HIGH PERFORMANCE RESEARCH COMPUTING

HPRC Primer
Grace

February 2, 2024

High Performance
Research Computing
DIVISION OF RESEARCH

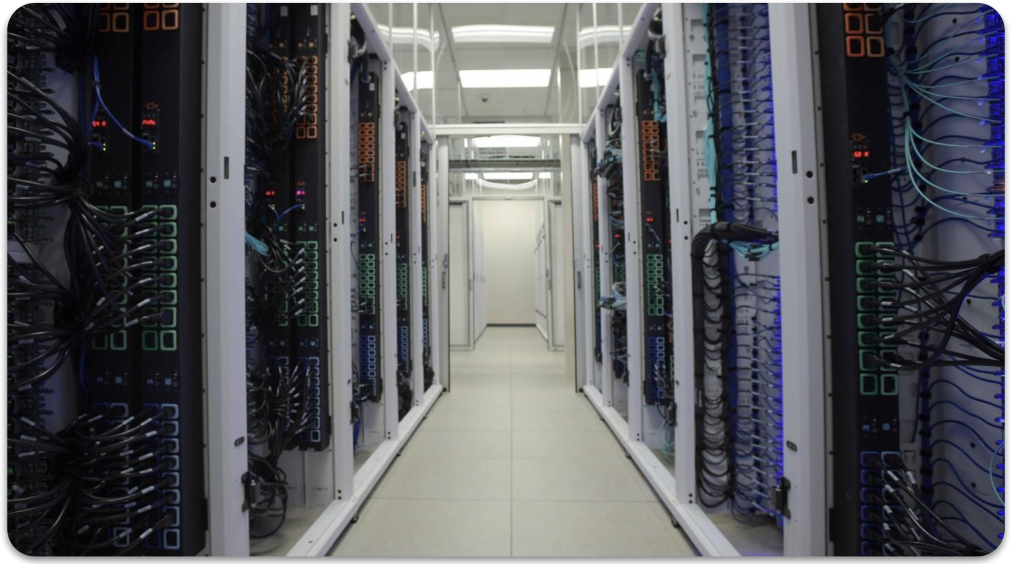
Grace Hardware

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

Resource	Count
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



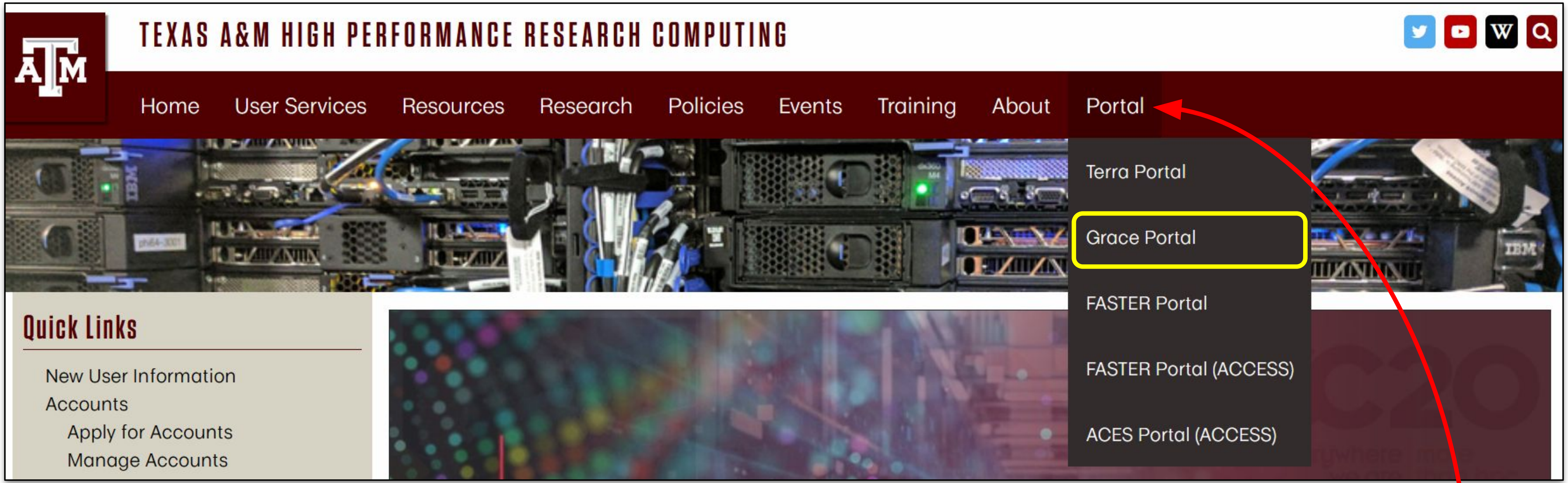
For more information:
https://hprc.tamu.edu/kb/User-Guides/Grace/

Accessing Grace: Setup

- If off-campus:
  Set up and start VPN (Virtual Private Network):
  u.tamu.edu/VPnetwork
- *Two-Factor Authentication* required

- Today we'll access Grace via the online Portal, but you can also use ssh.
- See https://hprc.tamu.edu/kb/User-Guides/Grace/Access/ for more details.

Accessing Grace via the Portal



Access the HPRC portals through most web browsers:

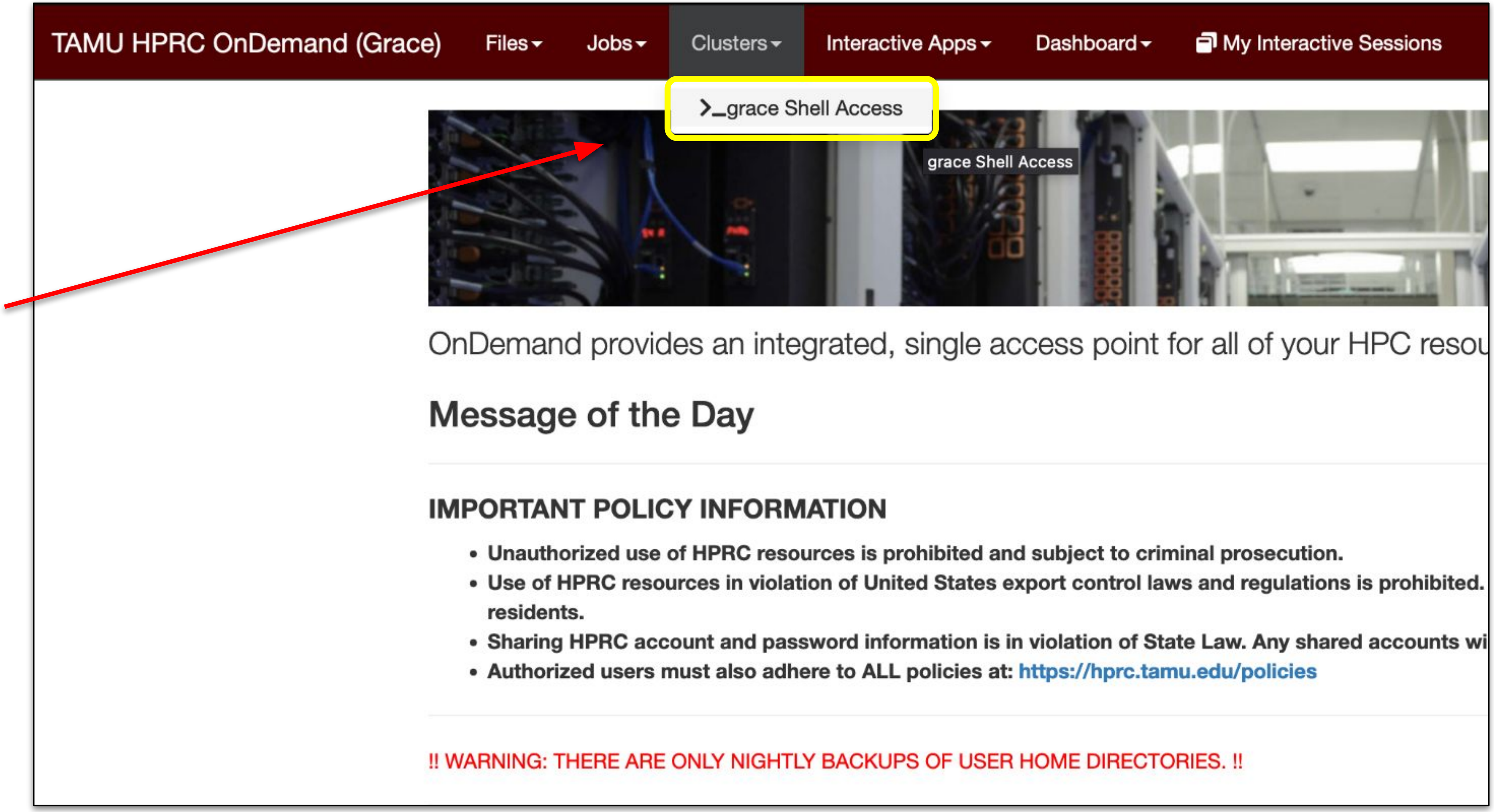
1. Go to portal.hprc.tamu.edu or use the Portal dropdown menu on the HPRC homepage: https://hprc.tamu.edu/
2. Choose **Grace Portal**

https://hprc.tamu.edu/kb/User-Guides/Grace/Access/

Accessing Grace via the Portal

Once in the Portal, select at the top: “Clusters” → “Grace Shell Access”

- *shell* is also called *terminal* or *command line*



OnDemand provides an integrated, single access point for all of your HPC resou

Message of the Day

IMPORTANT POLICY INFORMATION

- **Unauthorized use of HPRC resources is prohibited and subject to criminal prosecution.**
- **Use of HPRC resources in violation of United States export control laws and regulations is prohibited. residents.**
- **Sharing HPRC account and password information is in violation of State Law. Any shared accounts wi**
- **Authorized users must also adhere to ALL policies at: https://hprc.tamu.edu/policies**

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

https://hprc.tamu.edu/kb/User-Guides/Grace/Access/

Hands-On Activity - 2 Minutes

Try to access a *shell** on Grace now, either through portal.hprc.tamu.edu or hprc.tamu.edu

*(also called *terminal* or *command line*)

What message do you see when you login?

Remember Grace has 5 login nodes. Which one does your command prompt say you got?

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.

`$SCRATCH` is shared between the FASTER and Grace clusters.

View file usage and quota limits using the command: **`showquota`**

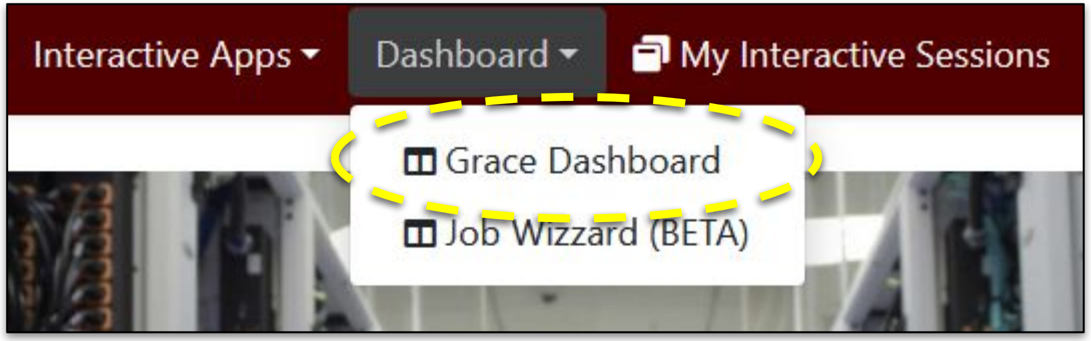
Do NOT share your home or scratch directories.
Request a group directory for sharing files.

https://hprc.tamu.edu/kb/User-Guides/Grace/Filesystems_and_Files/

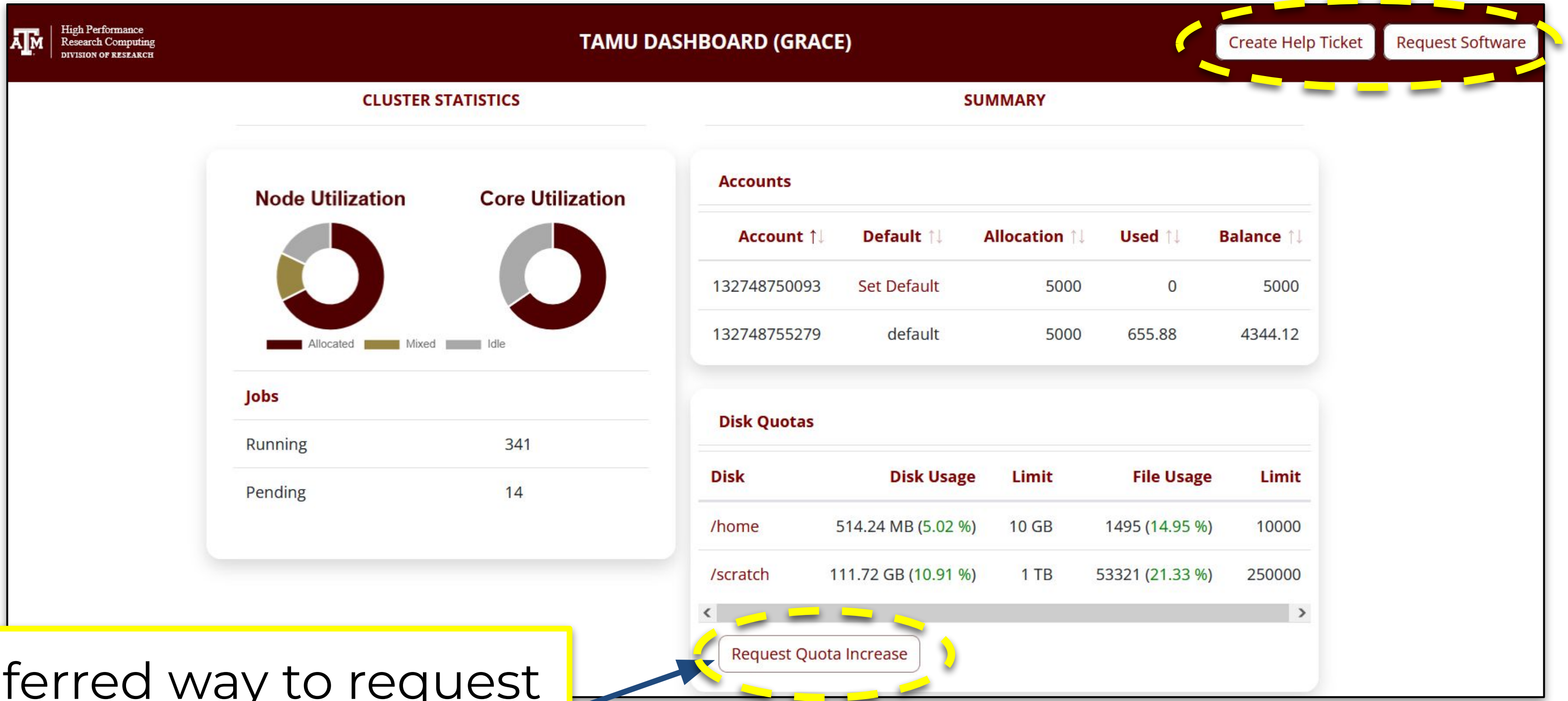
Portal: Grace Dashboard

Easily view Cluster utilization, Storage Quotas, & Allocation Balances

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



Request help or software



TAMU DASHBOARD (GRACE)

CLUSTER STATISTICS

Jobs	
Running	341
Pending	14

SUMMARY

Accounts

Account	Default	Allocation	Used	Balance
132748750093	Set Default	5000	0	5000
132748755279	default	5000	655.88	4344.12

Disk Quotas

Disk	Disk Usage	Limit	File Usage	Limit
/home	514.24 MB (5.02 %)	10 GB	1495 (14.95 %)	10000
/scratch	111.72 GB (10.91 %)	1 TB	53321 (21.33 %)	250000

Preferred way to request *Quota Increases*

Hands-On Activity - 2 Minutes

1. Please try to access dashboard now through the portal.

2. Check your quotas both on the command line and on the dashboard.

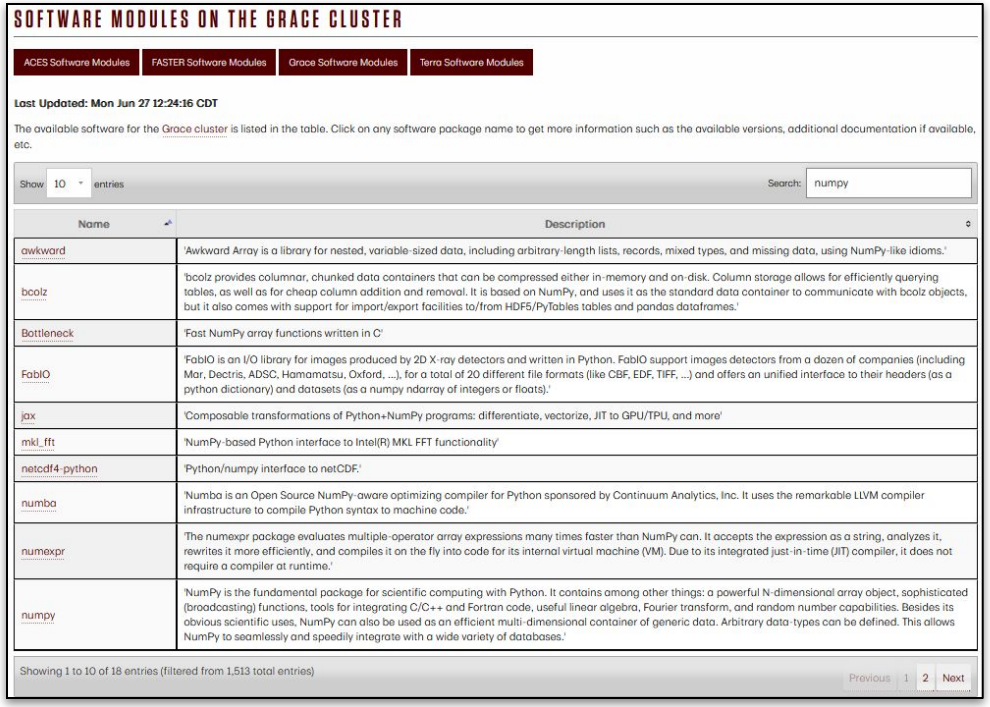
```
showquota
```

Software

- See the Software Knowledge Base page https://hprc.tamu.edu/kb/Software/ for instructions and examples

- Search for software modules on https://hprc.tamu.edu/software/grace/

- License-restricted software
  - Contact help@hprc.tamu.edu



SOFTWARE MODULES ON THE GRACE CLUSTER

Last Updated: Mon Jun 27 12:24:16 CDT

The available software for the Grace cluster is listed in the table. Click on any software package name to get more information such as the available versions, additional documentation if available, etc.

Show 10 entries

Search: numpy

Name	Description
awkward	'Awkward Array is a library for nested, variable-sized data, including arbitrary-length lists, records, mixed types, and missing data, using NumPy-like idioms.'
bcolz	'bcolz provides columnar, chunked data containers that can be compressed either in-memory and on-disk. Column storage allows for efficiently querying tables, as well as for cheap column addition and removal. It is based on NumPy, and uses it as the standard data container to communicate with bcolz objects, but it also comes with support for import/export facilities to/from HDF5/PyTables tables and pandas dataframes.'
Bottleneck	'Fast NumPy array functions written in C'
FabIO	'FabIO is an I/O library for images produced by 2D X-ray detectors and written in Python. FabIO support images detectors from a dozen of companies (including Mar, Dectris, ADSC, Hamamatsu, Oxford, ...), for a total of 20 different file formats (like CBF, EDF, TIFF, ...) and offers an unified interface to their headers (as a python dictionary) and datasets (as a numpy ndarray of integers or floats).'
jax	'Composable transformations of Python+NumPy programs: differentiate, vectorize, JIT to GPU/TPU, and more'
mkl_fft	'NumPy-based Python interface to Intel(R) MKL FFT functionality'
netcdf4-python	'Python/numpy interface to netCDF.'
numba	'Numba is an Open Source NumPy-aware optimizing compiler for Python sponsored by Continuum Analytics, Inc. It uses the remarkable LLVM compiler infrastructure to compile Python syntax to machine code.'
numexpr	'The numexpr package evaluates multiple-operator array expressions many times faster than NumPy can. It accepts the expression as a string, analyzes it, rewrites it more efficiently, and compiles it on the fly into code for its internal virtual machine (VM). Due to its integrated just-in-time (JIT) compiler, it does not require a compiler at runtime.'
numpy	'NumPy is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.'

Showing 1 to 10 of 18 entries (filtered from 1,513 total entries)

- Contact HPRC (can use the dashboard) 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 GCC/10.3.0  OpenMPI/4.1.1
```

← load GCC compiler version 10.3.0 and OpenMPI version 4.1.1

```
module avail
```

← show which software is available to use with the loaded modules

```
module load PyTorch/1.12.1
```

← load PyTorch version 1.12.1

```
module list
```

← see what software you've loaded already

Software: Modules and Toolchains

- Toolchains are what we call groups of compilers & libraries
- There’s a variety of toolchains available on the clusters:
  - intel/2022a
  - iomkl/2020a
  - foss/2022a
  - GCCcore/11.3.0

(more than just these versions)

- Module management:

```
module spider
module purge
```

← search for modules and their dependencies
← removes all loaded modules

Hands-On Activity - 5 Minutes

Remember: `module load` `module purge`

1. Please search for and load the following module:

```
OpenMPI/4.1.4
```

(Tip) Type this to show which compiler needs to be loaded:

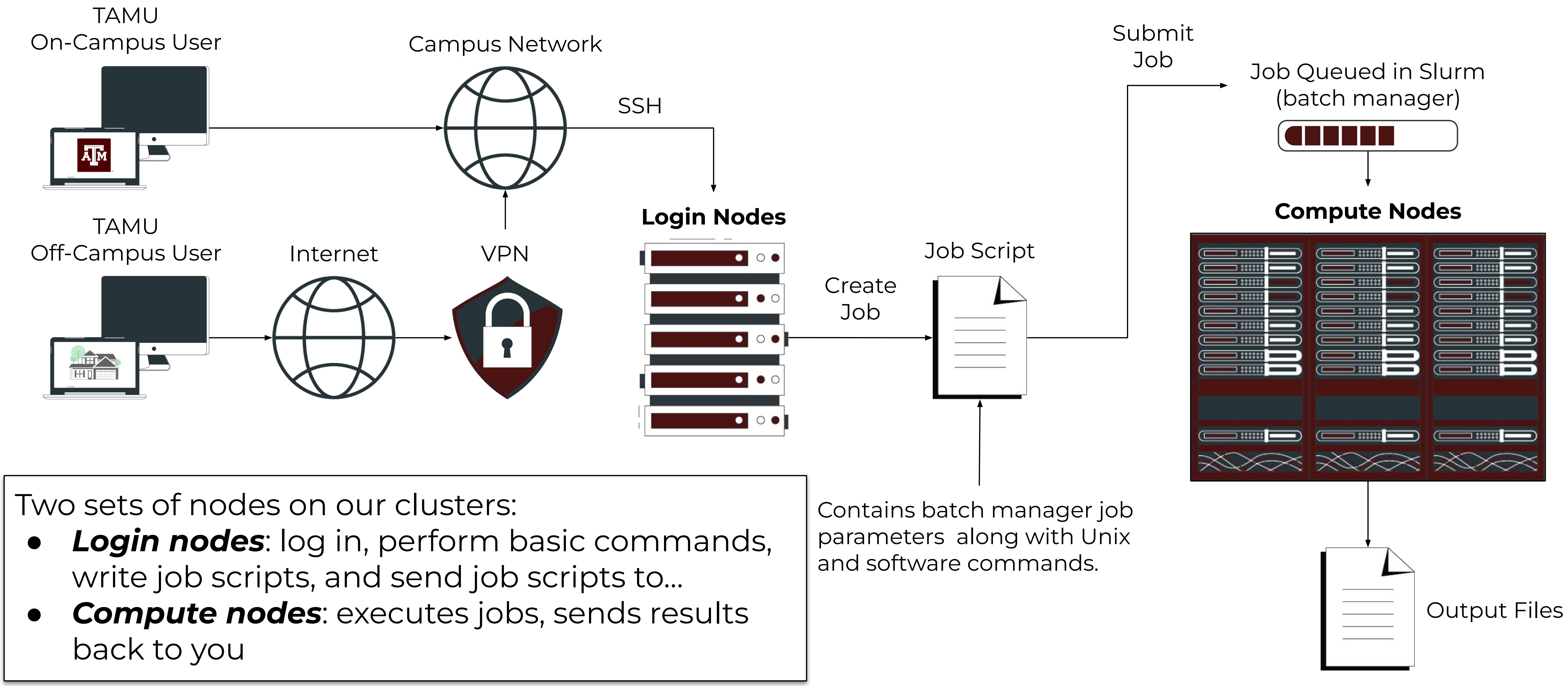
```
module spider OpenMPI/4.1.4
```

(Tip) And check that it’s been loaded with:

```
module list
```

2. Next remove (unload) all your current modules.

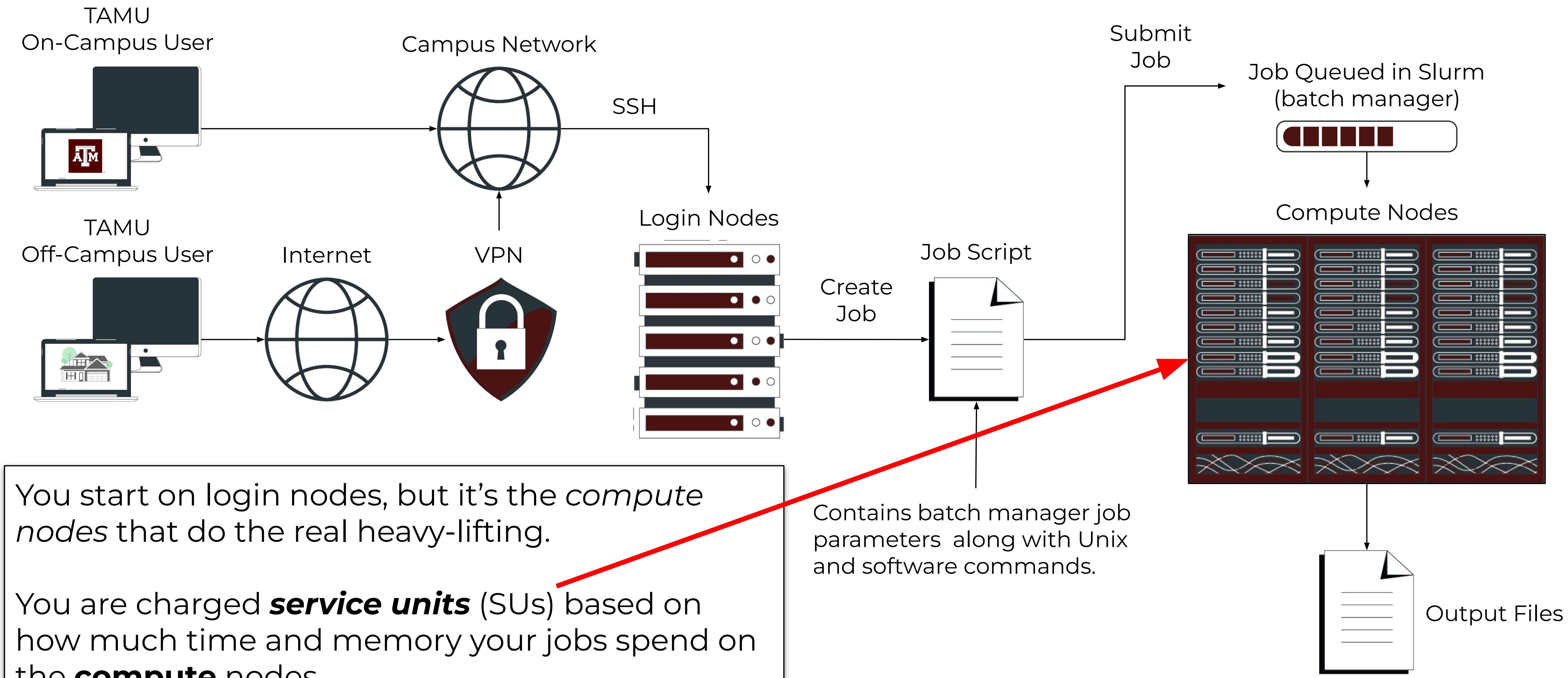
Computing on HPRC Clusters



Two sets of nodes on our clusters:

- ***Login nodes***: log in, perform basic commands, write job scripts, and send job scripts to...
- ***Compute nodes***: executes jobs, sends results back to you

Batch Jobs on HPRC Clusters

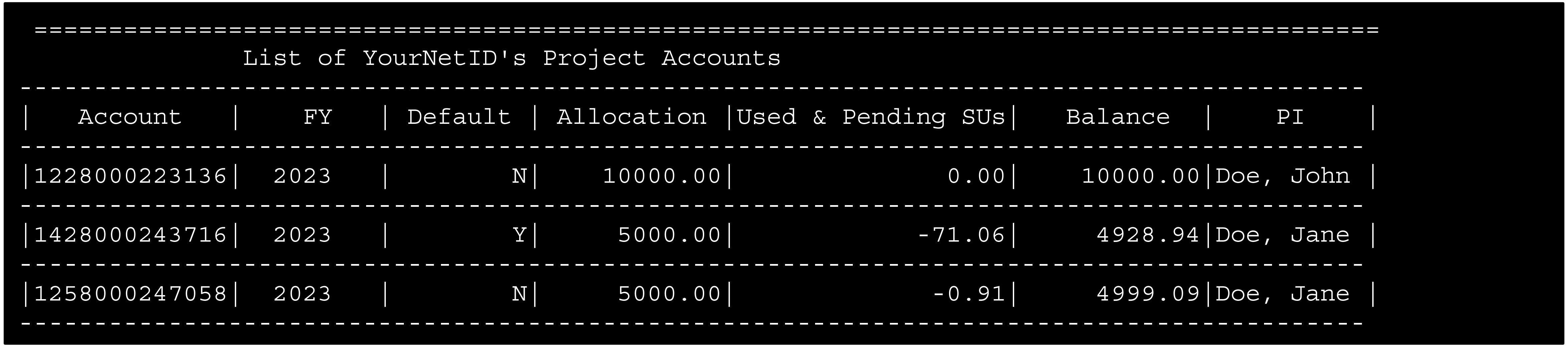


You start on login nodes, but it's the *compute nodes* that do the real heavy-lifting.

You are charged ***service units*** (SUs) based on how much time and memory your jobs spend on the **compute** nodes.

Check your Service Unit (SU) Balance

- 1 SU = 1 core-hour (GPUs are more expensive per-hour!)
- List the SU Balance of your Account(s) with: `myproject`



```
 ===============================================================================
             List of YourNetID's Project Accounts
-------------------------------------------------------------------------------
|   Account    |    FY    | Default | Allocation |Used & Pending SUs|   Balance   |    PI    |
-------------------------------------------------------------------------------
|1228000223136|  2023    |        N|    10000.00|                0.00|    10000.00|Doe, John |
-------------------------------------------------------------------------------
|1428000243716|  2023    |        Y|     5000.00|              -71.06|     4928.94|Doe, Jane |
-------------------------------------------------------------------------------
|1258000247058|  2023    |        N|     5000.00|               -0.91|     4999.09|Doe, Jane |
-------------------------------------------------------------------------------
```

- Run **`myproject -d`** `<Account#>` to change default project account

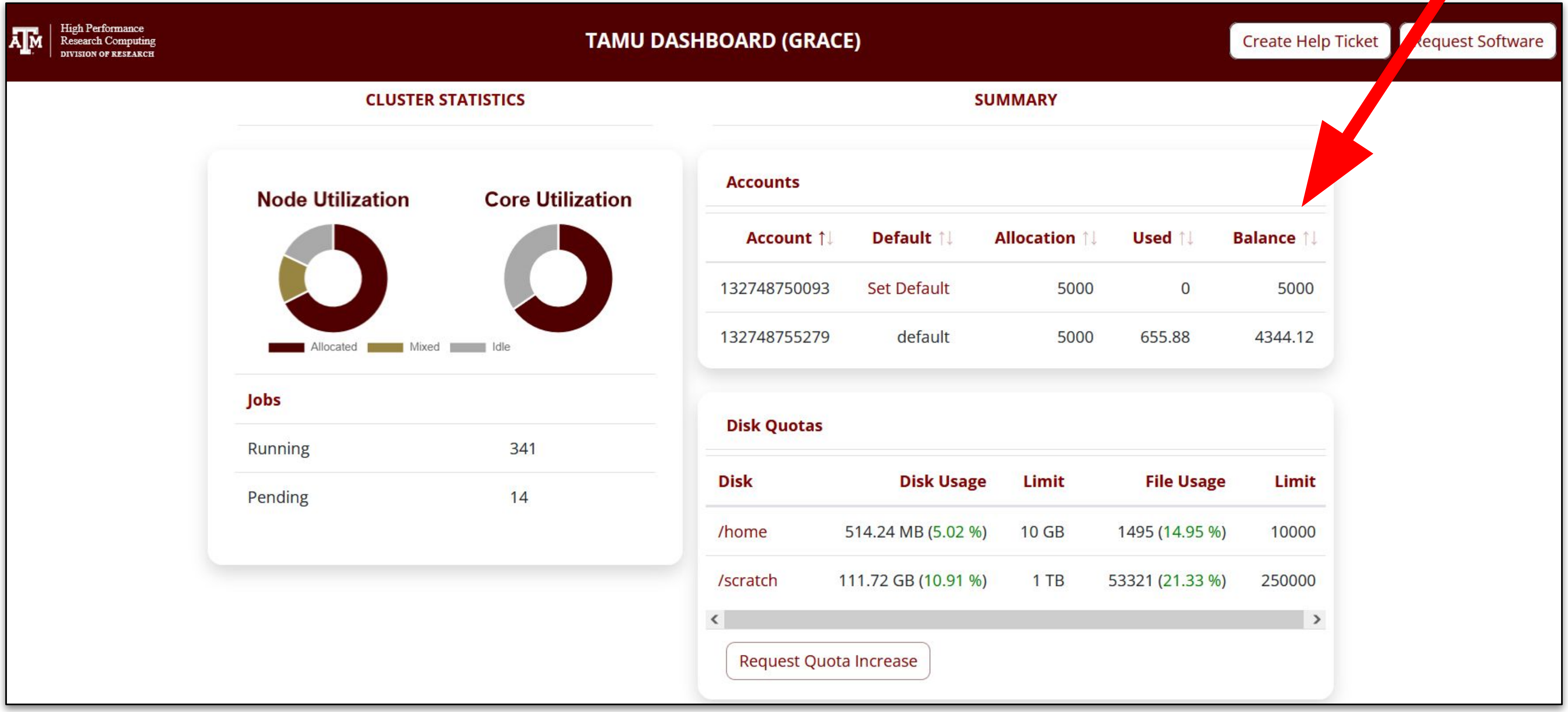
  (replace <Account#> with your number!)

- Run **`myproject -h`** to see more options

  https://hprc.tamu.edu/kb/User-Guides/AMS/Service_Unit/
  https://hprc.tamu.edu/kb/User-Guides/AMS/UI/

SUs in the Grace Dashboard

The same functionality can be found in the Dashboard:



TAMU DASHBOARD (GRACE)

CLUSTER STATISTICS

Jobs	
Running	341
Pending	14

SUMMARY

Accounts

Account	Default	Allocation	Used	Balance
132748750093	Set Default	5000	0	5000
132748755279	default	5000	655.88	4344.12

Disk Quotas

Disk	Disk Usage	Limit	File Usage	Limit
/home	514.24 MB (5.02 %)	10 GB	1495 (14.95 %)	10000
/scratch	111.72 GB (10.91 %)	1 TB	53321 (21.33 %)	250000

Hands-On Activity - 2 Minutes

1. Use **`myproject`** to check the SU balance of your accounts.

2. Use the dashboard the check the same information.

Sample Job Script Structure

```
#!/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 purge
module load GCCcore/11.3.0 Python/3.10.4

# Run your program
python my_program.py
```

These *parameters* describe your job to the Slurm job scheduler.
The lines starting with #SBATCH are NOT comments!
See the Knowledge Base for more info

← Account number to be charged

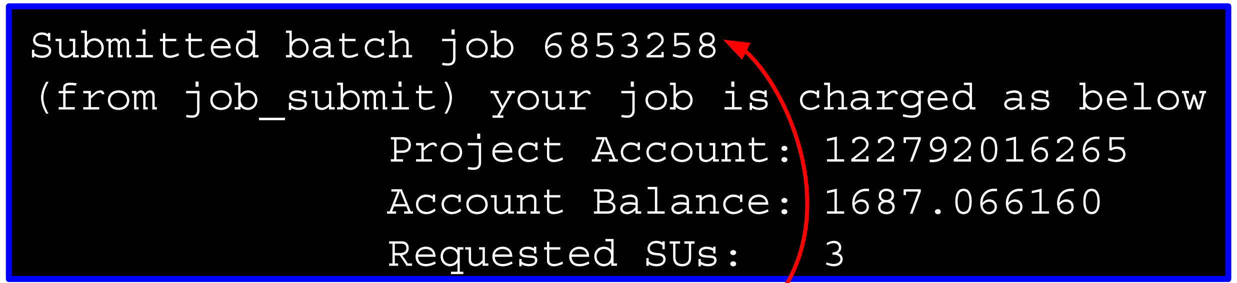
Whatever commands or scripts you want to run. Here, we set up the modules we need for our environment and run a python program.

(We will practice with job files in a few slides)!

Submit a Job and Check Job Status

Submit job

```
sbatch example01.job
```

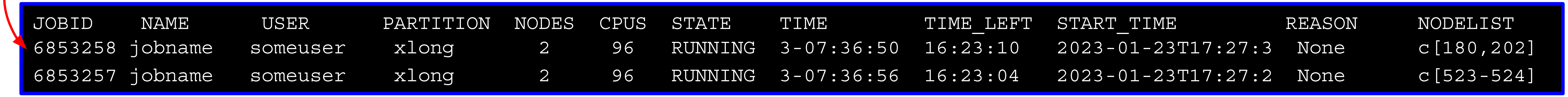


```
Submitted batch job 6853258
(from job_submit) your job is charged as below
              Project Account: 122792016265
              Account Balance: 1687.066160
              Requested SUs:    3
```

matching JOBID

Check status

```
squeue -u netID
```



```
JOBID     NAME      USER       PARTITION  NODES  CPUS  STATE     TIME        TIME_LEFT  START_TIME          REASON     NODELIST
6853258 jobname   someuser    xlong        2     96   RUNNING  3-07:36:50  16:23:10   2023-01-23T17:27:3  None      c[180,202]
6853257 jobname   someuser    xlong        2     96   RUNNING  3-07:36:56  16:23:04   2023-01-23T17:27:2  None      c[523-524]
```

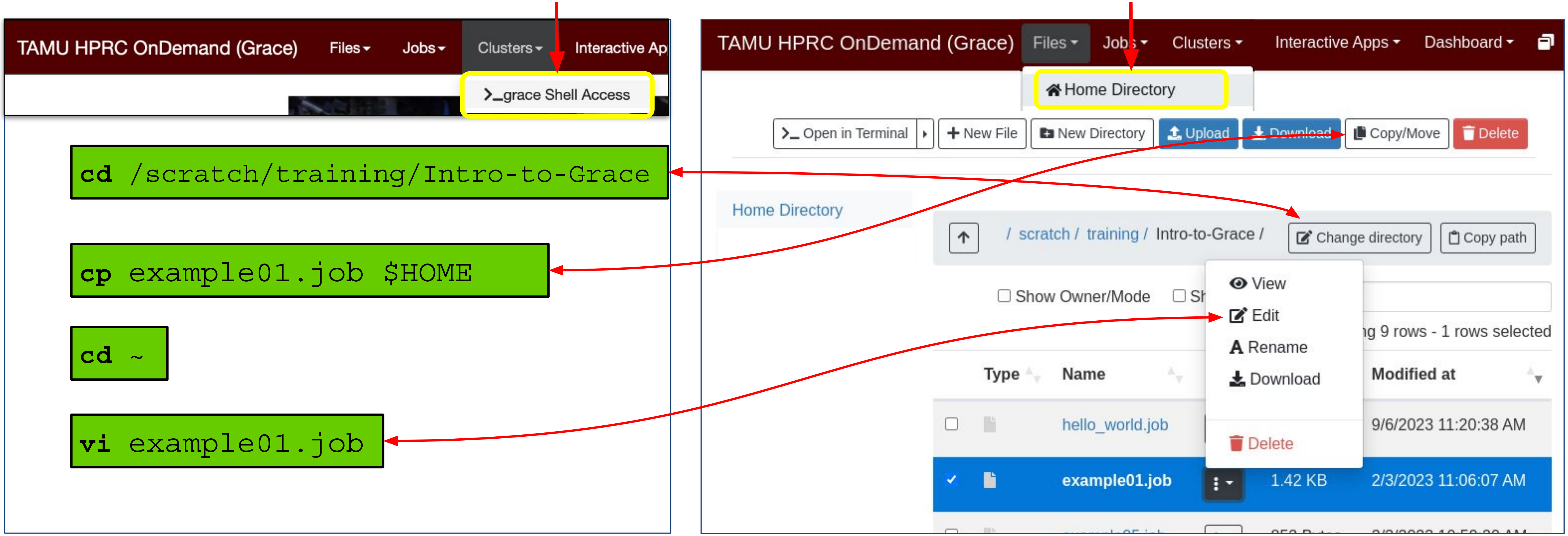
Hands-On Activity

1. Navigate to **`/scratch/training/Intro-to-Grace`**
2. Copy **`hello_world.job`** to your home directory
3. Return to your home directory and submit the job file using **`sbatch`**.
4. Check that the job is running in a Slurm queue with **`squeue`**.
5. When your job completes, check the contents of the output file.

Hands-On Activity

Linux command line tools

Portal file navigator



```
cd /scratch/training/Intro-to-Grace
```

```
cp example01.job $HOME
```

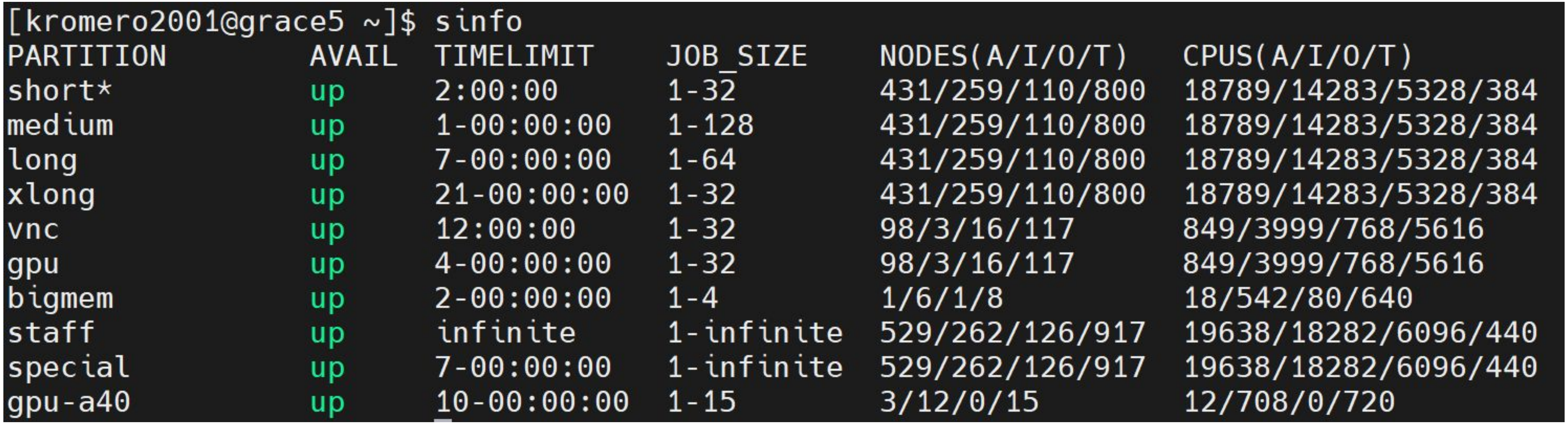
```
cd ~
```

```
vi example01.job
```

Then submit and check on the command line with `sbatch` and `squeue`.

Batch Queues

- Job submissions are auto-assigned to batch queues based on the resources requested (e.g. number of cores/nodes and walltime limit)
- Use **`sinfo`** to check their status:



```
[kromero2001@grace5 ~]$ sinfo
PARTITION        AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)   CPUS(A/I/O/T)
short*           up     2:00:00      1-32        431/259/110/800  18789/14283/5328/384
medium           up     1-00:00:00   1-128       431/259/110/800  18789/14283/5328/384
long             up     7-00:00:00   1-64        431/259/110/800  18789/14283/5328/384
xlong            up     21-00:00:00  1-32        431/259/110/800  18789/14283/5328/384
vnc              up     12:00:00     1-32        98/3/16/117      849/3999/768/5616
gpu              up     4-00:00:00   1-32        98/3/16/117      849/3999/768/5616
bigmem           up     2-00:00:00   1-4         1/6/1/8          18/542/80/640
staff            up     infinite     1-infinite  529/262/126/917  19638/18282/6096/440
special          up     7-00:00:00   1-infinite  529/262/126/917  19638/18282/6096/440
gpu-a40          up     10-00:00:00  1-15        3/12/0/15        12/708/0/720
```

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**

https://hprc.tamu.edu/kb/User-Guides/Grace/Batch/#batch-queues

Job Submission and Tracking

Slurm queue command	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

https://hprc.tamu.edu/kb/Helpful-Pages/Batch-Translation/

Need Help?

First check the FAQ

- Grace User Guide
- Email your questions to help@hprc.tamu.edu

Help us help you -- when you contact us, tell us:

- Which Cluster you're using
- Your username
- 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

Continued Learning

Intro to HPRC Video Tutorial Series

HPRC’s Knowledge Base

High Performance
Research Computing
DIVISION OF RESEARCH

Thank you.

Any questions?