/write data to `/local_scratch`, and doing so will not affect other users. (see example [here])
(https://www.palmetto.clemson.edu/palmetto/userguide_howto_choose_right_filesystem.html).

CONDOMINIUM NODE

Condominium model

Palmetto cluster operates in a condominium model which allows faculty to purchase immediate access to compute nodes on the cluster. More information can be found in the [Owner's Guide](#).

<https://www.palmetto.clemson.edu/palmetto/owners.html>

Node pricing \$7500/4 years

Storage pricing: \$150/1TB/4 years

Note:

- Palmetto office hour: every Wednesday 8-11am Cooper 412
- Palmetto onboarding: every Friday 9-10am Barre Hall 2108

THANK YOU