Introduction to HPC Using zcluster at GACRC

Georgia Advanced Computing Resource Center
University of Georgia
Suchitra Pakala
pakala@uga.edu
Slides courtesy: Zhoufei Hou
OVERVIEW

- GACRC
- High Performance Computing (HPC)
- zcluster – Architecture, Operation
- Access and Working with zcluster
Georgia Advanced Computing Resource Center

Who Are We?
- **Georgia Advanced Computing Resource Center (GACRC)**
- Collaboration between the Office of Vice President for Research (OVPR) and the Office of the Vice President for Information Technology (OVPIT)
- Guided by a faculty advisory committee (GACRC-AC)

Why Are We Here?
- To provide computing hardware and network infrastructure in support of high-performance computing (HPC) at UGA

Where Are We?
- [http://gacrc.uga.edu](http://gacrc.uga.edu) (Web)
- [http://wiki.gacrc.uga.edu](http://wiki.gacrc.uga.edu) (Wiki)
- [http://gacrc.uga.edu/help/](http://gacrc.uga.edu/help/) (Web Help)
- [https://wiki.gacrc.uga.edu/wiki/Getting_Help](https://wiki.gacrc.uga.edu/wiki/Getting_Help) (Wiki Help)
Computer Cluster

- Cluster is a widely used term meaning independent computers combined into a unified system through software and high speed dedicated network
  - It provides greater computational power than a single computer can provide – LARGER problems can be solved

- Used for:
  - High Availability (HA)
  - Greater reliability
  - High Performance Computing (HPC)
High Performance Computing (HPC)

Serial Computing

- A problem is broken into a discrete series of instructions
- Instructions are executed sequentially
- Executed on a single processor
- Only one instruction may execute at any moment in time

Parallel Computing

- A problem is broken into discrete parts that can be solved concurrently
- Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different processors
- An overall control/coordination mechanism is employed
Zcluster Overview

- zcluster Structure
- General Information
- Computing Resources
- Storage Environment
zcluster – General Information

GACRC zcluster is a Linux high performance computing (HPC) cluster:

- **Operating System:** 64-bit Red Hat Enterprise Linux 5 (RHEL 5)
- **Login Node:** zcluster.rcc.uga.edu
- **Transfer Node:** xfer.gacrc.uga.edu
- **Interactive Node:** compute-14-7/9

**Internodal Communication:** 1Gbit network

- compute nodes ↔ compute nodes
- compute nodes ↔ storage systems

**NOTE:** Please Do Not run jobs on the zcluster login node - use the Queues or the Interactive Nodes.
zcluster – General Information

- Batch-queueing System:
  - Jobs can be started (submitted), monitored, and controlled
  - Determine which compute node is the best place to run a job
  - Determine appropriate execution priority for a job to run

- On zcluster: Sun Grid Engine (SGE)
## zcluster – Computing Resources

<table>
<thead>
<tr>
<th>Queue Type</th>
<th>Queue Name</th>
<th>Nodes</th>
<th>Processor</th>
<th>Cores/Node</th>
<th>RAM(GB)/Node</th>
<th>Cores</th>
<th>NVIDIA GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>rcc-30d</td>
<td>45</td>
<td>Intel Xeon</td>
<td>12</td>
<td>48</td>
<td>540</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>150</td>
<td></td>
<td>8</td>
<td>16</td>
<td>1200</td>
<td></td>
</tr>
<tr>
<td>High Memory</td>
<td>rcc-m128-30d</td>
<td>1</td>
<td>Intel Xeon</td>
<td>8</td>
<td>128</td>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td>8</td>
<td>192</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td>12</td>
<td>256</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rcc-m512-30d</td>
<td>2</td>
<td></td>
<td>32</td>
<td>512</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Multi Core</td>
<td>rcc-mc-30d</td>
<td>6</td>
<td>AMD Opteron</td>
<td>32</td>
<td>64</td>
<td>192</td>
<td>N/A</td>
</tr>
<tr>
<td>Interactive</td>
<td>interq</td>
<td>2</td>
<td>AMD Opteron</td>
<td>48</td>
<td>132</td>
<td>96</td>
<td>N/A</td>
</tr>
<tr>
<td>GPU</td>
<td>rcc-sgpu-30d</td>
<td>2</td>
<td>Intel Xeon</td>
<td>8</td>
<td>48</td>
<td>16</td>
<td>4 Tesla S1070 cards</td>
</tr>
<tr>
<td></td>
<td>rcc-mgpu-30d</td>
<td>2</td>
<td></td>
<td>12</td>
<td>48</td>
<td>24</td>
<td>9 Tesla (Fermi) M2070 cards</td>
</tr>
<tr>
<td></td>
<td>rcc-kgpu-30d</td>
<td>4</td>
<td></td>
<td>12</td>
<td>96</td>
<td>24</td>
<td>32 Tesla (Kepler) K20Xm cards</td>
</tr>
</tbody>
</table>

**Total peak performance: 23 Tflops**
Layer 1: Node
- A standalone “computer in a box”
- Multiple processors, e.g. 2, sharing memory
- Local disk storage, network interface, etc.
- Networked into a cluster

Layer 2: Processor
- A single computing component
- Multicore processor, e.g. 4 cores

Layer 3: Core
- Central processing unit (CPU) reading and executing instructions independently
- Each core is assigned to a software thread
zcluster – Storage Environment

Mainly, there are 4 different storage locations:

- **Home directory** - where you would land after logging in to the zcluster
- **Iscratch** - is the storage that is local to the computation nodes
- **escratch4** - is temporary scratch which is visible to all nodes
- **Project** storage area - is long term, and is created for a lab.
zcluster – Storage Environment

- **Home directory** ➞ `/home/groupname/username`
  - Mounted and visible on all nodes, with a quota of ~100GB
  - Any directory on /home has **snapshot** backups
    - Taken once a day, and maintained **4 daily** ones and **1 weekly** one
    - Name: **.snapshot**, e.g., `/home/abclab/jsmith/.snapshot`
    - Completely invisible, however, user can “cd” into it and then “ls”:

```
pakala@zcluster:~$ pwd
/home/rccstaff/pakala
pakala@zcluster:~$ ls -a
.
..  .bash_history.compute-14-7  .bash_history.zhead  .bashrc  cmd_kill  .java  RNA_SEQ
.bash_history.compute-14-9  .bash_logout  batchsub_demo  e4  .mozilla  .ssh
.pakala@zcluster:~$ cd .snapshot  ➡️ Can “cd” into “.snapshot”
pakala@zcluster:~/.snapshot$ ls  ➡️ And “ls” to list its contents
2015.11.29.00.00.01.weekly  2015.12.06.00.00.01.weekly  2015.12.07.01.00.01.daily  2015.12.09.01.00.01.daily
2015.12.05.01.00.01.daily  2015.12.06.01.00.01.daily  2015.12.08.01.00.01.daily
```
zcluster – Storage Environment

- **Local scratch** ➔ /lscratch/username
  - On **local disk** of each **compute** node ➔ node-local storage
  - rcc-30d 8-core nodes: ~18GB, rcc-30d 12-core nodes: ~370GB
  - **No snapshot backup**
  - **Usage Suggestion:** *If your job writes results to /lscratch, job submission script should move the data to your home or escratch before exit*

- **Ephemeral Scratch** ➔ /escratch4/pakala/pakala_Nov_13
  - Create with **make_escratch** command at Login Node
  - Visible to **all nodes** with a quota of **4TB**
  - **No snapshot backup**
  - To be deleted after **37 days**
# zcluster – Storage Environment

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Role</th>
<th>Quota</th>
<th>Accessible from</th>
<th>Intended Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/abclab/username</td>
<td>Home</td>
<td>100GB</td>
<td>zcluster.rcc.uga.edu (Login) xfer.gacrc.uga.edu (Transfer) Interactive nodes (Interactive) compute nodes (Compute)</td>
<td>Highly static data being used frequently</td>
<td>Snapshots</td>
</tr>
<tr>
<td>/escratch4/username</td>
<td>Scratch</td>
<td>4TB</td>
<td></td>
<td>Temporarily storing large data being used by jobs</td>
<td>Auto-deleted in 37 days</td>
</tr>
<tr>
<td>/lscratch/username</td>
<td>Local Scratch 18 ~ 370GB</td>
<td>Individual compute node</td>
<td></td>
<td>Jobs with heavy disk I/O</td>
<td>User to clean up</td>
</tr>
<tr>
<td>/project/abclab</td>
<td>Storage</td>
<td>Variable</td>
<td>xfer.gacrc.uga.edu (Transfer)</td>
<td>Long-term data storage</td>
<td>Group sharing possible</td>
</tr>
</tbody>
</table>

Note: 1. /usr/local : Software installation directory  
   /db : bioinformatics database installation directory  
2. To login to Interactive nodes, use qlogin from Login node
# zcluster – Storage Environment

<table>
<thead>
<tr>
<th>6 Main Function</th>
<th>On/From-Node</th>
<th>Related Filesystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Landing</td>
<td>Login Transfer</td>
<td>/home/abclab/username (Home) (Always!) /home/username (Transfer)</td>
</tr>
<tr>
<td>Batch Job Submitting</td>
<td>Login or Interactive</td>
<td>/escratch4/username/username_mth_date (Scratch) (Suggested!) /home/abclab/username (Home)</td>
</tr>
<tr>
<td>Interactive Job Running</td>
<td>Interactive</td>
<td>/escratch4/username/username_mth_date (Scratch) /home/abclab/username (Home)</td>
</tr>
<tr>
<td>Data Archiving, Compressing and Transferring</td>
<td>Transfer</td>
<td>/escratch4/username/username_mth_date (Scratch) /panfs/pstor.storage/home/abclab/username (Home)</td>
</tr>
<tr>
<td>Job Data Temporarily Storing</td>
<td>Compute</td>
<td>/lscratch/username (Local Scratch) /escratch4/username/username_mth_date (Scratch)</td>
</tr>
<tr>
<td>Long-term Data Storing</td>
<td>Login or Transfer</td>
<td>/project/abclab</td>
</tr>
</tbody>
</table>
How does zcluster operate?

GACRC zcluster Operational Diagram

1. Gbit Ethernet

**Login Node**
- `ssh username@zcluster.rcc.uga.edu` with zcluster password
  - (for script editing, batch job submission)

**Interactive Node**

**Transfer Node**
- `ssh username@xfer.gacrc.uga.edu` with UGA MyID password
  - (for data transferring, compression, and packaging)
Zcluster – Tips, Dos and Don’ts

Before we start:

- To get zcluster to be your best HPC buddy, go to
  GACRC Wiki (http://wiki.gacrc.uga.edu)
  GACRC Web (http://gacrc.uga.edu)

- To get the most effective and qualified support from us, go to
  GACRC Support (https://wiki.gacrc.uga.edu/wiki/Getting_Help)

- To work happily and productively, follow the cluster’s
  Community Code of Conduct (CCOC)
zcluster – Tips, Dos and Don’ts - continued

- Cluster’s CCOC:

  On cluster, you are not alone... Each user is sharing finite resources, e.g., CPU cycles, RAM, disk storage, network bandwidth, with other researchers. *What you do may affect other researchers on the cluster.*

  6 rules of thumb to remember:
  - NO jobs running on login node
  - NO multi-threaded job running with only 1 core requested
  - NO large memory job running on regular nodes
  - NO long job running on interactive node
  - NO small memory job running on large memory nodes
  - Use the copy node for file transfer and compression
How to work with zcluster? - Overview

- Start with zcluster
- Connect & Login
- Transfer Files
- Run Interactive Jobs
- Submit Batch Jobs
  - How to submit *serial*, *threaded*, and *MPI* batch jobs
  - How to check job status, cancel a job, etc.
- Software Installed
Getting Started with zcluster

- You need a **User Account**: `username@zcluster.rcc.uga.edu`
- Procedure: [https://wiki.gacrc.uga.edu/wiki/User_Accounts](https://wiki.gacrc.uga.edu/wiki/User_Accounts)
- User receives an email notification once the account is ready
- User can use `passwd` command to change initial temporary password
- A UGA faculty member (**PI**) may register a computing lab: [http://help.gacrc.uga.edu/labAcct.php](http://help.gacrc.uga.edu/labAcct.php)
- The PI of a computing lab may request user accounts for members of his/her computing lab: [http://help.gacrc.uga.edu/userAcct.php](http://help.gacrc.uga.edu/userAcct.php)
Connection & Login @ zcluster

- Open a connection: Open a terminal and `ssh` to your account
  ```
  ssh pakala@zcluster.rcc.uga.edu
  or
  ssh -X pakala@zcluster.rcc.uga.edu
  ```
  (1) `-X` is for X windows application running on the cluster to be forwarded to your local machine
  (2) If using Windows, use SSH client to open connection, get from UGA download software page

- Logging in: You will be prompted for your `zcluster password`
  ```
  pakala@zcluster.rcc.uga.edu’s password:
  ```
  (3) On Linux/Mac, when you type in the password, the prompt blinks and does not move

- Logging out: `exit` to leave the system
  ```
  pakala@zcluster:~$ exit
  ```
Transfer Files Using Transfer Node: xfer.gacrc.uga.edu

- SSH `username@xfer.gacrc.uga.edu` with your UGA MyID password
- Landing directory: `/home/username` (Sapelo home)
- Move data into/out of zcluster (`scp`, `sftp`, SSH Secure File Transfer, FileZilla)
- Filesystems you can access:
  - `/home/username/`: Sapelo home (landing spot)
  - `/lustre1/username/`: Sapelo global scratch
  - `/panfs/pstor.storage/home/abclab/username/`: zcluster home
  - `/escratch4/username/`: zcluster scratch
  - `/project/abclab/`: long-term active data storage
- Most file systems on Transfer are *auto-mounted* upon *the first time full-path access*, e.g., `cd /project/abclab/`
Transfer Files @ zcluster

User’s local ➔ Transfer Node (xfer.gacrc.uga.edu)

- On Linux, Mac or cygwin on Windows: `scp [Source] [Target]`
  
  E.g. 1: On local machine, do Local ➔ zcluster
  ```
  scp file1 username@xfer.gacrc.uga.edu:~/subdir
  scp *.dat username@xfer.gacrc.uga.edu:~/subdir
  ```

  E.g. 2: On local machine, do zcluster ➔ Local
  ```
  scp username@xfer.gacrc.uga.edu:~/subdir/file ./
  scp username@xfer.gacrc.uga.edu:~/subdir/\*.dat ./
  ```

- On Windows: FileZilla, WinSCP, SSH Secure Client, etc.
SSH Secure Client: Connecting to Transfer Node

1. Quick Connect
2. Connect to Remote Host
3. Enter details
4. Connect

Host Name: example.uga.edu
User Name: username
Port Number: 22
Authentication Method: Login/Password
SSH Secure Client: Connecting to Transfer Node

Connecting to xfer.gacrc.uga.edu...
SSH Secure: Connected to Home Directory

The image shows a file transfer interface with a directory listing. The Home Directory is highlighted with an arrow. The directory structure includes various folders and files, indicating a secure file transfer environment.

### Directory Structure

<table>
<thead>
<tr>
<th>Local Name</th>
<th>Size</th>
<th>Type</th>
<th>Modified</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OneDrive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homegroup</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssuchi.pakala</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>My Computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Panel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Panel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control Panel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Font Reader</td>
<td>1,282</td>
<td>Shortcut</td>
<td>07/23/2015 04:57:19</td>
<td></td>
</tr>
<tr>
<td>Google Chrome</td>
<td>2,214</td>
<td>Shortcut</td>
<td>05/23/2015 10:12:31</td>
<td></td>
</tr>
<tr>
<td>Lenovo Solution Center</td>
<td>2,018</td>
<td></td>
<td>03/06/2015 11:03:10</td>
<td></td>
</tr>
<tr>
<td>Skype</td>
<td>2,713</td>
<td>Shortcut</td>
<td>03/09/2015 07:20:10</td>
<td></td>
</tr>
<tr>
<td>SSH Secure File Transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSH Secure Shell Client</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO NOT DELETE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GACRC_10152015_Suchi</td>
<td>1,363</td>
<td></td>
<td>07/06/2015 10:32:03</td>
<td></td>
</tr>
<tr>
<td>North_Ocone_Girls</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shivank</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suchi Samsung Pics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Remote Directory

- **Files**
  - .bash_history
  - .bash_history.compute-14-7
  - .bash_history.compute-14-9
  - .bash_history.compute-18-8
  - .bash_history.zcluster
  - .bash_history.zmed
- **Directories**
  - Folder
    - 11/17/2015 12:00:00
    - rw-xr-xr-x
  - Folder
    - 10/18/2015 04:30:00
    - rwxr-xr-x
  - Folder
    - 12/12/2015 12:40:40
    - rwxr-xr-x

This directory structure is typical for a Home Directory, with essential files and directories organized for a secure environment.
SSH Secure: Navigated to /escratch4/pakala
Run Interactive Jobs @ zcluster

❖ To run an interactive job, you need to open a session on an interactive node using `qlogin` command:

```
pakala@zcluster:~$ qlogin
Your job 9559204 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 9559204 has been successfully scheduled.
...
compute-14-7.local$
```

❖ Current maximum runtime is **12 hours**
❖ When you are done, remember to **exit** the session!
❖ Detailed information, about interactive parallel jobs.
Software Installed @ zcluster

- Perl, Python, Java, awk, sed, C/C++ and Fortran compilers
- Matlab, Maple, R
- Many Bioinformatics applications: NCBI Blast+, Velvet, Trinity, TopHat, MrBayes, SoapDeNovo, Samtools, RaxML, etc.
- RCCBatchBlast (RCCBatchBlastPlus) to distribute NCBI Blast (NCBI Blast+) searches to multiple nodes.
- Many Bioinformatics Databases: NCBI Blast, Pfam, uniprot, etc. [https://wiki.gacrc.uga.edu/wiki/Bioinformatics_Databases](https://wiki.gacrc.uga.edu/wiki/Bioinformatics_Databases)
- For a complete list of applications: [https://wiki.gacrc.uga.edu/wiki/Software](https://wiki.gacrc.uga.edu/wiki/Software)
Submit Batch Jobs @ zcluster

- Components you need to submit a batch job:
  - **Software** already installed on zcluster
  - **Job submission script** to run the software,
    - Specifying working directory
    - Exporting environment variables, e.g.,
      - OMP_NUM_THREADS (OpenMP threads number)
      - LD_LIBRARY_PATH (searching paths for shared libraries)

- Common commands you need:
  - `qsub` with specifying **queue name, threads or MPI rank number**
  - `qstat`, `qdel`
  - `qacct`, `qsj`, etc.
Batch Serial Job @ zcluster

Step 1: Create a job submission script fastqc_sub.sh running Fastqc tool:

```bash
#!/bin/bash

cd /escratch4/pakala/pakala_Nov_13
export PATH=/usr/local/fastqc/latest:$PATH
time fastqc SRR1369670.fastq -o Output_File
```

- **Linux shell** (bash)
- Specify and enter (cd) the working directory
- Export command helps to set the Environment variables
- Run Fastqc with ‘time’ command to measure amount of time it takes to run the application

Step 2: Submit it to the queue:

- `qsub -q rcc-30d fastqc_sub.sh`

OR

- `qsub -q rcc-30d -l mem_total=20g fastqc_sub.sh`

Submit a job to the queue rcc-30d with **16GB** RAM/Node

Your job submission script

Submit a job to the queue rcc-30d with **48GB** RAM/Node
Batch **Threaded** Job @ zcluster

- **Step 1:** Create a job submission script `blast.sh` running Blast:

  ```bash
  #!/bin/bash
  cd /escratch4/pakala/pakala_Nov_13
  time /usr/local/ncbiblast/latest/bin/blastall -p 2 [options]  ➡️ Run Blastall with 2 threads
  ```

- **Step 2:** Submit it to the queue:

  ```bash
  $ qsub -q rcc-30d -l mem_total=20g -pe thread 2 ./blast.sh
  ```

  Number of Threads = Number of Cores Requested

  to the queue rcc-30d with **48GB RAM/Node**

  2 cores requested

**Note:**
Please use the `rcc-mc-30d` queue, If using threads more than 8!
Batch **MPI** Job @ zcluster

**Step 1**: Create a job submission script `sub.sh` running RAxML:

```bash
#!/bin/bash

cd /escratch4/pakala/pakala_Nov_13

export MPIRUN=/usr/local/mpich2/1.4.1p1/gcc 4.5.3/bin/mpirun

$MPIRUN -np $NSLOTS /usr/local/raxml/latest/raxmlHPC-MPI-SSE3 [options]
```

- Define and export environment variable (`MPIRUN`) for convenient usage
- Run RAxML with 20 MPI processes (`-np $NSLOTS`)

**Step 2**: Submit it to the queue:

```bash
$ qsub -q rcc-30d -pe mpi 20 sub.sh
```

20 cores requested,
$NSLOTS will be assigned to 20 automatically, before the job submission script is interpreted
Check and Cancel Jobs @ zcluster

- To check the status of all queued and running jobs: `qstat`
  
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qstat</code></td>
<td>shows your job in the pool</td>
</tr>
<tr>
<td><code>qstat -u &quot;*&quot;</code></td>
<td>shows all the jobs in the pool</td>
</tr>
<tr>
<td><code>qstat -j 12345</code></td>
<td>shows detailed information, e.g., <code>maxvmem</code>, about the job with JOBID 12345</td>
</tr>
<tr>
<td><code>qstat -g t</code></td>
<td>list all nodes used by your jobs</td>
</tr>
</tbody>
</table>

- To cancel a queued or running job: `qdel`
  
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qdel -u pakala</code></td>
<td>deleted all your jobs</td>
</tr>
<tr>
<td><code>qdel 12345</code></td>
<td>deletes your job with JOBID 12345</td>
</tr>
</tbody>
</table>

- To list detailed information about a job: `qsj`, `qacct`
  
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>qsj 12345</code></td>
<td>shows information, e.g., <code>maxvmem</code>, about the <strong>RUNNING</strong> job with JOBID 12345</td>
</tr>
<tr>
<td><code>qacct -j 12345</code></td>
<td>shows information, e.g., <code>maxvmem</code>, about the <strong>ENDED</strong> job with JOBID 12345</td>
</tr>
</tbody>
</table>
How to Submit Tickets to GACRC

- For Installation/Downloading Software:
  - User needs to provide the name, version (or latest), and website
  - Applications need to be compatible with Linux
  - Note – only FREE software will be installed

- For Troubleshooting:
  - List the path of the working directory, path of the script that is producing errors, Job ID, and the command sent to the queue or interactive node
  - No need to attach the script or huge error messages

- For Testing:
  - Please have a sample dataset at your working directory, so that it can be used for debugging

- These steps will help us in responding quickly and efficiently
THANK YOU for your patience
Questions?