High Performance Computing (HPC) Using zcluster at GACRC

On-class STAT8060

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

GACRC

Outline

- What is GACRC?
- What is HPC Concept?
- What is zcluster?
- How to work with zcluster?

What is GACRC?

Who Are We?

- **G**eorgia **A**dvanced **C**omputing **R**esource **C**enter
- 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?

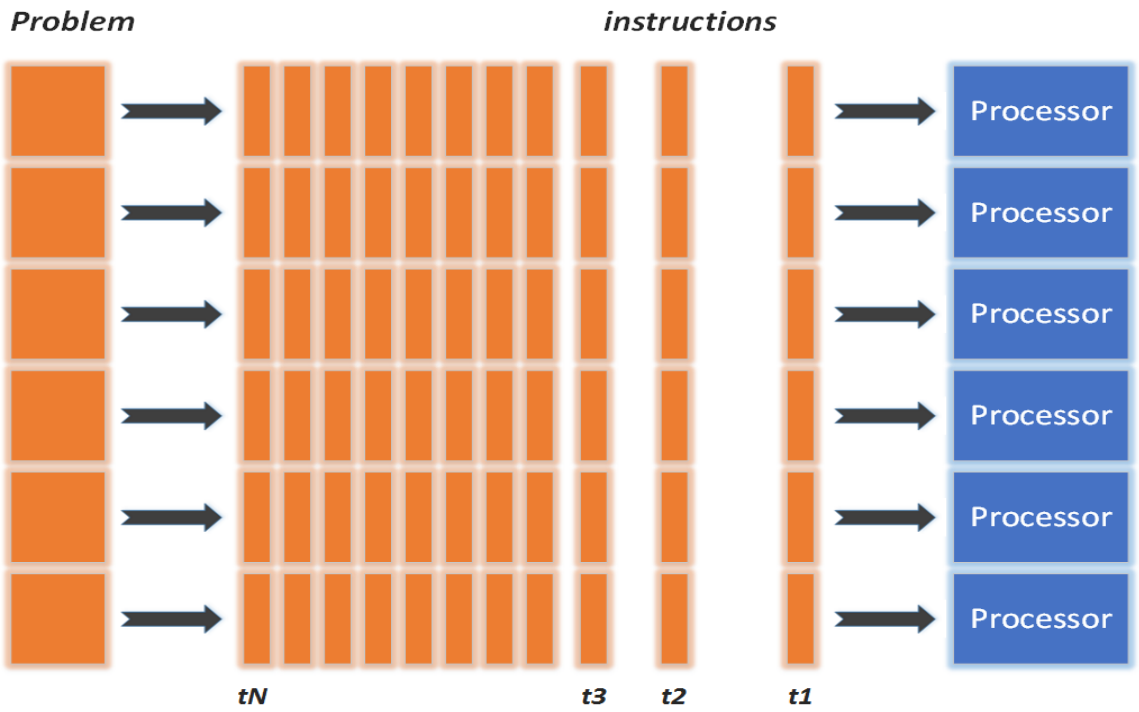
- http://gacrc.uga.edu (Web) http://wiki.gacrc.uga.edu (Wiki)
- https://wiki.gacrc.uga.edu/wiki/Getting_Help (Support)
- https://blog.gacrc.uga.edu (Blog) http://forums.gacrc.uga.edu (Forums)

What is HPC Concept

Serial:

- ✓ Problem broken into discrete instructions
- ✓ Instructions executed sequentially
- ✓ Only 1 instruction executed at any moment on a single processor

Parallel:



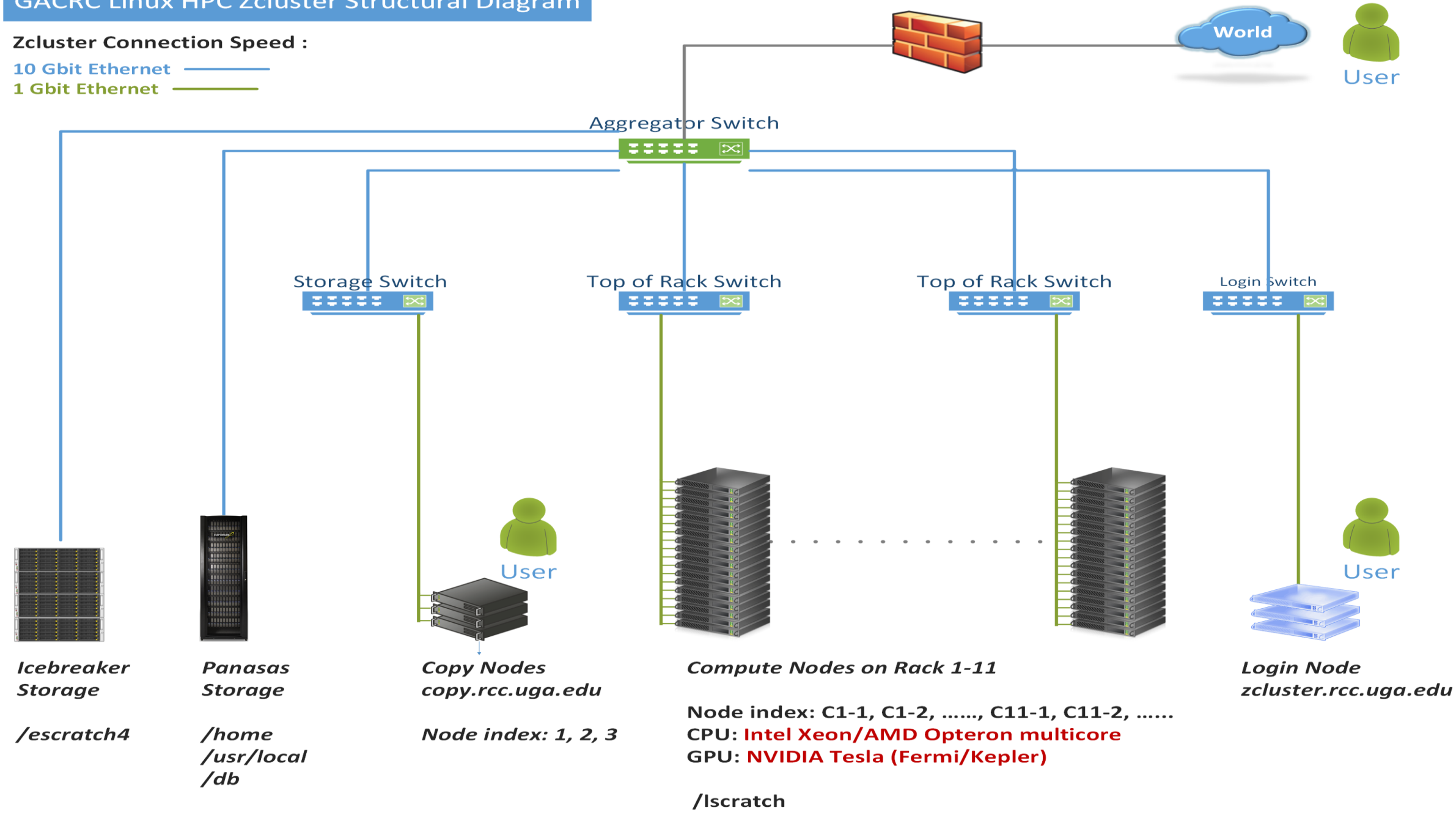
- ✓ Problem broken into parts can be solved concurrently
- ✓ Further broken into a series of instructions
- ✓ Instructions executed simultaneously on multiply processors
- ✓ Synchronization/communication mechanism employed

GACRC

What is zcluster?

- Cluster Structural Diagram
- General Information
- Computing Resources
- Disk Storage

GACRC Linux HPC Zcluster Structural Diagram



What is zcluster – General Information

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

- Operating System: 64-bit Red Hat Enterprise Linux 5 (RHEL 5)
- Login Node: zcluster.rcc.uga.edu
  Copy Node: copy.rcc.uga.edu
- Internodal Communication: 1Gbit network
  compute nodes ⇔ compute nodes
  compute nodes ⇔ storage systems

What is zcluster – General Information

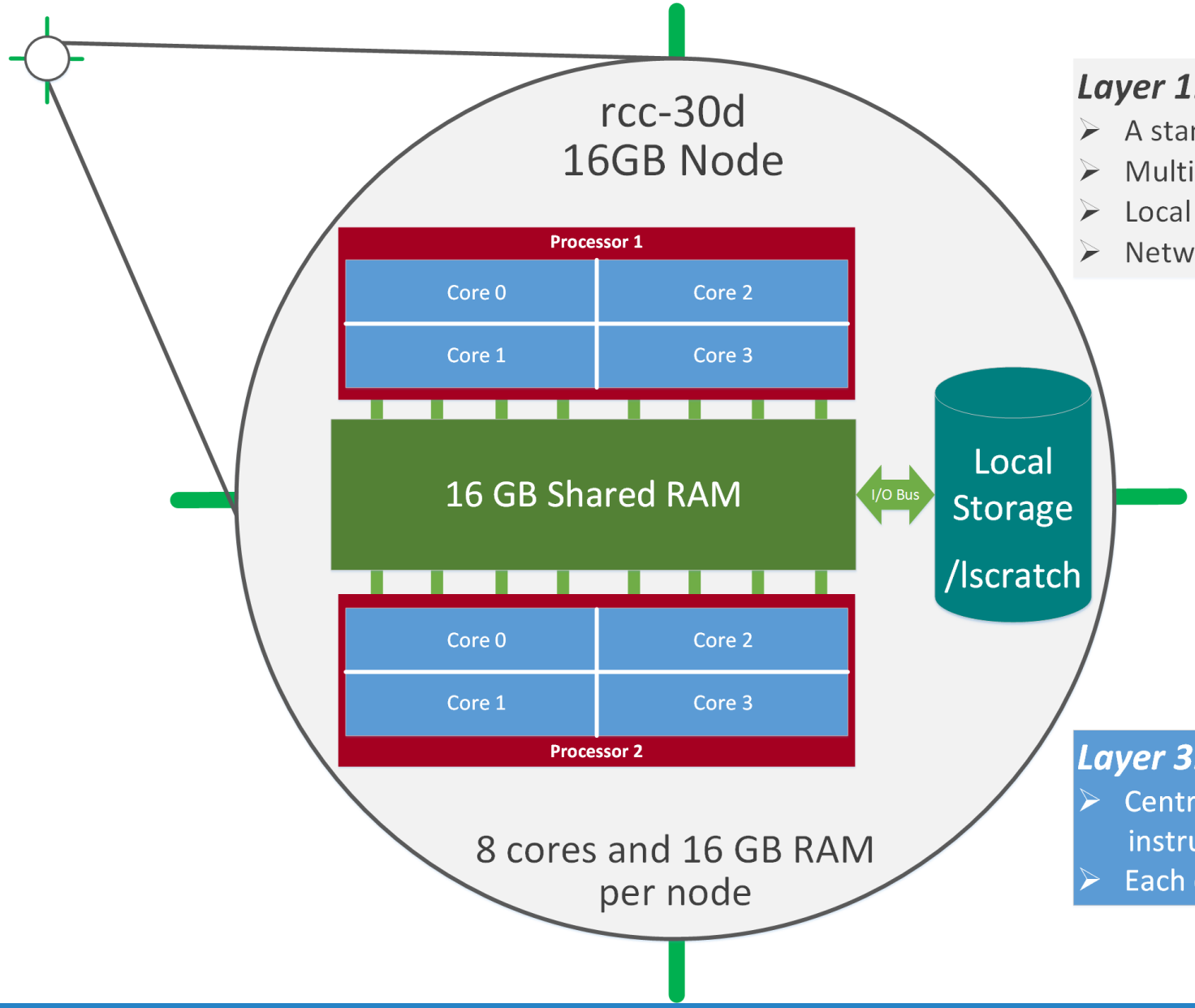
- Batch-queueing System:
  - Jobs can be started (submitted), monitored, and controlled
  - Determine which compute node is the best place to run a job
  - Determine appropriate execution priority for a job to run

- On zcluster: Sun Grid Engine (SGE)

What is zcluster – Computing Resources

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

***Total peak performance:* 23 Tflops**



Layer 1: Node

- A standalone "computer in a box"
- Multiple processors, e.g. 2, sharing memory
- Local disk storage, network interface, etc.
- Networked into a cluster

Layer 2: Processor

- A single computing component
- Multicore processor, e.g. 4 cores

Layer 3: Core

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

What is zcluster – Disk Storage

- **Home directory** ➔ /home/student/stat8060/username (e.g., s_01, s_02, …)
  - Mounted and visible on all nodes, with a quota of ~100GB
  - Any directory on /home has snapshot backups
    - Taken once a day, and maintained 4 daily ones and 1 weekly one
    - Name: .snapshot, e.g., /home/abclab/jsmith/.snapshot
    - Completely invisible, however, user can "cd" into it and then "ls":

```
zhuofei@zcluster:~$ ls -a
.                .bash_profile  .emacs.d   .fontconfig       .maple_history  MPIs      scripts  test.sh
..               .bashrc        .ENV_file  .gnuplot_history  .Mathematica    openMPs   serials  .viminfo
.bash_history    downloads      exe        .history          .mc             .profile  sht      .Xauthority
.bash_logout     .emacs         .flexlmrc  .lesshst          .mozilla        Pthreads  .ssh
zhuofei@zcluster:~$ cd .snapshot
zhuofei@zcluster:~/.snapshot$ ls
2015.06.21.00.00.01.weekly  2015.06.27.01.00.01.daily   2015.06.28.01.00.01.daily  2015.06.30.01.00.01.daily
2015.06.26.01.00.01.daily   2015.06.28.00.00.01.weekly  2015.06.29.01.00.01.daily
```

← .snapshot is NOT shown here!

← can "cd" into .snapshot

← then "ls" to list its contents

What is zcluster – Disk Storage

- **Local scratch** ➔ /lscratch/username (e.g., s_01, s_02, …)
  - On **local disk** of each **compute** node ➔ node-local storage
  - rcc-30d 8-core nodes: ~18GB, rcc-30d 12-core nodes: ~370GB
  - No snapshot backup
  - Usage Suggestion: *If your job writes results to /lscratch, job submission script should move the data to your home or escratch before exit*
- **Ephemeral Scratch** ➔ /escratch4/username_Aug_26 (e.g., s_01_Aug_26)
  - Create with `make_escratch` command
  - Visible to all nodes with a quota of 4TB
  - No snapshot backup
  - To be deleted after 37 days

How to work with zcluster?

Before we start:

- To get zcluster to be your best HPC buddy, go to
  GACRC Wiki (http://wiki.gacrc.uga.edu)
  GACRC Web (http://gacrc.uga.edu)
- To get the most effective and qualified support from us, go to
  GACRC Support (https://wiki.gacrc.uga.edu/wiki/Getting_Help)
- To work happily and productively, follow the cluster's Community Code of Conduct (CCOC)

How to work with it?

- Cluster's CCOC:

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

6 rules of thumb to remember:

- NO jobs running on login node
- NO multi-threaded job running with only 1 core requested
- NO large memory job running on regular nodes
- NO long job running on interactive node
- NO small memory job running on large memory nodes
- Use the copy node for file transfer and compression

How to work with zcluster?

- Start with zcluster
- Connect & Login
- Transfer Files
- Softwares Installed
- Run Interactive Jobs
- Run Batch Jobs
  - How to run ***serial*** jobs
  - How to run ***threaded*** jobs
  - How to run ***MPI*** jobs
  - How to check job status, cancel a job, etc.

GACRC

How to work with zcluster – Start with zcluster

- You need a User Account, e.g., : s_01@zcluster.rcc.uga.edu
- Procedure: https://wiki.gacrc.uga.edu/wiki/User_Accounts
- A UGA faculty member (PI) may register a computing lab: http://help.gacrc.uga.edu/labAcct.php
- The PI of a computing lab may request user accounts for members of his/her computing lab: http://help.gacrc.uga.edu/userAcct.php
- User receives an email notification once the account is ready
- User can use `passwd` command to change initial temporary password

How to work with zcluster – Connect & Login

- Open a connection: Open a terminal and `ssh` to your account

```
ssh s_40@zcluster.rcc.uga.edu
```

or

```
ssh -X s_40@zcluster.rcc.uga.edu
```

([1] `-X` is for X windows application running on the cluster to be forwarded to your local machine

[2] If using Windows, use SSH client to open connection, get from UGA download software page)

- Logging in: You will be prompted for your zcluster password

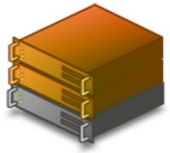
```
s_40@zcluster.rcc.uga.edu's password:
```

([3] On Linux/Mac, when you type in the password, the prompt blinks and does not move)

- Logging out: `exit` to leave the system

```
s_40@zcluster:~$ exit
```

How to work with zcluster – Transfer Files

User's local  SCP  Copy node (copy.rcc.uga.edu) of zcluster

- On Linux, Mac or cygwin on Windows : `scp [Source] [Target]`

  E.g. 1: On local machine, do Local → zcluster

```
scp file1 s_40@copy.rcc.uga.edu:~/subdir
```

```
scp *.dat s_40@copy.rcc.uga.edu:~/subdir
```

  E.g. 2: On local machine, do zcluster → Local

```
scp s_40@copy.rcc.uga.edu:~/subdir/file ./
```

```
scp s_40@copy.rcc.uga.edu:~/subdir/\*.dat ./
```

- On Window: FileZilla, WinSCP, etc.

GACRC

How to work with zcluster – Softwares Installed

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

How to work with zcluster – Run Interactive Jobs

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

```
zhuofei@zcluster:~$ 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$
```

← Now I am on compute-14-7, which is an interactive node

- Current maximum runtime is 12 hours
- When you are done, remember to `exit` the session! ⚠
- Note: Julia can NOT run on interactive nodes with AMD Opteron CPUs

GACRC

How to work with zcluster – Run Batch Jobs

- Components you need to run a batch job:
  - Software already installed on zcluster (Julia)
  - Job submission script to run the software,
    - ✓ Specifying working directory
    - ✓ Exporting environment variables, e.g.,
      OMP_NUM_THREADS (OpenMP threads number)
      LD_LIBRARY_PATH (searching paths for shared libraries)
- Common commands you need:
  - qsub with specifying queue name, threads or MPI rank number
  - qstat, qdel
  - qacct, qsj, etc.

How to work with zcluster – Run Batch *Serial* Jobs

- **Step 1**: Create a job submission script *subserial.sh* running julia:

```
#!/bin/bash                          ➔ Linux shell (bash)

cd `pwd`                             ➔ Specify and enter (cd) the working directory (pwd command gives the path of your current directory)

/usr/local/julia/0.3.3/julia test.jl ➔ Run julia script test.jl
```

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

```
$ qsub -q rcc-30d subserial.sh
```

OR

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

Submit a job

Your job submission script

to the queue rcc-30d
with 16GB RAM/Node

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

How to work with zcluster – Run Batch *Threaded* Jobs

- **Step 1**: Create a job submission script *subthread.sh* running julia:

```
#!/bin/bash

cd `pwd`

/usr/local/julia/0.3.3/julia -p 4 test.jl
```

➔ Run julia with 4 threads (-p 4)

Number of Threads = Number of Cores Requested

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

```
$ qsub -q rcc-30d -l mem_total=20g -pe thread 4 subthread.sh
```

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

4 cores requested

Note:
Please use the rcc-mc-30d queue, If using threads more than 8!

How to work with zcluster – Run Batch *MPI* Jobs

- **Step 1**: Create a job submission script *submpi.sh* running julia:

```
#!/bin/bash

cd `pwd`

/usr/local/julia/0.3.3/julia -p ${NSLOTS} --machinefile ${TMPDIR}/machines test.jl
```

➔ Run julia with 12 MPI processes (-np $NSLOTS)

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

```
$ qsub -q rcc-30d -pe mpi 12 submpi.sh
```

12 cores requested,
$NSLOTS will be assigned to 12 automatically, before the job submission script is interpreted

How to work with zcluster – Check and Cancel Jobs

- To check the status of all queued and running jobs: qstat

qstat	➔ shows your job in the pool
qstat –u "*"	➔ shows all the jobs in the pool
qstat –j 12345	➔ shows detailed information, e.g., maxvmem, about the job with JOBID 12345
qstat –g t	➔ list all nodes used by your jobs

- To cancel a queued or running job: qdel

qdel –u zhuofei	➔ deleted all your jobs
qdel 12345	➔ deletes your job with JOBID 12345

- To list detailed information about a job: qsj, qacct

qsj 12345	➔ shows information, e.g., maxvmem, about the RUNNING job with JOBID 12345
qacct –j 12345	➔ shows information, e.g., maxvmem, about the ENDED job with JOBID 12345

Thank You!

Good Luck STAT8060!

From Yecheng:

- Software support issue:

  What kind of software support we are responsible for the users? or definition of our software support

From Shan-Ho:

- rcc-30d : For MPI, max 75 cores total to be requested
- rcc-mc-30d : max 32 threads to be allowed
- rcc-m128-30d: max 5 cores to be requested

rcc-m512-30d: max 8 cores to be requested