Intro to the Sol Supercomputer

Gil Speyer

Director

KE Computational Research Accelerator

ASU Research Computing
Arizona State University

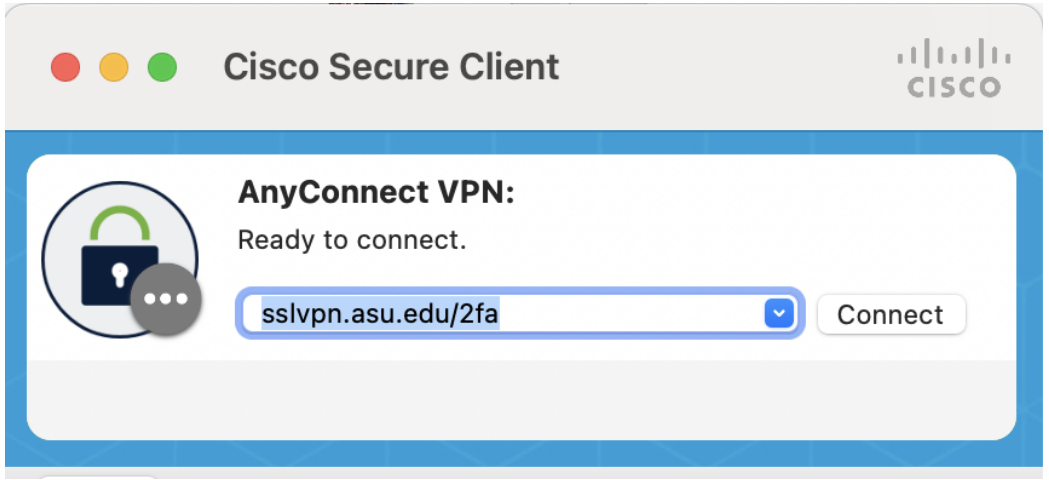
June 23, 2025

ASU Knowledge Enterprise
Arizona State University

Core Research Facilities

Before We Start

- For uninterrupted access, the Cisco AnyConnect Client VPN is required:
  - Go to sslvpn.asu.edu and install the Cisco AnyConnect Client VPN
  - Faculty, staff, and students will also need Duo Two-Factor Authentication
  - In the connect window, enter “sslvpn.asu.edu/2fa”
  - Sign in with ASURITE, ASU password, and DUO two-factor authentication method
    - If prompted, first password is your ASU password and second password is your DUO authentication method (push, phone, or sms)



What is Sol?

- Sol is a supercomputer, **FREE for faculty, staff, students, and affiliates**
- This supercomputer uses hundreds of compute servers, also called nodes, and their collective cores to help users advance their research
- This advancement can happen in several ways:
  - Free the researcher's local machine by running the program on a separate server
  - Allow for parallelization or for multiple operations, or "jobs", to run in parallel
  - Access to graphics processing units (GPUs)
  - Access to high memory computation and storage
  - Overall greater and more refined research through supercomputing

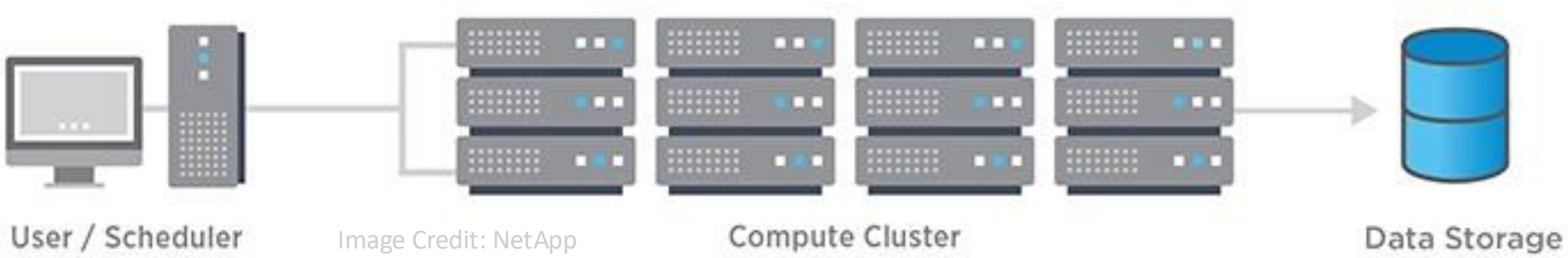


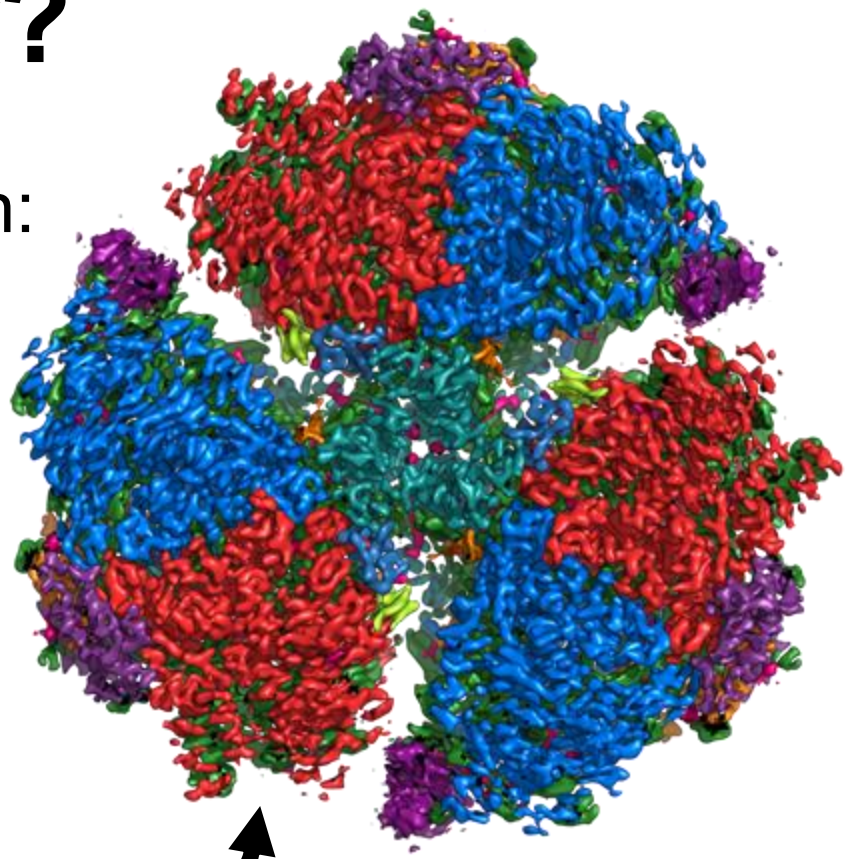
Image Credit: NetApp

What Is a Supercomputer?

ASU's supercomputers advance research with:

- Increased Compute Capacity
- More Processing Speed
- Modeling and Simulation
- Data and Artificial Intelligence (AI)
- Visualization
- Access to project and long-term data storage

Overall, supercomputers are powerful tools that help advance research

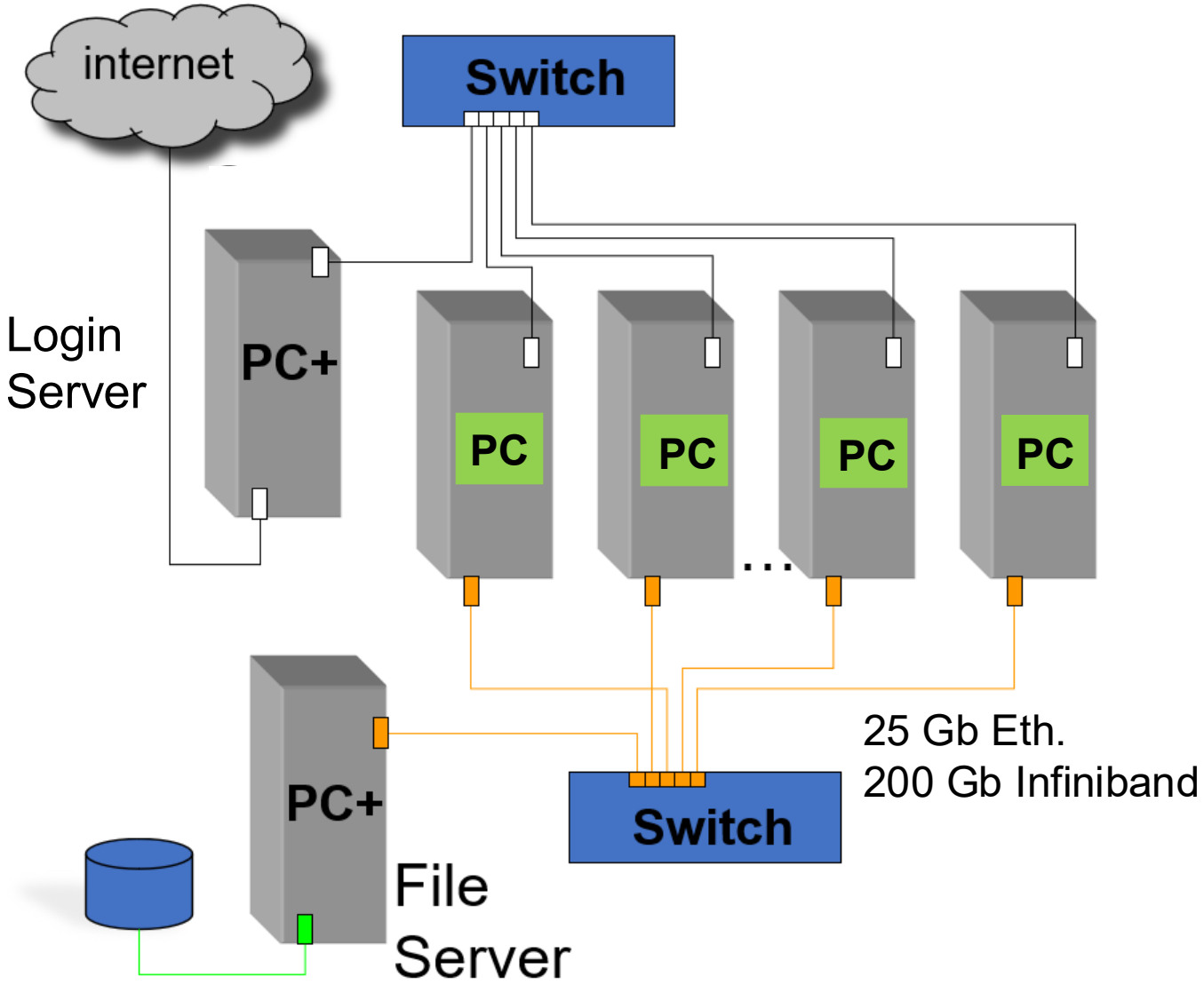


A visualization of a protein found in chlorophyll. Produced by Yuval Mazor, School of Molecular Sciences, on the Agave supercomputer

Helpful Terms

- **Node:** A single machine in a supercomputer. This will be either a physical machine or a virtual machine.
- **Login Node**: A node intended as a launching point to compute nodes. Login nodes have minimal resources and should not be used for any application that consumes a lot of CPU or memory. This is also known as a “head node”.
- **Compute Node**: Nodes intended for heavy compute. This is where all heavy processing should be done.
- **Job**: Work assigned to be done on a compute node. Any time a compute node is assigned a job is created.
- **Memory (RAM):** Short for “Random-Access Memory“. This is used for the amount of memory that each calculation or computation requires in order to execute and complete successfully. The term “memory“ is not used for disk space. This is another main component that defines a node.
- **CPU:** Short for “Central Processing Unit”, also called a core. This is one of the main components that defines a computing device, such as a node.
- **GPU:** Short for “Graphic Processing Unit”. This is a specialized piece of hardware that can enable and accelerate certain computational research.
- **Scheduler**: The application on our end that manages and assigns (allocates) compute resources for jobs. The scheduler used on the ASU Supercomputers is called Slurm.
- **SBATCH Script**: This is a script submitted through the sbatch (Slurm batch) process. It allows users to submit non-interactive jobs to the supercomputer.

How Does a Supercomputer Work?



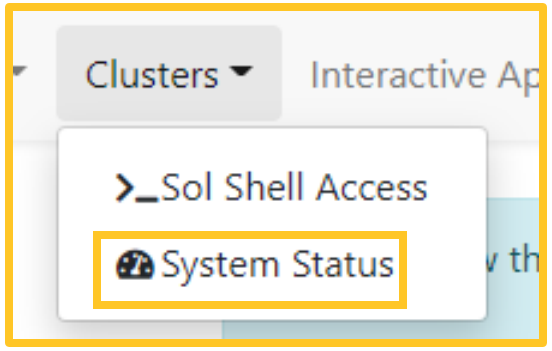
(Supercomputer)

Sol System Information

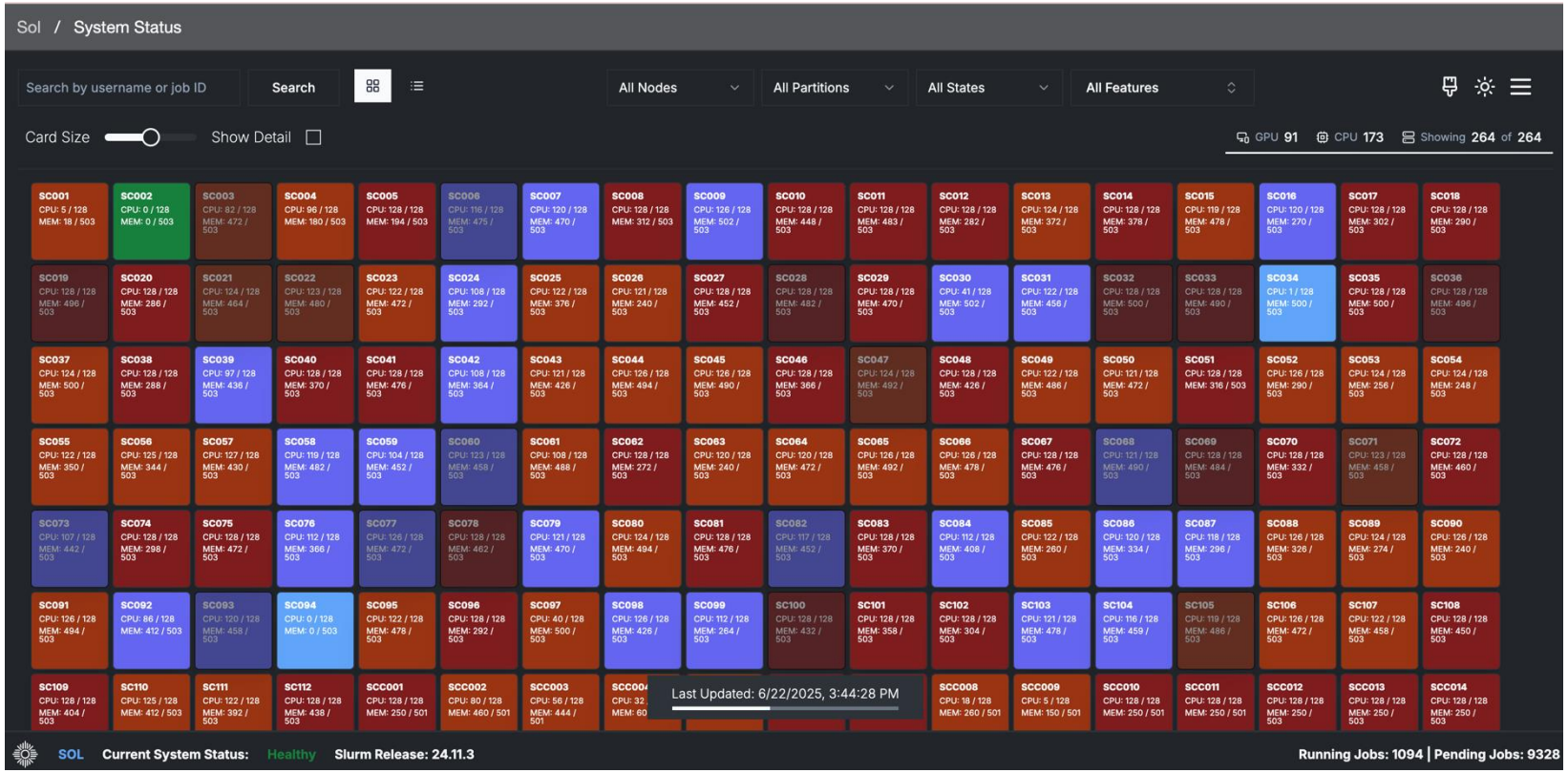
- Sol has over 24,132 *physical* cores (no hyperthreading)
  - AMD Zen3 CPUs
  - >2.55 quadrillion FLOPS
  - >98.5TB aggregate RAM
  - >4PB scratch and >2PB home data storage
  - Mellanox Infiniband – 200Gb/s HDR
  - Located in the Iron Mountain Data Center
- Over 337 GPUs (including 15 24GB A30s and 224 80GB A100s!)
- 5 high memory nodes (2TB each)
- Two FPGAs: Xilinx (AMD-based) and BittWare (Intel-based)
- Sol is a true **supercomputer** architecture
  - Nodes have 48-128 *physical* cores and 512+ GB of RAM.
  - Programs needing more resources **must** use node-parallel programming
  - Normal, single processor applications **do not necessarily go faster** on Sol

System Status

- Navigate to https://sol.asu.edu
- Select “Clusters” then “System Status”



- System Status may be reached by https://links.asu.edu/sol-status

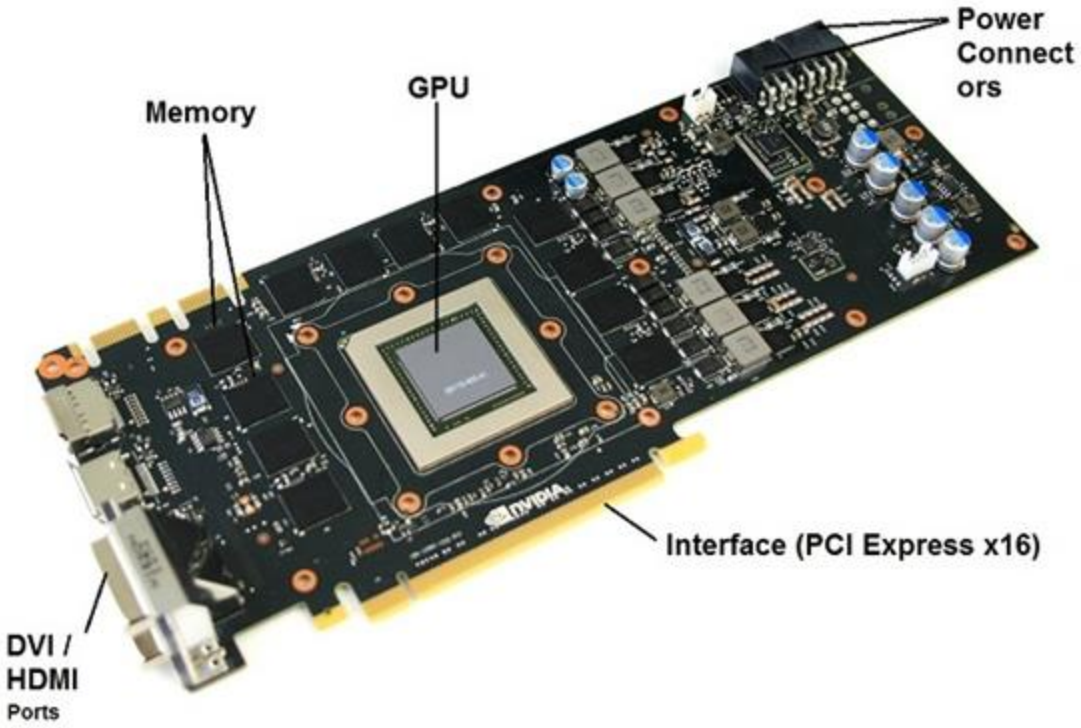


Graphics Processing Units (GPUs)

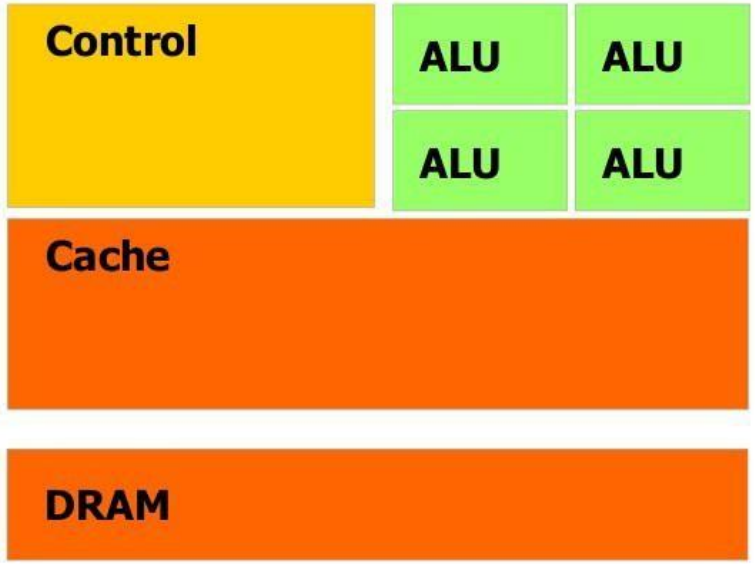
What Is a GPU?

- A **G**raphics **P**rocessing **U**nit is a specialized piece of hardware designed to rapidly manipulate and alter memory

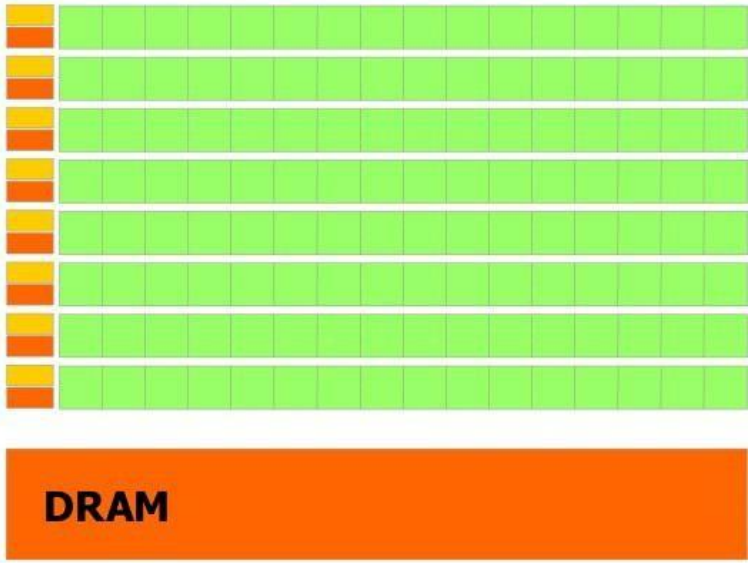
- Its parallel architecture has made GPUs ideal for massive parallel processing



Coprocessor Configuration



CPU

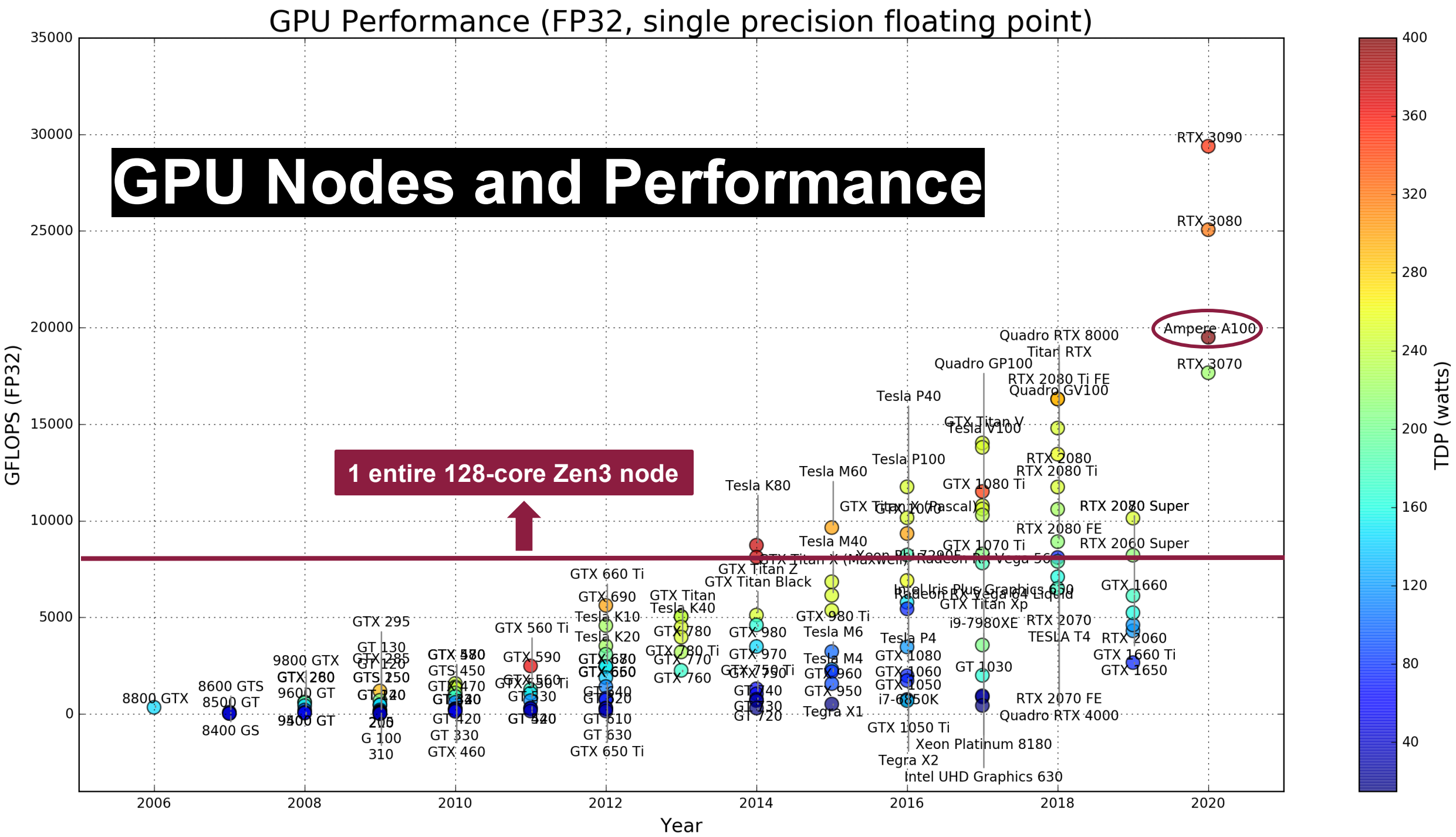


GPU

- *CPU* – **C**entral **P**rocessing **U**nit
- Low compute density
- Fast, dense operations
- Complex control logic

- *GPU* – **G**raphics **P**rocessing **U**nit
- High compute density
- Slow, shallow operations
- Simple control logic

GPU Nodes and Performance



Popular Software GPU Supported

MATLAB	silvaco	Parabricks	Trimmomatic
R	**GCC**	AlphaFold2	Samtools
Anaconda	**openmpi**	HDF5	rosetta
Python	**intel-mpi**	netcdf	vasp
SAS	PGI	PROJ	gaussian
Stata	**OpenACC**	GDAL	gromacs
perl	mvapich2	emboss	lammps
Java	singularity	SeqKit	blast_plus
cuDNN	**cuda**	DWGSIM	vmd
tensorflow	fftw	GATK	namd
pytorch	GSL	QUAST	vcftools
rapids	ANSYS	sratoolkit	mafft

Using the Web Portal

- Go to https://sol.asu.edu
- Login with ASURITE and password
- From here users can:
  - Manage file systems 

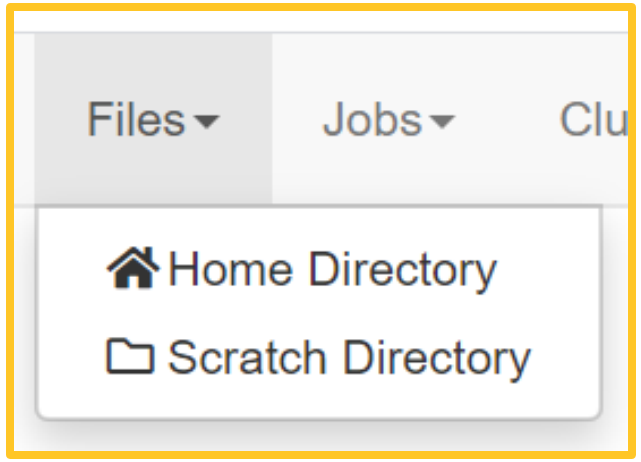
  - Create and monitor jobs 

  - Access the shell 

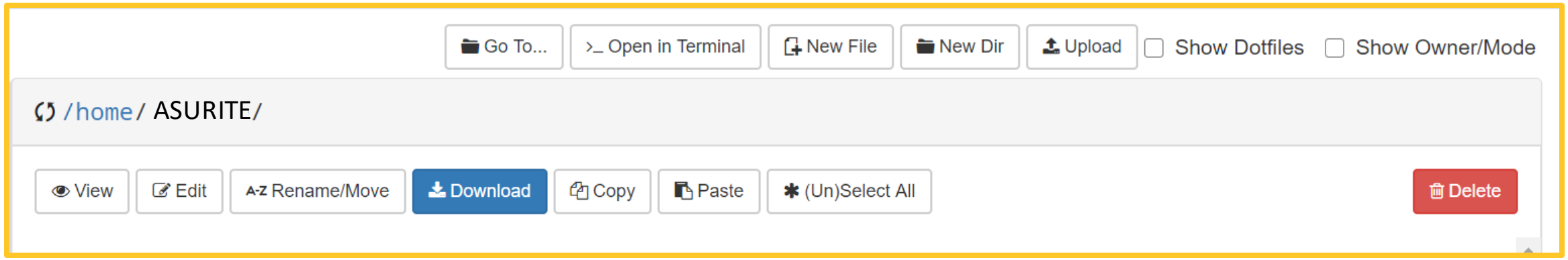
  - Access interactive apps 

  - View and manage interactive sessions 
  - View all apps 
  - Request help 

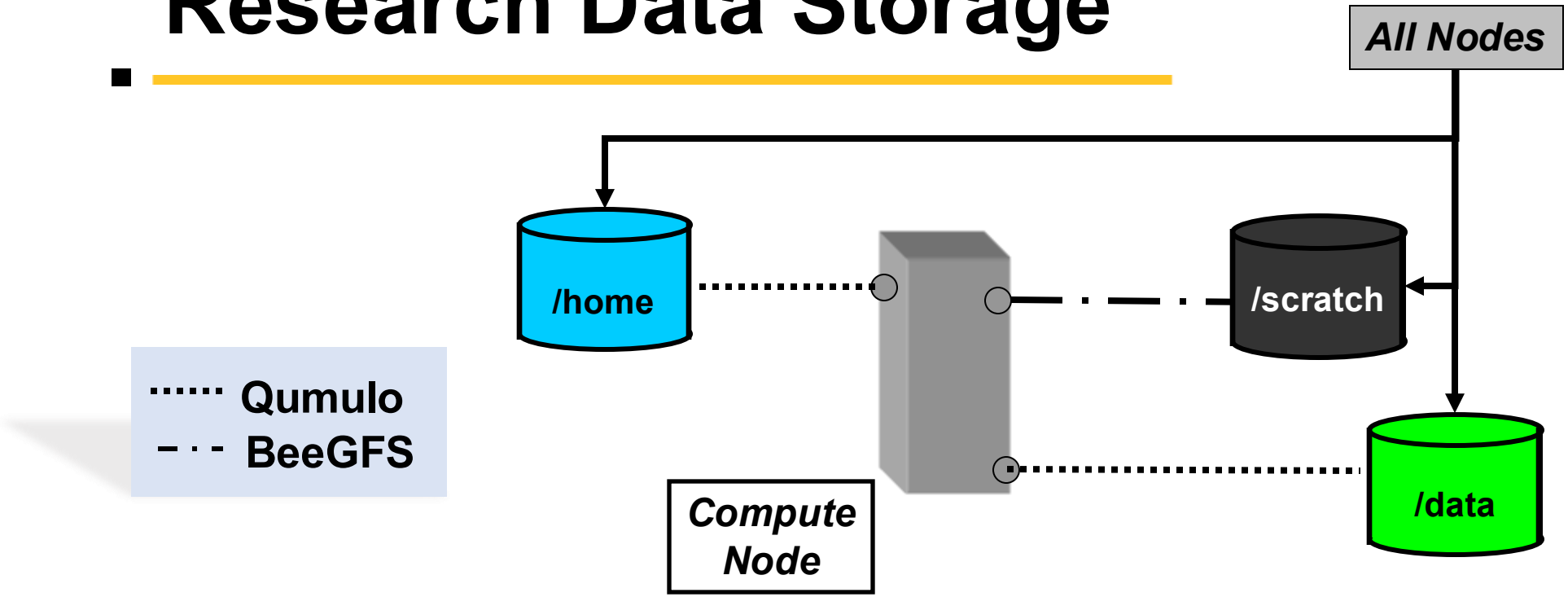
Managing Files



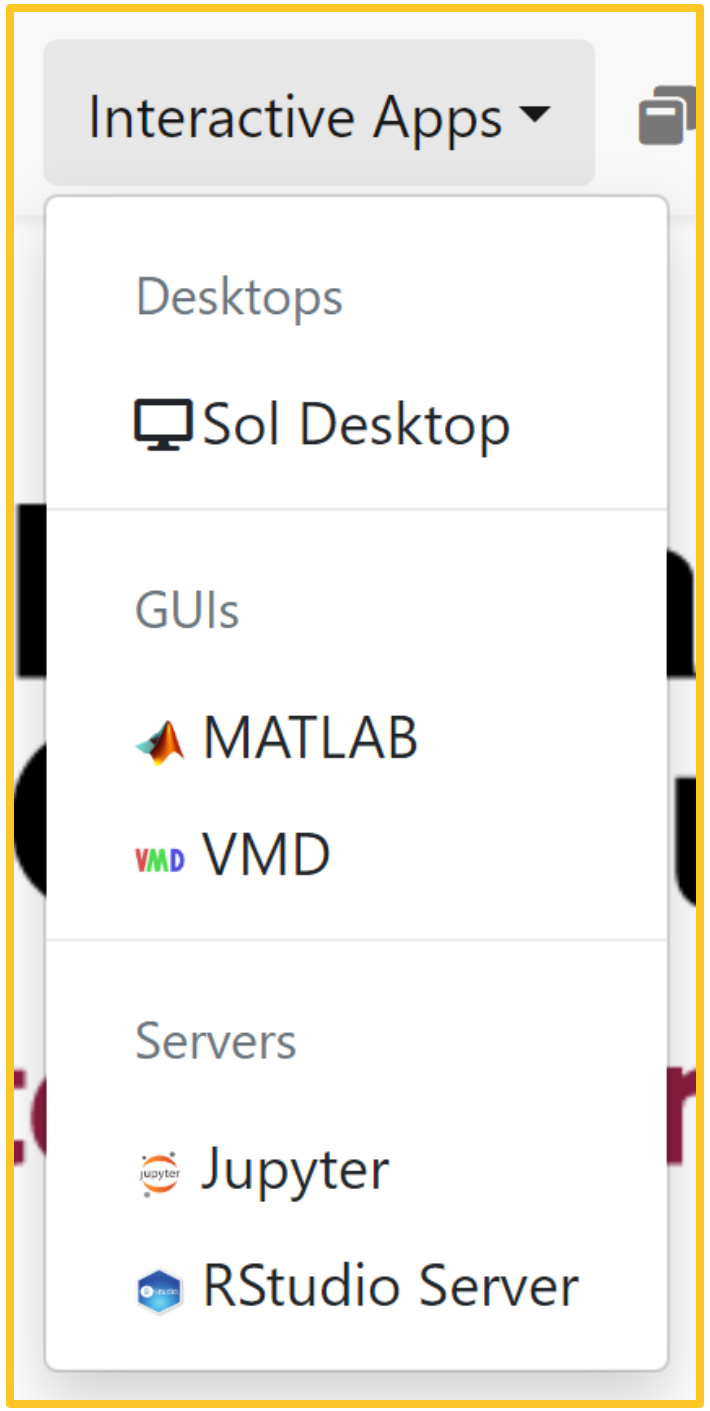
- Manage files between local machine and Sol
- Action buttons, drag and drop for upload
- See more under [Available File Systems]



Research Data Storage



Mount point	User Quota	Lifetime
/home	100 GB	Affiliation
/scratch	100 TB	90 days
/data	Purchased	Project



Interactive Apps & Virtual Desktops

- Software requiring **G**raphical **U**ser **I**nterfaces (**GUI**s) should use
  - Sol Virtual Desktop
  - Common GUIs are set up for convenience (MATLAB and VMD).

- Even if not using Python or R, Jupyter and RStudio Server provide modern interactive interfaces to the supercomputer.

Creating Interactive Sessions

- All Interactive Apps will allow you to specify any resources you wish to use for your session

Sol Desktop

This app will launch an interactive desktop on one or more compute nodes. You will have full access to the resources these nodes provide. This is analogous to an interactive batch job.

Partition

htc

- Specify a partition to run on, e.g. `htc`
- Default is `htc`

QOS

public

- Default is `public`

Cores

1

Session Wall Time

15

- Set an upper limit on the run time of the interactive desktop session.
- Wall time, not compute time.
- **time formats**: "minutes", "minutes:seconds", "hours:minutes:seconds", "days-hours", "days-hours:minutes", "days-hours:minutes:seconds"

☑ email when job starts

GPU Resources

Request GPU resources here if needed, e.g. `gpu:a30:1` or `gpu:1`.

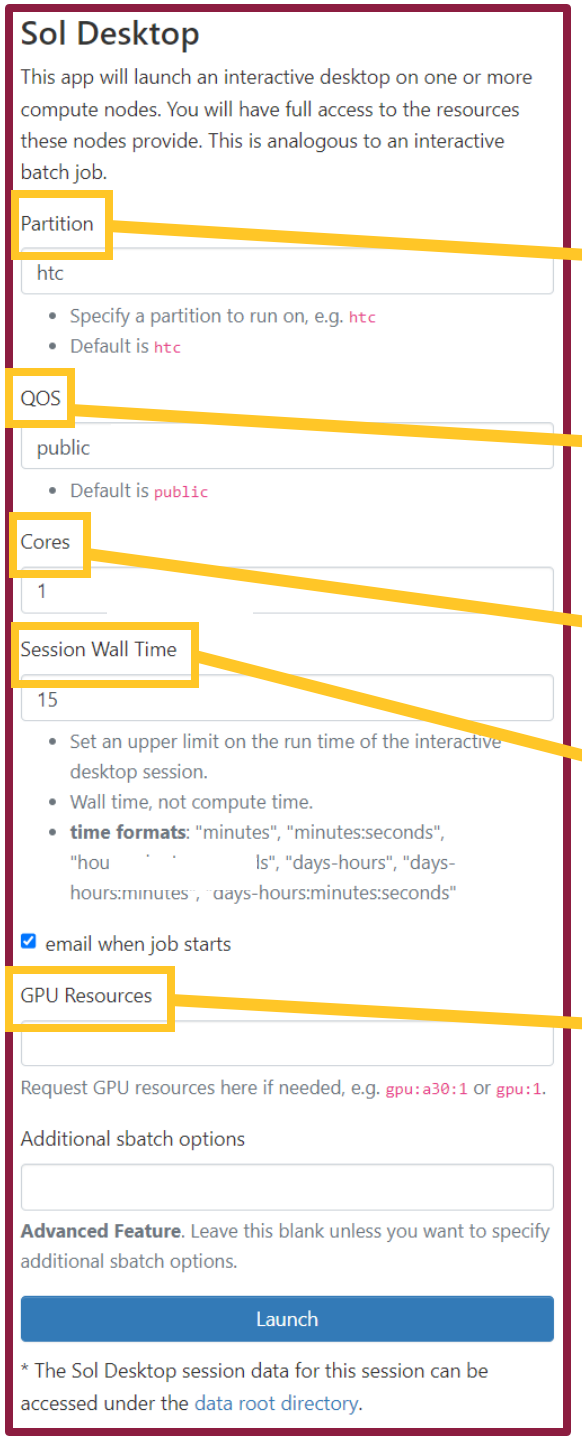
Additional sbatch options

Advanced Feature. Leave this blank unless you want to specify additional sbatch options.

Launch

* The Sol Desktop session data for this session can be accessed under the data root directory.

Understanding Resources



Partition – the node group you wish to use

QOS – the **q**uality **o**f **s**ervice required by that partition

Cores – the number of cores you expect to need

Session Wall Time – the amount of time you need

GPU Resources – the model and number of desired GPUs

Sol Partition Information

Partition	QOS	Maximum Wall Time
general	debug public private*	15 minutes 7 days 7 days*
htc	debug public private*	15 minutes 4 hours 7 days*

Request reservation with "--reservation=hackathon2025"

**Note*: Maximum wall time does not exclude risk from preemption for these * partitions

To request access for the two-week QOS, email rtshelp@asu.edu

Additional information can be found on the "Requesting Resources on Sol" page

Jupyter version: 63938ce

This app will launch a Jupyter Lab server on a compute node. From within the Jupyter interface you may find a list of preconfigured python kernels. To make your own kernels accessible through the interface, follow these instructions or request an install by emailing: rtshelp@asu.edu

Partition

general

- 2022-05-26: partitions are `general` and `htc`.
- **DOCS HERE ARE A WORK IN PROGRESS**: sol partition and QOS info

QOS

public

- 2022-05-26: QOS are `public`, `debug`, and `private`.
- **DOCS HERE ARE A WORK IN PROGRESS**: sol partition and QOS info

Number of Cores

1

Number of cores on node. Leave blank if requesting a full node's memory and cores (will set `--exclusive --mem=0`).

Jupyter Wall Time

2:00:00

- Set an upper limit on the run time of `jupyter`.
- **time formats**: "minutes", "minutes:seconds", "hours:minutes:seconds", "days-hours", "days-hours:minutes", "days-hours:minutes:seconds"

GPU Resources

gpu:a100:1

Request GPU resources here if needed, e.g. `gpu:a30:1`, `gpu:a100:1`, `gpu:2g.20gb:1`, `gpu:1g.20gb:1`, or `gpu:1`.

Jupyter Wall Time

0-1

Enter the time limit for your job. **Format:** "minutes", "minutes:seconds", "hours:minutes:seconds", "days-hours", "days-hours:minutes", "days-hours:minutes:seconds"

☐ I would like to receive an email when the session starts

Jupyter lab version

latest

Advanced Feature. Choose Jupyter version, default is `latest`,

Additional modules

Advanced Feature. These are additional modules to load in the background. Only include the module name. Separate modules with a space. For a list of available modules, see the "Available Modules" page under the "System" navbar item at the top of this page.

In the background this job will load an appropriate python module, **so do not request that here**.

Additional sbatch options

--reservation=hackathon2025

Advanced Feature. Leave this blank unless you want to specify additional sbatch options.

Example: GPU-Enabled Jupyter

Example: GPU-Enabled Jupyter

Jupyter (1449307) 1 node | 1 core | Running

Host: >_g001

Delete

Created at: 2022-11-10 09:37:15 MST

Time Remaining: 1 hour and 59 minutes

Session ID: 7df68d9f-4851-45e2-af30-1f81815ca875

Connect to Jupyter

File Edit View Run Kernel Tabs Settings Help

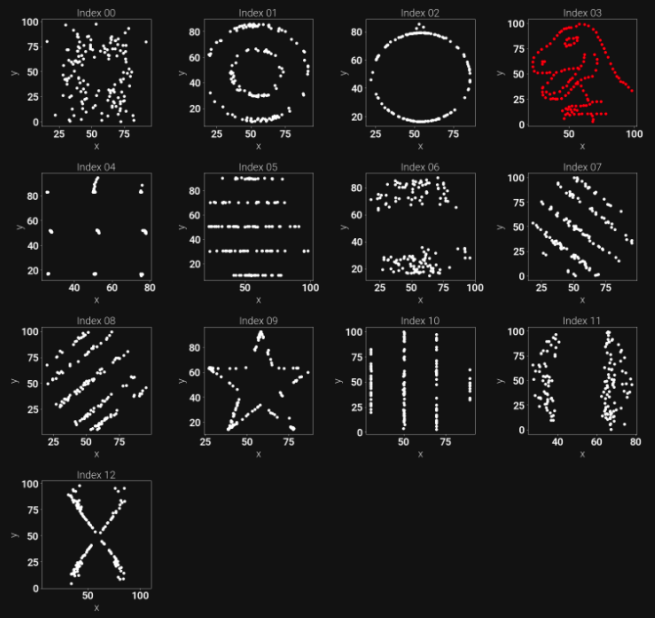
ML90-python > vis-discovery

Name	Last Modified
dat	2 years ago
fig	2 years ago
vis-discovery.ipynb	5 months ago
plt_styles.py	2 years ago

vis-discovery.ipynb

Markdown pyintel

```
# loop over dataframes and plot data
for k,df in enumerate(dfs):
    ax = axarr[k//4,k%4]
    ax.plot(df.x,df.y,'o',c='w' if k != 
    ax.set_xlabel(r'x',**lblprops)
    ax.set_ylabel(r'y',**lblprops)
    ax.set_title(f'Index {k:02d}')
plt.axis('equal')
plt.tight_layout()
```



plt_styles.py

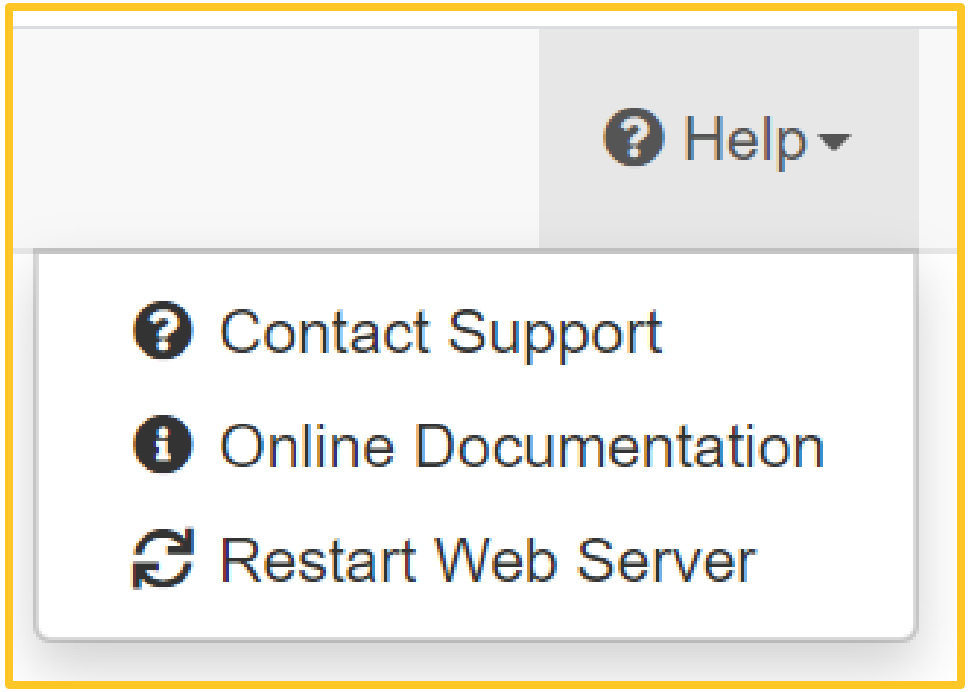
```
from matplotlib import rcParams

gld = '#FFC627'
mrn = '#8C1D40'
dpu = '#0C0C12'

rcParams['figure.figsize']      = (10,6)
rcParams['axes.facecolor']      = 'w'
rcParams['axes.edgecolor']      = 'k'
rcParams['figure.facecolor']    = 'w'
rcParams['xtick.color']         = 'k'
rcParams['ytick.color']         = 'k'
rcParams['text.color']          = 'k'
rcParams['font.size']           = 20
rcParams['axes.labelsize']      = 20
rcParams['axes.titlesize']      = 24
rcParams['legend.fontsize']     = 14
rcParams['xtick.direction']     = 'in'
rcParams['ytick.direction']     = 'in'
rcParams['xtick.major.pad']     = 10
rcParams['xtick.minor.pad']     = 10
rcParams['xtick.major.size']    = 8
rcParams['xtick.minor.size']    = 4
rcParams['xtick.major.width']   = 1.25
rcParams['xtick.minor.width']   = 1.25
rcParams['xtick.minor.visible'] = True
rcParams['xtick.direction']     = 'in'
rcParams['xtick.bottom']        = True
rcParams['xtick.top']           = True
rcParams['ytick.major.pad']     = 10
rcParams['ytick.minor.pad']     = 10
rcParams['ytick.major.size']    = 8
rcParams['ytick.minor.size']    = 4
rcParams['ytick.major.width']   = 1.25
```

rcsparky@gpu9-1:~

```
[rcsparky@gpu9-1:~]$
```

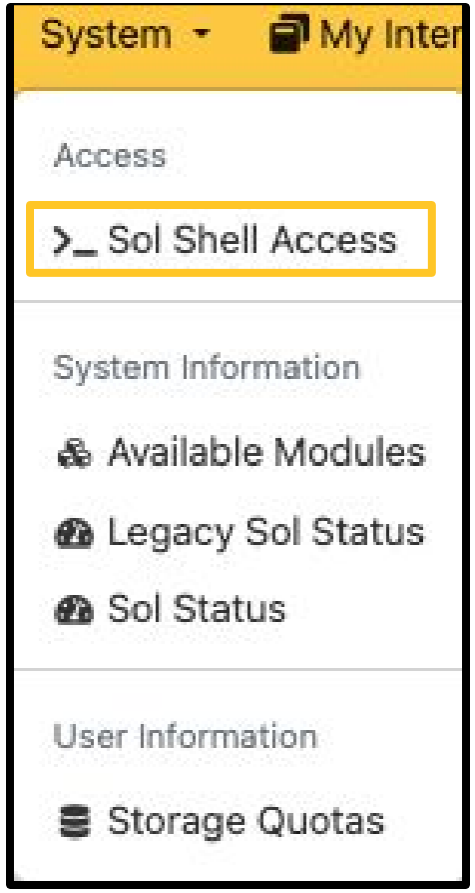
Example: GPU-Enabled Jupyter

Getting Help



- If issues do occur, select “Help” from top navbar
- Here users can contact support
  - Make sure to include Job ID, session-id, errors, and any other valuable information
- Also feel free to ask questions in #rc-support

Using GPUs through the Shell



- Go to https://sol.asu.edu
- Login with ASURITE and password
- From here, select **System** then **Sol Shell Access**

Using the Shell

```
Last login: Thu Mar 23 11:04:02 2023 from 10.139.120.3

The Sol scratch directory (/scratch) is not backed up, and ASU Research
Computing is not responsible for the deletion or loss of any files from the
scratch directory. To ensure the best system stability and performance, the
scratch directory has a limit of 100TB per user. Sol scratch files that have
not been accessed for 90 days will be removed from the system.

For more information, please refer to links.asu.edu/solscratch

                ---------- STORAGE ---------- ---------- INODES ----------
         quota       usage       limit     use  inodes      limit     use
 USER /scratch    0.0 Byte   100.0 TiB    0.0%       0   20000000    0.0%
GROUP /scratch     2.8 GiB  1024.0 TiB    0.0%   32017  unlimited      NA
 /home/rbelshe     0.0 GiB   100.0 GiB    0.0%      NA         NA      NA

================================================================================

  Upcoming Workshops

  REGISTRATION LINKS AND WORKSHOP DESCRIPTIONS AT links.asu.edu/learn

================================================================================

Sol documentation is available at links.asu.edu/sol-docs
```

Help Information, Supercomputer Status, and Usage Information

```
[rcsparky@login02:~]$
```

The Shell Prompt

The Shell Prompt

```
[rcsparky@login02:~]$ sbatch job_script.sh
```

```
[<username>@<hostname>:<working dir>]$ <shell commands>
```

Job Script

```
[rcsparky@login01:~]$ nano job.sh
```

```
#!/bin/bash
#SBATCH -c 1    ## number of cores
#SBATCH -t 15   ## minutes of execution time

module load intel/2020.2
./hello
```

Shell bang line (interprets execution commands)

Execution commands

```
[rcsparky@login01:~]$ sbatch job.sh
```

Python using GPU “job” Script

```
#!/bin/bash
#SBATCH -p general
#SBATCH -q public
#SBATCH -G 1
#SBATCH -o %j.out
#SBATCH -e %j.err
#SBATCH -t 0-4:00:00



module load mamba/latest
source activate genai-26.06
python3 llm.py
```

Module Management

- Modules are used to setup your PATH and other environment variables
- They are used to setup environments for packages and compilers

```
[rcsparky@login01:~]$ module            {lists options}
[rcsparky@login01:~]$ module avail      {lists available packages}
[rcsparky@login01:~]$ module load <package> <…> {add one or more packages}
[rcsparky@login01:~]$ module unload <package> {unload a package}
[rcsparky@login01:~]$ module list       {lists loaded packages}
[rcsparky@login01:~]$ module purge      {unloads all packages}
```

- Multiple compiler families available, so make sure you are consistent between libraries, source and run script!

Working with Python on Sol

- Working with Python requires an additional step to the module system
- First the Mamba package manager module must be loaded
- Then the appropriate environment (including custom environments) **must be activated**, e.g.

```
module load mamba/latest
source activate tensorflow-gpu-2.10.0
```

- Mamba is an open source and faster drop-in replacement for conda.

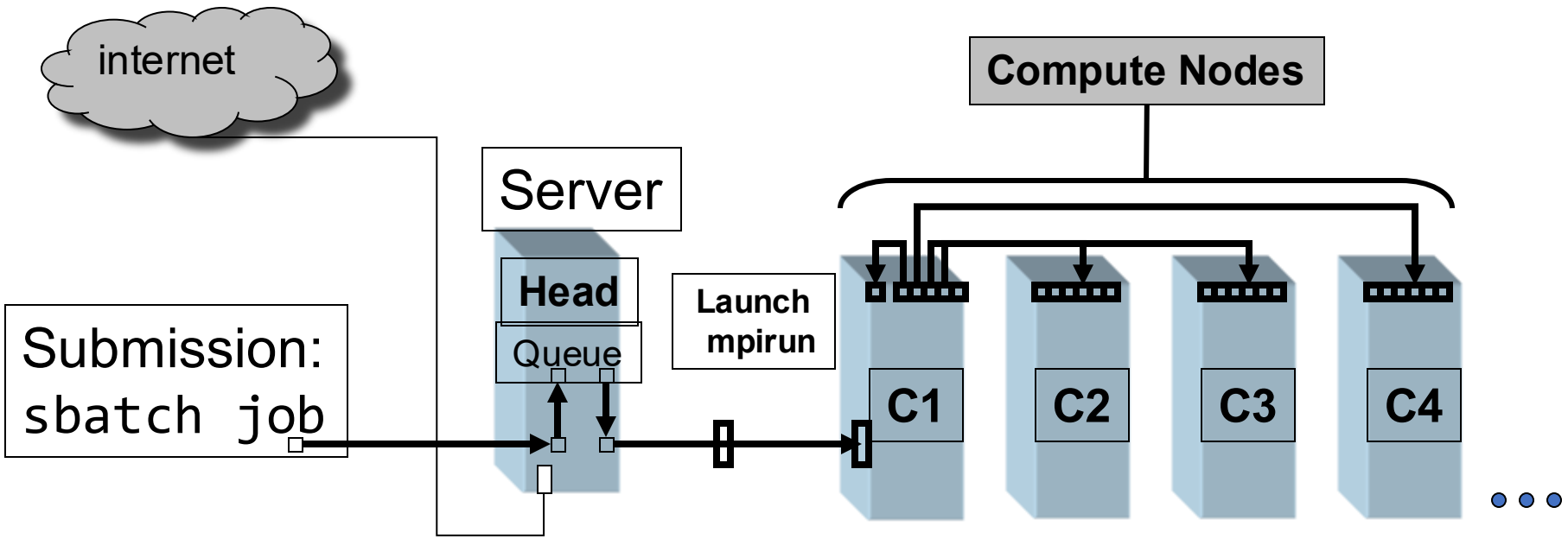
Additional documentation: **https://links.asu.edu/mamba**

Tensorflow GPU Job Script

```
#!/bin/bash
#SBATCH -p htc
#SBATCH -q public
#SBATCH -J tf
#SBATCH -o %j.out
#SBATCH -e %j.err
#SBATCH -G 1
#SBATCH -t 0-4:00:00

module load mamba/latest
source activate tensorflow-gpu-2.10.0
python train.py
```

```
[asurite@login01 ~]$ sbatch job
```

Batch Submission Process



Queue: Job script waits for resources on Server
Compute nodes execute the job script, launching MPI processes

Launch: Contact each compute node to start executable (e.g. a.out)

FairShare – Job scheduling based on priority score

Interactive Mode with GPUs

- GPU session on Sol: `interactive -G 1`
- `interactive -G a100:1`

  or `interactive --gres=gpu:a100:1`
- Entire Node: `interactive --exclusive --mem=0 -G 1`

`interactive` is modified by regular `sbatch` flags!

When using GPUs, make sure to consider…

- **How much time do you expect your job to run?**
  - ≤ 4 hours: use HTC partition
  - > 4: general partition
  - Since jobs can be canceled (preempted) when opportunely using a private QOS, checkpointing is highly encouraged if possible
- **How many GPUs do you expect to need?**
  - `-G {#}`
  - Keep in mind the maximum number of GPUs a node has available and if your program can use, or has been designed, to use multiple GPUs
- **Is model of GPU important?**
  - -G {a100, a30}:{#}
  - See Using GPU documentation

For Reference: Useful Commands

`nano, emacs, vim`	File editors, nano is most friendly
`sbatch <sh script>`	Submit a job
`myjobs` `squeue -u $USER`	Check on the status of your jobs
`showjobs -u $USER`	Show jobs with additional information (e.g. start time)
`scancel <jobid>`	Delete a running or queued job
`scancel -u $USER`	Delete all running or queued jobs for your user
`less, more, cat`	View files (ex: `more <file>`)
`man <command>`	Displays the manual page of a command
`ns`	Node status

For Reference: Useful Monitoring Commands

`thisjob <jobid>` `longjob <jobid>`	Print out queue and parameter information for a specified job
`showparts` `showgpus`	Print out information on partition availability and GPU availability
`seff <jobid>`	Prints job efficiency for completed jobs
`mysacct`	Prints out job hardware usage information for last 24 hours
`top -u $USER` `htop -u $USER`	Commands to monitor job CPU and MEM performance (must be run from compute node)
`nvidia-smi`	Command to monitor GPU performance (must be run from compute node)
`mybalance`	Returns current usage and FairShare score
`report_storage`	Returns GB used in scratch and home

Job Monitoring (squeue utility)

```
[rcsparky@login01:~]$ squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
208952_[0-199]	general	COLD_3	mrline	PD	0:00	1	(BeginTime)
207709	general	R-px-OSa	epopplet	PD	0:00	1	(AssocMaxJobsLimit)
207710	general	R-px-OSa	epopplet	PD	0:00	1	(AssocMaxJobsLimit)
207711	general	R-px-OSa	epopplet	PD	0:00	1	(AssocMaxJobsLimit)
207712	general	R-px-OSa	epopplet	PD	0:00	1	(AssocMaxJobsLimit)
207713	general	R-px-OSa	epopplet	PD	0:00	1	(AssocMaxJobsLimit)

Basic `squeue` options:

- `-u username` Display jobs belonging to specified user
- `-l, --long` Display extended job information

```
[rcsparky@login01:~]$ myjobs
```

OR

```
[rcsparky@login01:~]$ squeue --me
```

To kill a job:

```
[rcsparky@login01:~]$ scancel <JOBID>
```

Scheduling, Priority, and Fairshare

- Jobs in the queue are scheduled based on priority through the SLURM scheduler (try command **`myjobs`**)
- Priority is computed from linear combination of factors (job age, job size, job QOS, and FairShare) with admin defined weights
- FairShare ranges from 0 (lowest priority) to 1 (highest priority). Calculated from usage

$$\text{FairShare} = 2^{-usage}$$

- $usage$ decays exponentially, half-life of 1 week
- Usage tracks core, memory, and GPU utilization
- See more [in our documentation]

Things to Remember

- **Shared login node**: Do not compute on the login nodes
- **Shared network**: When using the shell, limit simultaneous transfers
- **Shared compute resource**: Give good estimate of runtime. Time extensions are the rare exception and are strongly discouraged. Test submission scripts before submitting them at large scale
- **Shared GPU reservation:** Each team should have access to one A100. If idle A100s are available feel free to use for short periods, but be mindful of depriving access to other teams.
- **Shared help desk**: Do some homework before submitting ticket. Do not submit multiple tickets on same topic. Describe issue in detail (e.g. job ID, full path to failing sbatch script, etc.). Be patient
- Review policies and guidelines here

Quick Links

Learn more about the new Sol supercomputer

For assistance: rtshelp@asu.edu

Office Hours: 1-3:30 PM Tuesday (and Wednesday during academic semesters)

Slack Channel: #rc-support

Documentation Page: links.asu.edu/docs

Workshops: links.asu.edu/learn

RC Expo: links.asu.edu/RCExpo

Available Software: links.asu.edu/modules

Thank you

Please visit

researchcomputing.asu.edu

for more information

ASU Research Computing
Arizona State University

June 23, 2025

ASU Knowledge Enterprise
Arizona State University

Core Research Facilities