



TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU



TEXAS

The University of Texas at Austin

Vista Introduction

Interacting with High Performance
Computing Systems at TACC

PRESENTED BY:

Jack Gaither, Engineering Scientist Associate
Texas Advanced Computing Center

<https://docs.tacc.utexas.edu/hpc/vista/>



Search docs

TACC Essentials

Documentation Overview

[Getting Started](#)

► **Managing Your TACC Account**

- ▶ Multi-Factor Authentication

► TACC Good Conduct Guide

► [Data Transfer Guide](#)

► Software at TACC

Help

HPC User Guides

 Chameleon

► Corral

► **Frontera**

► Lonestar6

 [Jetstream2](#)

► Ranch

- ▶ Stampede3

► Stallion

▼ Vista

Notices

Introduction

► **Account Administration**

► System Architecture

- ▶ Running Jobs

▶ Launching Applications

► NVIDIA MPS

► Machine Learning

► **Building Software**

► Building for Performance

HPC User Guides > Vista

[Suggest an update via GitHub](#)

VISTA USER GUIDE

Last update: February 18, 2025

Notices

- **New:** See TACC Staff's [notes on incorporating NVIDIA's Multi-Process Service](#). (MPS)
- **Important:** Please note [TACC's new SU charge policy](#). (09/20/2024)
- **Subscribe to Vista User News.** Stay up-to-date on Vista's status, scheduled maintenances and other notifications. (09/01/2024)

Introduction

TACC's new AI-centric system, Vista, is in full production for the open science community. Vista serves as a bridge from Frontera to Horizon, the primary system of the U.S. [NSF Leadership-Class Computing Facility](#) (LCCF), and marks a departure from the x86-based architecture to one with CPUs based on Advanced RISC Machines architecture. Vista expands the Frontera project's support of Machine Learning and GPU-enabled applications with a system based on NVIDIA Grace Hopper architecture and provides a path to more power efficient computing with NVIDIA's Grace Grace ARM CPUs.

The Grace Hopper Superchip introduces a novel architecture that combines the GPU and CPU in one module. This technology removes the bottleneck of the PCIe bus by connecting the CPU and GPU directly with NVLINK and exposing the CPU and GPU memory space as separate NUMA nodes. This allows the programmer to easily access CPU or GPU memory from either device. This greatly reduces the programming complexity of GPU programs while providing increased bandwidth and reduced latency between CPU and GPU.

The Grace Superchip connects two 72 core Grace CPUs using the same NVLINK technology used in the Grace Hopper Superchip to provide 144 ARM cores in 2 NUMA nodes. Using LPDDR memory, each Superchip offers over 850 GiB/s of memory bandwidth and up to 7 TFlops of double precision performance.

Vista is funded by the National Science Foundation (NSF) via a supplement to the Computing for the Endless Frontier award, [Award Abstract #1818253](#). Please [reference TACC](#) when providing any citations.

Account Administration

Crontabs

TACC allows cronjobs but be aware that crontab files are unique to the login node where they were created and are not shared across the login nodes. Crontab files are not allowed on the compute nodes.



Note

All TACC HPC systems host multiple login nodes. When you login, your connection is routed to the next available login node via round-robin DNS. This practice balances the user load across the system.

Logging In: What do I need?

- To access Vista, you will need the following:
 - Terminal application that supports SSH
 - TACC Account
 - Enable TACC MFA for your account on the TACC User Portal [here](#)
 - Member of a project with an active Vista allocation

Logging In: What do I see?

- Project Membership
- Available SUs (and expiration date)
- Disk Quotas

Welcome to the Vista Supercomputer
Texas Advanced Computing Center, The University of Texas at Austin

Unauthorized use/access is prohibited. **
If you log on to this computer system, you acknowledge your awareness
of and concurrence with the UT Austin Acceptable Use Policy. The
University will prosecute violators to the full extent of the law.

TACC Usage Policies:
<http://www.tacc.utexas.edu/user-services/usage-policies/>

Welcome to Vista, please read these important system notes:

--> Vista user documentation is available at:
<https://docs.tacc.utexas.edu/hpc/vista/>

----- Project balances for user jag22624 -----
Name Avail SUs Expires	Name Avail SUs Expires
TRA23003 4995 2025-12-31	DesignSafe-Co 0 2025-09-30
STA24005 100 2025-06-30	A-ccsc 175365 2027-11-30
----- Disk quotas for user jag22624 -----	
Disk Usage (GB) Limit %Used File Usage Limit %Used	
/scratch 0.0 0.0 0.00 1 0 0.00	
/home1 0.0 23.3 0.10 1849 500000 0.37	
/work 128.9 2048.0 6.29 593732 3000000 19.79	

login1.vista(170)\$ █

Vista File Management

- Comprised of three file systems:
 - \$HOME
 - Quota of 23GB or 500,000 files
 - Not intended for parallel or high-intensity file operations
 - Backed up regularly
 - Not purged
 - \$WORK
 - Quota of 1TB or 3,000,000 files across all TACC systems
 - Not intended for high-intensity file operations or jobs involving very large files.
 - Not backed up
 - Not purged
 - \$SCRATCH
 - No quota
 - Intended for high-intensity file operations or jobs involving very large files
 - Not backed up
 - Files are subject to purge if access time is more than 10 days old info found [here](#)

Using the File Systems

- Check quota with /usr/local/etc/taccinfo

```
login1.vista(170)$ /usr/local/etc/taccinfo
----- Project balances for user jag22624 -----
| Name                Avail SUs    Expires | Name                Avail SUs    Expires |
| A-ccsc              175365    2027-11-30 | DesignSafe-Co        0    2025-09-30 |
| STA24005            100    2025-06-30 | TRA23003             4995    2025-12-31 |
----- Disk quotas for user jag22624 -----
| Disk                Usage (GB)    Limit    %Used    File Usage    Limit    %Used |
| /scratch            0.0        0.0      0.00      1            0      0.00 |
| /home1              0.0       23.3     0.10     1849        500000    0.37 |
| /work               128.9    2048.0    6.29    593732     3000000    19.79 |
-----
```

Navigating the File System

- Basic bash commands:
 - cd
 - Change directory: `cd $WORK`
 - ls
 - List the contents of a directory: `ls -al`
 - mv
 - Move a file or directory to another directory
 - cp
 - Copy a file or directory to a directory: `cp </some/file> </destination/directory>`
 - pwd
 - Print working directory to get the file path to your current directory

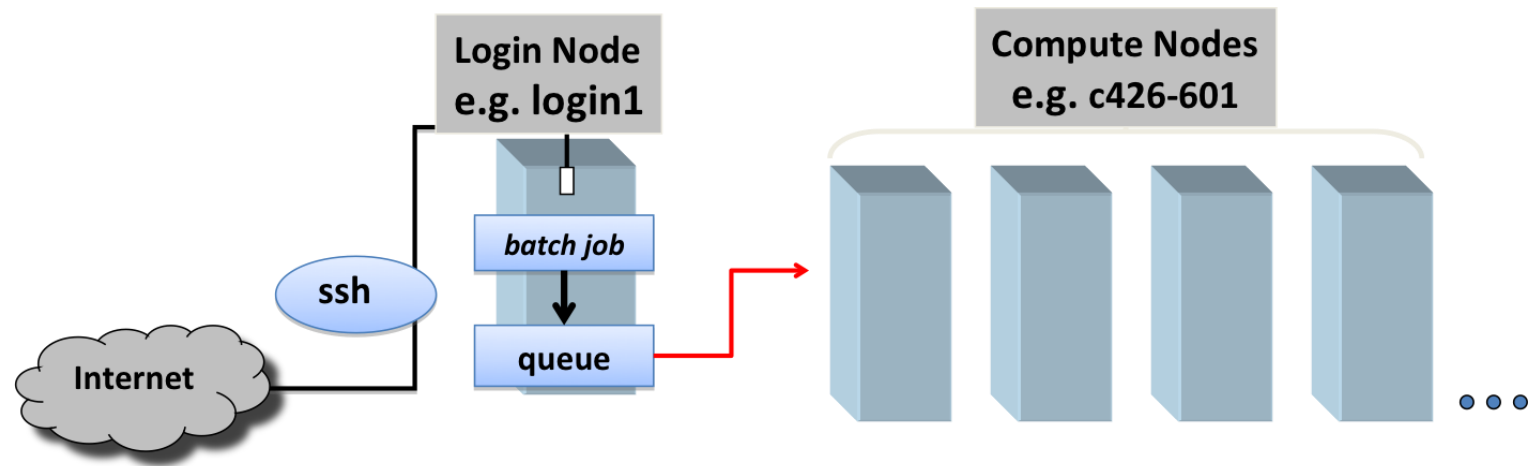
Transferring Files

- For SSH file transfer utilities we typically use scp or rsync
 - Docs for scp [here](#)
 - Docs for rsync [here](#)

Sharing Project Files

- The simplest way to have a shared workspace with collaborators that are in your group and part of your allocation is to use unix group level permissions on your files and directories
- To do this we use the command: `chmod 750`
 - This will allow full rwx access for the owner and r-x for your group, allowing them to read and execute those shared files without being able to edit
- For additional info on sharing project files visit [here](#)

Vista Overview



Good Citizenship on Vista

- Do NOT run jobs on the login nodes
- Do NOT stress the shared file system
- Limit Input/Output (I/O) activity
- Adhere to file transfer guidelines

Vista

- Production queues (gg, gh, gh-dev)
 - qlimits command displays info on queues
- SLURM Workload Manager (job scheduler)
 - jobID
 - Submit (sbatch, iddev)
 - Manage and monitor (squeue, scontrol, scancel)

idev

- Uses SLURM to start an interactive session on a compute node
- 30 minutes, single node, development queue
- Perfect for compiling, running i/o intensive applications, testing sbatch jobs before submitting them as an sbatch job

```
-> Checking on the status of gg queue. OK

-> Defaults file      : ~/.idevrc
-> System            : vista
-> Queue             : gg              (idev failover queue)
-> Nodes              : 1              (idev default      )
-> Tasks per Node    : 144            (Queue default    )
-> Time (minutes)     : 30            (idev default      )
-> Project            : A-ccsc         (~/.idevrc         )

-----
Welcome to the Vista Supercomputer
-----

No reservation for this job
-> Verifying valid submit host (login1)...OK
-> Verifying valid jobname...OK
-> Verifying valid ssh keys...OK
-> Verifying access to desired queue (gg)...OK
-> Checking available allocation (A-ccsc)...OK
-> Quotas are not currently enabled for filesystem /home1/05687/jag22624...OK
-> Verifying that quota for filesystem /work/05687/jag22624/vista is at 6.29% allocated...OK
Submitted batch job 149711

-> After your idev job begins to run, a command prompt will appear,
-> and you can begin your interactive development session.
-> We will report the job status every 4 seconds: (PD=pending, R=running).

-> job status: PD
-> job status: R

-> Job is now running on masternode= i617-024...OK
-> Checking to make sure your job has initialized an env for you....OK
-> Creating interactive terminal session (login) on master node i617-024.
-> Node access by: ssh -Y -A -o "StrictHostKeyChecking no" i617-024

i617-024[gg](170)$
```

Exit an idev session

```
[i617-024[gg](170)$ exit  
logout  
Connection to i617-024 closed.  
Cleaning up: submitted job (yes) removing job 149711.  
login1.vista(172)$ █
```

Submitting a job using sbatch

- Uses #SBATCH directives to request computing resources
- A table of common #SBATCH options can be found [here](#)
- Sbatch jobs are meant for applications that do not require any interactivity from the user

SLURM batch file

```
#!/bin/bash
```

```
#SBATCH -J myjob      # Job name
```

```
#SBATCH -o myjob.o%j  # Name of stdout output file
```

```
#SBATCH -e myjob.e%j  # Name of stderr error file
```

```
#SBATCH -p gg         # Queue (partition) name
```

```
#SBATCH -N 1          # Total # of nodes
```

```
#SBATCH -n 144        # Total # of mpi tasks
```

```
#SBATCH -t 00:05:00   # Run time (hh:mm:ss)
```

```
#SBATCH --mail-type=all # Send email at begin and end of job
```

```
#SBATCH -A <your allocation> # Project/Allocation name  
(req'd if you have more than 1)
```

```
#SBATCH --mail-user= <your email here>
```

```
# Any other commands must follow all #SBATCH directives...
```

```
module list
```

```
pwd
```

```
date
```

```
# Launch code...
```

```
echo "Hello World" # Use ibrun instead of mpirun or mpiexec
```

Submitting the job

- To submit the SLURM Batch job use the following command: `sbatch batchjob.sh`
- This should generate a text output to let you know about the parameters of the job and then provide you with a Job ID once the job has successfully been admitted to the queues

Submitting the job

```
login2.frontera(1031)$ sbatch batchjob.sh
```

```
-----  
Welcome to the Frontera Supercomputer  
-----
```

```
No reservation for this job
```

```
--> Verifying valid submit host (login2)...OK
```

```
--> Verifying valid jobname...OK
```

```
--> Verifying valid ssh keys...OK
```

```
--> Verifying access to desired queue (small)...OK
```

```
--> Checking available allocation (A-ccsc)...OK
```

```
--> Verifying that quota for filesystem /home1/02070/scottre is at 0.76% allocated...OK
```

```
--> Verifying that quota for filesystem /work2/02070/scottre/frontera is at 0.94% allocated...OK
```

```
Submitted batch job 3968716
```

```
login2.frontera(1032)$ █
```

Check Status of Job

- `queue -u <your username>`
- `queue -j <job id or comma delimited list of job ids>`
- Jobs have three states:
 - PD means Pending
 - R means Running
 - CG means Completing (cleaning up after exiting the job)

Cleaning up jobs

- When your batch job has finished running it will automatically be cleared from the queues.
- Your output will be in the folder you pointed it to within your batch job file.
- If for some reason you wish to cancel your job while it is still running, you can do so with `scancel <jobID>`

Tacc Analysis Portal (TAP)

- Web portal interface that allows users to run DCV, VNC, Jupyter Notebook, and Rstudio sessions interactively with an available display
- Very useful for interactive development with data visualization eliminating the need to copy files to local to view generated images/figures
- Documents page [here](#)

Monitoring Jobs IO Do's

- Do:
 - Use \$SCRATCH for production jobs and take advantage of /tmp space if possible info [here](#)
 - Work with large files rather than many tiny files if possible
 - Use a single stripe for small files and stripe large files on the Lustre file systems info on striping [here](#)
 - Open/close only once for each file if possible rather than conducting open/close/state operations repetitively
 - Use scalable Parallel I/O libraries, like phdf5, pnetcdf, PIO
 - Install Python/R modules under \$SCRATCH
 - Copy Python/R modules under /tmp for large-scale runs Use Python_Cacher
 - Use profilers or I/O monitoring tools when necessary

File Striping

- Lustre can stripe (distribute) large files over several physical disks, making it possible to deliver the high performance needed to service input/output (I/O) requests from hundreds of users across thousands of nodes
- a file with 16 stripes, for example, is distributed across 16 OSTs
- A good rule of thumb is to allow at least one stripe per 100GB in the file.
 - While the \$WORK file system has hundreds of OSTs \$SCRATCH has far fewer meaning we need to limit the stripe count to no more than 8
- \$WORK striping ex: `lfs setstripe -c 12 $PWD`
- \$SCRATCH striping ex: `lfs setstripe -c 8 $PWD`
- To get the stripe count for a file or directory use the command: `lfs getstripe myfile`

Job Tools

- REMORA or **RE**source **MO**nitoring for **R**emote **A**pplications link [here](#)
 - Remora provides a simple interface to gather important system utilization data while running on HPC systems. Remora monitors almost all user activities and provides per-node and per-job resource usage data for users
- CD Tools or Collect-Distribute link [here](#)
 - has been designed and developed to distribute files or directories to or from the /tmp directory
- OOOPS or **O**ptimal **O**verloaded **I/O** **P**rotection **S**ystem link [here](#)
 - easy-to-use tool to help HPC users optimize heavy I/O requests and reduce the impact of high I/O jobs
- Python_Cacher link [here](#)
 - For jobs that make use of large numbers of Python modules or use local installations of Python/Anaconda/MiniConda

Comments, Questions, Cool Stuff?

Thank you for attending!