Introduction to High Performance Computing (HPC) Resources at GACRC

Georgia Advanced Computing Resource Center
University of Georgia
Zhuofei Hou, HPC Trainer
zhuofei@uga.edu
Outline

• GACRC?
• High Performance Computing (HPC)
• GACRC Sapelo Cluster
GACRC

Who Are We?
- Georgia Advanced Computing Resource Center
- 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 (Web)
- http://wiki.gacrc.uga.edu (Wiki)
- http://gacrc.uga.edu/help/ (Web Help)
- https://wiki.gacrc.uga.edu/wiki/Getting_Help (Wiki Help)
High Performance Computing (HPC)

Serial:
- Serial problem can not be broken
- Discrete instructions executed sequentially
- Only 1 instruction executed at any moment on a single processor

Parallel:
- Problem broken into parallel parts can be solved concurrently
- Instructions executed simultaneously on multiply processors
- Synchronization/communication employed
- Shared-memory multithreaded job or MPI job (Message Passing Interface)
GACRC Sapelo Cluster

- Cluster Structural Diagram
- Cluster Overview
- Computing Resources
- Storage Environment
- Software Modules
- Run Batch Jobs: Workflow, Submit Job, and Job Submission Script
Network Connection Speed:

- InfiniBand
- 10 Gbit Ethernet
- 1 Gbit Ethernet

The New GACRC Linux HPC Cluster Structural Diagram

- Aggregator Switch
- Storage Switch
- Top of Rack Switch
- Login Switch

**InfiniBand Switch**

**Icebreaker Storage**
- Node index: sn3
- /project

**Seagate Storage**
- Lustre Parallel File System
- /lustre1

**Transfer Node**
- xfer.gacrc.uga.edu
- Build Node
  - build1.gacrc.uga.edu
  - AMD Opteron 16-core processor, 64 GB RAM

**Compute Nodes**
- Node index: n0, n1, n2, n165, n166, n167
- CPU: AMD Opteron / Intel Xeon multicore processors
- GPU: NVIDIA Tesla (Kepler) K40m GPU cards
- /lgscratch

**User**
- sapelo1.gacrc.uga.edu
- Intel Xeon processor
Cluster Overview

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

- OS: 64-bit CentOS Linux 6.5
- You can log on to:
  - Login (username@sapelo1.gacrc.uga.edu): edit script, submit batch job
  - Transfer (username@xfer.gacrc.uga.edu): transfer, compress, package data
- Login Interactive Node: run interactive job, edit script, submit batch job
- Queueing System: Torque + Moab with qsub, qstat, qdel commands
## Computing Resources

### Computing Resources Table

<table>
<thead>
<tr>
<th>Queue</th>
<th>Node Feature</th>
<th>Total</th>
<th>Processor</th>
<th>Cores /Node</th>
<th>RAM (GB) /Node</th>
<th>Max RAM (GB) /Single-node Job</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>AMD Opteron</td>
<td>48</td>
<td>128</td>
<td>126</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td>256</td>
<td>252</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHMEM</td>
<td>AMD</td>
<td>7</td>
<td>AMD Opteron</td>
<td>48</td>
<td>512 (6)</td>
<td>504</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1024 (1)</td>
<td>997</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>Intel Xeon</td>
<td>2</td>
<td></td>
<td>16</td>
<td>128</td>
<td>126</td>
<td>NVIDIA K40m</td>
<td>8</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Peak Performance per Node: 500Gflops/Node

Home: /home/username: **100GB**

Global scratch: /lustre1/username: **NO quota limit, auto-moved to /project if no modification in 30 days!**
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

I/O bus

Local Storage
/I/O scratch

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

<table>
<thead>
<tr>
<th>4 Filesystems</th>
<th>Role</th>
<th>Quota</th>
<th>Intended Use</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home/username</td>
<td>Home</td>
<td>100GB</td>
<td>Highly static data being used frequently</td>
<td>Snapshots</td>
</tr>
<tr>
<td>/lustre1/username</td>
<td>Global Scratch</td>
<td>No Limit</td>
<td>Temporarily storing large data being used by jobs</td>
<td>Auto-moved to /project if 30 days no modification</td>
</tr>
<tr>
<td>/project/abclab</td>
<td>Storage</td>
<td>Variable</td>
<td>Long-term data storage</td>
<td>Group sharing possible</td>
</tr>
</tbody>
</table>

**Note:**  
/ usr/local/apps : Software installation directory  
/db : Bioinformatics database installation directory
Storage Environment

Accessing Rule of 123:

1. Login
2. Interactive
3. Transfer

/qlogin
Software Modules

• Sapelo uses environment modules to define paths for software
• Current number of modules installed is ~300 and expanding daily!
• module avail ➔ List all modules available on Sapelo
  module list ➔ List modules currently being loaded for use
  module load ➔ Load modules needed
  module unload ➔ Unload modules not needed
#!/bin/bash

# Import necessary modules
source /usr/local/modulesfiles/mibosystem
source /usr/local/modulesfiles/core
source /usr/local/modulesfiles/net
source /usr/local/modulesfiles/db
source /usr/local/modulesfiles/python
source /usr/local/modulesfiles/perl
source /usr/local/modulesfiles/compile

# Set environment variables
MIBOSYSTEM=/usr/local/mibosystem
CORE=/usr/local/modulesfiles/core
NET=/usr/local/modulesfiles/net
DB=/usr/local/modulesfiles/db
PYTHON=/usr/local/modulesfiles/python
PERL=/usr/local/modulesfiles/perl
COMPILE=/usr/local/modulesfiles/compile

# Set default module
module load default

# Set arguments
arg1=1
arg2=2
arg3=3

# Execute commands
command1 arg1 arg2 arg3
command2 arg1 arg2 arg3

# Clean up
module unload default
exit 0
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10

[zhuofei@75-104 ~]$ module load python/2.7.8
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10   3) python/2.7.8

[zhuofei@75-104 ~]$ exit
logout
Connection to sapel01.gacrc.uga.edu closed.
zhuofei@zcluster:~$ ssh zhuofei@sapel01.gacrc.uga.edu
zhuofei@sapel01.gacrc.uga.edu's password:
The following have been reloaded with a version change:
  1) moab/8.1.1 => moab/7.2.10

[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10
Run Batch Jobs

• Components you need to run a job:
  - Software already loaded. If not, used module load
  - Job submission script to run the software, and specify computing resources:
    - Number of nodes and cores
    - Amount of memory
    - Type of nodes
    - Maximum wallclock time, etc.

• Common commands you need: qsub, qstat, qdel
Run Batch Jobs – Submit Job

[qhuofei@n15 workDir]$ pwd
© n15: interactive node
© /lustre1/zhuofei/workDir: global scratch

[qhuofei@n15 workDir]$ qsub sub.sh
1165617.pbs.scm

$qsub$ is to submit a job

$sub.sh$ is your job submission script specifying:

- Number of nodes and cores
- Amount of memory
- Type of nodes
- Maximum wallclock time, etc.
Run Batch Jobs – Job Submission Script

- Example: **Serial job submission script** `sub.sh` running NCBI Blast +

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>#PBS -S /bin/bash</code></td>
<td>Linux shell (bash)</td>
</tr>
<tr>
<td><code>#PBS -q batch</code></td>
<td>Queue name (batch)</td>
</tr>
<tr>
<td><code>#PBS -N testBlast</code></td>
<td>Name of the job (testBlast)</td>
</tr>
<tr>
<td><code>#PBS -l nodes=1:ppn=1:AMD</code></td>
<td>Number of nodes (1), number of cores/node (1), node type (AMD)</td>
</tr>
<tr>
<td><code>#PBS -l mem=20gb</code></td>
<td>Maximum amount of physical memory (20 GB) used by the job</td>
</tr>
<tr>
<td><code>#PBS -l walltime=48:00:00</code></td>
<td>Maximum wall clock time (48 hours) for the job, default 6 minutes</td>
</tr>
<tr>
<td><code>cd $PBS_O_WORKDIR</code></td>
<td>Use the directory from which the job is submitted as the working directory</td>
</tr>
<tr>
<td><code>module load ncbiblast+/2.2.29</code></td>
<td>Load the module of ncbiblast+, version 2.2.29</td>
</tr>
<tr>
<td><code>time blastn [options] &gt; outputfile</code></td>
<td>Run blastn with ‘time’ command to measure the amount of time it takes to run the application</td>
</tr>
</tbody>
</table>
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:

  2. User Account Request: http://help.gacrc.uga.edu/userAcct.php (for PI of an existing group)
  3. New User Training: http://gacrc.uga.edu/help/training/
  4. Welcome letter with whole package of information about your Sapelo user account
Useful Links

• GACRC Web: http://gacrc.uga.edu/
• GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
• GACRC Help: http://gacrc.uga.edu/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!