HiPerGator Usage Quick Reference

UCF Office of Research Cyberinfrastructure

Short internal reference for common usage and support questions

April 17, 2026

Contents

1 Purpose

This document is a short reference for common HiPerGator support topics that frequently come up when helping users. It is intentionally concise: the goal is to have the main ideas, common commands, and the most useful official links in one place rather than reproduce the full UF documentation.

Because HiPerGator hardware, partitions, and policies can change, it is still wise to check the linked UF documentation when giving final advice on edge cases or time-sensitive questions.

Source: https://docs.rc.ufl.edu/

2 System at a Glance

HiPerGator is the University of Florida's research computing system. The official HiPerGator overview page currently describes HiPerGator 4th Gen as having approximately:

- 60,000 CPU cores,
- 600 NVIDIA L4 GPUs,
- 504 NVIDIA B200 GPUs,
- 11 PB all-flash, high-performance parallel file system.

For support purposes, the most important practical points are:

- Jobs are scheduled with Slurm,
- Access is group/allocation based,
- `/blue` is the main high-performance storage location for job I/O,
- `/orange` is slower storage better suited for longer-term retention,
- Users should not run production work on the login nodes.

Source: https://it.ufl.edu/rc/hipergator/

Source: https://docs.rc.ufl.edu/training/new_user_training/

3 Access and Interfaces for UCF users

3.1 Federated users

UCF faculty and researchers are considered 'federated users'. Web access is through institutional login via InCommon. SSH access for federated users requires SSH keys, and they must first connect through eduVPN.

3.2 Command line access

- Only **SSH key access** is available for SSH,
- Username/password SSH login is **not** available,
- The user must first connect to the **University of Florida eduVPN** service before using SSH.

3.2.1 Connecting to eduVPN

- Download the latest version of the eduVPN client from here and install it.
- After installing the eduVPN client, open the application.
- Search for and **select the University of Florida** (Not UCF). You will be presented with an InCommon login page.
- **Select UCF** from the dropdown and click on Select.
- After logging in, approve eduVPN access on the resulting page and close the web page. You are now connected to our instance of eduVPN.

Typical SSH command:

```
ssh username@hpg.rc.ufl.edu
```

Important notes:

- use the HiPerGator username sent in the account creation email,
- connect to **UF**'s eduVPN instance (not UCF's),
- never share the private SSH key.

Source: https://docs.rc.ufl.edu/access/federated_login/

Source: https://docs.rc.ufl.edu/interfaces/terminal/

3.3 Web interfaces

Interface	URL	Typical use
SSH terminal	`hpg.rc.ufl.edu`	Standard shell access for file management, software use, and job submission.
Open OnDemand	https://ood.rc.ufl.edu/	Browser-based terminal, files, interactive apps, GUI sessions, Jupyter, job composer.
JupyterHub	http://jhub.rc.ufl.edu/	Quick Jupyter notebooks with preset resource options.

Galaxy	https://galaxy.rc.ufl.edu/	Web-based biology/genomics workflows.

Important JupyterHub note: it can only use a user's **primary** group resources. If a user needs a secondary group/allocation, Open OnDemand is the better choice.

Source: https://docs.rc.ufl.edu/quickstart/interfaces/

Source: https://docs.rc.ufl.edu/interfaces/ood/

Source: https://docs.rc.ufl.edu/access/federated_login/

4 Groups, Sponsors, and Allocations

4.1 Core ideas

- A faculty member with a UFIT Research Computing investment is an **investor**.
- Every user must have a **sponsor**.
- A user's **primary group** is the sponsor's group.
- Access to other resources is granted through **secondary membership** in an **allocation group**.
- All groups must have an active **Blue Storage Unit (BSU)** allocation in order to use HiPerGator services.

4.2 Allocation types

A group's allocation can be built from either or both of:

- **Hardware investment**: purchase for the full useful lifetime (typically 5 years).
- **Service investment**: purchase the use of resources for a shorter period.

4.3 Trial Allocation

All new sponsors on HiPerGator are eligible to request a free trial resource allocation for their primary group. This allocation allows a research team that is just getting started on HPG to try using the system and determine what resources, applications, and services to invest in for successful and timely completion of their projects. A trial allocation is requested via the **trial allocation request form** (must be on the UF network or UF VPN to access the form.)

The default trial resources:

- 16 NCU
  (Normalized Compute Unit; one priority CPU core with 8 GB RAM, approximately 7 GB usable by applications)
- 2 NGU
  (Normalized GPU Unit; one NVIDIA GPU. Requires at least some CPU and storage investment)
- 1 BSU
  (Blue Storage Unit; 1 TB of high-performance storage)
- 1 OSU
  (Orange Storage Unit; 1 TB of slower, longer-term storage)

Trial allocation duration is **3 months**. If you need different resources or a longer trial allocation duration please add a comment to the request when submitting the form.

Source: https://docs.rc.ufl.edu/resources/trial/

5 Storage

5.1 Main storage locations

Location	Quota level	Best use	Notes
`/home/USER`	User	Scripts, code, config files, compiled applications	40 GB per user; do **not** use for job input/output; snapshots available for seven days.
`/blue/GROUP`	Group	Main job input/output, active project data	Highest-performance shared filesystem; this should be the default location for heavy job I/O.
`/orange/GROUP`	Group	Long-term storage, raw data, archival-style storage	Slower than `/blue`; not intended for large job I/O.

5.2 Backup and quota notes

- Unless a separate protection option is purchased, **nothing is backed up by default**.
- `/blue` and `/orange` quotas are at the **group** level.
- `/home` quotas are at the **user** level.
- `/blue` and `/orange` are automounted; users may need to type the full path before the directory becomes visible.

5.3 Useful storage commands

```
home_quota
blue_quota
orange_quota
ncdu
```

Source: https://docs.rc.ufl.edu/training/new_user_training/

Source: https://docs.rc.ufl.edu/software/ufrc_tools/

6 Login Nodes and Job Submission

6.1 Login node limits

Login nodes are for file management, job submission, short testing, and development only. UF training documentation gives these acceptable use limits:

- no more than 16 cores,
- no more than 64 GB RAM,
- no longer than 10 minutes.

Do **not** run production jobs on the login nodes.

Source: https://docs.rc.ufl.edu/training/new_user_training/

6.2 Basic Slurm workflow

```
sbatch my_script.sh
squeue --me
```

Common support reminders:

- Users should request only the CPU, memory, GPU, and walltime they actually need.
- Over-requesting resources does not make jobs run faster and can delay scheduling.
- For work beyond login-node limits, users should request interactive resources or submit a batch job.

Source: https://docs.rc.ufl.edu/training/new_user_training/

6.3 Common time limits

Partition / QoS	Default	Max
Interactive & GUI Sessions		
`hpg-dev`	10 minutes	12 hours
`hwgui`	10 minutes	4 days
Compute Partitions		
`hpg-default`	10 minutes	31 days
`bigmem`	10 minutes	31 days
`hpg-milan`	10 minutes	31 days
`hpg-turin`	10 minutes	14 days
`hpg-b200`	10 minutes	14 days
QOS		
Burst QoS	10 minutes	4 days
Investment QoS	10 minutes	31 days

Additional note from UFRC documentation: Open OnDemand GPU jobs are limited to 12 hours.

Source: https://docs.rc.ufl.edu/scheduler/partition_limits/

6.4 Burst QoS

UF documents a **burst capacity** equal to **9 times** the number of CPU cores in the group's investment. Burst jobs run when idle cores are available. They may wait in queue for an indeterminate amount of time. There is **no burst QoS for GPU partitions**.

Source: https://it.ufl.edu/rc/documentation/procedures/groups-and-allocations/

Source: https://it.ufl.edu/rc/about/our-services/computation/

Source: https://docs.rc.ufl.edu/scheduler/partition_limits/

Source: https://it.ufl.edu/rc/get-started/price-list/

7 GPU Quick Reference

7.1 Common GPU batch directives

GPU type	Typical directives	Notes
L4 (Default)	`##SBATCH -partition=hpg-turin` `##SBATCH -gpus=2`	General L4 GPU jobs on one node.
B200	`##SBATCH -partition=hpg-b200` `##SBATCH -gpus=2`	B200 jobs on one node.
Alternative format	`##SBATCH -gres=gpu:1` or `##SBATCH -gres=gpu:b200:1`	Note that `-gpus=` does **not** report GPU usage data to `slurmInfo`.

7.2 Interactive GPU examples

```
srun --partition=hpg-turin --gpus=1 --pty -u bash -i
srun --partition=hpg-b200 --gpus=2 --mem=300gb --time=3:00:00 --ntasks=1 --pty bash -i
```

Note that the default time Interactive GPU sessions default to 10 minutes and are limited to 12 hours.

Source: https://docs.rc.ufl.edu/scheduler/gpu_use/

7.3 Sample VASP GPU script

The following is adapted from the University of Florida Research Computing (UFRC) GPU access documentation and is useful as a support reference:

```
#!/bin/bash
#SBATCH --job-name=vasptest
#SBATCH --output=vasp.out
#SBATCH --error=vasp.err
#SBATCH --mail-type=ALL
#SBATCH --mail-user=email@ufl.edu
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --ntasks-per-node=8
#SBATCH --ntasks-per-socket=4
#SBATCH --mem-per-cpu=7000mb
#SBATCH --distribution=cyclic:cyclic
#SBATCH --partition=hpg-turin
#SBATCH --gres=gpu:4
#SBATCH --time=00:30:00

echo "Date      = $(date)"
echo "host      = $(hostname -s)"
echo "Directory = $(pwd)"

module purge
module load cuda intel openmpi vasp

T1=$(date +%s)
srun --mpi=${HPC_PMIX} vasp_gpu
T2=$(date +%s)

ELAPSED=$((T2 - T1))
echo "Elapsed Time = $ELAPSED"
```

Helpful reminder: `module spider vasp` can be used to inspect available versions and compatible dependency stacks.

Source: https://docs.rc.ufl.edu/scheduler/gpu_use/

8 NVIDIA Omniverse

NVIDIA Omniverse^{TM} is a collection of libraries and microservices for developing **physical AI** applications such as industrial digital twins and robotics simulation. Omniverse librairews enable software makers to integrate pre-built functionality into their solutions.

Source: https://www.nvidia.com/en-us/omniverse

HiPerGator Users from UCF that have an active allocation can request access to OVX systems by submitting a request (only PI to submit request) and including all the users who need accesss to OVX systems for their application. Few things to note:

- A user does not need any additional licenses
- To run Omniverse, you will need to **apply for the OVX access**. OVX is NVIDIA server infrastructure designed for 3D visualization, simulation, and AI inference for digital twin workflows. On HiPerGator, it uses L40 GPUs with 48 GB memory to provide low-latency remote desktop environments compatible with advanced digital twin tools.
- OVX access itself is free, and you can submit a request here:https://docs.rc.ufl.edu/services/OVX/#requesting-ovx-resources
- OVX eligibility is based on a group's HPG resources. No new purchase is necessary if a current allocation includes the storage, CPU and GPU resources.

8.1 How to apply for OVX access

Apply by submitting a ticket using (https://support.rc.ufl.edu/enter_bug.cgi) and add the following info :

- The number of VMs you wish to create
- This represents the number of simultaneous connections you may have
- The GPU resources for each VM GPUs can be split between VMs in factors of 2, such as 1/2 or 1/4 of a GPU to a VM. Splitting GPUs may be desirable if resources are limited, but multiple VMs are required.
- Recall that each L40 GPU has 48 gigabytes of VRAM.
- The CPU resources for each VM (aka 'cores')
- The memory for each VM
- The local storage for each VM
- Blue storage is encouraged to be utilized in general cases. The local VM storage should not be large, and should not exceed 100 gigabytes. Local storage will be cleared when a user logs off a VM.
- A description of the use case that requires OVX
- Please provide as much relevant information as necessary.

Source: https://docs.rc.ufl.edu/services/OVX/#requesting-ovx-resources

8.2 How many resources to request for OVX access

The number of VMs depends on the number of **simultaneous users** you expect. For example, if only 1–2 people will be using the system at the same time, requesting 1–2 VMs should be sufficient. For GPU allocation per VM, it depends on the workload and memory requirements. For lighter visualization workloads, you can start with $\frac{1}{2}$ GPU per VM. For heavier or more memory-intensive workloads, 1 GPU per VM is a better choice. Each L40 GPU has 48GB of VRAM, and we've seen users successfully run Omniverse workloads with both $\frac{1}{2}$ GPU and 1 GPU configurations. You may also want to consider requesting Blue storage based on your project's data size, since local VM storage is limited and temporary. The configuration you submit does not have to be final. Once we receive your application, our team will review your use case and may suggest adjustments if needed.

9 File Transfer

This section provides a brief overview of common ways to transfer data to and from HiPerGator. In practice, the best tool depends on the size of the transfer, whether folders are involved, and whether convenience or robustness is more important. For detailed instructions on how to use each method, click the method name in the table below to open the official UF documentation.

UCF has a Globus subscription for secure and reliable research data management across systems. Using Globus, you can transfer, share and discover data via a single interface. For more information, visit this link.

9.1 General guidelines for selecting a tool

Method	Best for	Considerations	Resume interrupted transfers?	Speed
Globus	Large datasets	Requires Globus on both ends of the transfer	Auto-resumes	Fastest
rsync	Most use cases	Good general option, but command line only	Manual resume	Fast
SFTP	Small to medium transfers	Convenient CLI and GUI options	No	Fast
Samba	Browsing or small transfers	Convenient because folders can be mounted on a local desktop, but can have issues	No	Slow
OOD Files menu	Small uploads/-downloads	Does not handle large files or folders well	No	Medium
Jupyter upload	Single small files	Cannot upload or download folders; limited to files under 2 GB	No	Slow

For large transfers, Globus is generally the preferred option. For routine command-line transfers, rsync is often the most practical choice. SFTP, Samba, the Open OnDemand Files menu, and Jupyter upload are more convenient for smaller and simpler transfers, but they are not ideal for large datasets or complex directory structures.

Source: https://docs.rc.ufl.edu/data_transfer/overview/

10 Useful UFRC Helper Commands

The `ufrc` module is loaded automatically at login and provides many helper commands.

Command	Purpose
`id`	Show real and effective user and group IDs.
`groups`	Show the groups that the user is in.
`showAllocation`	Show group resource allocations.
`showAssoc`	Show Slurm account associations.
`showQos`	Show QoS limits.
`slurmInfo`	Show allocation, pending, and running job information for a Slurm account.
`jobhtop`	Real-time display of job processes.
`jobnvtop`	Real-time display of GPU usage/processes for a job.
`nodeInfo`	Show node types and capacities.
`blue_quota`	Show Blue usage/quota.
`orange_quota`	Show Orange usage/quota.
`home_quota`	Show home quota.
`ncdu`	Inspect storage usage by directory.
`qos_to_burst JOBID`	Move a pending job to burst QoS.
`qos_to_main JOBID`	Move a pending job back to investment QoS.

Source: https://docs.rc.ufl.edu/software/ufrc_tools/

11 Common Support Reminders

1. **Do not run heavy work from SSH IDE sessions.** VS Code, PyCharm, and similar SSH sessions land on a login node unless the user explicitly launches compute resources another way.
2. **Use `/blue` for job I/O.** Writing large outputs to `/home` or `/orange` is a common mistake.
3. **Check group/QoS first.** Many scheduling questions are really account/QoS/allocation issues.
4. **For software install requests**, ask for the application name and download URL.
5. **For job debugging**, ask for JobID, job script path, log path, and a short description of the failure.

Source: https://docs.rc.ufl.edu/training/new_user_training/

Source: https://docs.rc.ufl.edu/support/faq/

Source: https://docs.rc.ufl.edu/interfaces/jupyter/

12 HiPerGator Support for UCF

UF states that front-line support for partner universities comes from the local research computing or IT support unit, with UFIT Research Computing acting as the escalation path when needed.

Dr. Nandan Tandon (nandan.tandon@ucf.edu) and **Dr. Glenn Martin** (glenn.martin@ucf.edu) are the UCF sponsors for any HiPerGator needs.

Acronyms

BSU Blue Storage Unit; 1 TB of high-performance storage. 5, 6

CLI Command Line Interface. 11

GUI Graphical User Interface. 4, 8, 11

NCU Normalized Compute Unit; one priority CPU core with 8 GB RAM (approximately 7 GB usable by applications). 6

NGU Normalized GPU Unit; one NVIDIA GPU. Requires at least some CPU and storage investment. 6

OSU Orange Storage Unit; 1 TB of slower, longer-term storage. 6

13 Primary Official References

- HiPerGator overview: https://it.ufl.edu/rc/hipergator/
- UFRC documentation home: https://docs.rc.ufl.edu/
- New user training: https://docs.rc.ufl.edu/training/new_user_training/
- Groups and allocations: https://it.ufl.edu/rc/documentation/procedures/groups-and-allocations/
- Interfaces overview: https://docs.rc.ufl.edu/quickstart/interfaces/
- SSH access: https://docs.rc.ufl.edu/interfaces/terminal/
- Open OnDemand: https://docs.rc.ufl.edu/interfaces/ood/
- Federated login: https://docs.rc.ufl.edu/access/federated_login/
- GPU use: https://docs.rc.ufl.edu/scheduler/gpu_use/
- Partition time limits: https://docs.rc.ufl.edu/scheduler/partition_limits/
- UFRC tools: https://docs.rc.ufl.edu/software/ufrc_tools/
- Price list: https://it.ufl.edu/rc/get-started/price-list/
- Other universities / partner policy: https://it.ufl.edu/rc/documentation/policies/other-universities/