Oak Ridge National Laboratory

Introduction to Andes

OLCF Data Analysis and Visualization Cluster

Leah Huk

User Assistance – Production

August 2022

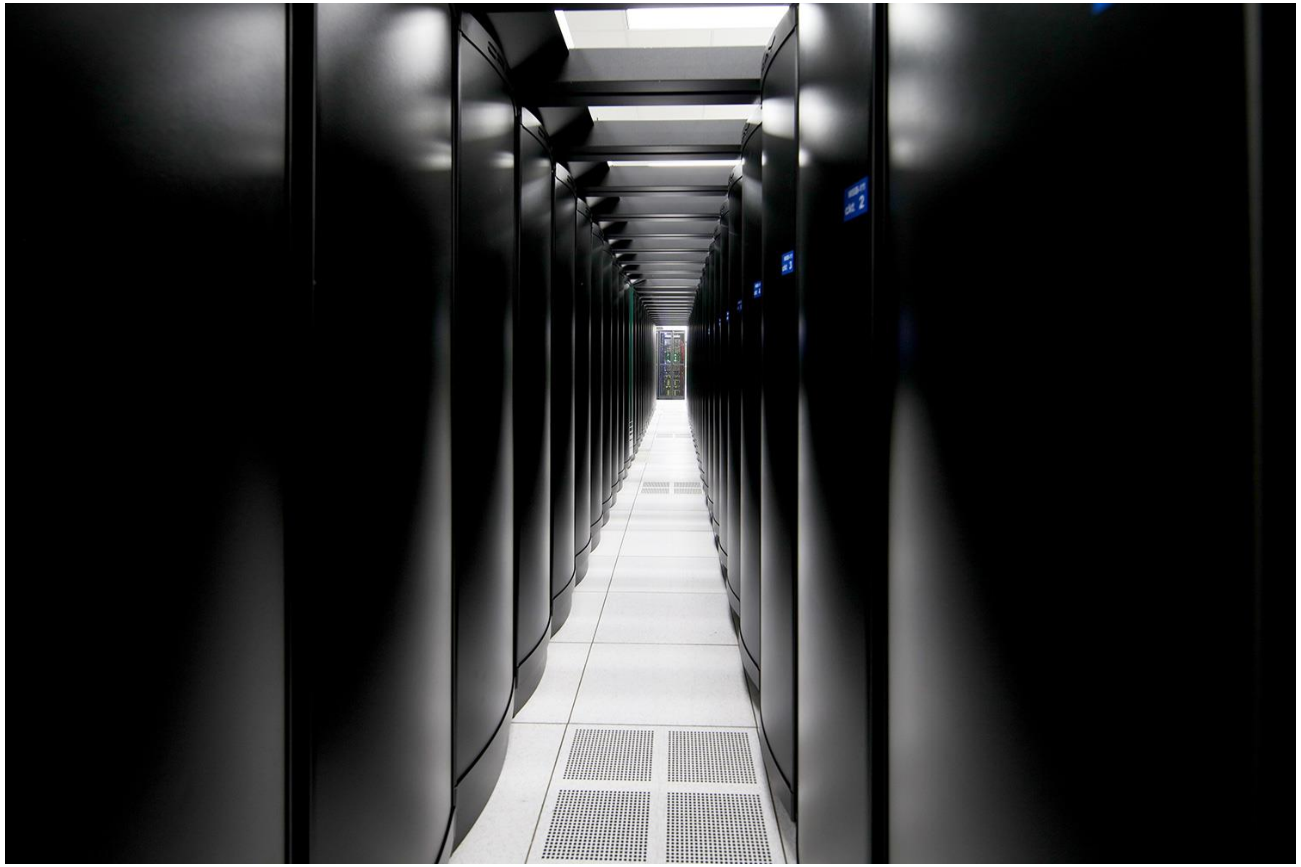
ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

What is Andes for?

Andes prioritizes small jobs and schedules in such a way that there should always be nodes open for the work of smaller tasks

"Andes is a 704-compute node commodity-type linux cluster. The primary purpose of Andes is to provide a **conduit for large-scale scientific discovery via pre/post processing and analysis of simulation data generated on Summit**."



What type of tasks can I do on Andes?

Any type of ancillary analysis or processing work that does not require the large-scale leadership class mandate resources on Summit are ideal for running on Andes!

- **Preprocessing**: cleaning, concatenating, or reshaping large data, or otherwise preparing input for a large-scale run, etc.
- **Postprocessing**: running scripts to analyze, reshape, or restructure model output.
- **Data Visualization**: Running VisIt, Paraview, TurboVNC. Creating plots and videos using scripts written in R, Python, Julia, etc.
- Debugging, troubleshooting, and lots more!

System Overview: Connecting to Login Nodes

To connect to Andes, ssh to andes.olcf.ornl.gov using your OLCF username and Passcode (PIN+Tokencode)

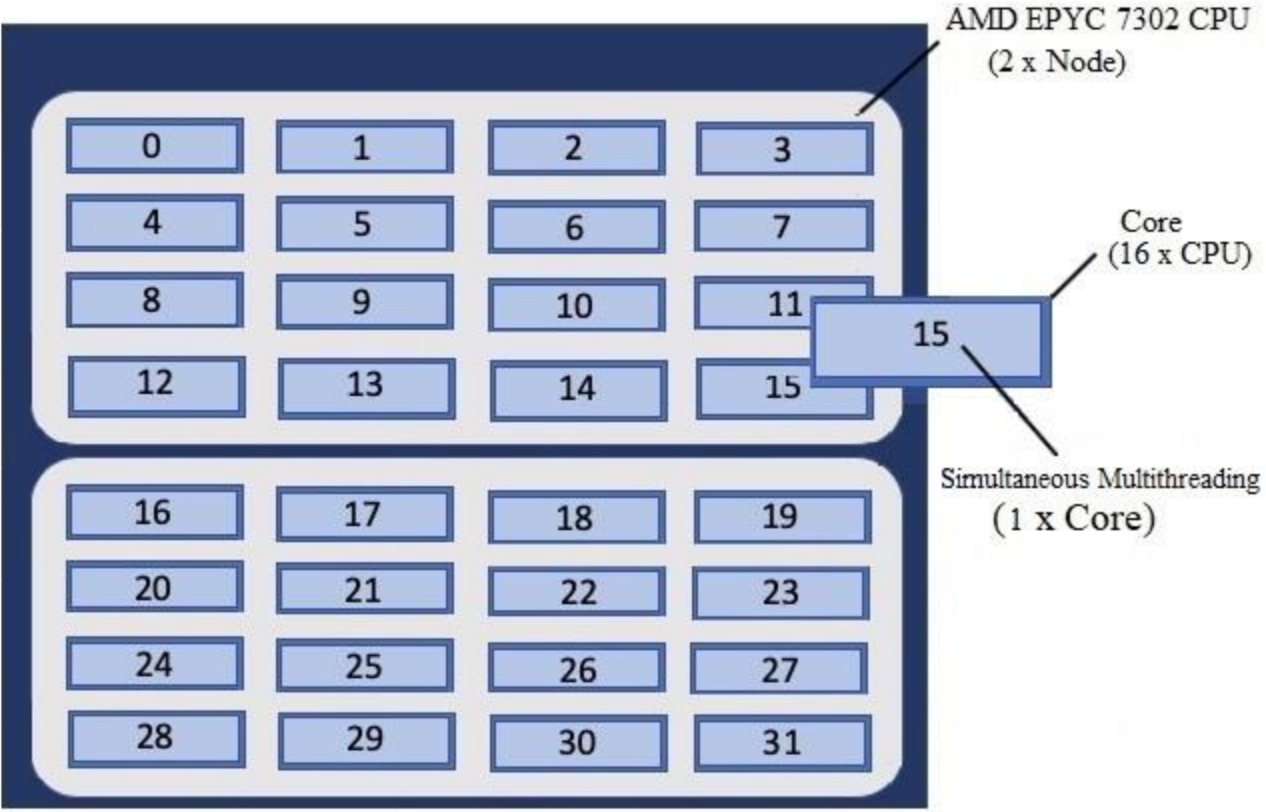
```
$ ssh username@andes.olcf.ornl.gov
```

- There are 8 login nodes (identical to the batch partition compute nodes) which provide an environment for editing, compiling, and launching codes onto the compute nodes.
- All Andes users access the system through these login nodes, and as such, any CPU- or memory-intensive tasks on these nodes could interrupt service to other users.

PLEASE REFRAIN FROM DOING ANY ANALYSIS OR VISUALIZATION TASKS ON THE LOGIN NODES.

System Overview: Compute Nodes

Partition	batch (default)	gpu
Node Count	704	9
Memory	256 GB	1 TB
GPU	N/A	[2x] NVIDIA® K80
CPU	[2x] AMD EPYC 7302 16Core Processor 3.0 GHz, 16 cores (total 32 cores per node)	[2x]Intel® Xeon® E5-2695 @2.3 GHz 14 cores, 28 HT (total 28 cores, 56 HT per node)



System Overview: File System

- Users can access the moderate security enclave NFS, Alpine, and HPSS filesystems from Andes.
- These are the same filesystems accessible on Summit, making paired analysis from work on Summit more convenient.

	Area	Path	Permissions	Quota	Backups	Purged	Retention	On Compute Nodes
NFS	User Home	/ccs/home/[userid]	User set	50 GB	Yes	No	90 days	Read-only
	Project Home	/ccs/proj/[projid]	770 (rwxrwx---)	50 GB	Yes	No	90 days	Read-only
Alpine	Member Work	/gpfs/alpine/[projid]/scratch/[userid]	700 (rwx------)	50 TB	No	90 days	N/A	Read/Write
	Project Work	/gpfs/alpine/[projid]/proj-shared	770 (rwxrwx---)	50 TB	No	90 days	N/A	Read/Write
	World Work	/gpfs/alpine/[projid]/world-shared	775 (rwxrwxr-x)	50 TB	No	90 days	N/A	Read/Write
HPSS	Member Archive	/hpss/prod/[projid]/users/[userid]	700 (rwx------)	100 TB	No	No	90 days	No
	Project Archive	/hpss/prod/[projid]/proj-shared	770 (rwxrwx---)	100 TB	No	No	90 days	No
	World Archive	/hpss/prod/[projid]/world-shared	775 (rwxrwxr-x)	100 TB	No	No	90 days	No

Shell and Programming Environment

- **Shells**: bash (default), tsch, csh, ksh. You can request a change in shell by submitting a ticket to help@olcf.ornl.gov.

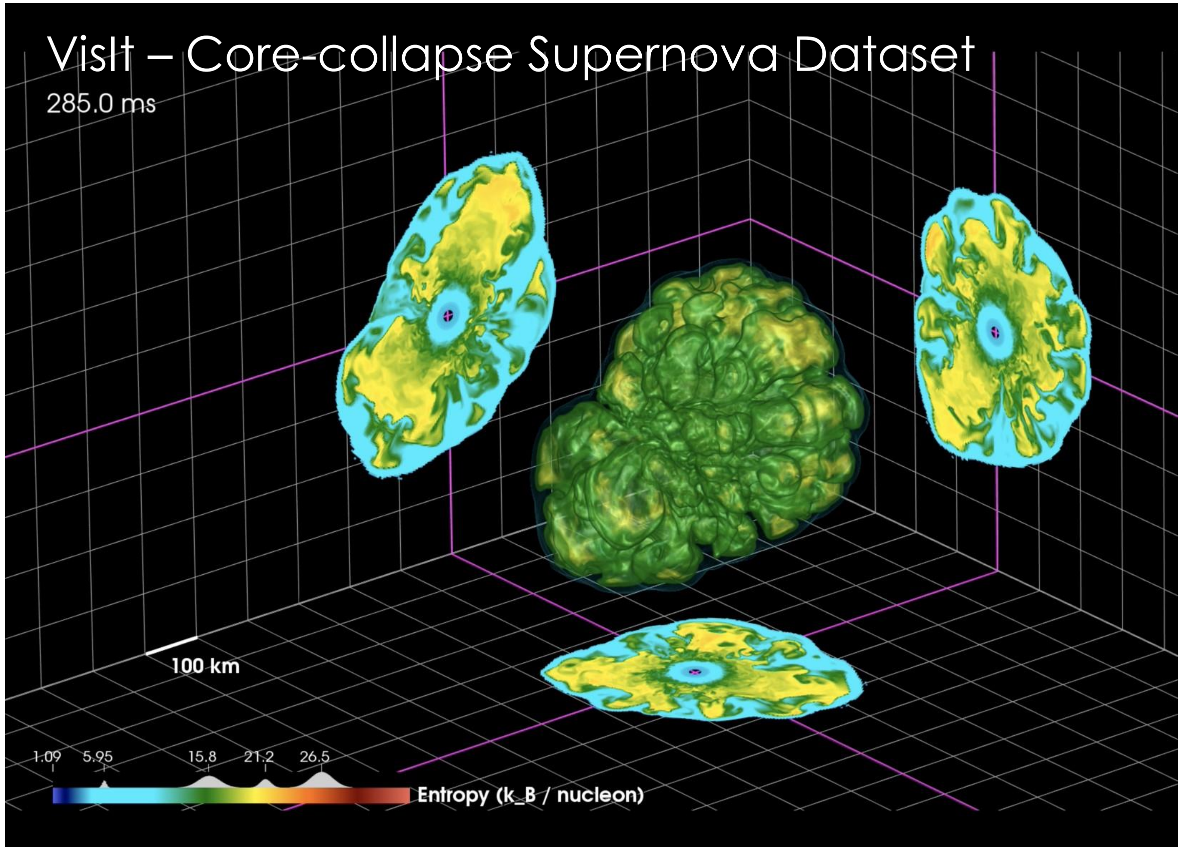
- **Lmod**:Use the `module` command to manage your modules

Command	Description
module -t list	Shows a terse list of the currently loaded modules.
module avail	Shows a table of the currently available modules
module spider <string>	Searches all possible modules according to <string>
module load <modulename> […]	Loads the given <modulename>(s) into the current environment
module help <modulename>	Shows help information about <modulename>
module show <modulename>	Shows the environment changes made by the <modulename> modulefile
module use <path>	Adds <path> to the modulefile search cache and MODULESPATH
module unuse <path>	Removes <path> from the modulefile search cache and MODULESPATH
module purge	Unloads all modules

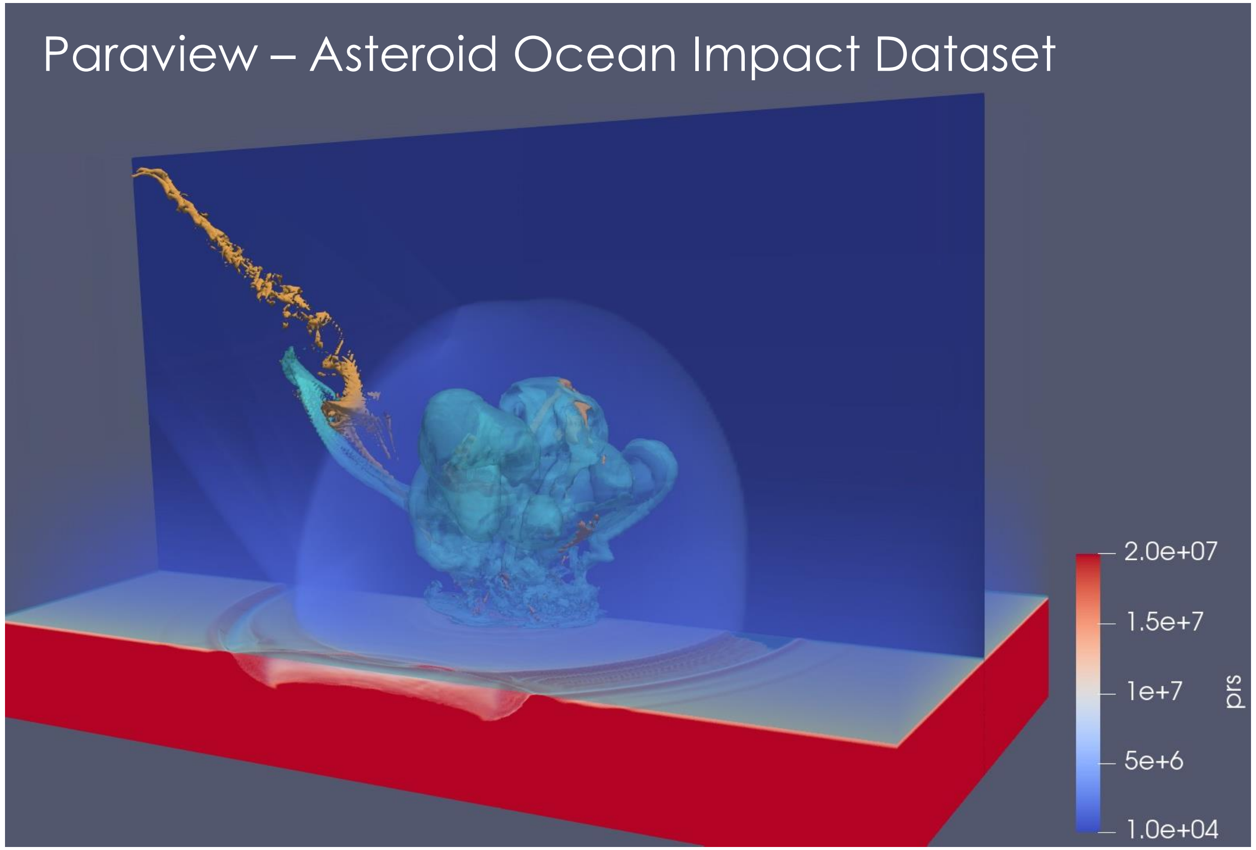
- **Software**: Type `module avail` to see all available software. A definitive guide to Python on Andes can be found in our docs.

Visualization with Paraview and Visit

VisIt and Paraview are interactive, parallel analysis and visualization tools for exploring large scientific datasets. Both are available on Andes



VisIt – Core-collapse Supernova Dataset



Paraview – Asteroid Ocean Impact Dataset

See the Software -> Visualization Tools section of the docs for in-depth guides on installation and use

Python on Andes

Andes is the best place to run Python

- It is the easiest place to install if you need customization
- It plays nicely with the Slurm scheduler

```
$ module avail python

------------- /sw/andes/modulefiles/core ----------------------
   python/3.7-anaconda3

$ module load python
$ conda list
```

If a package you require isn’t available, you can create and load a custom environment and install yourself.

Step by step instructions on custom environments and running python on Andes are in the docs under Software -> Python

Compilers

The **intel** (default), **pgi**, and **gcc** compilers are available on Andes, and can be loaded or changed using the `module` command

```
$ module avail intel
---------------------- /sw/andes/spack-envs/base/modules/site/Core -----------------------------------
        intel/19.0.3 (D)    intel/2021.4.0
$ module avail pgi
---------------------- /sw/andes/spack-envs/base/modules/site/Core -----------------------------------
        pgi/19.10
$ module avail gcc
---------------------- /sw/andes/spack-envs/base/modules/site/Core -----------------------------------
        gcc/6.5.0    gcc/9.3.0 (D)    gcc/10.1.0    gcc/10.3.0
```

The following wrapper programs are cognizant of your currently loaded modules:

- `mpicc` to invoke the c compiler
- `mpicc`, `mpicxx`, or `mpic++` to invoke the c++ compiler
- `mpif77` or `mpif90` to invoke appropriate versions of the fortran compiler

Compilers, cont.

When building threaded codes with Openmp, compiler-specific flags must be included to ensure a proper build.

For example, if compiling C code with the various compiler modules, add the following compiler specific flags:

- For pgi, add “-mp” to the build line.

```
$ mpicc -mp test.c -o test.x
$ export OMP_NUM_THREADS=2
```

- For gnu, add “-fopenmp” to the build line.

```
$ mpicc -fopenmp test.c -o test.x
$ export OMP_NUM_THREADS=2
```

- For intel, add “-qopenmp” to the build line.

```
$ mpicc -qopenmp test.c -o test.x
$ export OMP_NUM_THREADS=2
```

Running Jobs with Slurm

- Only a limited number of simultaneous basic tasks (editing files, compiling codes) should be done on the login nodes. For compute- or memory-intensive tasks, users must submit as jobs to the compute nodes.

- Unlike Summit, Andes uses the Slurm batch scheduler.

Task	LSF (Summit)	Slurm
View batch queue	jobstat	squeue
Submit batch script	bsub	sbatch
Submit interactive batch job	bsub -Is $SHELL	salloc
Run parallel code within batch job	jsrun	srun

- There are two ways to submit to the computes:
  1. Request nodes for an interactive batch job with `salloc`
  2. Write and submit a batch script with `sbatch`

Interactive Jobs

- To submit an interactive job, use the `salloc` command:

```
$ salloc -A abc123 -p gpu -N 4 -t 1:00:00
```

salloc	Start an interactive session
-A	Charge to the abc123 project
-p gpu	Run in the gpu partition
-N 4	request (4) nodes...
-t 1:00:00	...for (1) hour

- The job will wait for nodes to be available. An interactive prompt will appear once the job starts.

- Use the `sbatch --test-only` command to see when a job of a specific size could be scheduled:

```
$ sbatch --test-only -N2 -t1:00:00 batch-script.slurm

sbatch: Job 1375 to start at 2022-08-29T10:54:01 using 64 processors on nodes andes[499-500] in
partition batch
```

Writing a Batch Script

Consider the following batch script

```
#!/bin/bash
#SBATCH -A ABC123
#SBATCH -J RunSim123
#SBATCH -o %x-%j.out
#SBATCH –e %x-%j.err
#SBATCH -t 1:00:00
#SBATCH -p batch
#SBATCH -N 128
#SBATCH --mail-type=ALL
#SBATCH --mail-user=email.address@provider.com

cd $MEMBERWORK/abc123/Run.456

srun -N # –n # –c # –cpu-bind=cores ./a.out
```

Interpreter line
-A - Project to charge
-J - Job Name
-o – stdout, %x=job_name, %j=job_number
-e – stderr
-t – walltime in HH:MM:SS
-p – partition (queue)
-N – number of nodes
Email notifications (ALL, BEGIN, END, etc.)
Email address to notify

Change into the run directory

Run the job

Where the srun command takes the following common options:

`–N [#nodes] –n [#ntasks] –c [#cores_per_task] –cpu-bind=[cores,threads,no]`

Submit your batch script on the command line with `sbatch`:

```
$ sbatch myscript.slurm
```

Monitoring and Modifying Jobs

Command	Action/Task	LSF Equivalent
`squeue`	Show the current queue	bjobs
`sinfo`	Show node/partition info	bqueues or bhosts
`sacct`	Show queued and historical job info. and steps	bacct
`scancel`	Cancel a job or job step	bkill
`scontrol`	View or modify job configuration.	bstop, bresume, bmod

To see all jobs currently in the queue:

```
$ squeue –l
```

To see all of your queued jobs:

```
$ squeue -l -u $USER
```

To see all your queued jobs (plus steps) in the queue

```
$ sacct -u $USER
```

To see all steps submitted to job 123456:

```
$ sacct -j 123456
```

To cancel a job in the queue:

```
$ scancel –j <job_id_#>
```

To place a queued job on hold:

```
$ scontrol hold <job_id_#>
```

To release a held job:

```
$ scontrol release <job_id_#>
```

To see additional details for a job:

```
$ scontrol show job <job_id_#>
```

Queue Policies

There are two compute node partitions: “batch” and “gpu”. Each has different queue policies for the number of jobs that can be run and wall times per number of nodes requested.

Batch Partition:

Bin	Node Count	Duration	Policy
A	1 - 16 Nodes	0 - 48 hr	max 4 jobs running and 4 jobs eligible **per user** in bins A, B, and C
B	17 - 64 Nodes	0 - 36 hr	
C	65 - 384 Nodes	0 - 3 hr	

GPU Partition:

Node Count	Duration	Policy
1-2 Nodes	0 - 48 hrs	max 1 job running **per user**

Note: to access the gpu partition, specify `–p gpu` in your script or interactive node request.

Questions?

- Full documentation for using Andes can be found here:
  https://docs.olcf.ornl.gov/systems/andes_user_guide.html
  https://docs.olcf.ornl.gov/software/python/index.html

- Trouble? Submit a help-desk ticket to help@olcf.ornl.gov