Vermont Advanced Computing Center

UVM Research Week Open House 2024

Research Computing Organization

Mike

Chris

Andi

Pat

- UVM Chief Technology Officer: Mike Austin
- VACC Director: Chris Danforth
- VACC Director of Operations: Andrea Elledge
- UVM Director, Large Research Initiatives: Patrick Clemins
- Research Computing Team (see next slide)

Research Computing Team

- Team Lead: Jim Lawson
- Systems Engineers
  - Bennet Fauber
  - Travis Bartlett
  - Katy Czar
  - Mac Mansfield-Parisi
- Research Facilitators
  - Shelly Johnson
  - Terry Barrett

Katy

Terry

Travis

Mac

Shelly

UVM Research Computing Framework

After the computing needs of a researcher or student outgrows their personal machine and/or their research group's current computing resources:

- The VACC is the default research computing environment and should be used whenever possible for all compute needs.
- UVM's Virtual Server Environment should be used in those cases where the VACC does not meet a researcher's needs.
- https://www.uvm.edu/it/research-computing

Research Computing

https://www.uvm.edu/it/research-computing

VACC Clusters

BlueMoon
200 CPU Nodes
8340 Cores

DeepGreen
10 GPU Nodes
80 x NVIDIA V100

BlackDiamond
6 GPU Nodes
48 x AMD Radeon M150

DataMountain
Large memory cluster
64 TB RAM

MPI workloads AI/ML workflows Large scale

VACC Services

High-Performance Computing
Onboarding
Programming Assistance

Virtual Machine Cluster

CPU
10 x AMD EPYC 75F3
320 cores @ 2.95 GHz

Memory
5 TB

GPU
2 x NVIDIA A100 40 GB

Smaller scale Prototyping Scalable

Network Storage

NetFiles
2.4 Petabytes
(2400 Terabytes)

ETS Services

Virtual Machine Hosting
Network Storage
Globus
Consulting

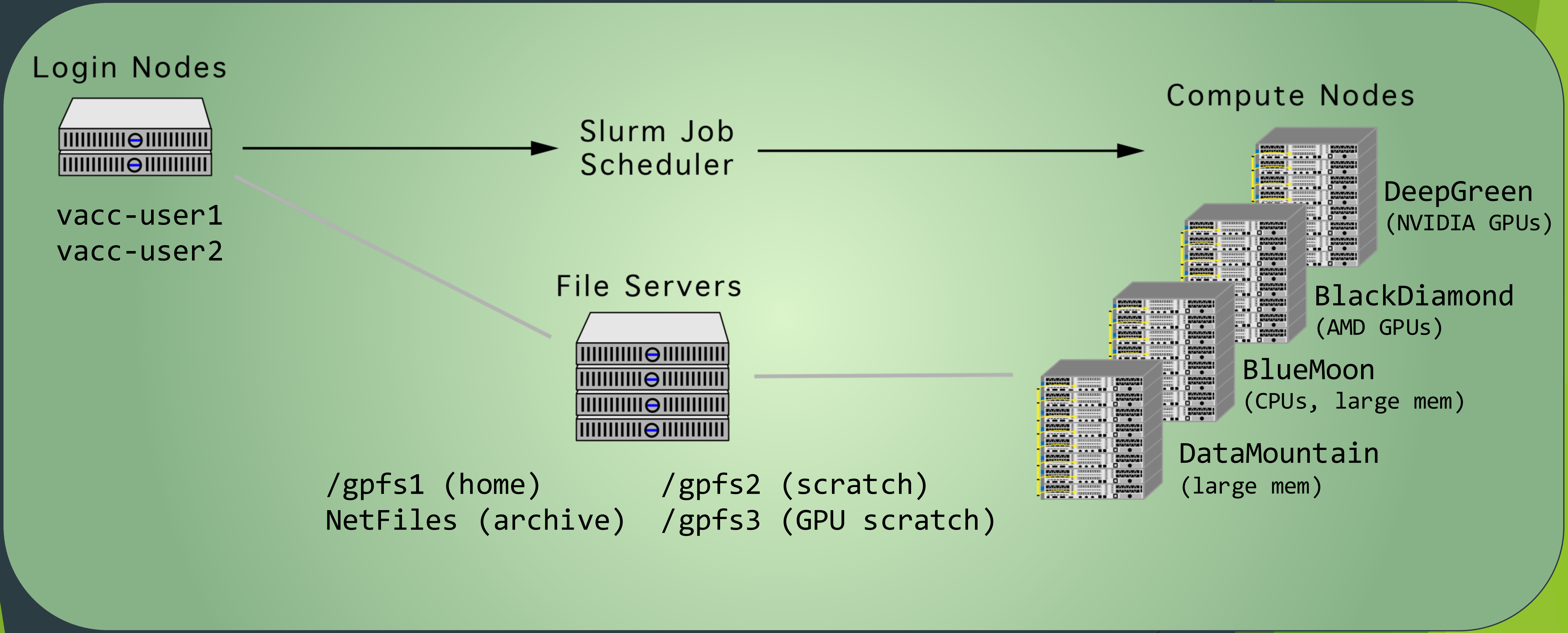
THE UNIVERSITY OF VERMONT
ADVANCED COMPUTING CENTER

THE UNIVERSITY OF VERMONT
ENTERPRISE TECHNOLOGY SERVICES

High Performance Computing

- A High Performance Computing (HPC) cluster is a collection of many computers networked together
- Each computer is called a node
- HPC clusters run computations managed by a scheduler program (we use Slurm)
- The scheduler keeps track of available resources, allowing job requests to be efficiently assigned to compute resources
  - memory, processing power (cpu or gpu), time, or some combination

The VACC Cluster



Research VMs

- While the VACC strives to be very flexible and meet as many researchers needs as possible, sometimes something different is needed.
- UVM has had a “private cloud” VMware virtualization infrastructure for many years. We have recently significantly expanded that environment with new compute nodes to be able to host virtual servers (Windows, Linux) for researchers who have needs where they would normally want a dedicated research workstation or cluster.
  - Each node in new cluster has 64 processor cores, 2TB RAM, 25Gb networking, access to significant storage resources. GPU resources for compute and graphics are also available.
- One free VM available to all PIs (12 CPU cores, 48 GB RAM, 100 GB storage)
  - Scalable with modest cost
  - Great alternative to dedicated physical hardware

VACC Cluster

- BlueMoon – 200 CPU nodes, 8340 CPU cores, multi-TB RAM nodes, MPI
- DeepGreen – 80 NVIDIA V100 GPUs
- BlackDiamond – 48 AMD Radeon M150 GPUs
- DataMountain - Large memory cluster 64TB RAM, sharded MongoDB cluster

Apps to connect to VACC

- SSH through a text interface
  - Command Prompt on Windows (or other terminal clients: PowerShell, PuTTY, etc)
  - Terminal on Mac, *nix, and others
- Web browsers
  - Open OnDemand https://vacc-ondemand.uvm.edu/
- Interactive Development Environments (IDEs)
  - Visual Studio Code
  - PyCharm

Node Types

Login Node

When you log onto a cluster you land on a login node. Edit scripts, run short tests, and submit jobs.

```
[ <netID>@vacc-user1 ~] $
```

Rough guidelines: If testing, the test should run for less than 5 minutes, use less than 8 GB of memory, and use 4 CPUs or fewer.

If more resources are needed, use a batch job, an interactive job, or Open OnDemand.

Compute Node

Compute nodes are where all the heavy lifting is done. Once on a login node, you can submit your jobs either in batch form or interactively to the compute nodes.

Slurm workload manager

All jobs are run on the compute nodes using the
Slurm Workload Manager

Slurm needs to have the following information about what you want to run

- The partition (group of nodes) to use
- Number of CPUs
- Amount of memory
- Amount of time needed to run
- Whether you need a GPU, and if so, how many

You will need to specify theses values when starting a job

For further information about Slurm, see our Knowledge Base:

https://www.uvm.edu/vacc/kb/knowledge-base/submit-a-job/

Modules on VACC

- Modules are used to modify your environment to access software packages installed on the cluster

- Modules can be loaded on the command line or in your batch script
  - It is preferred to load them in your batch script to have a record of which modules were loaded if there is a problem with your job

Load Software on VACC

- Modules:
  - `module load R/4.3.2`
- Spack:
  - `spack load bowtie@1.2.3`
- Conda:
  - `conda activate <environment_name>`

See details in next slides

Load Software on VACC: MODULES

Show which modules are available:

```
$ module av
```

Load a module and list the loaded modules (including dependencies):

```
$ module load Rtidyverse/4.3.2
$ module load rstudio
$ module list
```

Unload modules:

```
$ module unload Rtidyverse rstudio
$ module purge
```

Load Software on VACC: Spack

Show which packages are available:

```
   $ spack find
```

Load a package and list the loaded software (including dependencies):

```
   $ spack load bowtie@1.2.3
   $ spack find --loaded
```

Unload a Spack package:

```
   $ spack unload bowtie
```

Load Software on VACC: Conda

Load conda:

```
    $ module purge
    $ module load python10-anaconda/2023.03-1
    $ source ${ANACONDA_ROOT}/etc/profile.d/conda.sh
```

Show which environments you have installed:

```
    $ conda info -e
```

Activate the environment and list the loaded packages (including dependencies):

```
    $ conda activate <environment name>
    $ conda list
```

Exit the environment:

```
    $ conda deactivate
```

Open OnDemand

- https://vacc-ondemand.uvm.edu/
- Browser-based access to VACC resources
- File management
- Job management and monitoring
- Command-line shell access
- Choose between a Linux Desktop or any of the following interactive apps
  - Python via Jupyter Notebook / JupyterLab
  - R via RStudio
  - MATLAB

Guide to using the VACC Open OnDemand

https://www.uvm.edu/vacc/kb/knowledge-base/ondemand/

Creating a Batch Script for Slurm

- A batch script is the script used to submit a job to the cluster. Batch scripts end with either *.sbat* or *.sh*
- The example shown is a simple batch script that loads a Python module and runs a Python script
- Modules should be loaded after the #SBATCH lines and before any commands

```
#!/bin/bash
#SBATCH --job-name=example_job
#SBATCH --time=00:15:00
#SBATCH --ntasks=2
#SBATCH --mem=8G
#SBATCH --nodes=1
#SBATCH --account=<account_name>
#SBATCH --partition=bluemoon
#SBATCH --mail-user=<netID>@uvm.edu
#SBATCH --mail-type=BEGIN,END

module purge
module load python3.11-anaconda
module list

python my_python_script.py
```

Submitting a Job to the Cluster

Consider using the `sinfo` command to check partition status

Check your that your script specifies all required resources, the account and partition are accurate, and all of your modules loaded

To submit your job to the cluster

```
    $ sbatch <script_name>.sbat (or.sh)
```

Check the status of your job

```
    $ squeue -u <user>
```

```
 JOBID PARTITION     NAME     USER  ST    TIME  NODES NODELIST(REASON)
325511  bluemoon  testjob  mjohns89  R    0:00      1 node200
```

Helpful Commands

```
my_accounts                  : list your cluster accounts
groupquota                   : show the filesystem quota for your group and users in that group
my_job_statistics <jobID>    : show resources used for a completed job
squeue --me                  : view info about any jobs you have in the queue
scancel <jobID>              : cancel a job
```

Research storage

- Cluster storage - fastest storage for computing at scale
  - gpfs1 - Primary, backed up
    - Up to 20TB / 6M files per research group
  - gpfs2 - Scratch
    - Up to 35TB / 8M files per research group
  - gpfs3 - GPU scratch
    - Up to 20TB / 8M files per research group

Research storage

- Netfiles – multi-PB storage solution for researchers
  - 10TB for free for each faculty lab, can self-provision
    - Scalable for modest cost. Some PIs have 100s of TBs
  - Daily backups for 30 days, replicated between multiple datacenters
  - Can be accessed from your laptop, virtual machines, and the VACC
  - Easy to share with others at UVM or with external collaborators using Globus
  - Self-service webpage available at https://research-storage.uvm.edu/

Data transfer and sharing

- Globus https://www.uvm.edu/it/kb/article/globus-data-transfer-service/
  - Web-browser based large-scale (many TB) data transfers between UVM systems and external research organizations
  - We provide Globus endpoints for VACC home and scratch, Netfiles shares, and UVM OneDrive
  - Supports sharing of data with external collaborators
  - Unattended/scheduled transfers
- scp - command line file transfer
  - Copy a file from your computer to the cluster

    ```
    scp myfile.sh username@vacc-user1.uvm.edu:myfile.sh
    ```
  - Copy a file from the cluster to your computer

    ```
    scp username@vacc-user1.uvm.edu:myfile.sh myfile.sh
    ```
- Open OnDemand - browser-based graphical user interface
- FileZilla - Fast, cross-platform SFTP Client with graphical user interface
- Rsync - fast remote (and local) command line file-copying tool
  - Incremental updates, good for backups, as well as everyday copying

Help with Research Computing

- How our team supports you
  - Help you figure out how to store and protect your research data
  - Assistance migrating and scaling-up desktop computing to the cluster environment
  - Optimizing use of resources for fast, efficient processing
  - Troubleshooting blockers
  - Collaborative planning for future needs and research interests

Help with Research Computing

- Research Computing homepage https://www.uvm.edu/it/research-computing
  - Contact Us link on this page
- VACC Knowledge Base https://www.uvm.edu/vacc/kb/
- Open OnDemand (web interface) https://vacc-ondemand.uvm.edu/
- VACC Help requests *vacchelp@uvm.edu*
- Regularly-scheduled virtual drop-in help sessions
  - Teams links for sessions are available at https://www.uvm.edu/it/research-computing/office-hours-workshops
    - Current schedule is one session a week, usually alternating Monday afternoons and Thursday mornings

Exercises - Try on your own and bring back your questions!

- Interactive job using `srun` (see following slides)
  - Start interactive job with a GPU
  - Load required modules
  - Install tensorflow and run test script
- Python - Geopandas map analysis
  - Run as a script in a batch run
  - Run as a Jupyter Notebook in OOD
  - Instructions and code: https://gitlab.uvm.edu/Terence.Barrett/geopandas-example
- Open OnDemand Desktop/Rstudio (see following slides)
  - Start an OOD Desktop job
  - Open a terminal and load required modules
  - Launch Rstudio and generate a plot

Example Interactive Job using srun

Start an interactive job with the Slurm `srun` command:

```
 -p, --partition=<partition>  -N, --nodes=<node_count>
 -n, --ntasks=<number>        -t, --time=<time>
```

```
$ srun -p dggpu -N 1 --ntasks=1 -t 01:00:00 --mem=8G --gpus=1 --pty /bin/bash
```

Load Anaconda, cuda, and cudnn modules:

```
$ module purge
$ module load python3.11-anaconda cuda/12.2.2  cudnn/12.2-v8.9.6
```

Install TensorFlow and run test script:

```
$ pip install --user tensorflow
$ python /gpfs1/sw/examples/tensorflow/test-tf.py
```

It should print the number of GPUs and output the answer:

```
[[4 6 8]
 [4 6 8]]

$ exit
```

Example Open OnDemand Desktop Job

Go to UVM VACC Open OnDemand: https://vacc-ondemand.uvm.edu/
If not connected to UVM network, you must use UVM's VPN.
Under the Interactive Apps menu, select `Desktop`.

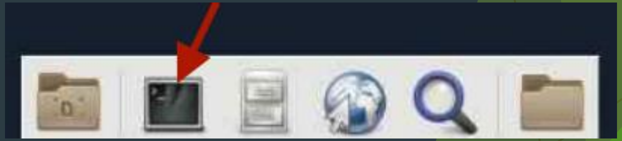
Use the following form completions:

Slurm Account	leave blank	Partition	Bluemoon (short)
Number of nodes	1	Number of hours	1
Cores per node	2	Memory per node	8

Click blue Launch button at bottom of the form. This will open the Interactive Sessions page.

When your job is ready, probably only after a short delay, click the blue Launch Desktop button.

Open a terminal by clicking the black icon at the bottom of the page



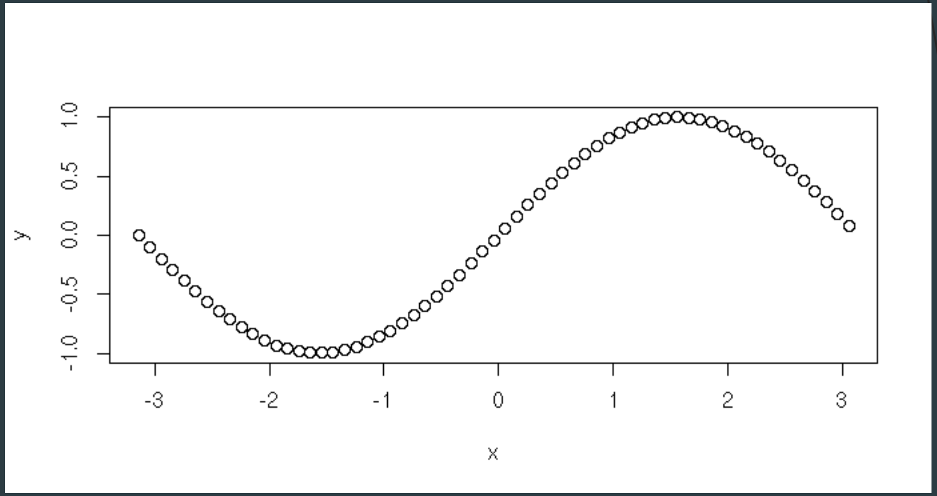
Load modules and launch Rstudio by typing the following into the terminal after the prompt (`$`):

```
$ module load Rtidyverse/4.3.2
$ module load rstudio/2023.09.1
$ rstudio &
```

Example Open OnDemand Desktop Job

When Rstudio opens, enter the following into the Console in the bottom-left panel after the > prompt:

```
> x=seq(-pi,pi,0.1)
> y=sin(x)
> plot(x,y)
```



As you define the x and y values, they will be stored in your environment showing in the top-right panel. When you enter the plot command, the sine wave plot will display in the Plots tab of the lower-right window.

VACC OOD KnowledgeBase: https://www.uvm.edu/vacc/kb/knowledge-base/ondemand/