Introduction to HPC Using Sapelo Cluster at GACRC

Georgia Advanced Computing Resource Center (GACRC)
EITS/University of Georgia
Zhuofei Hou  zhuofei@uga.edu
Outline

• GACRC
• Sapelo Cluster
• Job Submission Workflow
• Work with Sapelo
• Guideline and Practical Tips
GACRC

- We are a high-performance-computing (HPC) center at UGA
- We 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

- [http://wiki.gacrc.uga.edu](http://wiki.gacrc.uga.edu) (GACRC Wiki)
- [https://wiki.gacrc.uga.edu/wiki/Getting_Help](https://wiki.gacrc.uga.edu/wiki/Getting_Help) (GACRC Support)
- [http://gacrc.uga.edu](http://gacrc.uga.edu) (GACRC Web)
Sapelo Cluster

- Cluster Diagrams
- Cluster Overview
- Computing Resources
- Storage Environment
What is a Cluster?

When you work on cluster, 3 roles are working:

- User: to submit a job
- Queueing System: to dispatch a job to run on cluster
- Cluster: to run a job
Sapelo Cluster Diagram

- Node: **Computer** for a specific function on cluster, e.g., login node, work node
- Queue: Organization of **compute nodes** for a specific computing need
- Cluster: **Nodes + Drives**, all connected by network

**Login Node (sape101)**

**Work Node (Interactive Node)**

**Compute Nodes**

- AMD Compute Nodes
- HIGHMEM Compute Nodes
- GPU Compute Nodes

**batch queue**

**Home Drive /home** 100 GB

**Work Drive /lustre1** No quota limit

**Transfer Node (xier)**

**Project Drive /project** 1 TB

UGA MyID
Cluster Overview

Sapelo is a Linux high performance computing (HPC) cluster:

- **OS:** 64-bit CentOS Linux 6.5
- **You can log on (open connection) to:**
  - **Login (username@sapelo1.gacrc.uga.edu):** edit script, submit batch job *(not suggested!)*
  - **Transfer (username@xfer.gacrc.uga.edu):** transfer, compress, package data
  - **Login Work Node (qlogin):** edit script, submit batch job, run interactive job
  - **Internodal Communication:** **InfiniBand network**
    - compute nodes ↔ compute nodes ↔ storage systems
Cluster Overview

• 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

• On Sapelo:
  ➢ Torque Resource Manager + Moab Workload Manager
  ➢ Queueing commands:
    - qsub, qstat_me, qstat, qdel
    - showjobs, checkjob, showq
## Computing Resources

<table>
<thead>
<tr>
<th>Queue</th>
<th>Node Feature</th>
<th>Total Nodes</th>
<th>RAM(GB)/Node</th>
<th>Max RAM(GB)/Single-node Job</th>
<th>Cores /Node</th>
<th>Processor</th>
<th>GPU</th>
<th>GPU Cards /Node</th>
<th>InfiniBand</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>AMD</td>
<td>112</td>
<td>128</td>
<td>126</td>
<td></td>
<td>AMD Opteron</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>256</td>
<td>252</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HIGHMEM</td>
<td>6</td>
<td>512</td>
<td>504</td>
<td>48</td>
<td>AMD Opteron</td>
<td></td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1024 (1)</td>
<td>997</td>
<td>28</td>
<td>Intel Xeon</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1024 (3)</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td></td>
<td>2</td>
<td>128</td>
<td>126</td>
<td>16</td>
<td>NVIDIA K40m</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>abcnod (buy-in)</td>
<td></td>
<td>85</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
<td>variable</td>
<td></td>
</tr>
</tbody>
</table>

10/26/2017
Layer 1: Node
- A standalone “computer in a box”
- Multiple processors, e.g. 4, sharing memory
- Local disk storage, network interface, etc.
- Networked into a cluster

Layer 2: Processor
- A single computing component
- Multicore processor, e.g. 12 cores

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

Node 167

128 GB Shared RAM

48 cores and 128 GB RAM per node
## Storage Environment

<table>
<thead>
<tr>
<th>4 Filesystems</th>
<th>Role</th>
<th>Quota</th>
<th>Accessible from</th>
<th>Intended Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/username/</td>
<td>Home</td>
<td>100GB</td>
<td>sapelo1.gacrc.uga.edu (Login) Interactive nodes (Interactive) xfer.gacrc.uga.edu (Transfer) All compute nodes</td>
<td>Static data being used frequently, e.g., scripts, source codes, local software</td>
<td>Snapshots</td>
</tr>
<tr>
<td>/lustre1/username/</td>
<td>Global Scratch</td>
<td>No Limit</td>
<td>Interactive nodes (Interactive) xfer.gacrc.uga.edu (Transfer) All compute nodes</td>
<td>Current job data being read from/written into by jobs running on cluster</td>
<td>User to clean up! Subject to deletion in 30 days*</td>
</tr>
<tr>
<td>/tmp/lscratch/</td>
<td>Local Scratch</td>
<td>N/A</td>
<td>Individual compute node</td>
<td>Jobs with heavy disk I/O</td>
<td>User to clean up! When job exists from the node!</td>
</tr>
<tr>
<td>/project/abclab/</td>
<td>Storage</td>
<td>1TB (Initial)</td>
<td>xfer.gacrc.uga.edu (Transfer)</td>
<td>Temporary data parking for non-current active data</td>
<td>Group sharing possible</td>
</tr>
</tbody>
</table>

Note: /usr/local/apps : Software installation directory ; /db : Bioinformatics database installation directory
* denotes component or policy to be implemented in the future
Storage Environment - *Data Storing Rule*

I/O Performance

**Fast**
- `/lustre1/username/` ← **Current Job Data** being used by current running jobs on cluster
- `/home/username/` ← **Static Data** being used frequently, e.g., scripts or local software

**Slow**
- `/project/abclab/` ← **Non-current Active Data** to be analyzed in the future, e.g., 1 month

- **User Local Computer** ← **Final Data**, e.g., final outputs and results
Storage Environment - Accessing Rule of 123

```
/home/username
Landing
Login
ssh

/lustre1/username
Interactive
cd
qlogin

/project/abclab
Transfer
cd
ssh
```
## Storage Environment

<table>
<thead>
<tr>
<th>7 Main Functions</th>
<th>Related Filesystem</th>
<th>Related Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Landing</td>
<td>/home/username (Home) (Always!)</td>
<td>Login or Transfer</td>
</tr>
<tr>
<td>Batch Job Submitting</td>
<td>/home/username (Home)</td>
<td>Login</td>
</tr>
<tr>
<td></td>
<td>/lustre1/username (Global Scratch) (Suggested!)</td>
<td>Interactive</td>
</tr>
<tr>
<td>Interactive Job Running</td>
<td>/lustre1/username (Global Scratch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>/home/username (Home)</td>
<td></td>
</tr>
<tr>
<td>Data Transferring, Archiving, Compressing</td>
<td>/lustre1/username (Global Scratch)</td>
<td>Transfer</td>
</tr>
<tr>
<td></td>
<td>/home/username (Home)</td>
<td></td>
</tr>
<tr>
<td>Long-term Active Data Storing</td>
<td>/project/abclab (Project Storage)</td>
<td></td>
</tr>
<tr>
<td>Job Data Temporarily Storing</td>
<td>/lustre1/username (Global Scratch)</td>
<td>Compute</td>
</tr>
<tr>
<td></td>
<td>/tmp/lscratch (Local Scratch)</td>
<td></td>
</tr>
</tbody>
</table>
Job Submission Workflow

Job Working Space is Global Scratch: /lustre1/username/

- Why?
  No quota limit + The I/O fastest filesystem

- How to access?
  From Interactive Node, `cd /lustre1/username/`

- What need to pay attention?
  Clean up! Non-current Active Data ➔ /project/abclab/
  Non-active Data ➔ local storage

  using Transfer Node
Job Submission Workflow

1. Linux/Mac user:
   `ssh username@sapelo1.gacrc.uga.edu`

Windows user:

2. `qlogin`

3. `cd /lustre1/username`

Interactive

4. `mkdir ./workDir`

5. `cd ./workDir`

6. Linux/Mac user:
   `scp file username@xfer.gacrc.uga.edu:/lustre1/username/workDir`

Windows user:

7. `nano ./sub.sh`

   `#PBS -S /bin/bash`
   `#PBS -q batch`
   `#PBS -N testBlast`
   `#PBS -l nodes=1:ppn=1:AMD`
   `#PBS -l mem=20gb`
   `#PBS -l walltime=48:00:00`
   `cd $PBS_O_WORKDIR`
   `module load ncbiblast+/2.2.29`
   `blastn [options] > outputfile`

8. `$ qsub sub.sh`

Note: `-r` option of `scp` command will recursively copy a directory
Job Submission Workflow

1. Log on to Sapelo Login node: `ssh username@sapelo1.gacrc.uga.edu`
2. From Sapelo Login node, transfer to Interactive node by issuing the command: `qlogin`
3. From Interactive node: Change directory to global scratch: `cd /lustre1/username`
4. Create a working subdirectory on global scratch: `mkdir ./workDir`
5. Change directory to workDir: `cd ./workDir`
6. Transfer data to workDir using `scp` or SSH File Transfer (with `tar` or `gzip`)
7. Make a Sapelo job submission script: `nano ./sub.sh`
8. Submit job: `qsub ./sub.sh`

Useful PBS headers:  
`#PBS -l mem=200gb` : request 256GB high memory AMD nodes if mem>126gb!  
`#PBS -l nodes=1:ppn=24:AMD` : request 24 cores for 24 threads, max 48!
Work on Sapelo

Before we start:

GACRC Wiki: [http://wiki.gacrc.uga.edu](http://wiki.gacrc.uga.edu)

GACRC Wiki Running Jobs: [https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo](https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo)

GACRC Wiki Software: [https://wiki.gacrc.uga.edu/wiki/Software](https://wiki.gacrc.uga.edu/wiki/Software)

GACRC Wiki Linux Command: [https://wiki.gacrc.uga.edu/wiki/Command_List](https://wiki.gacrc.uga.edu/wiki/Command_List)

GACRC Support: [https://wiki.gacrc.uga.edu/wiki/Getting_Help](https://wiki.gacrc.uga.edu/wiki/Getting_Help)
Work on Sapelo

To submit a ticket to us?

- **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

---

Note:

It’s USER’s responsibility to make sure the correctness of datasets being used by jobs!
Work on Sapelo

- User Account
- Connect and Login
- Transfer Files Using Transfer Node
- Software Packages
- Run Interactive Jobs
- Run Batch Jobs
  - Submit Batch Jobs
  - **Serial, Threaded,** and **MPI** Batch Job Submission Scripts
  - Check Job Status and Cancel a Job
  - Check Computing Resources Used by a Job
User Account

- User Account: UGAMyID@sapelo1.gacrc.uga.edu

  A valid official UGA MyID is a MUST to create a user account!

- To get a user account:

  4. **Welcome letter** with whole package of information about your Sapelo user account
Connect and Login

• On Linux/Mac: use Terminal utility and `ssh` to your account:

```
ssh zhuofei@sapelol.gacrc.uga.edu
```

or

```
ssh -X zhuofei@sapelol.gacrc.uga.edu
```

(\(^1\) `X` is for *X windows application* running on the cluster with its UGI to be forwarded to local

(\(^2\) On Windows, use a *SSH client* to open the connection (next page))

• Logging in: You will be prompted for your *UGA MyID password*:

```
zhuofei@sapelol.gacrc.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:

```
[zhuofei@75-104 ~]$ exit
```
Connect and Login

1. To download:
   
   [link](http://eits.uga.edu/hardware_and_software/software/)
   
   with your UGA MyID and password

2. After connection is built, working environment is Linux, same as Linux/Mac users’
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 Sapelo (scp, sftp, rsync, SSH File Transfer, FileZilla, WinSCP)
✓ Compress or package data on Sapelo (tar, gzip)
✓ Transfer data between Sapelo and zcluster (cp, mv)
✓ Filesystems you can access using full path:
  ➢ /home/username/ : Sapelo home directory (landing directory)
  ➢ /lustre1/username/ : Sapelo global scratch directory
  ➢ /project/abclab/ : long-term active data storage for group
  ➢ /panfs/pstor.storage/home/abclab/username/ : zcluster home directory
  ➢ /escratch4/username/ : zcluster global scratch directory
✓ Most file systems on Transfer are auto-mounted upon the first time full-path access, e.g.,
  cd /project/abclab/
Linux/Mac Users Transfer Files Using scp/sftp

User’s local \(\xrightarrow{scp/sftp/rsync}\) Transfer (xfer.gacrc.uga.edu)

Type in Terminal utility on your local computer: \texttt{scp (-r) [Source] [Target]}

\textit{E.g. 1:} working on local machine, from Local \(\rightarrow\) Sapelo global scratch

\begin{verbatim}
scp ./file zhuofei@xfer.gacrc.uga.edu:/lustrel1/zhuofei/workDir/
scp -r ./folder/ zhuofei@xfer.gacrc.uga.edu:/lustrel1/zhuofei/workDir/
\end{verbatim}

\textit{E.g. 2:} working on local machine, from Sapelo global scratch \(\rightarrow\) Local

\begin{verbatim}
scp zhuofei@xfer.gacrc.uga.edu:/lustrel1/zhuofei/workDir/file ./
scp -r zhuofei@xfer.gacrc.uga.edu:/lustrel1/zhuofei/workDir/folder/ ./
\end{verbatim}
Windows Users Transfer Files Using SSH File Transfer/FileZilla

User’s local ssh client app ➔ Transfer (xfer.gacrc.uga.edu)

1. Connect to Remote Host
2. Host Name: xfer.gacrc.uga.edu
3. User Name: [email]
4. Port Number: 22
5. Transfer node!
6. Remote directory view
7. Local directory view
Software Packages

- The cluster uses environment modules to define the various paths for software packages (more than 600 and expanding daily!)
- **module avail**: List all available modules (centralized installation):

```
$ module avail

------------------------------------------------------------------- /usr/local/modulefiles
-------------------------------------------------------------------

Core/StdEnv         fftw/3.3.4/sgi149-mvapich200          mafft/7.273-e          python/2.7.8
Data/cache/moduleT.new  fftw/3.3.4/sgi149-omp183 (D)   mageck/0.5             python/2.7.13
Data/cache/moduleT   find_cns/07212016              mageck-vispr/02242017    python/3.4.3
(D)                   fineradstructure/12092016    magickwand/0.2          python/3.5.1 (D)
Data/system.txt      finestructure/2.0.7             magma/2.22-9            pytorch/0.1.9
R/3.0.3               flash/1.2.11                  mahotas/1.4.3           pytorch/02272017 (D)
R/3.1.2               flex/2.6.0/gcc/4.7.4            magma/1.3.3             qiime/1.9.1
R/3.2.1               flex/2.6.0/pgi/14.10            matplotlib/3.0.2        qt/4.8
R/3.3.1 (D)           freebayes/1.0.1                matplotlib/3.1           quast/4.1
StdEnv (D)            freesurfer/6.0.0               matplotlib/3.2          quast/4.2 (D)
......
```
Software Packages

• **module list**: List modules currently loaded:

```
$ module list
Currently Loaded Modules:
  1) moab/8.1.1  2) StdEnv
```

• **module load**: Load the needed module:

```
$ module load python/2.7.8
$ module load R/3.2.3
$ module list
Currently Loaded Modules:
  1) moab/8.1.1  2) StdEnv  3) python/2.7.8  4) R/3.2.3
```

• **module unload**: Remove a module

• **module spider**: Search a module:

```
$ module spider R/3.2.3
---------------------------------------------------------------
R: R/3.2.3
---------------------------------------------------------------
Description:
  statistics package ......
Run Interactive Jobs

- To run an interactive job, using `qlogin` command from Login node:

```
[zhuoifei@75-104 ~]$ qlogin
qsub: waiting for job 1058157.pbs.scm to start
qsub: job 1058157.pbs.scm ready
[zhuoifei@n15 ~]
[zhuoifei@n15 ~]$ module load R/3.2.3
[zhuoifei@n15 ~]$ R
```

Now I am on n15, which is an interactive node; Session time is 12 hours

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

...
[Previously saved workspace restored]
```

- When you are done, remember to `exit` the session
Run Batch Jobs

• Components you need to run a job:
  ➢ Software already installed (cluster software or the one installed by yourself)
  ➢ Job submission script to
    1. specify computing resources:
       ✓ number of nodes and cores
       ✓ amount of memory
       ✓ node’s feature
       ✓ maximum wallclock time
    2. load software using module load (for cluster software)
    3. run any Linux commands you want to run, e.g., pwd, mkdir, cd, echo, etc.
    4. run the software

• Common queueing commands you need:
  ➢ qsub, qstat_me, qstat, qdel
  ➢ showjobs, checkjob, showq
Submit Batch Jobs

[zhufei@n15 workDir]$ pwd  ➔ n15: interactive node
/lustre1/zhuofei/workDir  ➔ /lustre1/zhuofei/: global scratch

[zhufei@n15 workDir]$ qsub sub.sh

1165617.pbs.scm

**qsub** is to submit a job

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

1. specify computing resources:
2. load software using module load
3. run any Linux commands you want to run
4. run the software
Example 1: **Serial job script** *sub.sh* running NCBI Blast +

```plaintext
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=1:AMD
#PBS -l mem=20gb
#PBS -l walltime=48:00:00

cd $PBS_O_WORKDIR

module load ncbiblast+/2.2.29

time blastn [options] >outputfile 2>&1
```

- Linux shell (bash)
- Queue name (batch)
- Name of the job (testBlast)
- Number of nodes (1), number of cores/node (1), node type (AMD)
- Maximum amount of physical memory (20 GB) used by the job
- Maximum wall clock time (48 hours) for the job, default 6 minutes
- Use the directory from which the job is submitted as the working directory
- Load the module of ncbiblast+, version 2.2.29
- Run blastn with ‘time’ command to measure the amount of time it takes to run the application
Example 2: Threaded job script `sub.sh` running NCBI Blast+

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=24:AMD
#PBS -l mem=20gb
#PBS -l walltime=480:00:00

#PBS -M jSmith@uga.edu
#PBS -m ae
#PBS -j oe

cd $PBS_O_WORKDIR

module load ncbiblast+/2.2.29

```

- **Number of nodes (1), number of cores/node (24), node type (AMD)**
  - *Number of threads (24) = Number of cores requested (24)*

- **Email address to receive a notification for computing resources**
- **Send email notification when job aborts (a) or terminates (e)**
- **Standard error file (testBlast.e1234) will be merged into standard out file (testBlast.o1234)**

```
time blastn -num_threads 24 [options] >outputfile 2>&1
```

- **Run blastn with 24 threads (-num_threads 24)**
Example 3: MPI job script `sub.sh` running RAxML

```bash
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=48:AMD
#PBS -l walltime=480:00:00
#PBS -l mem=20gb

cd $PBS_O_WORKDIR

module load raxml/8.1.20
module load intel/15.0.2
module load openmpi/1.8.3/intel/15.0.2

mpirun –np 50 raxmlHPC-MPI-AVX [options] >outputfile 2>&1
```

- Number of nodes (2), number of cores/node (48), node type (AMD)
- Total cores requested = 2 × 48 = 96
  
*We suggest, Number of MPI Processes (50) ≤ Number of cores requested (96)*

- To run raxmlHPC-MPI-AVX, MPI version using OpenMPI 1.8.3/Intel 15.0.2
- Run raxmlHPC-MPI-AVX with 50 MPI processes (–np 50), default 96
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=27:AMD
#PBS -l walltime=480:00:00
#PBS -l mem=20gb

cd $PBS_O_WORKDIR

# Context Sharing
CONTEXTS=$(/usr/local/bin/set_contexts.sh $PBS_NUM_PPN)
if [[ "$?" -eq "0" ]]; then
  export PSM_SHAREDCONTEXTS_MAX=$CONTEXTS
fi

module load raxml/8.1.20
module load intel/15.0.2
module load openmpi/1.8.3/intel/15.0.2

mpirun --np 50 raxmlHPC-MPI-AVX [options] >outputfile 2>&1

ppn number (27) fewer than 48 MUST be a multiplier of 3!

Run raxmlHPC-MPI-AVX with 50 MPI processes (--np 50), default 54
Check Job Status (\texttt{qstat\_me}) and Cancel a Job (\texttt{qdel})

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Name</th>
<th>User</th>
<th>Time Use</th>
<th>S</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>481929.pbs</td>
<td>testJob1</td>
<td>jSmith</td>
<td>900:58:10</td>
<td>C</td>
<td>batch</td>
</tr>
<tr>
<td>481931.pbs</td>
<td>testJob2</td>
<td>jSmith</td>
<td>04:00:03</td>
<td>R</td>
<td>batch</td>
</tr>
<tr>
<td>481934.pbs</td>
<td>testJob3</td>
<td>jSmith</td>
<td>0</td>
<td>C</td>
<td>batch</td>
</tr>
</tbody>
</table>

R: job is running
C: job completed (or canceled or crashed) and is not longer running. Jobs stay in this state for 1 hour
Q: job is pending, waiting for resources to become available
Note: “Time Use” is the CPU time, instead of the wall-clock time!

$ \texttt{qdel} 481934$

$ \texttt{qstat\_me}$

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Name</th>
<th>User</th>
<th>Time Use</th>
<th>S</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>481929.pbs</td>
<td>testJob1</td>
<td>jSmith</td>
<td>900:58:00</td>
<td>C</td>
<td>batch</td>
</tr>
<tr>
<td>481931.pbs</td>
<td>testJob2</td>
<td>jSmith</td>
<td>04:00:03</td>
<td>R</td>
<td>batch</td>
</tr>
</tbody>
</table>
| 481934.pbs | testJob3   | jSmith| 0        | C  | batch   | 1hr on list
Check All Jobs on Cluster (**qstat**)  

```
$ qstat
```

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Name</th>
<th>User</th>
<th>Time Use S</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>2392787.pbs</td>
<td>G_s-Butyl_0_opt</td>
<td>gtp72866</td>
<td>101:34:5</td>
<td>R batch</td>
</tr>
<tr>
<td>2392792.pbs</td>
<td>G_s-Butyl_20_opt</td>
<td>gtp72866</td>
<td>56:14:32</td>
<td>R batch</td>
</tr>
<tr>
<td>2393599.pbs</td>
<td>...80-oz0.6-K220</td>
<td>rbchan</td>
<td>89:99:59</td>
<td>R batch</td>
</tr>
<tr>
<td>2395891.pbs</td>
<td>NoRiver_jul_2008</td>
<td>hluo76</td>
<td>0</td>
<td>Q batch</td>
</tr>
<tr>
<td>2395893.pbs</td>
<td>NoRiver_jul_2012</td>
<td>hluo76</td>
<td>0</td>
<td>Q batch</td>
</tr>
<tr>
<td>2396261.pbs</td>
<td>...opyl_freq_005</td>
<td>gtp72866</td>
<td>31:55:56</td>
<td>R batch</td>
</tr>
<tr>
<td>2396297.pbs</td>
<td>...opyl_freq_041</td>
<td>gtp72866</td>
<td>202:53:1</td>
<td>R batch</td>
</tr>
<tr>
<td>2400998.pbs</td>
<td>...rrect_lab4.sh</td>
<td>xwwang</td>
<td>844:06:3</td>
<td>R batch</td>
</tr>
<tr>
<td>2406268.pbs</td>
<td>abundTrinity3</td>
<td>mpodio</td>
<td>0</td>
<td>Q batch</td>
</tr>
<tr>
<td>2423833.pbs</td>
<td>L80-307</td>
<td>jx57780</td>
<td>393:37:1</td>
<td>C batch</td>
</tr>
<tr>
<td>2423838.pbs</td>
<td>L80-312</td>
<td>jx57780</td>
<td>373:57:4</td>
<td>C batch</td>
</tr>
<tr>
<td>2513741.pbs</td>
<td>...3012_ah100_00</td>
<td>byang</td>
<td>00:00:00</td>
<td>R batch</td>
</tr>
<tr>
<td>2513743.pbs</td>
<td>...4012_ah070_01</td>
<td>byang</td>
<td>138:22:3</td>
<td>R batch</td>
</tr>
</tbody>
</table>

Note: `qstat` command will give you a long list of all jobs from all users on cluster!
Check Jobs on Which Nodes (`qstat -n -u MyID`)

```
$ qstat -n -u mheiges
pbs.scm:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Req'd Memory</th>
<th>Req'd Time</th>
<th>S</th>
<th>Elap Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2554422.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>STDIN</td>
<td>395315</td>
<td>1</td>
<td>1</td>
<td></td>
<td>480:00:00</td>
<td>R</td>
<td>00:01:16</td>
</tr>
<tr>
<td>n558/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554425.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>STDIN</td>
<td>395393</td>
<td>1</td>
<td>1</td>
<td></td>
<td>480:00:00</td>
<td>R</td>
<td>00:01:16</td>
</tr>
<tr>
<td>n558/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554432.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>STDIN</td>
<td>395497</td>
<td>1</td>
<td>1</td>
<td></td>
<td>480:00:00</td>
<td>R</td>
<td>00:01:15</td>
</tr>
<tr>
<td>n558/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554436.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>STDIN</td>
<td>395690</td>
<td>1</td>
<td>1</td>
<td></td>
<td>480:00:00</td>
<td>R</td>
<td>00:01:15</td>
</tr>
<tr>
<td>n558/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554467.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>DistribJob</td>
<td>404662</td>
<td>1</td>
<td>1</td>
<td>8gb</td>
<td>240:00:00</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>n558/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554468.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>DistribJob</td>
<td>404695</td>
<td>1</td>
<td>1</td>
<td>8gb</td>
<td>240:00:00</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>n558/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2554469.pbs.scm</td>
<td>mheiges</td>
<td>batch</td>
<td>DistribJob</td>
<td>405046</td>
<td>1</td>
<td>1</td>
<td>8gb</td>
<td>240:00:00</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>
```

Note: “Elap Time” is the wall-clock time, instead of the CPU time, which qstat_me can give you!
Check Computing Resources Used by a Job

**Option 1:** `qstat -f JobID` for *running jobs* or *finished jobs in 1 hour*

**Option 2:** `showjobs JobID` for *finished jobs over 1 hour, but ≤ 7 days*

**Option 3:** Email notification from *finished jobs (completed, canceled, or crashed)*,

if using:

```
#PBS -M jSmith@uga.edu
#PBS -m ae
```
qstat -f JobID for running jobs or finished jobs in 1 hour

```bash
$ qstat -f 1426778
Job Id: 1426778.pbs.scm
  Job_Name = BS2_GFF_range_methylator5-10.py
  Job_Owner = gent@uga-2f0f976.scm
  resources_used.cput = 76:54:46
  resources_used.energy_used = 0
  resources_used.mem = 44136kb
  resources_used.vmem = 90200kb
  resources_used.walltime = 76:50:04
  job_state = C
  queue = batch
  Error_Path = uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.e1426778
  exec_host = n2/7
  Output_Path = uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.o1426778
  Resource_List.mem = 2gb
  Resource_List.nodect = 1
  Resource_List.nodes = 1:ppn=1:AMD
  Resource_List.walltime = 480:00:00
  Variable_List = PBS_O_QUEUE=batch,PBS_O_HOME=/home/gent, .... ,
  PBS_O_WORKDIR=/home/gent/scripts,
```
showjobs  JobID  for finished jobs over 1 hour, but ≤ 7 days

$ showjobs 1426778

Job Id            : 1426778.pbs.scm
Job Name          : BS2_GFF_range_methylator5-10.py
Output File       : uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.o1426778
Error File        : uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.e1426778
Working Directory : /home/gent/scripts
Home Directory    : /home/gent
Submit Arguments  : BS2_GFF_range_methylator5-10.py.sh
User Name         : gent
Group Name        : krdlab
Queue Name        : batch
Wallclock Limit   : 20:00:00:00
Wallclock Duration: 3:04:50:04
CPUTime           : 3:04:54:46
Memory Used       : 43.1Mb
Memory Limit      : 2gb
vmem Used         : 88.1Mb
Submit Time       : Mon Mar 27 20:51:01 2017
Start Time        : Tue Mar 28 03:06:41 2017
End Time          : Fri Mar 31 07:58:32 2017
Exit Code         : 0
Master Host       : n2
Email notification from finished jobs

PBS Job Id: 700009.pbs.scm
Job Name: testJob
Exec host: n1/4-27
Execution terminated
Exit_status=0
resources_used.cput=00:05:12
resources_used.energy_used=0
resources_used.mem=410984kb
resources_used.vmem=6548516kb
resources_used.walltime=04:00:59
Error_Path: uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.o700009
Output_Path: uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.o700009

PBS Job Id: 700097.pbs.scm
Job Name: testJob
Exec host: n5/4-27
Execution terminated
Exit_status=271
resources_used.cput=00:11:22
resources_used.energy_used=0
resources_used.mem=412304kb
resources_used.vmem=6548524kb
resources_used.walltime=05:00:41
Error_Path: uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.o700097
Output_Path: uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.o700097
Check Queue Status (*showq*)

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USERNAME</th>
<th>STATE</th>
<th>PROCS</th>
<th>REMAINING</th>
<th>STARTTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>481914</td>
<td>brant</td>
<td>Running</td>
<td>1</td>
<td>20:46:21</td>
<td>Fri Jun 12 11:32:23</td>
</tr>
<tr>
<td>481915</td>
<td>brant</td>
<td>Running</td>
<td>1</td>
<td>20:48:56</td>
<td>Fri Jun 12 11:34:58</td>
</tr>
<tr>
<td>481567</td>
<td>becton</td>
<td>Running</td>
<td>288</td>
<td>2:04:15:48</td>
<td>Wed Jun 10 15:01:50</td>
</tr>
</tbody>
</table>

108 active jobs

5141 of 5740 processors in use by local jobs (89.56%)  
121 of 122 nodes active (99.18%)

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USERNAME</th>
<th>STATE</th>
<th>PROCS</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>481821</td>
<td>joykai</td>
<td>Idle</td>
<td>48</td>
<td>50:00:00:00:00:00</td>
<td>Thu Jun 11 13:41:20</td>
</tr>
<tr>
<td>481813</td>
<td>joykai</td>
<td>Idle</td>
<td>48</td>
<td>50:00:00:00:00:00</td>
<td>Thu Jun 11 13:41:19</td>
</tr>
<tr>
<td>481811</td>
<td>joykai</td>
<td>Idle</td>
<td>48</td>
<td>50:00:00:00:00:00</td>
<td>Thu Jun 11 13:41:19</td>
</tr>
</tbody>
</table>

50 eligible jobs

0 blocked jobs

Total jobs: 158
Guideline Tips

• Do NOT use Login Node to run jobs ➔ Interactive Node or the queue

• Do NOT use Login Node upload or download data to/from cluster

• Do NOT use Login Node to transfer data to the home dir

• NO large memory job running on AMD nodes ➔ HIGHMEM nodes

• NO small memory job running on HIGHMEM nodes ➔ AMD nodes

• As a general rule, threads # = cores # requested
Practical Tips

• Each 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!
Practical Tips

- Job name should have a specific computational meaning
  
  **Good Examples:** 
  
  #PBS -N blastn_dataSet1_trail2 ; #PBS -N M-10-1121
  
  **Bad Examples:** 
  
  #PBS -N job1 ; #PBS -N bowtie ; #PBS -N 20160930

- The stdout .o file and stderr .e file are to be written into files at the finishing time of a job.
  
  Redirect standard output and error of the application to a file, instead of letting it be written in the stdout .o file and stderr .e file of the job, e.g.:

  
  
  time application >file 2>&1

- Monitor job progress from time to time, to catch if a job gets stuck
Useful Links

• GACRC Web: http://gacrc.uga.edu/
• GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
• GACRC Support : https://wiki.gacrc.uga.edu/wiki/Getting_Help
• GACRC Training: https://wiki.gacrc.uga.edu/wiki/Training
• GACRC User Account: https://wiki.gacrc.uga.edu/wiki/User_Accounts
• GACRC Software: https://wiki.gacrc.uga.edu/wiki/Software
Thank You!