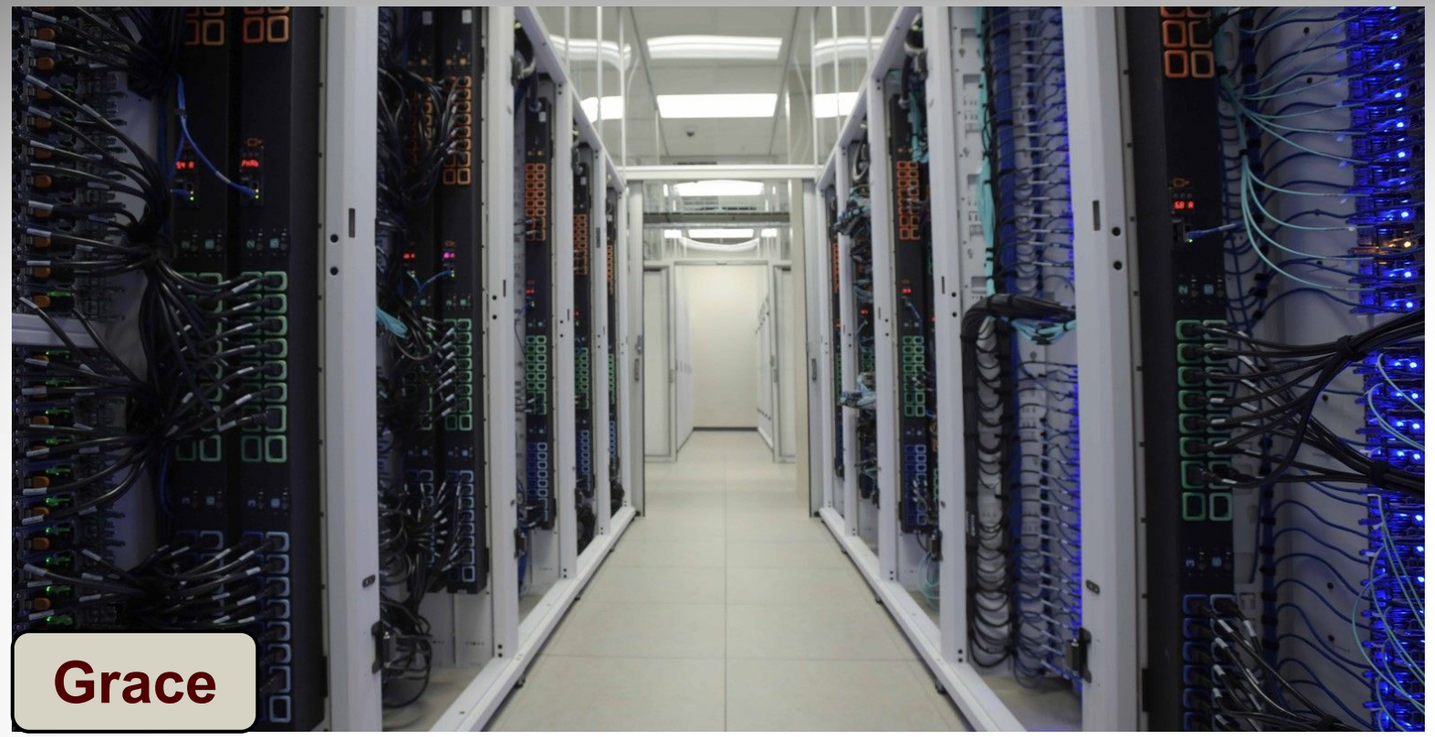
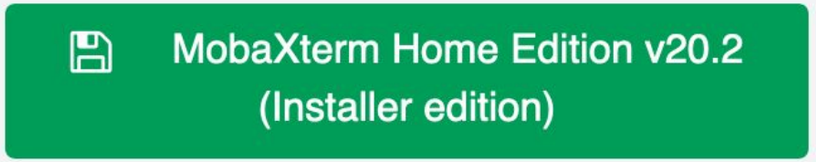


Windows Users:

To follow along, please download Mobaxterm
(a free SSH client)

Google Search “Mobaxterm” or navigate to:
https://mobaxterm.mobatek.net/download.html

Download the “Installer edition” (green button)



HPRC Clusters:

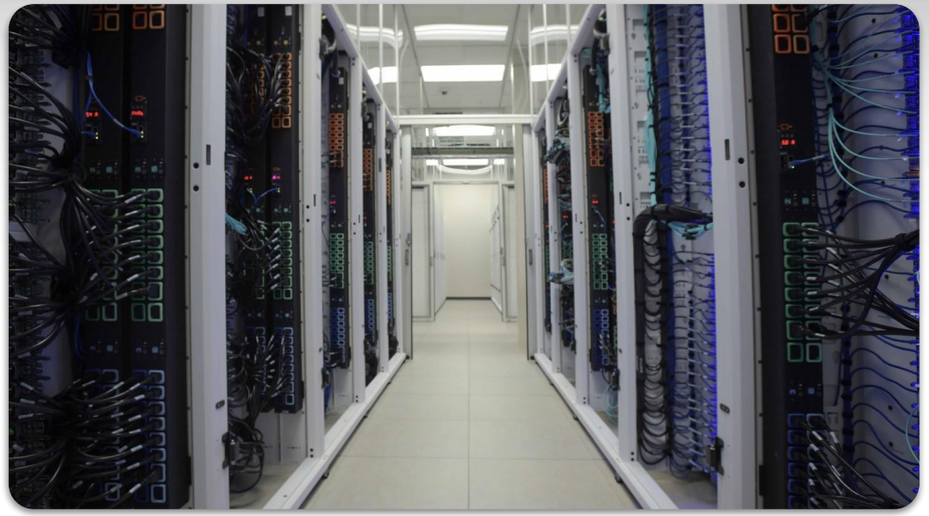
Grace Primer

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

Fall 2021

HPRC's Flagship Cluster

Grace is a 925-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect, A100 GPUs, RTX 6000 GPUs and T4 GPUs. The 925 nodes are based on the Intel Cascade Lake processor.



48 cores/node

3TB Large Memory-80 cores/nodes

Login Nodes:
10 GbE TAMU network connection

Login Nodes	5
384GB memory general compute nodes	800
GPU - A100 nodes with 384GB memory	100
GPU - RTX 6000 nodes with 384GB memory	9
GPU - T4 nodes with 384GB memory	8
3TB Large Memory	8

For more information:
https://hprc.tamu.edu/wiki/Grace:Intro

Batch Computing on HPRC Clusters

A batch job script is a text file that contains Unix and software commands and Batch manager job parameters

On-campus:

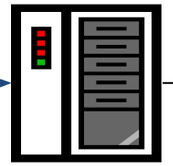
Campus Network

VPN

Internet

Off-campus:

SSH



Login nodes

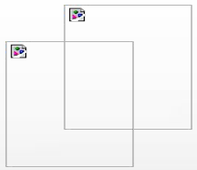
Create job

Job script file

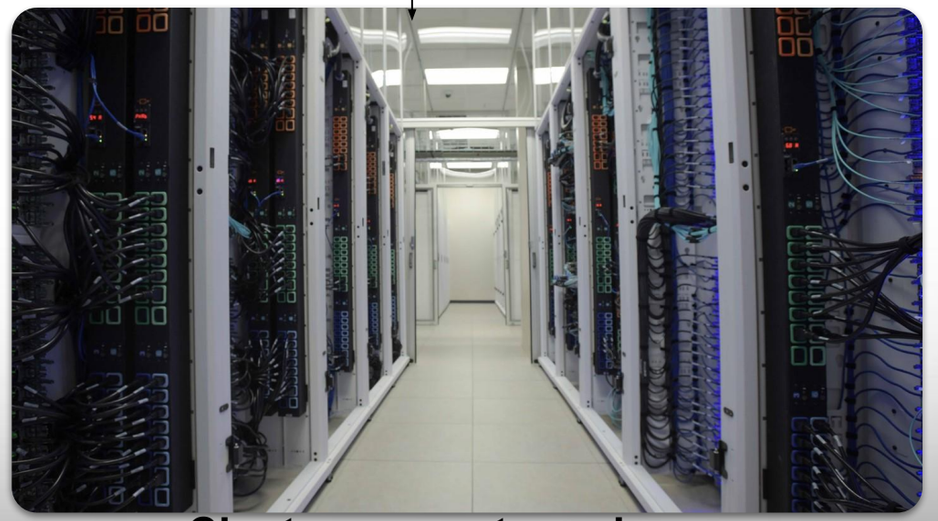
Submit job

Batch manager:
Slurm on Grace

Queue



Output Files



Cluster compute nodes

Accessing Grace: via SSH

- SSH command is required for accessing Grace / Terra:
  - On campus: `ssh NetID@grace.tamu.edu`
  - Off campus:
    - Set up and start VPN (Virtual Private Network): u.tamu.edu/VPnetwork
    - Then: `ssh NetID@grace.tamu.edu`
  - *Two-Factor Authentication* enabled for CAS, VPN, SSH

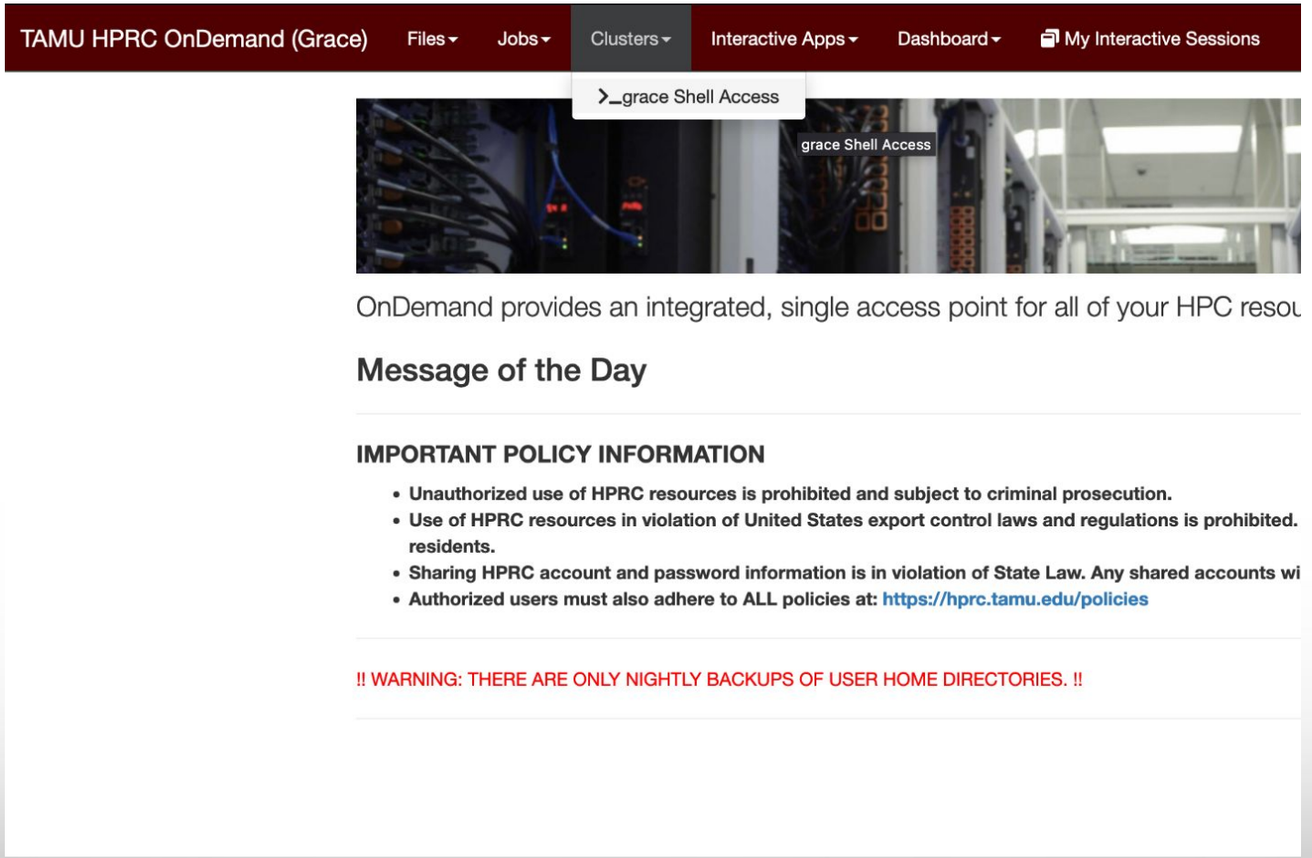
- SSH programs for Windows:
  - MobaXTerm (preferred, includes SSH and X11)
  - PuTTY SSH
  - Windows Subsystem for Linux
- Grace has 5 login nodes. Check the bash prompt.

Login sessions that are idle for **60** minutes will be closed automatically
Processes run longer than **60** minutes on login nodes will be killed automatically.
Do not use more than 8 cores on the login nodes!
Do not use the sudo command.

hprc.tamu.edu/wiki/HPRC:Access

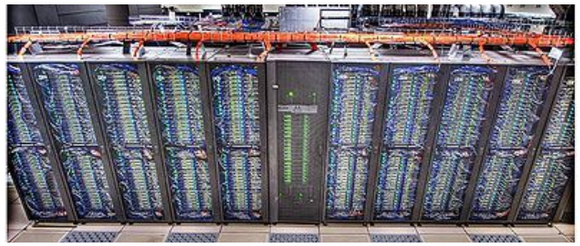
Accessing Grace: via the Portal

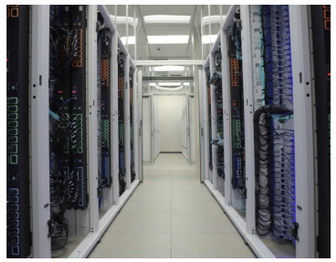
- Access through (most) web browsers:
  portal.hprc.tamu.edu
  Top Banner Menu “Clusters” -> “Grace Shell Access”



hprc.tamu.edu/wiki/HPRC:Access

Pop Quiz





How many login nodes are on Grace?

A. 1
B. 5
C. 3
D. 10

Two-Factor Authentication

- Duo NetID two-factor authentication to enhance security (it.tamu.edu/duo/)
  - All web login (howdy, portal.hprc.tamu.edu, Globus) through CAS
  - VPN to TAMU campus (since Oct 1st, 2018)
  - SSH/SFTP to HPRC clusters (since Nov 4th, 2019)

- See instructions in two-factor wiki page (hprc.tamu.edu/wiki/Two_Factor)
- SSH clients work with Duo
  - ssh command from Linux, macOS Terminal, Windows cmd
  - MobaXterm for Windows (click on “Session” icon or via local session: hit “enter” 3 times and wait for “Password:” prompt)
  - Putty for Windows
- SFTP clients work with Duo
  - scp/sftp command from Linux, macOS Terminal, Windows cmd
  - WinSCP for Windows
  - Cyberduck for macOS

- Not all software supports SSH+Duo: SFTP in Matlab

hprc.tamu.edu/wiki/Two_Factor

Example: SSH login with Duo

```
$ ssh grace.tamu.edu
**************************************************************
…. warning message (snipped) …….
**************************************************************

Password:
Duo two-factor login for UserNetID

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-1234
 2. Phone call to XXX-XXX-1234
 3. SMS passcodes to XXX-XXX-1234 (next code starts with: 9)

Passcode or option (1-3): 1
Success. Logging you in...
```

File Systems and User Directories

Directory	Environment Variable	Space Limit	File Limit	Intended Use
/home/$USER	$HOME	10 GB	10,000	Small to modest amounts of processing.
/scratch/user/$USER	$SCRATCH	1 TB	250,000	Temporary storage of large files for on-going computations. Not intended to be a long-term storage area.

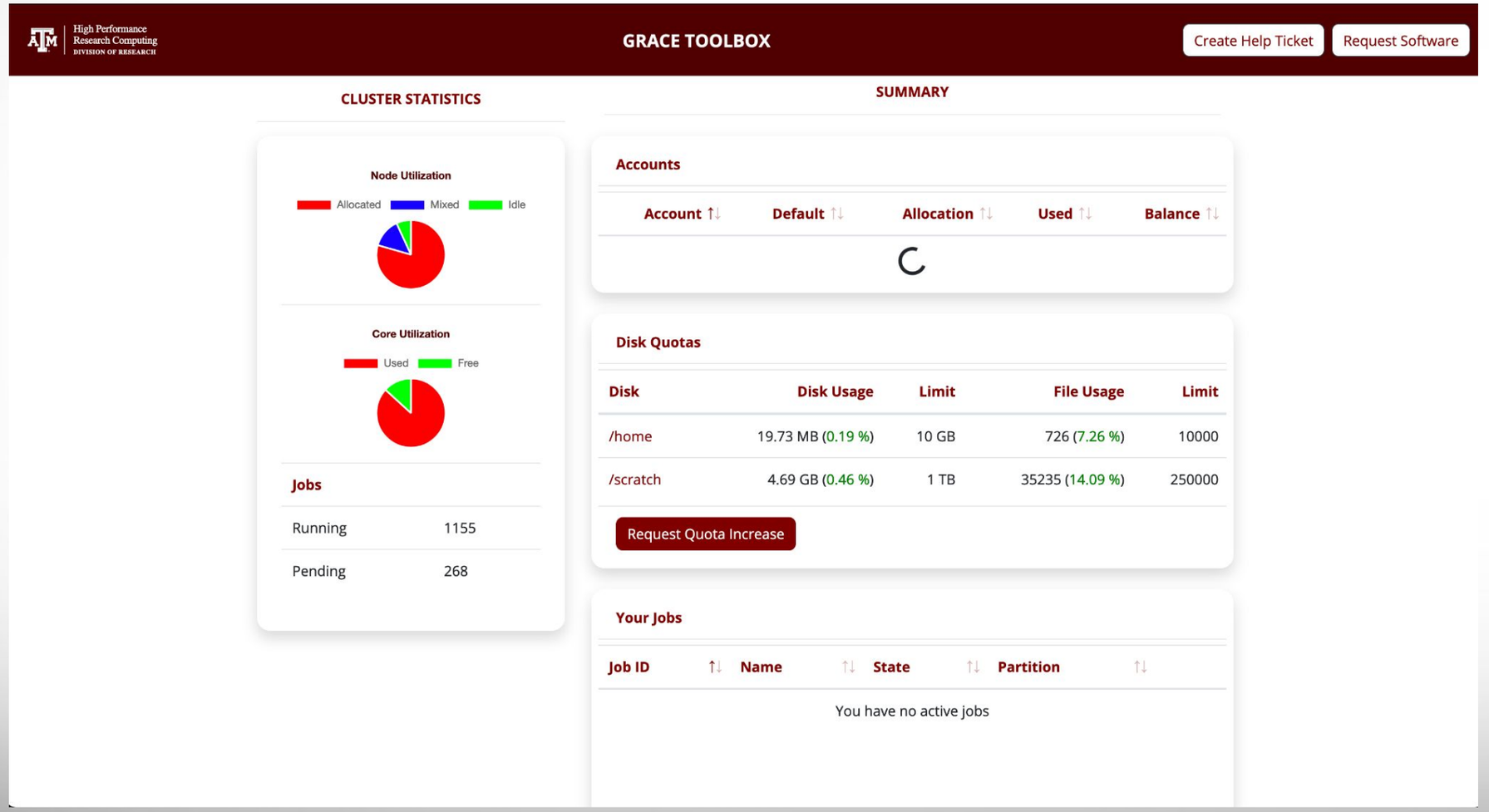
- **`$HOME`** and **`$SCRATCH`** directories are not shared between Grace and Terra clusters.
- View usage and quota limits using the command: **`showquota`**

- Request a group directory for sharing files.
- Do not share your home or scratch directories.

hprc.tamu.edu/wiki/Grace:Filesystems_and_Files

OOD Dashboard: Grace

Easily view Cluster utilization, Storage Quotas & Allocation Balances



Quota and file limit increases will only be considered for scratch directories

Preferred way to request *Quota Increases*

Software

- See the Software wiki page for instructions and examples
  - hprc.tamu.edu/wiki/SW
  - hprc.tamu.edu/software/grace

- License-restricted software
  - Contact license owner for approval
- Contact us for software installation help/request
  - User can install software in their home/scratch directory
  - Do not run the "*sudo*" command when installing software

Software: Application Modules

- Installed applications are made available with the module system
  Grace uses a *software hierarchy* inside the module system

  In this hierarchy, the user loads a compiler which then makes available Software built with the currently loaded compiler

```
module avail                                  # shows which software is available

module load GCCcore/9.3.0                     # load GCC compiler version 9.3.0

module avail                                  # show which software is now available

module load BeautifulSoup/4.9.1-Python-3.8.2  # loads BeautifulSoup version 4.9.1
```

Software: Modules and Toolchains

- Toolchains are what we call groups of compilers & libraries
- There's a variety of toolchains on the clusters:
    - `intel/2018b`
    - `iomkl/2018b`
    - `foss/2018b`
    - `GCCcore/7.3.0`

```
module purge                    # removes all loaded modules
```

Consumable Computing Resources

- Resources specified in a job file:
  - Processor cores
  - Memory
  - Wall time
  - GPU
- Service Unit (SU) - Billing Account
  - Use "**`myproject`**" to query
    hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

`myproject`

```
================================================================================
             List of YourNetID's Project Accounts
--------------------------------------------------------------------------------
|   Account    |   FY   | Default | Allocation |Used & Pending SUs|    Balance   |     PI     |
--------------------------------------------------------------------------------
|1228000223136|  2019 |        N|    10000.00|              0.00|    10000.00|Doe, John   |
--------------------------------------------------------------------------------
|1428000243716|  2019 |        Y|     5000.00|           -71.06|     4928.94|Doe, Jane   |
--------------------------------------------------------------------------------
```

- Other resources:
  - Software license/token
    - Use "**`license_status`**" to query
    - hprc.tamu.edu/wiki/SW:License_Checker

`license_status -a`

Find available license for "ansys":

`license_status -s ansys`

```
License status for ANSYS:
----------------------------------------------------------------
|License Name                |  # Issued|  # In Use|# Available|
----------------------------------------------------------------
|aa_mcad                     |        50|         0|         50|
|aa_r                        |        50|        32|         18|
|aim_mp1                     |        50|         0|         50|
|    ......                                                    |
----------------------------------------------------------------
```

Find detail options:

`license_status -h`

Check your Service Unit (SU) Balance

- List the SU Balance of your Account(s)

```
myproject
```

```
 ================================================================================================
              List of YourNetID's Project Accounts
------------------------------------------------------------------------------------------------
|   Account    |    FY    | Default | Allocation |Used & Pending SUs|   Balance   |    PI     |
------------------------------------------------------------------------------------------------
|1228000223136|  2019    |       N|    10000.00|              0.00|    10000.00|Doe, John |
------------------------------------------------------------------------------------------------
|1428000243716|  2019    |       Y|     5000.00|            -71.06|     4928.94|Doe, Jane |
------------------------------------------------------------------------------------------------
|1258000247058|  2019    |       N|     5000.00|             -0.91|     4999.09|Doe, Jane |
------------------------------------------------------------------------------------------------
```

- Run "**`myproject -d Account#`**" to change default project account
- Run "**`myproject -h`**" to see more options

hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
hprc.tamu.edu/wiki/HPRC:AMS:UI

Batch Queues

Job submissions are auto-assigned to batch queues based on the resources requested (number of cores/nodes and walltime limit)

hprc.tamu.edu/wiki/Terra:Batch#Queues

sinfo : Current Queues on **Grace**

dylan — dylan@login4:~ — ssh

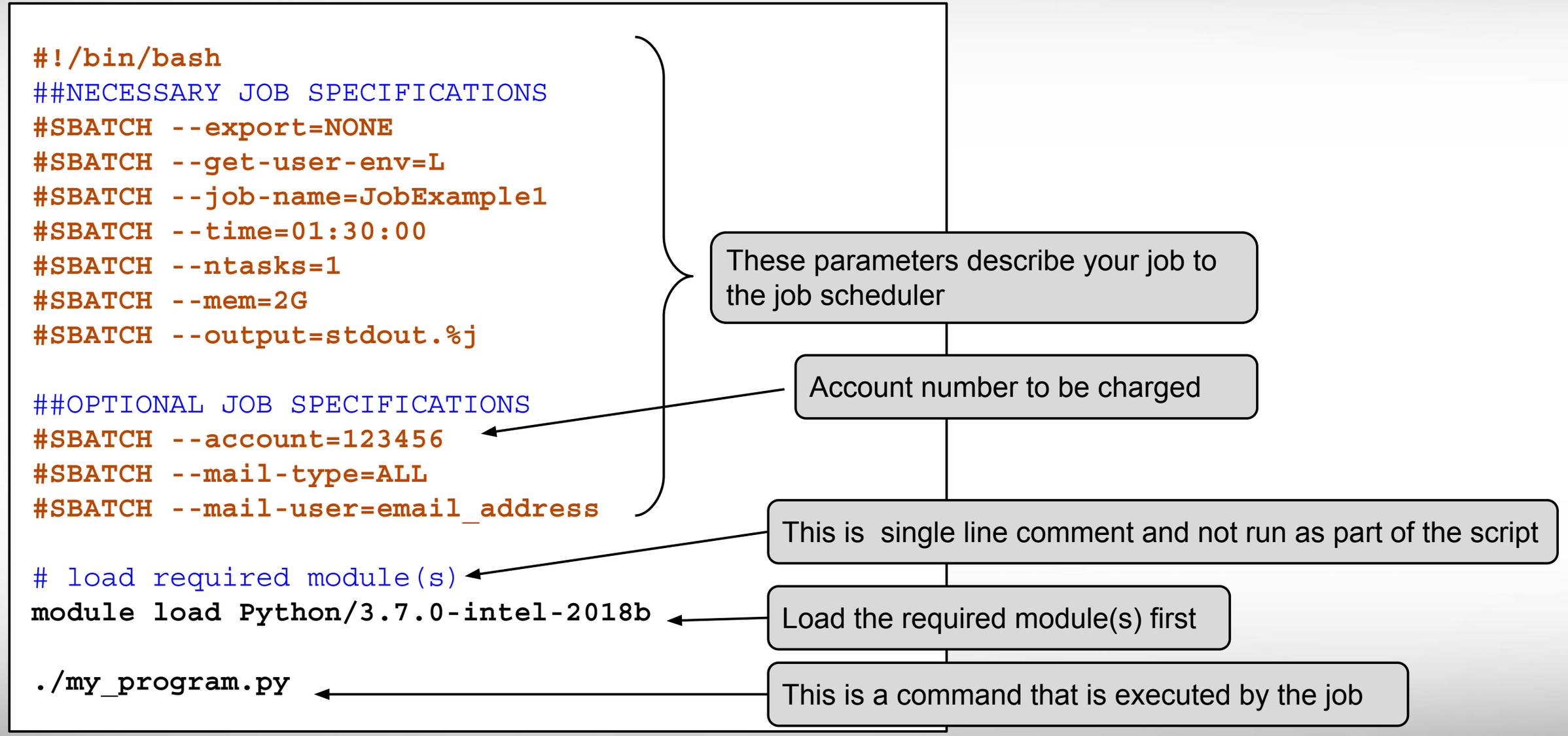
```
[dylan@grace4 ~]$ sinfo
PARTITION        AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)   CPUS(A/I/O/T)
short*           up     2:00:00      1-32        768/0/32/800     34768/1836/1796/3840
medium           up     1-00:00:00   1-128       768/0/32/800     34768/1836/1796/3840
long             up     7-00:00:00   1-64        768/0/32/800     34768/1836/1796/3840
xlong            up     21-00:00:00  1-32        768/0/32/800     34768/1836/1796/3840
vnc              up     12:00:00     1-32        49/63/5/117      2192/3184/240/5616
gpu              up     4-00:00:00   1-32        49/63/5/117      2192/3184/240/5616
bigmem           up     2-00:00:00   1-4         2/5/1/8          160/400/80/640
staff            up     infinite     1-infinite  817/63/37/917    36960/5020/2036/4401
special          up     7-00:00:00   1-infinite  817/63/37/917    36960/5020/2036/4401
[dylan@grace4 ~]$
```

For the NODES and CPUS columns:

A = Active (in use by running jobs)
I = Idle (available for jobs)
O = Offline (unavailable for jobs)
T = Total

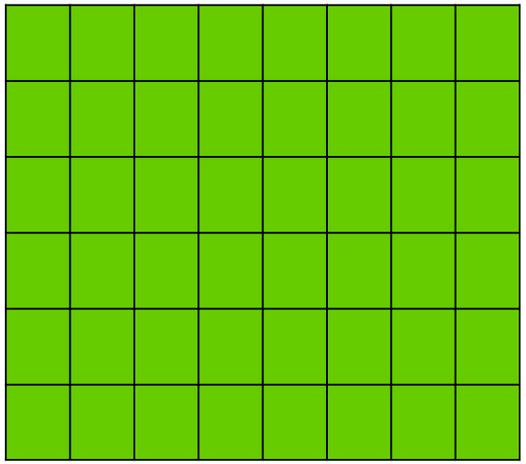
Sample Job Script Structure (**Grace**)

```
#!/bin/bash
##NECESSARY JOB SPECIFICATIONS
#SBATCH --export=NONE
#SBATCH --get-user-env=L
#SBATCH --job-name=JobExample1
#SBATCH --time=01:30:00
#SBATCH --ntasks=1
#SBATCH --mem=2G
#SBATCH --output=stdout.%j

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456
#SBATCH --mail-type=ALL
#SBATCH --mail-user=email_address

# load required module(s)
module load Python/3.7.0-intel-2018b

./my_program.py
```



Mapping Jobs to Cores per Node on **Grace**

A.

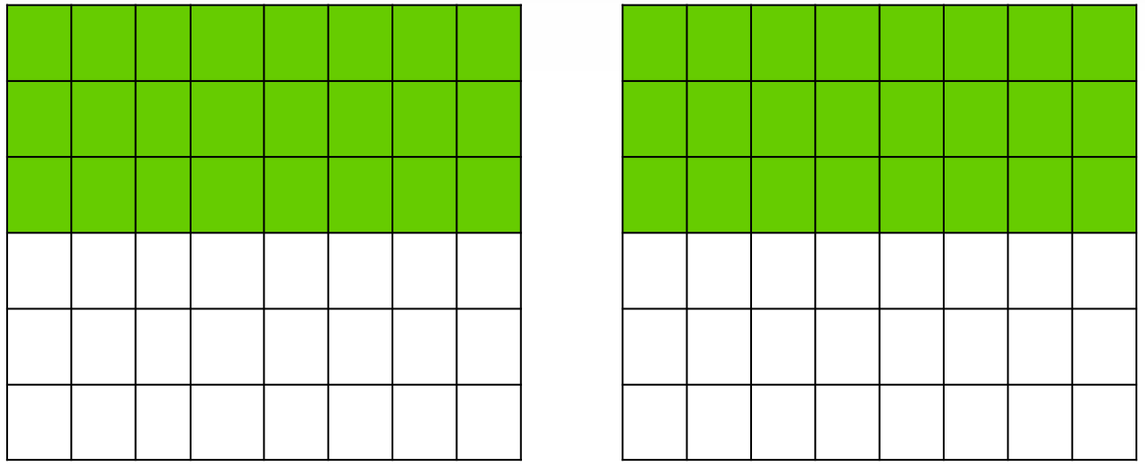


48 cores on
1 compute node

#SBATCH --ntasks=48
#SBATCH --tasks-per-node=48

Preferred Mapping
(if applicable)

B.



48 cores on
2 compute nodes

#SBATCH --ntasks=28
#SBATCH --tasks-per-node=24

Important Batch Job Parameters

Grace	Comment
`#SBATCH --export=NONE` `#SBATCH --get-user-env=L`	Initialize job environment.
`#SBATCH --time=HH:MM:SS`	Specifies the time limit for the job.
`#SBATCH --ntasks=NNN`	Total number of tasks (cores) for the job.
`#SBATCH --ntasks-per-node=XX`	Specifies the maximum number of tasks (cores) to allocate per node

`#SBATCH --mem=nnnnM`
`or`
`#SBATCH --mem=nG`
 	
`(memory per NODE)`
	Sets the maximum amount of memory (MB).
	
	G for GB is supported on Grace

hprc.tamu.edu/wiki/HPRC:Batch_Translation

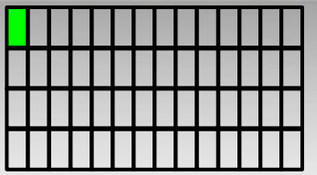
Pop Quiz

```
#SBATCH --export=NONE
#SBATCH --get-user-env=L
#SBATCH --job-name=stacks_S2
#SBATCH --ntasks=80
#SBATCH --ntasks-per-node=20
#SBATCH --mem=40G
#SBATCH --time=48:00:00
#SBATCH --output=/scratch/user/dylan/stdout.%J
#SBATCH --error stderr.%J
```

How many nodes is this job requesting?

A. 1600
B. 80
C. 20
D. 4

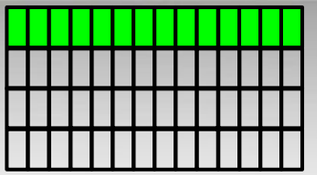
Grace Job File (Serial Example)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample1       # Set the job name to "JobExample1"
#SBATCH --time=01:30:00              # Set the wall clock limit to 1hr and 30min
#SBATCH --ntasks=1                   # Request 1 task (core)
#SBATCH --mem=2G                     # Request 2GB per node
#SBATCH --output=stdout.%j           # Send stdout and stderr to "stdout.[jobID]"

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# load required module(s)
module load intel/2018b

# run your program
./myprogram
```

SUs = 1.5

Grace Job File (multi core, single node)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample2       # Set the job name to "JobExample2"
#SBATCH --time=6:30:00               # Set the wall clock limit to 6hr and 30min
#SBATCH --nodes=1                    # Request 1 node
#SBATCH --ntasks-per-node=14         # Request 14 tasks(cores) per node
#SBATCH --mem=28G                    # Request 28GB per node
#SBATCH --output=stdout.%j           # Send stdout to "stdout.[jobID]"
#SBATCH --error=stderr.%j            # Send stderr to "stderr.[jobID]"
##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456     # Set billing account to 123456  #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# load required module(s)
module load intel/2018b

# run your program
./my_multicore_program
```

SUs = 91

Job Memory Requests on **Grace**

- Specify memory request based on memory per node:

  #SBATCH --mem=xxxxM # memory per node in MB

  or

  #SBATCH --mem=xG # memory per node in GB

- On 384GB nodes, usable memory is at most 360 GB.

  The per-process memory limit should not exceed ~7500 MB for a 48-core job.

- On 3TB nodes, usable memory is at most 2900 GB.

  The per-process memory limit should not exceed 37120 MB for a 48-core job.

Submitting Your Job and Check Job Status

Submit job

```
sbatch example01.job
```

```
Submitted batch job 161997
(from job_submit) your job is charged as below
                Project Account: 122792016265
                Account Balance: 1687.066160
                Requested SUs:   3
```

Check status

```
squeue -u netID
```

```
JOBID   NAME      USER      PARTITION  NODES  CPUS  STATE    TIME   TIME_LEFT  START_TIME          REASON      NODELIST
64039   somejob   someuser  medium     4      112   PENDING  0:00   20:00      2017-01-30T21:00:4  Resources
64038   somejob   someuser  medium     4      112   RUNNING  2:49   17:11      2017-01-30T20:40:4  None        tnxt-[0401-0404]
```

Job Submission and Tracking

Grace	Description
`sbatch jobfile1`	Submit jobfile1 to batch system
`squeue [-u user_name] [-j job_id]`	List jobs
`scancel job_id`	Kill a job
`sacct -X -j job_id`	Show information for a job (can be when job is running or recently finished)
`sacct -X -S YYYY-HH-MM`	Show information for all of your jobs since YYYY-HH-MM
`lnu job_id`	Show resource usage for a job
`pestat -u $USER`	Show resource usage for a running job
`seff job_id`	Check CPU/memory efficiency for a job

hprc.tamu.edu/wiki/HPRC:Batch_Translation

Job submission issue: insufficient SUs

Grace:

```
$ sbatch myjob
sbatch: error: (from job_submit) your account's balance is not sufficient to submit your job
               Project Account: 123940134739
               Account Balance: 382.803877
               Requested SUs:   18218.666666667
```

- What to do if you need more SUs
  - Ask your PI to transfer SUs to your account
  - Apply for more SUs (if you are eligible, as a PI or permanent researcher)

hprc.tamu.edu/wiki/HPRC:CommonProblems#Q:_How_do_I_get_more_SUs.3F
hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
hprc.tamu.edu/wiki/HPRC:AMS:UI

CRLF Line Terminators

Windows editors such as Notepad will add hidden Carriage Return Line Feed (CRLF) characters that will cause problems with many applications

```
cd $SCRATCH/batch_examples
file dos_text.txt
```

use `file` command to check

```
dos_text.txt: ASCII English text, with CRLF line terminators
```

```
cat -v dos_text.txt
```

use `cat` command to see CRLF characters

```
dos2unix dos_text.txt
file dos_text.txt
```

use `dos2unix` command to correct

```
dos_text.txt: ASCII English text
```

Need Help?

- First check the FAQ hprc.tamu.edu/wiki/HPRC:CommonProblems
  - Grace User Guide hprc.tamu.edu/wiki/Grace
    - Exercises hprc.tamu.edu/wiki/Grace:Exercises
  - Email your questions to **help@hprc.tamu.edu**. (Managed by a ticketing system)

- Help us, help you -- we need more info
  - Which Cluster
  - UserID/NetID (*UIN is not needed!*)
  - Job id(s) if any
  - Location of your jobfile, input/output files
  - Application used if any
  - Module(s) loaded if any
  - Error messages
  - Steps you have taken, so we can reproduce the problem

HIGH PERFORMANCE RESEARCH COMPUTING

TEXAS A&M UNIVERSITY

Thank you.

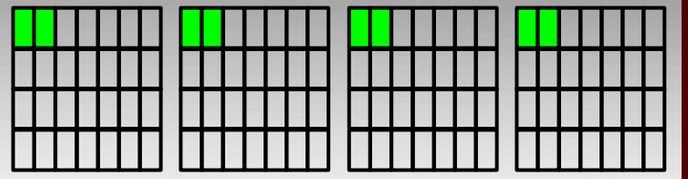
Any questions?

Job Environment Variables

- **Terra:**
  - **$SLURM_JOBID** = job id
  - **$SLURM_SUBMIT_DIR** = directory where job was submitted from
  - **$SCRATCH** = /scratch/user/NetID
  - **$TMPDIR** = /work/job.$SLURM_JOBID
    - $TMPDIR is local to each assigned compute node for the job and is about 850GB

hprc.tamu.edu/wiki/Terra:Batch#Environment_Variables

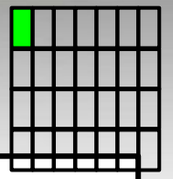
Terra Job File (multi core, multi node)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample3       # Set the job name to "JobExample3"
#SBATCH --time=1-12:00:00            # Set the wall clock limit to 1 Day and 12hr
#SBATCH --ntasks=8                   # Request 8 tasks (cores)
#SBATCH --ntasks-per-node=2          # Request 2 tasks(cores) per node
#SBATCH --mem=2.5G                   # Request 2.5 GB  per node
#SBATCH --output=stdout.%j           # Send stdout and stderr to "stdout.[jobID]"

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456 #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# this intel toolchain is just an example.  recommended toolchain is TBD
module load intel/2017A

# run program with MPI
mpirun my_multicore_multinode_program
```

SUs = 288

Terra Job File (serial GPU)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample4       # Set the job name to "JobExample4"
#SBATCH --time=01:00:00              # Set the wall clock limit to 1hr
#SBATCH --ntasks=1                   # Request 1 task (core)
#SBATCH --mem=2G                     # Request 2GB per node
#SBATCH --output=stdout.%j           # Send stdout and stderr to "stdout.[jobID]"
#SBATCH --gres=gpu:1                 # Request 1 GPU
#SBATCH --partition=gpu              # Request the GPU partition/queue

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456          # Set billing account to 123456  #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# load required module(s)
module load intel/2017A
module load CUDA/9.2.148.1

# run your program
./my_gpu_program
```

SUs = 28

File Transfers and **Terra**

- Simple File Transfers:

  Two-factor authentication required

  - scp: command line (Linux, Mac, Windows cmd)
  - rsync: command line (Linux, Mac, Windows); ***can resume transfer***
  - MobaXterm: GUI (Windows)
  - WinSCP: GUI (Windows)
  - Cyberduck: GUI (Mac)
  - Portal: portal.hprc.tamu.edu (web page; through "Files" menu)
  - rclone: move files to/from cloud storage; command line (HPRC clusters)
- Bulk data transfers:
  - Use fast transfer nodes
    - data transfer processes will not timeout at 60 minutes
    - on **Terra**: `terra-ftn.hprc.tamu.edu`
    - Globus Connect (hprc.tamu.edu/wiki/SW:GlobusConnect)

hprc.tamu.edu/wiki/HPRC:FileTransfers

Job Memory Requests on **Terra**

- Specify memory request based on memory per node:

  #SBATCH --mem=xxxxM # memory per node in MB

  or

  #SBATCH --mem=xG # memory per node in GB

- On 64GB nodes, usable memory is at most 56 GB. The per-process memory limit should not exceed 2000 MB for a 28-core job.
- On 128GB nodes, usable memory is at most 112 GB. The per-process memory limit should not exceed 4000 MB for a 28-core job.

Continued Learning

Intro to HPRC Video Tutorial Series

HPRC's Wiki Page