Using the computational resources at the GACRC
An introduction to zcluster

Shan-Ho Tsai

Georgia Advanced Computing Resource Center (GACRC)
University of Georgia

Dr. Landau’s PHYS8602 course - Spring 2017
What is GACRC?

- Georgia Advanced Computing Resource Center (GACRC)
- Provides computing hardware and network infrastructure in support of high-performance computing at UGA.
- Collaboration between the Office of the Vice President for Research (OVPR) and the Office of the Vice President for Information Technology (OVPIT).
- Guided by a faculty advisory committee (GACRC-AC).
- Over 1000 users, from more than 230 groups and over 50 departments.

GACRC’s Website: http://gacrc.uga.edu

GACRC’s Wiki: https://wiki.gacrc.uga.edu

GACRC’s Linux cluster: zcluster
What is zcluster?

GACRC zcluster is a Linux high performance computing cluster.

Operating system: 64-bit Red Hat Enterprise Linux 5

zcluster consists of:

- A login node
- Many compute nodes
- Some interactive nodes
- Queueing system software to run jobs on the compute nodes and on the interactive nodes.
## CPU resources on zcluster

The zcluster has the following nodes:

<table>
<thead>
<tr>
<th>Nodes</th>
<th>cores/node</th>
<th>CPU type</th>
<th>RAM/node</th>
<th>Total cores</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>12</td>
<td>Intel Xeon</td>
<td>48GB</td>
<td>540</td>
<td>Regular</td>
</tr>
<tr>
<td>150</td>
<td>8</td>
<td>Intel Xeon</td>
<td>16GB</td>
<td>1200</td>
<td>Regular</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>Intel Xeon</td>
<td>192GB</td>
<td>32</td>
<td>Medium memory</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>Intel Xeon</td>
<td>256GB</td>
<td>120</td>
<td>Medium memory</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>Intel Xeon</td>
<td>512GB</td>
<td>64</td>
<td>Big memory</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>AMD Opteron</td>
<td>64GB</td>
<td>192</td>
<td>Many cores</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>AMD Opteron</td>
<td>132GB</td>
<td>96</td>
<td>Interactive</td>
</tr>
</tbody>
</table>

Total peak performance: 23 Tflops
Hardware resources

GPU resources on zcluster

The zcluster has the following NVIDIA GPU resources:

- Tesla S1070, 4 GPU cards, each with 30 multiprocessors (MP) and 8 cores/MP = 960 cores
- Tesla M2070 (Fermi), 12 GPU cards, each with 14 multiprocessors and 32 cores/MP = 5376 cores
- Tesla C2075 (Fermi), 1 GPU card = 448 cores
- Tesla K20Xm (Kepler), 32 GPU cards, each with 14 multiprocessors and 192 cores/MP = 86,016 cores
How do I connect to zcluster?

**From Mac OSX and Unix/Linux:**
- Open a Terminal window
- Use SSH to login to `zcluster.rcc.uga.edu` (login node).

```
local$ ssh -X username@zcluster.rcc.uga.edu
```

where `username` is your zcluster username (e.g. s_21). Enter your password when prompted for it.

**From Windows:**
- Install an SSH Secure Shell Utility for Windows, such as the one available at
  
  [https://eits.uga.edu/hardware_and_software/software](https://eits.uga.edu/hardware_and_software/software).
- Open an SSH connection to `zcluster.rcc.uga.edu`
How do I transfer files to/from zcluster?

From Mac OS X and Unix/Linux:

- Use SCP to transfer files between your local machine and zcluster.
- Open a connection to copy.rcc.uga.edu (copy nodes).

Examples:
- To copy a file from your local machine to zcluster, type the following at a terminal on your local machine

  ```
  local$ scp filename username@copy.rcc.uga.edu:
  ```

- To copy a file from zcluster to your local machine, type the following at a terminal on your local machine

  ```
  local$ scp username@copy.rcc.uga.edu:filename
  ```
How do I connect to zcluster? - continued

**From Windows:**

- Use SSH Secure File transfer and connect to `copy.rcc.uga.edu` (copy nodes).
- Use the drag and drop feature to transfer files between your local machine and zcluster.

For more information on how to transfer files to/from zcluster and between filesystems on zcluster, please see https://wiki.gacrc.uga.edu/wiki/Transferring_Files
Introduction

Linux Cluster - zcluster

Software installed

Running Jobs

Support and Reporting Problems

OS and directory structure

- Operating System: 64-bit Linux (RHEL5)

Once you login, you will get a shell prompt

Shell prompt

s_21@zcluster:/home/student/phys8601/s_21#

And you will be at the top level of your home directory.

Check your path

s_21@zcluster:/home/student/phys8601/s_21# pwd
/home/student/phys8601/s_21

The directory path for the top level of your home directory can be referenced as ${HOME} in your scripts.
You can create directories within other directories. Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkdir dirname</td>
<td>Create a directory called dirname.</td>
</tr>
<tr>
<td>cd dirname</td>
<td>Change to the dirname directory.</td>
</tr>
<tr>
<td>cd ..</td>
<td>Go one level up.</td>
</tr>
<tr>
<td>cd</td>
<td>Go to the top level of your home dir.</td>
</tr>
</tbody>
</table>

```bash
s_21@zcluster:/home/student/phys8601/s_21# mkdir test
s_21@zcluster:/home/student/phys8601/s_21# cd test
s_21@zcluster:/home/student/phys8601/s_21/test# pwd
/home/student/phys8601/s_21/test
s_21@zcluster:/home/student/phys8601/s_21/test# mkdir projectA
s_21@zcluster:/home/student/phys8601/s_21/test# mkdir projectB
s_21@zcluster:/home/student/phys8601/s_21/test# cd projectA
s_21@zcluster:/home/student/phys8601/s_21/test/projectA# cd ../projectB
s_21@zcluster:/home/student/phys8601/s_21/test/projectB#
```
Unix Shell

- What is the Unix Shell: program that defines the environment of a terminal and interprets commands typed at the prompt.
- Basic shells: bash, sh, csh, tcsh, ksh, zsh

Check the shell in use

```
s_21@zcluster:/home/student/phys8601/s_21# echo $SHELL
/bin/bash
```

For simplicity, in the examples below the shell prompt used will be

```
zcluster#
```
Changing the password

Use the `passwd` command to change your password.

To change the password

```
zcluster# passwd
Changing password for user s_21.
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```
Listing files and file properties

- **ls**: List the contents (files and subdirectories) of a directory.
- **ls -a**: List all files, including hidden ones that start with dot.
- **ls -l**: Show additional info of the files and dirs.
- **ls -lh**: Shows sizes in human readable format.
- **ls -lt**: List the contents of a dir in cronological order.

```
zcluster# ls -lha
```
```
  drwx------  2 s_21 student  4.0K Dec 11 2012 s_empty
  -rw-------  1 s_21 student  1.3K Mar 13 15:35 .bashrc
  -rw-------  1 s_21 student  79M Feb 20 2013 demo.fa
  -rw-------  1 s_21 student  513 Dec 11 2012 sub.sh
  lrwxrwxrwx  1 root root  46 Jan 10 04:26 submitted_homework
```

File permissions: 
```
d: directory, r: read, w: write, x: execute
```
```
drwxrwxrwx: by user by group by others
```
Use the **chmod** command to change permissions.
Manipulating files and directories

- `cp filename dirname` : Copy a file into a directory.
- `cp file1 file2` : Copy a file into another.
- `cp -i file dir` : Copy a file into dir, ask before overwriting.
- `cp -r dir1 dir2` : Copy a directory with its contents.
- `mv filename dirname` : Move a file into a directory.
- `mv file1 file2` : Renames a file.
- `mv -i file dir` : Move a file and ask before overwriting.
- `rm filename` : Remove a file.
- `rm -r dirname` : Remove a directory with its contents.
- `rm -i filename` : Ask before removing a file.
- `rm -i -r dirname` : Confirm deletion of dir and its contents.
- `mkdir dirname` : Make a directory.
- `rmdir dirname` : Remove an empty directory.
View file contents and text editors

- **file filename**: Determine the type of a file.
- **less filename**: View text files and paginate them if needed.

Converting files to Linux format

- **dos2unix filename**: Removes DOS/Windows line endings in file.
- **mac2unix filename**: Removes Mac line endings in file.

Text Editors

- vi, vim, emacs, xemacs, nano, nedit.

Manual pages for Linux commands: **man command**

For example:

- **man ls**
- **man chmod**
- **man dos2unix**
Applications

Applications installed on zcluster

- Compilers: Intel, Portland Group and GNU C/C++ and Fortran; LLVM-Clang, Java
  https://wiki.gacrc.uga.edu/wiki/Code_Comilation_on_zcluster
- Debuggers: GNU gdb, PGI pgdbg (with MPI support, GUI)
- Profilers: GNU gprof, PGI pgprof
- Math libraries: BLAS, LAPACK, FFTW, GSL, etc
- MPI libraries: MPICH, MPICH2, OpenMPI
  https://wiki.gacrc.uga.edu/wiki/MPI
- Perl, Python, awk, sed
- Matlab, Maple, R
- Queueing system: Univa Grid Engine

For a list of applications installed see:
https://wiki.gacrc.uga.edu/wiki/Software
Write a Fortran77 code

Option 1:
- Write the code on your local machine
- Transfer it to zcluster
- Use dos2unix or mac2unix to convert it to Linux

Option 2:
- Use a text editor on zcluster to create/edit the code. Example:

```
zcluster# nano mult.f
```

Sample code: `/home/student/phys8601/ins_data/mult.f`
You can copy it to your home directory:

```
zcluster# cp ../ins_data/mult.f .
```
How to compile a code

Sample code compilation commands on zcluster

To compile a Fortran77 code called *mult.f*

```
zcluster#  pgf77  mult.f  -o  mult.x
```

To compile a C code called *mult.c)*

```
zcluster#  pgcc  mult.c  -o  mult.x
```

These will create an executable named *mult.x*.

On zcluster, type `man pgf77` or `man pgcc` at a terminal prompt to get the documentation for these compilers.

For more information, please see https://wiki.gacrc.uga.edu/wiki/CodeCompilation_on_zcluster.
Running interactive jobs on zcluser

- No jobs should be run on the zcluser login node!
- To run an interactive job: use the `qlogin` command:

```
zcluser# qlogin
Your job 1391816 ("QLOGIN") has been submitted waiting for interactive job to be scheduled ...
Your interactive job 1391816 has been successfully scheduled.
...
compute-14-7.local#
```

- This will open a session on an interactive node.
- Current maximum runtime is 12 hours.
- Please remember to exit the session when you are done.
- Detailed example of how to run an interactive parallel job is at https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_zcluser
Running serial batch jobs on zcluster

Step 1: Create a job submission script file. Example:

**Script file: sub.sh**

```
#!/bin/bash
cd ${HOME}/testdir
time ./mult.x
```

- **Line 1: #!/bin/bash**
  Specifies the Unix shell to use

- **Line 2: cd ${HOME}/testdir**
  Goes to a working directory called testdir in your home dir.

- **Line 3: time ./mult.x**
  Runs the executable mult.x. The `time` command was added to record the amount of time used by the application (mult.x).

```
zcluster# cp ../ins_data/sub.sh .
```
Step 2: Submit it to the queue

Submit a job to a batch queue

```
zcluster# qsub -q fsr7 sub.sh
Your job 12345 ("sub.sh") has been submitted
```

Options for `qsub`:

- `-q`: specify a queuename, e.g. `fsr7`

The batch queue used in this class is called `fsr7` and the maximum runtime of any job in this queue is 10 minutes.
Step 3: To check the status of your jobs

zcluster# qstat

<table>
<thead>
<tr>
<th>job-ID</th>
<th>name</th>
<th>user</th>
<th>state</th>
<th>submit/start at</th>
<th>queue</th>
<th>slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>131</td>
<td>sub.sh</td>
<td>s_21</td>
<td>r</td>
<td>01/11/2016 1:32:12</td>
<td><a href="mailto:fsr7@compute-11-3.local">fsr7@compute-11-3.local</a></td>
<td>1</td>
</tr>
<tr>
<td>132</td>
<td>sub.sh</td>
<td>s_21</td>
<td>qw</td>
<td>01/11/2016 1:34:23</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>130</td>
<td>sub.sh</td>
<td>s_21</td>
<td>Eqw</td>
<td>01/11/2016 1:10:23</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Common states for jobs:
- **r**: job is running, job start time shown
- **qw**: job is pending, job submission time is shown
- **Eqw**: job is in error state, needs attention

To view more details of a job

zcluster# qsj jobid
zcluster# qstat -j jobid
More queueing system commands

You can cancel a running or pending job.

**To cancel a job**

```
zcluster# qdel jobid
```

Example: To cancel a job that has jobid 12345, use the command

```
zcluster# qdel 12345
```

You can obtain a report with information such as the wall-clock time, the cpu time, etc for a job that finished.

**To get information about a finished job**

```
zcluster# qacct -j jobid >summary.log
```
How do I monitor memory usage of running jobs?

Use the command

```
zcluster# qsj jobid
```

The maximum virtual memory used so far by the job is given by the parameter `maxvmem`. Example:

```
zcluster# qsj 12345
subscription_time: Fri Jan 8 17:24:25 2016
owner: s_21
cwd: /home/student/phys8601/s_21/testdir
hard_queue_list: fsr7
script_file: sub.sh
usage 1: cpu=00:01:15, mem=115.36 GBs, io=0.01083, vmem=1.648G,
maxvmem=1.648G
```
Running the executable

How do I view memory used by job that ended?

Use the command

```
  zcluster# qacct -j jobid
```

The maximum virtual memory used by the job is given by the parameter `maxvmem`. Example:

```
  zcluster# qacct -j 12345
  ================================================================
  hostname  compute-11-3.local
  owner      s_21
  jobname    job1.sh
  start_time Fri Jan 8 17:17:12 2016
  end_time   Fri Jan 8 17:18:40 2016
  ru_wallclock 137
  cpu         136.713
  maxvmem     1.966G
```
How to get support?

The preferred method to ask questions and report problems is to use the online form available at:

http://help.gacrc.uga.edu

When you request support, please let us know your zcluster username (e.g. s_21) as well as details of the problem or question.

Training opportunities: The GACRC offers regular training workshops, please see https://wiki.gacrc.uga.edu/wiki/Training

GACRC communication resources:

- **Web Site** - general overview: http://gacrc.uga.edu/
- **Wiki** - software docs, how-to’s: https://wiki.gacrc.uga.edu/