Terms and Conditions/Information for Users

Grace is a shared resource, and for the most part, a 'self-service' resource. In order to be fair to all users, we have set the following rules. Ignorance of the rules is not an acceptable excuse for bad behavior. To request a new account please e-mail hpcadmin@uams.edu after reading these rules acknowledging that you agree to abide by them.

1. The login nodes are not meant for computation. Computationally intensive programs should be run as jobs using the queueing system.
    a. Users caught running computationally extensive programs (including containers) on the login nodes without permission from the system administrators may have their privileges on Grace suspended temporarily until after they have a conversation with a system administrator.
    b. The system administrators may kill computationally intensive programs running on the login nodes without warning to insure that other users are not adversely affected by the computationally intensive program.
    c. Compiling programs is allowed, if you have no facilities on your personal computer for compiling programs.
2. Users may not login (e.g. use ssh) directly into any of the computational nodes, unless the user has an active job running on that node.
    a. Users caught logging into computational nodes may have their Grace accounts temporarily suspended until they have a conversation with a system administrator.
    b. The one exception is if you have a non-interactive job running on a particular computational node, you may ssh into that node to check that job's status, nothing else.
    c. Users should only log into the login nodes, or utilize the Open OnDemand portal (https://portal.hpc.uams.edu), and should only access the computational nodes through a scheduled and running job, nothing else.
3. If you need an interactive shell or similar on a computational node, please run an interactive job..
    a. Launching a non-interactive job to do interactive work is strongly discouraged, and may lead to a conversation with a system administrator. Interactive work includes, for example, manually typing in shell commands, and should only be done from an interactive job, not a regular non-interactive job.
    b. Grace's Web Portal has a number options for graphical interactive jobs, such as Jupyter Lab/Notebook, R Studio Server, or a Linux desktop.
4. Users shall accurately estimate the resources needed for a job, such as number of cores, amount of memory per core, wall clock time needed to run the job, etc.
    a. Failure to accurately estimate these resource requests, especially underestimating them, could lead to problems both for you and others in running jobs. Jobs that are exceeding their requested resources can be immediately terminated by the system administrators, to avoid damaging the system.
    b. Inaccuracies in resource requests also can lead to inefficiencies in the scheduler, which potentially impacts everyone.
    c. Users who consistently overestimate their resource requests, essentially reserving large blocks of resources with a job, but then not using them, could be penalized, for example by having the priorities of all of their jobs lowered. Flagrant violators, particularly users who 'camp' on an interactive job (e.g. Jupyter Lab or Notebook) without actively using it may have their accounts temporarily suspended until they have a conversation with a system administrator.
5. Users may not log into any of the management nodes by any means for any purpose without permission from a system administrator.
    a. Users caught doing so may have their Grace accounts temporarily suspended until they have a conversation with the system administrator.
6. Grace's storage system (e.g. /home, /scratch, /storage) is not an appropriate place to archive or permanently store data or programs.
    a. Data or programs that are not in routine use by running jobs should be offloaded by the user to other storage, such as ROSS (the Research Object Storage System), the Research NAS, a user's own workstation, lab or departmental storage, or even to cloud storage, such as Box.
    b. Keep in mind that Grace's storage system, though highly redundant and quite reliable, is not backed up. The system operators take no responsibility for any data left on Grace that hasn't been backed up by the user, or that cannot be recovered by external means.
7. The system administrators have the option to remove data in any directory on Grace's storage systems that has not be accessed for 4 months or more (14 days if in /scratch, /local-scratch or /tmp).
    a. The data or programs in a user's or a group's home directory, up to 1 TB, is exempt from this rule and may be kept indefinitely.
    b. Users or groups that maintain more than 1 TB in their home or group directory may be asked to remove data to come within the 1 TB exemption limit. If the user or group does not drop usage to less than 1 TB after the request from the system administrators, then the 4 week rule will apply to that user's or group's home directory.
    c. If you have a special project that requires that data be held on Grace's internal storage systems for more than 4 months, please submit a proposal in writing to the system administrators (e-mail to hpcadmin@uams.edu is sufficient) , detailing:
        i. Succinctly, what the project is and why the space needs to be held on Grace's internal storage for more than 4 months.
        ii. A short name for the project or group that will be used as the name of the top level directory where this data will reside.
            1. For a user-specific project, the top level directory is the user's home directory.
            2. For a group project, the top level directory is the group's shared directory, typically found in /storage.
        iii. How much space the project anticipates needing.
        iv. For how long the project or group anticipates keeping the space before archiving it elsewhere.
        v. What is the backup or archiving plan for that data. (The plan could be, 'this data does not require backup nor archiving.')
    d. Requests for additional space or longer storage durations will be reviewed, with possible further negotiations, before being approved. Please wait for an approval notice before exceeding the space limitation or expecting longer durations. Once approved, only data in the project directory (the short name) will be exempted from the 1 TB rule.
    e. Users who attempt to circumvent the space use policies, for example by creating bogus accounts, projects or groups, or by running scripts that 'touch' files to make them look like they are being used, when really they are not in active use, may have their accounts suspended until after they have a conversation with a system administrator.
8. You should never put sensitive information, such as fully identified patient records, onto Grace or any of her storage systems. Grace is not considered a HIPAA or FERPA repository.
9. Similarly, you should never attempt to access sensitive data on the UAMS network via Grace. Such activity would be reported, and could result in disciplinary actions. If you attempt unauthorized access to data protected by HIPAA or FERPA regulations, there could also be criminal charges brought by the government. Bottom line – just don't do it.
10. You agree to maintain the confidentiality of any information that you may encounter that does not belong to you. If the data is not yours, you may not disclose it to anyone without permission.
11. Although the upper limit to the number of jobs that slurm can comfortably manage is large, it is not infinite. Please check with a system administrator before submitting a bolus of more than 2,000 jobs within a 24 hour period, so that the system administrator can check to make sure sufficient resources are available. Also be aware that you could easily overwhelm the slurm scheduler if you submit jobs too rapidly, which could impact other users.

    a. If you are submitting a large number of related jobs, please consider using a Job Array to submit the entire set at once, rather than submitting all of the jobs individually. Job Arrays are much easier to administer and monitor, and often are more efficient than submitting individual jobs. Grace is configured to handle millions of tasks in a job array.
12. The system administrators do have the right to look into your directories and jobs as needed to assure the smooth operation of Grace. They may suggest potential improvements or changes to the way you work, either to help your jobs be better "HPC Citizens" or to improve their efficiency. Since Grace is a shared resource, it is important to use the system gracefully (pun intended). Please do not feel offended if a system administrator approaches you. They are only trying to help you complete your work in a fashion that is fair to everyone.
13. If Grace is misbehaving, please let the system administrators know. E-mail to hpcadmin@uams.edu is currently the best way to reach the administrators. They cannot fix things that they do not know about.
14. The HPC system administration staff certainly is willing to answer questions and to assist with small problems, as long as such questions are infrequent. But the system administrators do not have the bandwidth to do major work for you, or to answer dozens of tiny questions whose answers could be found by simply reading the documents or searching the internet. Being largely 'self-service', Grace's users are expected to do their own research, do their own programming, do their own optimizing, and manage their own education on how to use an HPC. Users should only bother the system administrators if there is a problem with the system, and occasionally when the user is stuck or needs a little hint.
15. If you need extensive assistance, there may be a need to set up a research partnership or project, e.g. with faculty or staff in the Department of Biomedical Informatics, or in the UAMS IT department. Obviously such partnerships or projects often require a financial arrangement to cover people's time.

Some general information about Grace that might be helpful:

- If you are not familiar with HPC scheduling and the various parameters that determine which job goes next, Dr. Tarbox gave a presentation at the Biomedical Informatics Seminar on March 15th which gives an introduction to HPCs and scheduling that might be helpful. We encourage anyone not familiar with how an HPC such as Grace operate to listen to the presentation.
- Periodically the HPC team holds training sessions on various aspects of HPC use and scientific computing. Please check the web site (https://hpc.uams.edu) for announcements.
- There are also a lot of web resources available to learn how HPC operate, and how to effectively use them. Just moving a program from your favorite laptop to Grace is not going to give very satisfactory results. A researcher often has to rethink and refactor the problem to make effective use of an HPC.
- The easiest way to access Grace is through her web-based, Open OnDemand portal, found at https://portal.hpc.uams.edu. Open OnDemand allows you to get a web-based command line interface with Grace, manipulate files in your home directory, and submit and monitor job scripts. Open OnDemand also has the capability to launch interactive jobs, for example using Jupyter Lab or Notebooks, or RStudio Server. It is a work in progress, so more interactive options may appear from time to time.
- You may need to log into Grace at least once by ssh to set your password before using Open OnDemand. It may be possible to manually reset your password via Open OnDemand by launching a terminal window and using the "passwd" command at the prompt.
- Many researchers access Grace by using ssh directed to login.hpc.uams.edu. Note that the login nodes are only accessible inside the UAMS firewall. If you need to access the login nodes (e.g. via ssh) from outside of the UAMS network you will need to first set up a VPN connenction into the UAMS network.
- Over the October 2, 2020 weekend Grace was switched from the Moab/Torque scheduler/resource manager to slurm. The slurm website includes a table that shows the mapping between slurm commands and the Torque commands. Some, but not all, of the old Torque commands (e.g. qsub) are emulated in slurm, but may have somewhat different behavior from the Torque variant. Switching to slurm, makes Grace operations more consistent with HPC services at the University of Arkansas in Fayetteville (e.g. the Pinnacle cluster).
- The scheduler/resource manager works best if your wallclock, memory, and processing core requests are close to real. If you are unsure what to use, please try one or two small sample jobs, then check what the actual runtime, memory, and core usage were, and use that to guide your estimates. Note that your job priorities are penalized if you consistently overestimate these values.
- The compute nodes are outfitted with the OpenHPC libraries. Many options require issuing a *module* command to load appropriate versions of software libraries before use.
- The compute nodes are also outfitted with Conda. Conda environments are a good option for dealing with program dependencies, regardless of the language a program is written in (i.e. conda is not just for Python or R).
- If you need a special software configuration (i.e. the OpenHPC libraries or Conda are not enough) we recommend using Singularity containers. Most Docker containers can run under Singularity, and there are a lot of Singularity and Docker containers available on their respective hubs. If you can't find a suitable container, you can choose a base container, and then configure it to your liking on your local workstation before loading the container to the HPC.
- Each HPC user gets a home directory, which is visible to all of the compute nodes and the login node, with a 1 TB quota. Users who consistently exceed their 1 TB quota without permission from the system administrators may find that they can no longer run jobs due to a lack of disk space.
- Users may temporarily store data in the /scratch directory, which is shared across all nodes. Note that data in the /scratch directory is not considered permanent, and could be purged of older data at any time. So do not count on keeping things in /scratch for more than 14 days after a job ends. Hence the strong recommendation that users utilize data staging options to copy input data to /scratch before the job starts, then copy output data to outside storage (e.g. the research NAS, the Research Object Storage System (ROSS), the user's personal computer, or a cloud storage provider) at the end of a job.
- Each compute job also has the option to use local scratch storage (/tmp or /local-scratch) on each node. Note that local scratch areas are not shared between nodes (i.e. they are on a disk local to a particular compute node). Their big advantage over /scratch is that accessing local scratch storage incurs no network overhead, so can be faster than /scratch if a program is doing lots of small, random reads and writes. The shared /scratch area may be faster for long sequential reads and writes due to the large block size and parallelism on the shared storage system. If you use local scratch, be sure to save output delete what you put into it when done.
- A good habit to get into is to stage (move) data into a scratch area (local or shared, as needed) prior to the start of a job, and to copy data out of the scratch areas after the end of the job. We cannot emphasize this enough.
- There is a shared area of library files (files common to a number of projects) currently called /storage. (We may rename it.) Please do not store personal files there. At some point in the future, /storage will be made read-only, and personal files will be moved to the user's home directory, or into a group's shared directory space, potentially impacting that user's or group's quota and ability to run jobs.
- Projects or labs may contact the HPC Administrators to set up shared project-specific area accessible by all compute nodes. There may be a fee for this service, dependent on the amount of storage reserved (i.e. the quota requested).
- UAMS has bulk campus storage that is available to researchers often for a fee. The research campus storage facilities are accessible to the login node, as well as the data transfer nodes that can used stage data for jobs, but are not accessible to the compute nodes (i.e. cannot be accessed inside a compute job). The bulk campus storage (e.g. ROSS) is not designed for I/O intensive access by the compute nodes, hence a user is better off staging the data to/from bulk campus storage and Grace's cluster storage for use inside a compute job.
- There are three categories of processors, selectable via feature requests when submitting jobs, in Grace:

    - **Xeon Phi** (phi) – ideal for massively parallel operations that have lots of threads. Each Xeon Phi node has 64 physical cores and either 384 GB of memory (80 of the Xeon Phi nodes), or 192 GB of memory (16 of the nodes). Although the clock speed of each individual processor is slower than a regular xeon, since there are more than double the number of cores on a Xeon Phi compared to regular Xeon, appropriately configured multi-threaded jobs can run faster on the Xeon Phi. A program optimized for Xeon Phi can gain significant speed advantage over regular Xeon. Optimized programs also typically will run faster on regular Xeon as well, though the speedup is not as dramatic. Programs that are not optimized for Xeon Phi processors often suffer poor performance, unless they are highly mulit-threaded or can be broken down into lots of parallel tasks.
    - **Regular Xeon** (xeon) – use where single thread performance is critical. Each regular Xeon node has 28 physical cores and 128 GB of memory. Unoptimized programs generally do better on regular Xeon.
    - **Nvidia Tesla GPU nodes** (gpu) – for jobs that can capitalize on GPU performance. Each node has 24 regular Xeon cores, 128 GB of memory, and a pair of 12GB P100 GPUs. We have Nvidia CUDA 10 drivers installed on the GPU nodes. The GPU nodes are in a special class (a.k.a. queue) named “gpu”, separate from the other nodes. So when submitting jobs that depend on the GPU, make sure to select the “gpu” class, not the default “batch” class. Otherwise your job could sit in limbo asking for GPU resources on nodes that don’t have them.
- Generally, if your program is not geared specifically for one of the above categories, it is best to just let the scheduler decide either by using the "any" partition, or by not specifying the partition (i.e. don’t put in a feature request for a specific class of system), so that the job gets routed to the first available processor.