GACRC Sapelo2 Cluster New User Training

Georgia Advanced Computing Resource Center (GACRC)
Enterprise Information Technology Services (EITS)
The University of Georgia
Outline

- GACRC
- Sapelo2 Cluster
  - Cluster Diagram and Overview
  - Five Directories
  - Five Computational Partitions
  - Software Environment
- Batch Job Submission Workflow
- Useful Commands: sq --me, sacct-gacrc -X, interact
- GACRC Wiki and User Support
- Appendices
GACRC

- A high-performance-computing (HPC) center at the UGA
- Provide to the UGA research and education community an advanced computing environment:
  - HPC computing and networking infrastructure located at the Boyd Data Center
  - Comprehensive collection of scientific, engineering and business applications
  - Consulting and training services

Wiki: http://wiki.gacrc.uga.edu
Help and Support: http://help.gacrc.uga.edu
Web Site: http://gacrc.uga.edu
Kaltura Channel: https://kaltura.uga.edu/channel/GACRC/176125031
**Sapelo2 Cluster**

1. **Login/Submit Nodes**
   - sapelo2.gacrc.uga.edu
   - (ss-sub1, ss-sub2, ss-sub3, ss-sub4, ss-sub5)

2. **Transfer Nodes**
   - xfer.gacrc.uga.edu
   - (xfer4, xfer5, xfer6, xfer7)

**Home Storage**
- /home/MyID
- 200 GB quota

**Scratch Storage**
- /scratch/MyID
- No quota

**Work Storage**
- /work/abclab
- 500 GB or $10^5$ files quota

**Project Storage**
- /project/abclab
- 1 TB quota

**Compute Nodes**

- **Cluster**: Nodes + Storage, all connected by network
- **Node**: Computer with a particular purpose within the cluster

---

**Note**: You need to connect to the UGA network using VPN when accessing from outside of the UGA main campus.

UGA VPN: [https://eits.uga.edu/access_and_security/infosec/tools/vpn/](https://eits.uga.edu/access_and_security/infosec/tools/vpn/)
## Five Directories
[https://wiki.gacrc.uga.edu/wiki/Disk_Storage](https://wiki.gacrc.uga.edu/wiki/Disk_Storage)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Name</th>
<th>Quota</th>
<th>Accessible from</th>
<th>Intended Use</th>
<th>Backed-up</th>
<th>Important Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/MyID</td>
<td>Home</td>
<td>200GB</td>
<td>Login</td>
<td>Static data, e.g. 1. Scripts, source codes 2. Local software</td>
<td>Yes</td>
<td>Not for storing data of your jobs!</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Transfer Compute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/scratch/MyID</td>
<td>Scratch</td>
<td>No Limit</td>
<td>Login</td>
<td>Temporary files needed for currently running jobs</td>
<td>No</td>
<td>Clean up when your job finishes! Subject to “30-day purge” policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Transfer Compute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/work/abclab</td>
<td>Work</td>
<td>500GB 10^5 files</td>
<td>Login</td>
<td>Input files needed for repeated jobs</td>
<td>No</td>
<td>Clean up when your job finishes! Group sharing is possible</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Transfer Compute</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/project/abclab</td>
<td>Project</td>
<td>1TB (initial)</td>
<td>Transfer</td>
<td>Temporary data parking</td>
<td>Yes</td>
<td>Group sharing is possible</td>
</tr>
<tr>
<td>/lscratch</td>
<td>Local Scratch</td>
<td>200GB - 800GB</td>
<td>Compute</td>
<td>Jobs with heavy disk I/O operations</td>
<td>No</td>
<td>Clean up when job exits from node!</td>
</tr>
</tbody>
</table>
Scratch File System 30-Day Purge Policy

https://wiki.gacrc.uga.edu/wiki/Disk_Storage#Scratch_file_system

Any file that is not accessed or modified by a compute job in a time period no longer than 30 days will be automatically deleted off the /scratch file system.

Measures circumventing this policy will be monitored and actively discouraged.

- You are suggested to copy files from /scratch to /project or outside of GACRC.
- You should first move all unnecessary files and folders to /scratch/trash/\$USER.
- The fastest way to save old files is to copy them to /project area, using the fpsync utility on xfer.gacrc.uga.edu.
- When you archive data using tar on /scratch, please do not use z option (compression option). After you archive data with tar, you can use gzip to compress it.
Compute nodes are divided into groups called **partitions**. A **partition** is a collection of compute nodes for a particular computing need.

- **batch/batch_30d**: For regular jobs
- **highmem_p/highmem_30d_p**: For high memory jobs
- **gpu_p/gpu_30d_p**: For GPU-enabled jobs
- **inter_p**: For interactive jobs

_p = “partition”  
_30d = 30 day (time limit)
<table>
<thead>
<tr>
<th>Type</th>
<th>Partition</th>
<th>Time limit</th>
<th>Max jobs Running</th>
<th>Max jobs Submit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>batch</td>
<td>7 days</td>
<td>250</td>
<td>10,000</td>
<td>Regular nodes</td>
</tr>
<tr>
<td></td>
<td>highmem_p</td>
<td></td>
<td>15</td>
<td>100</td>
<td>For running high memory jobs</td>
</tr>
<tr>
<td></td>
<td>hugemem_p</td>
<td></td>
<td>4</td>
<td>4</td>
<td>For running huge memory jobs</td>
</tr>
<tr>
<td></td>
<td>gpu_p</td>
<td></td>
<td>18</td>
<td>20</td>
<td>For running GPU-enabled jobs</td>
</tr>
<tr>
<td>Long-term</td>
<td>batch_30d</td>
<td>30 days</td>
<td>1</td>
<td>2</td>
<td>30-day partition counterparts</td>
</tr>
<tr>
<td></td>
<td>highmem_30d_p</td>
<td>30 days</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hugemem_30d_p</td>
<td></td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>gpu_30d_p</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>inter_p</td>
<td>2 days</td>
<td>3</td>
<td>20</td>
<td>Regular nodes, for interactive jobs.</td>
</tr>
<tr>
<td>Buy-in</td>
<td>name_p</td>
<td>variable</td>
<td></td>
<td></td>
<td>Partitions that target different groups' buy-in nodes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The name string is specific to each group.</td>
</tr>
<tr>
<td>Partition</td>
<td>Total Nodes</td>
<td>Max Mem(GB)/Single-node job</td>
<td>Cores /Node</td>
<td>Processor Type</td>
<td>GPU Cards /Node</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
<td>-------------</td>
<td>------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>batch</td>
<td>119</td>
<td>500</td>
<td>128</td>
<td>AMD EPYC Milan</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>250</td>
<td>64</td>
<td>AMD EPYC Milan</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>120</td>
<td>64</td>
<td>AMD EPYC Rome</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>123</td>
<td>120</td>
<td>64</td>
<td>AMD EPYC Rome</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td></td>
<td>32</td>
<td>AMD EPYC Naples</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>180</td>
<td>32</td>
<td>Intel Xeon Skylake</td>
<td>N/A</td>
</tr>
<tr>
<td>highmem_p</td>
<td>18</td>
<td>500</td>
<td>32</td>
<td>AMD EPYC Naples</td>
<td>N/A</td>
</tr>
<tr>
<td>highmem_30d_p</td>
<td>4</td>
<td></td>
<td>64</td>
<td>AMD EPYC Naples</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>990</td>
<td>32</td>
<td>AMD EPYC Milan</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>128</td>
<td>AMD EPYC Milan</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>28</td>
<td>Intel Xeon Broadwell</td>
<td>N/A</td>
</tr>
<tr>
<td>hugemem_p,</td>
<td>2</td>
<td>2000</td>
<td>32</td>
<td>AMD EPYC Rome</td>
<td>N/A</td>
</tr>
<tr>
<td>hugemem_30d_p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpu_p</td>
<td>4</td>
<td>180</td>
<td>32</td>
<td>Intel Xeon Skylake</td>
<td>N/A</td>
</tr>
<tr>
<td>gpu_30d_p</td>
<td>2</td>
<td>120</td>
<td>16</td>
<td>Intel Xeon</td>
<td>1 NVIDIA P100</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1000</td>
<td>64</td>
<td>AMD EPYC Milan</td>
<td>4 NVIDIA A100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Software Environment
https://wiki.gacrc.uga.edu/wiki/Software

• Approximately 900 software modules are installed (as of March 2021)
• Most modules are compiled EasyBuild (EB) toolchains GCC-8.3.0 or foss-2019b.
• Name format: Name/Version-Toolchain, e.g., Python/3.8.2-GCCcore-8.3.0 (case-sensitive)
• Module commands:
  ➢ ml spider pattern: Search module names matching a pattern
  ➢ ml moduleName: Load a module into your working environment
  ➢ DO NOT LOAD/USE MODULES ON THE LOGIN/SUBMIT NODES! (ss-sub1, ss-sub2, ss-sub3, etc...)
  ➢ ml av: List all available software modules installed on cluster
  ➢ ml: List modules currently loaded
  ➢ ml-moduleName: Remove a module from working environment
  ➢ ml purge: Remove all modules from working environment
Important Tip using Software

https://wiki.gacrc.uga.edu/wiki/Available_Toolchains_and_Toolchain_Compatibility

• When you load more than one software modules, toolchain compatibility is the most important thing you need to pay attention to

• If you load more than one module and some toolchains are incompatible, your job will end up with failing dependencies or Lmod errors, such as:

Lmod has detected the following error:
These module(s) exist but cannot be loaded as requested

ml Python/3.7.4-GCCcore-8.3.0
ml Perl/5.28.0-GCCcore-7.3.0

ml Beast/2.6.3-foss-2019b
ml Perl/5.28.0-GCCcore-7.3.0

ml Python/3.7.4-GCCcore-8.3.0
ml Perl/5.30.0-GCCcore-8.3.0

ml Beast/2.6.3-foss-2019b
ml Perl/5.30.0-GCCcore-8.3.0
Job Submission Workflow

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2

1. Log on to Login node using MyID and password, and two-factor authentication with Archpass Duo:
   
   `ssh MyID@sapelo2.gacrc.uga.edu`

2. On Login node, change directory to your scratch space:
   
   `cd /scratch/MyID`

3. Create a working subdirectory for a job:
   
   `mkdir workDir`

4. Change directory to `workDir`:
   
   `cd workDir`

5. Transfer data from local computer to `workDir`:
   
   Use **Globus** to transfer data to the cluster.

   Transfer data on cluster to `workDir`:
   
   Use **Globus** or log on to Transfer node and then use **cp** or **mv**

6. Make a job submission script in `workDir`:
   
   `nano sub.sh`

7. Submit a job from `workDir`:
   
   `sbatch sub.sh`

8. Check job status:
   
   `squeue --me` or Cancel a job:
   
   `scancel jobID`
Step 1: Log on to Login node - Mac/Linux using ssh

https://wiki.gacrc.uga.edu/wiki/Connecting

1. Open **Terminal** utility

2. Type command line: `ssh MyID@sapelo2.gacrc.uga.edu`

3. You will be prompted for your **MyID password**

4. Sapelo2 access requires ID verification using two-factor authentication with Archpass Duo. If you are not enrolled in Archpass Duo, please refer to

   https://eits.uga.edu/access_and_security/infosec/tools/archpass_duo/ on how to enroll
Step 1 (Cont.): Mac/Linux using ssh

Use Terminal utility on Mac or Linux!

[zhufei@localhost ~]$ ssh zhufei@sapelo2.gacrc.uga.edu
Password: ➔ Input MyID password!

Enter a passcode or select one of the following options:
1. Duo Push to XXX-XXX-5758
2. Phone call to XXX-XXX-5758
3. Phone call to XXX-XXX-1925
4. SMS passcodes to XXX-XXX-5758

Passcode or option (1-5): 1 ➔ Select Duo authentication option!
Success. Logging you in...
Last login: Tue Sep 15 11:22:42 2020 from 128.192.75.65

zhufei@ss-sub1 ~$ ➔ I am on login node ss-sub1!
Step 1 (Cont.): Windows using PuTTY

1. Download and install PuTTY: [https://www.putty.org/](https://www.putty.org/)

2. Detailed downloading and installation instructions:
   
   [https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY](https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY)

3. Detailed configuring and usage instructions:
   
   [https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY#Configuring_PuTTY](https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY#Configuring_PuTTY)
Step 1 (Cont.): Windows using PuTTY

The first time you connect to a login node, PuTTY will give you this security alert window. Please click “Yes” or “Accept”
Next you will enter your UGA MyID password and initiate DUO authentication procedure:

Password: [enter UGA MyID password]

DUO option

For additional help with UGA DUO authentication or to report an issue please visit:

https://eits.uga.edu/access_and_security/infosec/tools/archpass/
Step 2: On Login node change directory to scratch

• Once you logged on, your current directory will be your home directory

```
zhuofei@ss-sub1 ~$ pwd
/home/zhuofei
```

→ this is my home directory!

• Use cd command to change your current directory to /scratch/MyID

```
zhuofei@ss-sub1 ~$ cd /scratch/zhuofei/
zhuofei@ss-sub1 zhuofei$ pwd
/scratch/zhuofei
```

→ this is my scratch space!

• Use ls command to take a look in /scratch/MyID

```
zhuofei@ss-sub1 zhuofei$ ls
user_test
```
Step 3 - 4: Create and cd to a working subdirectory

• Use `mkdir` command to make a subdirectory in `/scratch/MyID`

  ```bash
  zhuofei@ss-sub1 zhuofei$ mkdir workDir
  zhuofei@ss-sub1 zhuofei$ ls
  user_test workDir
  ```

• Use `cd` command to change your current directory to `/scratch/MyID/workDir`

  ```bash
  zhuofei@ss-sub1 zhuofei$ cd workDir
  zhuofei@ss-sub1 workDir$ pwd
  /scratch/zhuofei/workDir
  zhuofei@ss-sub1 workDir$ ls
  ❯ it is empty!
  ```
Step 5: Transfer Data to the Cluster using Globus

https://wiki.gacrc.uga.edu/wiki/Globus

1. Create a Globus Identity Account at globus.org following the instructions at
https://wiki.gacrc.uga.edu/wiki/Globus#Getting_Started

2. From the File Manager tab at globus.org, search for the “UGA GACRC Collection”
Step 5 (Cont.): Transfer Data to the Cluster using Globus
https://wiki.gacrc.uga.edu/wiki/Globus

When you open the UGA GACRC Collection in Globus, by default you will be in your home directory. You can change to any of your or your lab’s other directories by typing a path in the Path bar.

Use the **Upload/Download** options on the right for transferring **small** or a **few** files.

Use the **Transfer or Sync** option for **many** or **large** files (requires Globus Connect Personal on your computer; https://wiki.gacrc.uga.edu/wiki/Globus_Connect_Personal)
Step 5 (Cont.): Transfer Data to the Cluster using Globus

https://wiki.gacrc.uga.edu/wiki/Globus

Example 1:
Using the Transfer or Sync option from between one’s personal computer and a /project directory on the cluster
Step 5 (Cont.): Transfer Data to the Cluster using Globus

https://wiki.gacrc.uga.edu/wiki/Globus

Example 1 (Cont.):

Using the Transfer or Sync option from between one’s personal computer and a /project directory on the cluster
Step 5 (Cont.): Transfer Data on the Cluster to workDir using Globus

https://wiki.gacrc.uga.edu/wiki/Globus

Example 2:

Using the Transfer or Sync option between two places on the cluster
Step 6: Make a job submission script in workDir
https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2

```
$ nano sub.sh
```

nano is a simple text editor on Linux. You are welcome to use other editors like vim or emacs.

Ctrl-x to save file and quit from nano
To run the demo example, please copy these files into your working dir:
```
cp -r /usr/local/training/Sapelo2/* .
```
Submission scripts do not have to be complex!

https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2

The only required Slurm headers are:

- --partition
- --ntasks
- --mem
- --time

(Slurm will let you know if you forgot one when you try to submit your submission script)

Some default values if not specified:

- --cpus-per-task=1
- --nodes=1

#!/bin/bash

#SBATCH --partition=batch
#SBATCH --ntasks=1
#SBATCH --mem=4G
#SBATCH --time=1:00:00

ml Bowtie2/2.4.1-GCC-8.3.0

bowtie2 -x index/lambda_virus -U myreads.fq
Step 7: Submit a job from workDir using `sbatch`

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_submit_a_job_to_the_batch_partition

```
zhuofei@ss-sub1 workDir$ pwd
/scratch/zhuofei/workDir
zhuofei@ss-sub1 workDir$ ls
index  myreads.fq  sub.sh
zhuofei@ss-sub1 workDir$ sbatch sub.sh
Submitted batch job 32860
```

**sub.sh** is a job submission script to

1. specify computing resources:
2. load software using `ml moduleName`
3. run any Linux commands you want to run
4. run the software
Step 8: Check pending/running job status using `squeue/sq --me`

https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2

<table>
<thead>
<tr>
<th>JOBID</th>
<th>PARTITION</th>
<th>NAME</th>
<th>USER</th>
<th>NODES</th>
<th>TIME</th>
<th>ST</th>
<th>NODELIST (REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4618668</td>
<td>highmem_p</td>
<td>test-job</td>
<td>zhuofei</td>
<td>1</td>
<td>1:16</td>
<td>R</td>
<td>d4-11</td>
</tr>
<tr>
<td>4618666</td>
<td>batch</td>
<td>Bowtie2-2cpu</td>
<td>zhuofei</td>
<td>1</td>
<td>1:19</td>
<td>R</td>
<td>c1-23</td>
</tr>
<tr>
<td>4618665</td>
<td>batch</td>
<td>testBowtie2</td>
<td>zhuofei</td>
<td>1</td>
<td>5:23</td>
<td>R</td>
<td>c1-23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JOBID</th>
<th>TIME</th>
<th>TIME_LIMIT</th>
<th>NAME</th>
<th>PARTITION</th>
<th>USER</th>
<th>NODES</th>
<th>CPUS</th>
<th>MIN_MEMORY</th>
<th>PRIORITY</th>
<th>STATE</th>
<th>NODELIST (REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4618668</td>
<td>1:18</td>
<td>4:00:00</td>
<td>test-job</td>
<td>highmem_p</td>
<td>zhuofei</td>
<td>1</td>
<td>1</td>
<td>200G</td>
<td>5993</td>
<td>RUNNING</td>
<td>d4-11</td>
</tr>
<tr>
<td>4618666</td>
<td>1:21</td>
<td>1:00:00</td>
<td>Bowtie2-2cpu</td>
<td>batch</td>
<td>zhuofei</td>
<td>1</td>
<td>2</td>
<td>4G</td>
<td>5991</td>
<td>RUNNING</td>
<td>c1-23</td>
</tr>
<tr>
<td>4618665</td>
<td>5:25</td>
<td>2:00:00</td>
<td>testBowtie2</td>
<td>batch</td>
<td>zhuofei</td>
<td>1</td>
<td>1</td>
<td>4G</td>
<td>5991</td>
<td>RUNNING</td>
<td>c1-23</td>
</tr>
</tbody>
</table>
sq --help output
https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2

zhuofei@ss-sub1 workDir$ sq --help

Usage: sq [OPTIONS]

Descriptions: sq - preformatted wrapper for squeue. See man squeue for more information.

- j Displays squeue output for a given job
--me Displays squeue output for the user executing this command
- p Displays squeue output for a given partition
- u Displays squeue output for a given user
- T Displays submit and start time columns
-h, --help Displays this help output
Step8 (Cont.): Cancel job using `scancel`

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_cancel_.28delete.29_a_running_or_pending_job

```
zhuofei@ss-sub1 workDir$ squeue --me -l
Tue Sep 15 15:08:27 2020
JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
32869     batch testBowt  zhuofei  RUNNING       0:01   1:00:00      1 c5-19
32868     batch testBowt  zhuofei  RUNNING       0:05   1:00:00      1 c5-19
32867     batch testBowt  zhuofei  RUNNING       0:10   1:00:00      1 c5-19
zhuofei@ss-sub1 workDir$
squeue --me -l
Tue Sep 15 15:08:45 2020
JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
32867     batch testBowt  zhuofei COMPLETI       0:26   1:00:00      1 c5-19
32869     batch testBowt  zhuofei  RUNNING       0:19   1:00:00      1 c5-19
32868     batch testBowt  zhuofei  RUNNING       0:23   1:00:00      1 c5-19
zhuofei@ss-sub1 workDir$
squeue --me -l
Tue Sep 15 15:08:50 2020
JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
32869     batch testBowt  zhuofei  RUNNING       0:19   1:00:00      1 c5-19
32868     batch testBowt  zhuofei  RUNNING       0:23   1:00:00      1 c5-19
```
### Step 8 (Cont.):

Check running or finished job status using `sacct -X / sacct-gacrc -X`

[https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)

---

<table>
<thead>
<tr>
<th>JobID</th>
<th>JobName</th>
<th>Partition</th>
<th>Account</th>
<th>AllocCPUS</th>
<th>State</th>
<th>ExitCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>4618312</td>
<td>interact</td>
<td>inter_p</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>COMPLETED</td>
<td>0:0</td>
</tr>
<tr>
<td>4618665</td>
<td>testBowtie2</td>
<td>batch</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>COMPLETED</td>
<td>0:0</td>
</tr>
<tr>
<td>4618666</td>
<td>Bowtie2-2+</td>
<td>batch</td>
<td>gacrc-ins+</td>
<td>2</td>
<td>COMPLETED</td>
<td>0:0</td>
</tr>
<tr>
<td>4618668</td>
<td>test-job</td>
<td>highmem_p</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>COMPLETED</td>
<td>0:0</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>JobID</th>
<th>JobName</th>
<th>User</th>
<th>Partition</th>
<th>ACCT</th>
<th>AllocCPUS</th>
<th>State</th>
<th>ExitCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>4618312</td>
<td>interact</td>
<td>bc06026</td>
<td>inter_p</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>2Gn</td>
<td>00:07:01</td>
</tr>
<tr>
<td>4618665</td>
<td>testBowtie2</td>
<td>bc06026</td>
<td>batch</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>4Gn</td>
<td>00:23:12</td>
</tr>
<tr>
<td>4618666</td>
<td>Bowtie2-2cpu</td>
<td>bc06026</td>
<td>batch</td>
<td>gacrc-ins+</td>
<td>2</td>
<td>4Gn</td>
<td>00:20:24</td>
</tr>
<tr>
<td>4618668</td>
<td>test-job</td>
<td>bc06026</td>
<td>highmem_p</td>
<td>gacrc-ins+</td>
<td>1</td>
<td>200Gn</td>
<td>00:09:46</td>
</tr>
</tbody>
</table>
sacct-gacrc --help output

https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2

zhuofei@ss-sub1 workDir$ sacct-gacrc --help

Usage: sacct-gacrc [OPTIONS]

Description: preformatted wrapper for sacct. See man sacct for more information.

-E, --endtime         Display information about jobs up to a date, in the format of yyyy-mm-dd (default: now)
-j, --jobs            Display information about a particular job or jobs (comma-separated list if more than one job)
-r, --partition       Display information about jobs from a particular partition
-S, --starttime       Display information about jobs starting from a date in the format of yyyy-mm-dd (default: Midnight of today)
-u, --user            Display information about a particular user's job(s) (default: current user)
-X, --allocations     Only show one line per job (do not display job steps)
--debug               Display the sacct command being executed
-h, --help            Display this help output
Step 8 (Cont.): Check resource usage of finished jobs using `seff`

https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2

```
zhuofei@ss-sub1 workDir$ seff 37259
```

Job ID: 37259
Cluster: tc2
User/Group: zhuofei/gacrc-instruction
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:09:45
CPU Efficiency: 99.66% of 00:09:47 core-walltime
Job Wall-clock time: 00:09:47
Memory Utilized: 197.34 MB
Memory Efficiency: 4.82% of 4.00 GB
Obtain Job Details

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_check_resource_utilization_of_a_running_or_finished_job

Option 1: `squeue` or `sq --me` for details of a pending or running jobs

Option 2: `sacct -X` or `sacct-gacrc -X` for details of running or finished jobs

Option 3: `seff` for details of computing resource usage of a finished job

Option 4: Email notification from finished jobs (completed, canceled, or crashed), if using:

```
#SBATCH --mail-user=username@uga.edu
#SBATCH --mail-type=ALL
```
## Interactive jobs

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_open_an_interactive_session

https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2#How_to_run_an_interactive_job_with_Graphical_User_Interface_capabilities

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start an interactive session</td>
<td><code>interact</code></td>
</tr>
<tr>
<td>Start an interactive session with X forwarding</td>
<td><code>interact --x11</code></td>
</tr>
</tbody>
</table>

```bash
interact
srn --pty --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=2GB /bin/bash -l
```

```bash
interact --x11
srn --pty --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=2GB --x11 /bin/bash -l
```
Usage: interact [OPTIONS]

Description: Start an interactive job

-c, --cpus-per-task       CPU cores per task (default: 1)
-J, --job-name            Job name (default: interact)
-n, --ntasks              Number of tasks (default: 1)
-N, --nodes               Number of nodes (default: 1)
-p, --partition           Partition for interactive job (default: inter_p)
-q, --qos                 Request a quality of service for the job.
-t, --time                Maximum run time for interactive job (default: 12:00:00)
-w, --nodelist            List of node name(s) on which your job should run
--constraint              Job constraints
--gres                    Generic consumable resources
--mem                     Memory per node (default 2GB)
--shell                   Absolute path to the shell to be used in your interactive job (default: /bin/bash)
--wckey                   Wckey to be used with job
--x11                     Start an interactive job with X Forwarding
-h, --help                Display this help output
GACRC Wiki [http://wiki.gacrc.uga.edu](http://wiki.gacrc.uga.edu)
Kaltura channel [https://kaltura.uga.edu/channel/GACRC/176125031](https://kaltura.uga.edu/channel/GACRC/176125031)

System: [https://wiki.gacrc.uga.edu/wiki/Systems#Sapelo2](https://wiki.gacrc.uga.edu/wiki/Systems#Sapelo2)
Connection: [https://wiki.gacrc.uga.edu/wiki/Connecting#Connecting_to_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Connecting#Connecting_to_Sapelo2)
Software: [https://wiki.gacrc.uga.edu/wiki/Software_on_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Software_on_Sapelo2)
Running Jobs: [https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo2)
Monitoring Jobs: [https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_Sapelo2)
Sample scripts: [https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2](https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2)
Transferring Files: [https://wiki.gacrc.uga.edu/wiki/Globus](https://wiki.gacrc.uga.edu/wiki/Globus)
Linux Commands: [https://wiki.gacrc.uga.edu/wiki/Command_List](https://wiki.gacrc.uga.edu/wiki/Command_List)
Open OnDemand: [https://wiki.gacrc.uga.edu/wiki/OnDemand](https://wiki.gacrc.uga.edu/wiki/OnDemand)
Training: [https://wiki.gacrc.uga.edu/wiki/Training](https://wiki.gacrc.uga.edu/wiki/Training)
GACRC Help and Support
https://wiki.gacrc.uga.edu/wiki/Getting_Help

- **Job Troubleshooting:**
  - Please tell us details of your question or problem, including but not limited to:
    - Your user name
    - Your job ID
    - Your working directory
    - The queue name and command you used to submit the job

- **Software Installation:**
  - Specific name and version of the software
  - Download website
  - Supporting package information if have

When you ask GACRC to test or troubleshoot your jobs, Please make sure of the correctness of your datasets being used!
Georgia Advanced Computing Resource Center (GACRC) service catalog.

If you would like to reach out to GACRC and do not have a UGA MyID, please send an email to gacrc-help@uga.edu, and we will respond promptly.

Categories (3)

**Services For Users**
General user support, request software installation or update, request training.

**Services for PIs**
For PIs only: Lab registration, user account creation/modification, class account requests, storage quota modifications.

**For GACRC Staff**
For GACRC's internal use only.
Services For Users

General user support, request software installation or update, request training.

Services (3)

General Support
Report issues and request help with GACRC systems, except for software installation requests and account/lab creation requests.

Software Installation/Update
Request software and common application database (e.g. NCBI blast databases) installation and upgrade.

Training
Request support related to training provided by the GACRC.
## Slurm job states

<table>
<thead>
<tr>
<th>Code</th>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Running</td>
<td>Job is running on compute node(s)</td>
</tr>
<tr>
<td>PD</td>
<td>Pending</td>
<td>Job is waiting for compute node(s)</td>
</tr>
<tr>
<td>CD</td>
<td>Completed</td>
<td>Job completed</td>
</tr>
<tr>
<td>CG</td>
<td>Completing</td>
<td>Job is completing</td>
</tr>
<tr>
<td>CA</td>
<td>Canceled</td>
<td>Job was canceled</td>
</tr>
<tr>
<td>F</td>
<td>Failed</td>
<td>Job terminated with non-zero exit code</td>
</tr>
<tr>
<td>NF</td>
<td>Node Fail</td>
<td>Job terminated due to failure of node(s)</td>
</tr>
</tbody>
</table>
## Commands for submitting and canceling jobs

<table>
<thead>
<tr>
<th>Description</th>
<th>Slurm Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit a batch job to queue</td>
<td><code>sbatch sub.sh</code></td>
</tr>
<tr>
<td>Delete a job from queue</td>
<td><code>scancel &lt;jobID&gt;</code></td>
</tr>
<tr>
<td>Cancel all your job(s)</td>
<td><code>scancel -u &lt;username&gt;</code></td>
</tr>
<tr>
<td>Cancel all your pending job(s)</td>
<td><code>scancel -t PENDING -u &lt;username&gt;</code></td>
</tr>
<tr>
<td>Cancel your job(s) by job name</td>
<td><code>scancel --name &lt;jobname&gt;</code></td>
</tr>
<tr>
<td>Cancel an element (index) of an array job (jobID)</td>
<td><code>scancel &lt;jobID&gt;_&lt;index&gt;</code></td>
</tr>
</tbody>
</table>
## Commands for monitoring jobs

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information about currently running jobs</td>
<td><code>squeue</code> or <code>sq</code></td>
</tr>
<tr>
<td>Information about <strong>your</strong> currently running jobs</td>
<td><code>squeue --me</code> or <code>sq --me</code></td>
</tr>
<tr>
<td>Information about a user’s currently running jobs</td>
<td><code>squeue -u &lt;username&gt;</code></td>
</tr>
<tr>
<td>Information about your running or finished jobs</td>
<td><code>sacct</code> or <code>sacct-gacrc</code></td>
</tr>
<tr>
<td>Job’s resource usage</td>
<td><code>seff &lt;job ID&gt;</code></td>
</tr>
</tbody>
</table>
Slurm headers for running a **Serial (single-core) job**

https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_Sapelo2

```bash
#!/bin/bash

#SBATCH --job-name=testBowtie2 # Job name (testBowtie2)
#SBATCH --partition=batch # Queue name (batch)
#SBATCH --ntasks=1 # Run in a single task using one CPU core on a single node
#SBATCH --mem=4G # Job memory limit (4 GB)
#SBATCH --time=1:00:00 # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.out # Standard output log, e.g., testBowtie2_1234.out
#SBATCH --error=%x_%j.err # Standard error log, e.g., testBowtie2_1234.err
#SBATCH --mail-type=END,FAIL # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu # Where to send mail

cd $SLURM_SUBMIT_DIR # Change directory to job submission directory
ml Bowtie2/2.4.1-GCC-8.3.0 # Load software module and run bowtie2 below
bowtie2 -x ./index/lambda_virus -U ./myreads.fq -S output.sam
```
Slurm headers for running a **Threaded** job

```bash
#!/bin/bash

#SBATCH --job-name=testBowtie2           # Job name (testBowtie2)
#SBATCH --partition=batch                # Queue name (batch)
#SBATCH --nodes=1                         # Run all processes on a single node
#SBATCH --ntasks=1                       # Run in a single task on a single node
#SBATCH --cpus-per-task=8                # Number of CPU cores per task (8)
#SBATCH --mem=10G                        # Job memory limit (10 GB)
#SBATCH --time=1:00:00                   # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE                    # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.out               # Standard output log, e.g., testBowtie2_1234.out
#SBATCH --error=%x_%j.err                # Standard error log, e.g., testBowtie2_1234.err
#SBATCH --mail-type=END,FAIL             # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu     # Where to send mail

cd $SLURM_SUBMIT_DIR
ml Bowtie2/2.4.1-GCC-8.3.0

bowtie2 -p 8 -x ./index/lambda_virus -U ./myreads.fq -S output.sam
```
Slurm headers for running an Array job

```bash
#!/bin/bash
#SBATCH --job-name=testBowtie2Array # Job name (testBowtie2Array)
#SBATCH --partition=batch # Queue name (batch)
#SBATCH --ntasks=1 # Run in a single task using one CPU core on a single node
#SBATCH --mem=4G # Job memory limit (4 GB)
#SBATCH --time=1:00:00 # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.out # Standard output log, e.g., testBowtie2Array_1234.out
#SBATCH --error=%x_%j.err # Standard error log, e.g., testBowtie2Array_1234.err
#SBATCH --array=0-9 # Array element range from 0 to 9, i.e. 10 element jobs

cd $SLURM_SUBMIT_DIR
ml Bowtie2/2.4.1-GCC-8.3.0 # Original data is split into 10 pieces and run in each element job
bowtie2 -x ./index/lambda_virus -U ./myreads_$SLURM_ARRAY_TASK_ID.fq \
-S output_$SLURM_ARRAY_TASK_ID.sam
```
Slurm headers for running a Threaded (OpenMP) job

```
#!/bin/bash
#SBATCH --job-name=testOpenMP # Job name (testOpenMP)
#SBATCH --partition=batch # Queue name (batch)
#SBATCH --nodes=1 # Run all processes on a single node
#SBATCH --ntasks=1 # Run in a single task on a single node
#SBATCH --cpus-per-task=12 # Number of CPU cores per task (12)
#SBATCH --mem=10G # Job memory limit (10 GB)
#SBATCH --export=NONE # Do not load any users’ explicit environment variables
#SBATCH --time=24:00:00 # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --output=%x_%j.log # Standard output and error log, e.g., testOpenMP_1234.log
#SBATCH --mail-type=END,FAIL # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu # Where to send mail

cd $SLURM_SUBMIT_DIR
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK # Sets the number of threads to use for OpenMP parallel regions
ml foss/2019b # Load toolchain module
./myProgram.x # Run your program binary compiled with OpenMP
```
Slurm headers for running an MPI job

```
#!/bin/bash
#SBATCH --job-name=testMPI # Job name (tesMPI)
#SBATCH --partition=batch # Queue name (batch)
#SBATCH --nodes=2 # Run on two nodes
#SBATCH --ntasks-per-node=16 # How many tasks on each node; Number of tasks=32=mpi ranks
#SBATCH --cpus-per-task=1 # Number of CPU cores per task; 16 CPU cores per node
#SBATCH --mem-per-cpu=500M # Memory per allocated CPU; 8GB (500MB*16) memory per node
#SBATCH --time=24:00:00 # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.log # Standard output and error log, e.g., testMPI_1234.log
#SBATCH --mail-type=END,FAIL # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu # Where to send mail

cd $SLURM_SUBMIT_DIR
ml foss/2019b # Load toolchain module
mpirun -n 32 ./myProgram.x # Run your program binary compiled with OpenMPI with 32 ranks
```
#!/bin/bash

#SBATCH --job-name=testHybrid  # Job name (testHybrid)
#SBATCH --partition=batch       # Queue name (batch)
#SBATCH --nodes=2              # Run on two nodes
#SBATCH --ntasks-per-node=8    # How many tasks on each node; Number of tasks=16=mpi ranks
#SBATCH --cpus-per-task=4      # Number of CPU cores per task; 32 CPU cores per node
#SBATCH --mem-per-cpu=500M     # Memory per allocated CPU; 16GB (500MB*32) memory per node
#SBATCH --time=24:00:00        # Time limit hrs:min:sec or days-hours:minutes:seconds
#SBATCH --export=NONE          # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.log      # Standard output and error log
#SBATCH --mail-type=END,FAIL   # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu  # Where to send mail

cd $SLURM_SUBMIT_DIR  # Change directory to job submission directory
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK  # Sets the number of threads to use for OpenMP parallel regions
ml foss/2019b        # Load toolchain module
mpirun -n 16 ./myProgram.x  # Run your program binary compiled with OpenMPI with 16 ranks
Slurm headers for running a GPU job

#!/bin/bash

#SBATCH --job-name=amber  # Job name
#SBATCH --partition=gpu_p  # Partition (queue) name, i.e., gpu_p
#SBATCH --gres=gpu:1  # Requests one GPU device; --gres=gpu:P100:1, --gres=gpu:K40:1
#SBATCH --ntasks=1  # Run a single task
#SBATCH --cpus-per-task=2  # Number of CPU cores per task
#SBATCH --mem=40gb  # Job memory request
#SBATCH --time=10:00:00  # Time limit hrs:min:sec
#SBATCH --export=NONE  # Do not load any users’ explicit environment variables
#SBATCH --output=%x_%j.out  # Standard output and error log, e.g., amber_1234.out
#SBATCH --mail-type=END,FAIL  # Mail events (BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu  # Where to send mail

cd $SLURM_SUBMIT_DIR
ml Amber/18-fosscuda-2018b-AmberTools-18-patchlevel-10-8
mpiexec $AMBERHOME/bin/pmemd.cuda -O -i ./prod.in -o prod_c4-23.out -p ./dimerFBP_GOL.prmtop -c ./restart.rst \ -r prod.rst -x prod.mdcrd
Slurm headers for running a Singularity container

```bash
#!/bin/bash
#SBATCH --job-name=test_sortmerna          # Job name
#SBATCH --partition=batch                  # Partition (queue) name
#SBATCH --ntasks=1                         # Run on a single CPU
#SBATCH --mem=8gb                          # Job memory request
#SBATCH --time=02:00:00                     # Time limit hrs:min:sec
#SBATCH --export=NONE                      # Do not load any users’ explicit environment variables
#SBATCH --output=sortmerna.%%j.out         # Standard output log, e.g., sortmerna.1234.out
#SBATCH --error=sortmerna.%%j.err          # Standard error log, e.g., sortmerna.1234.err
#SBATCH --cpus-per-task=4                  # Number of CPU cores per task
#SBATCH --mail-type=END,FAIL               # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=username@uga.edu      # Where to send mail

cd $SLURM_SUBMIT_DIR
singularity exec /apps/singularity-images/sortmerna-3.0.3.simg sortmerna --threads 4 --ref db.fasta,db.idx --reads file.fa \  --aligned base_name_output
```
General guidelines

- Do NOT use Login nodes to run CPU/memory intensive tasks directly ➔ submit jobs to Compute nodes!
- Do NOT use Login nodes to transfer data between your local computer and cluster ➔ use Transfer nodes!
- Do NOT use Home for storing job data ➔ use /scratch/MyID
- Do NOT park data in Scratch or Local Scratch ➔ clean up when job finishes or exits from node
- Do NOT park data permanently in Project ➔ download data to your local drive
- NO large memory job running on batch partition ➔ use highmem_p
- NO small memory job running on highmem_p partition ➔ use batch
- In general, number of threads you want to run with a parallel job = number of cores requested
- When you archive data using tar on /scratch, please do not use the z option (compression option). After you archived data with tar, you can use gzip to compress it.
General guidelines

• No directory should not have too many files inside! A rule of thumb would be to try to keep no more than a few tens of thousands of files (<10000 would be even better) in any single directory which is accessed frequently.

All files are in ONE single dir! ❌

Files are organized in subdirs! ✔️
Thank You!

Telephone Support
EITS Help Desk: 706-542-3106
Monday – Thursday: 7:30 a.m. – 7:30 p.m.
Friday: 7:30 a.m. – 6 p.m.
Saturday – Sunday: 1 p.m. – 7 p.m.

Georgia Advanced Computing Resource Center
101-108 Computing Services building
University of Georgia
Athens, GA 30602
https://gacrc.uga.edu/