Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

Introduction to High Performance Computing

Using Sapelo2 at GACRC

Georgia Advanced Computing Resource Center
University of Georgia
Suchitra Pakala
pakala@uga.edu

Outline

- High Performance Computing (HPC)
- HPC at UGA - GACRC
- Sapelo2 Cluster Overview
  - Architecture
  - Computing resources, Storage Environment
  - Software on Cluster
  - Job Submission Workflow
  - Access and Working with Sapleo2

- High Performance Computing (HPC)
- Cluster Computing

What is HPC?

- **High Performance Computing**
  - Practice of aggregating computing power
  - Higher performance - when compared to regular Desktop or Laptops
  - Parallel processing for solving complex computational problems
  - Using advanced applications programs efficiently, reliably and quickly

Also… Cluster Computing

- A cluster:
  - Parallel or distributed processing system
  - Consists of a collection of interconnected stand alone computers
  - Working together as a single integrated computing resource
  - Provide better system reliability and performance
  - Appears to users as a single highly available system

Why use HPC?

- A single computer (processor) is limited in:
  - Memory
  - Speed
  - Overall performance

- A cluster of computers can overcome these limitations
  - Solves problems that cannot fit in a single processor's memory
  - Reduces computational time to reasonable expectations
  - Solves problems at finer resolution

RESEARCH Open Access

Scaling bioinformatics applications on HPC

Mike Mikailov[1], Fu-Jyh Luo[1], Stuart Barkley[1], Lohit Valleru[1], Stephen Whitney[1], Zhichao Liu[2], Shraddha Thakkar[2], Weida Tong[2] and Nicholas Petrick[1*]

From The 14th Annual MCBIOS Conference
Little Rock, AR, USA. 23-25 March 2017

Results: BLAST jobs that hitherto failed or slogged inefficiently to completion now finish with speeds that characteristically reduce wallclock time from 27 days on 40 CPUs to a single day using 4104 tasks, each task utilizing eight CPUs and taking less than 7 minutes to complete.

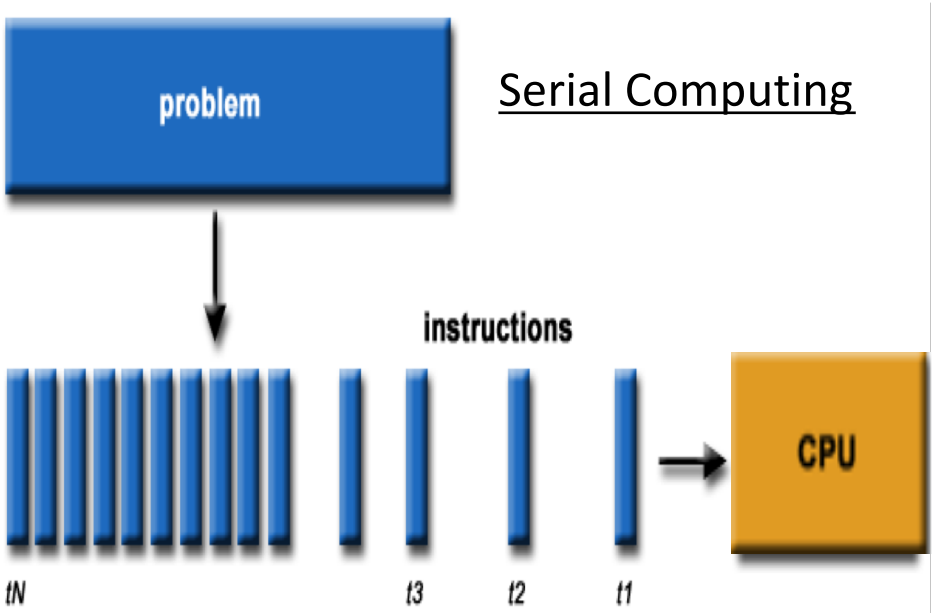
Components of HPC

- **Node** – Individual computer in a cluster
  - Eg: Login node, Transfer node
  - Individual nodes can work together, talk to each other
  - Faster problem solving

- **Queue** – Collection of compute nodes for specific computing needs on a cluster
  - Eg: batch, highmem_q, inter_q, gpu_q

- **Jobs** – User programs that run on a cluster
  - Managed through a queueing system (Torque/Moab)

HPC - Submitting Jobs :



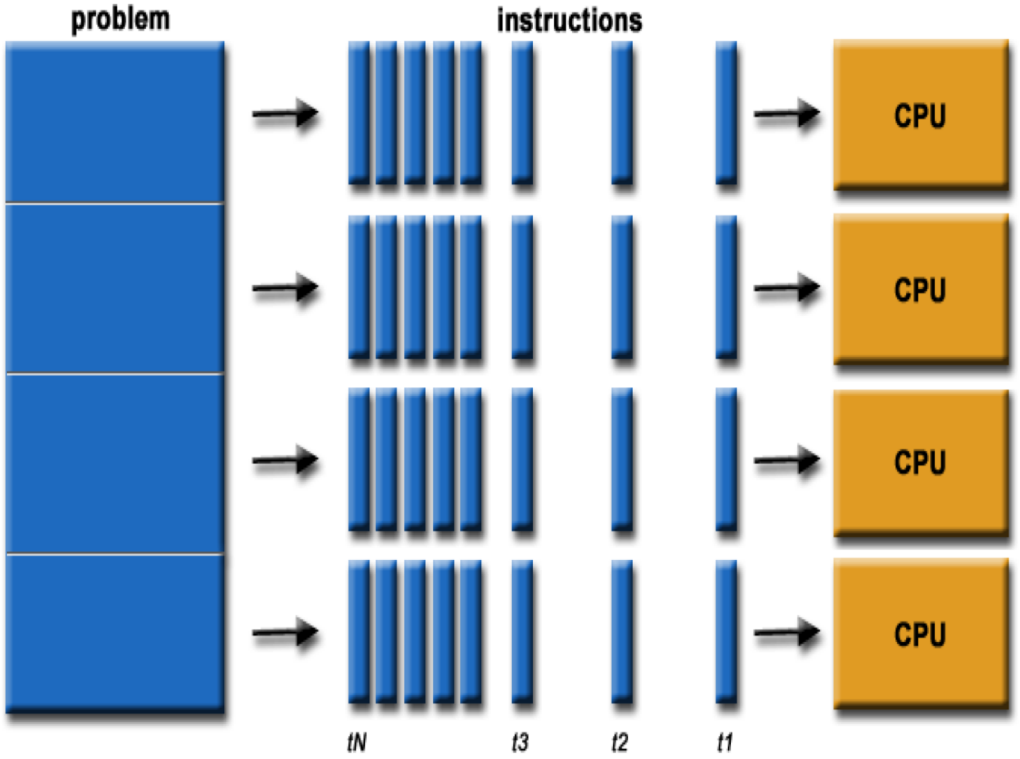
Serial Computing

- A problem is broken into a discrete series of instructions
- Instructions are executed sequentially
- Executed on a single processor
- Only one instruction may execute at any moment in time

HPC - Submitting Jobs:

Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

Parallel Computing

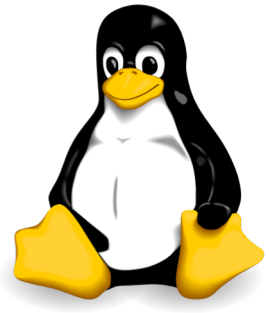


Parallel Computing

- A problem is broken into discrete parts that can be solved concurrently
- Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different processors
- An overall control/coordination mechanism is employed

Operating System: Linux

- Several distributions - Ubuntu, **CentOS**, Fedora, RedHat, etc
- Open Source, Multi-user, Multi-tasking operating system
- Free, Stable, Secure, Portable



High Performance Computing at GACRC

Sapelo2

Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

GACRC

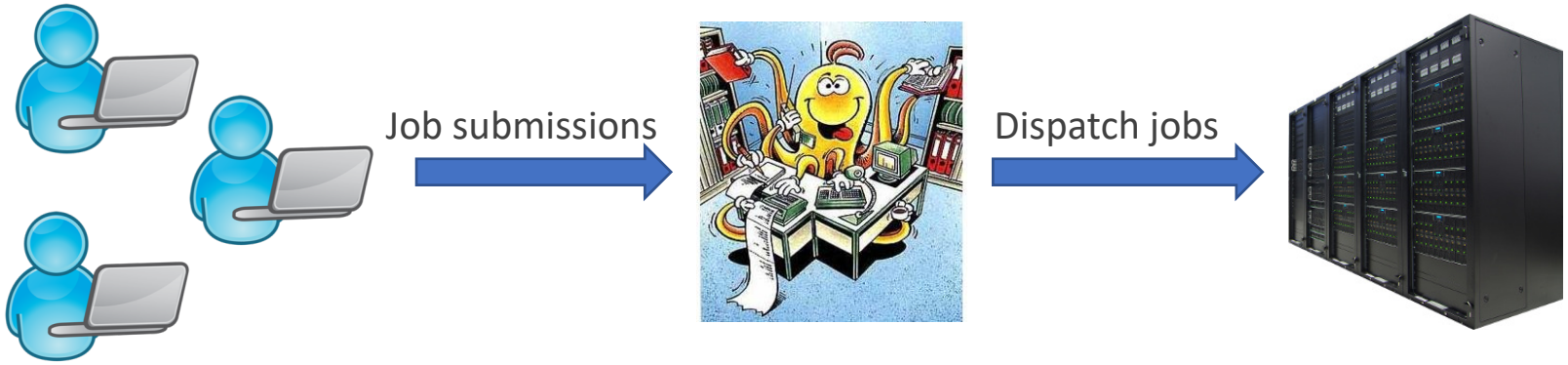
- We are the high-performance-computing (HPC) center at UGA
- We provide to the UGA research and education community an advanced computing environment:
  - HPC computing and networking infrastructure located at the Boyd Data Center
  - Comprehensive collection of scientific, engineering and business applications
  - Consulting and training services
- http://wiki.gacrc.uga.edu (GACRC Wiki)
- https://wiki.gacrc.uga.edu/wiki/Getting_Help (GACRC Support)
- http://gacrc.uga.edu (GACRC Web)

Sapelo2 Overview

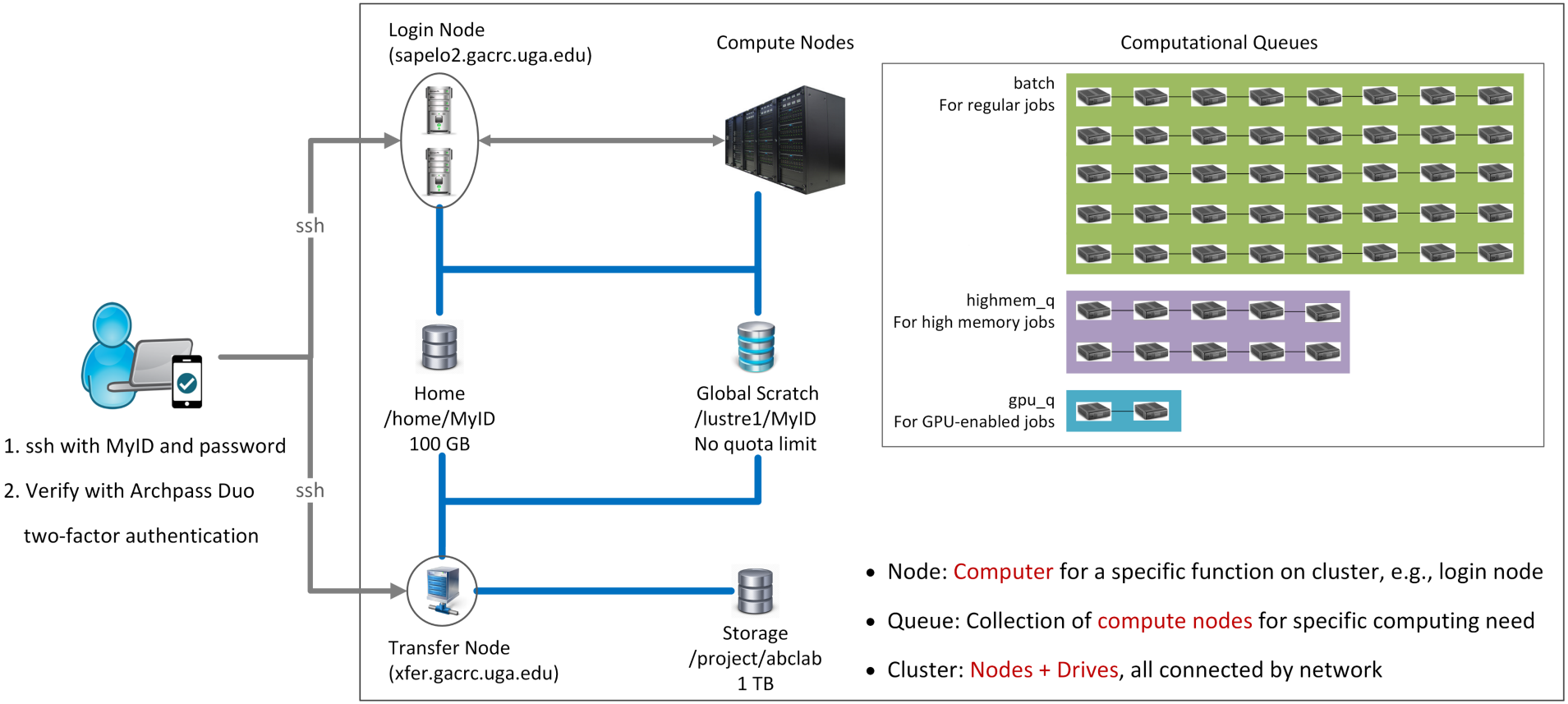
- Architecture
- General Information
- Computing resources
- Storage Environment
- Software on Cluster
- Job submission Workflow

Cluster

- Using a cluster involves 3 roles:
  - User(s): to submit jobs
  - Queueing System: to dispatch jobs to cluster, based on availability of resources
  - Cluster: to run jobs



Sapelo2 Cluster



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

Sapelo2: A Linux HPC cluster (64-bit Centos 7)

- Two Nodes:
  - Login node for batch job workflow: MyID@sapelo2.gacrc.uga.edu
  - Transfer node for data transferring: MyID@xfer.gacrc.uga.edu
- Three Directories:
  - Home: Landing spot; 100GB quota; Backed-up
  - Global Scratch: High performance job working space; NO quota; NOT backed-up
  - Storage: Temporary data parking; 1TB quota (for group); Backed-up (**ONLY accessible from Transfer node!**)
- Three Computational Queues: batch, highmem_q, gpu_q

Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

Four Computational Queues

Queue	Node Feature	Total Nodes	RAM(GB) /Node	Max RAM(GB) /Single-node Job	Cores /Node	Processor Type	GPU Cards /Node	InfiniBand
batch	Intel	30	64	**62**	**28**	Intel Xeon	N/A	Yes
		42	192	**188**	**32**	Intel Xeon (Skylake)		
	AMD	90	128	**125**	**48**	AMD Opteron		
highmem_q	Intel/AMD	4/1	1024	**997**	**28**	Intel Xeon		
	AMD/Intel	4/1	512	**503**	**48**	AMD Opteron		
gpu_q	GPU	2	128	**125**	**16**	Intel Xeon	8 NVIDIA K40	
		2	96/80	**92/76**	**12**		7 NVIDIA K20	
		4	192	**188**	**32**	Intel Xeon (Skylake)	1 NVDIA P100	
grpBuyin_q	variable							

Three Directories

Role	Directory	Intended Use	Quota	Accessible from	Backed-up	Notes
Home	**/home/MyID**	**Static data:** 1. Scripts, source codes 2. Local software	100GB	Login Transfer	Yes	
Global Scratch	**/lustre1/MyID**	**Current job data:** data being read/written by running jobs	No Limit		No	**User to clean up!** Subject to deletion in 30 days
Storage	**/project/abclab**	**Temporary data parking:** non-current active data	1TB (Initial)	Transfer	Yes	Group sharing possible

Software on Cluster

- The cluster uses environment modules to define the various paths for software packages
- Software names are long and have a EasyBuild toolchain name associated to it
- Complete module name: Name/Version-toolchain, e.g., BLAST+/2.6.0-foss-2016b-Python-2.7.14
- More than 600 modules currently installed on cluster
- Out of these, around 260 modules are Bioinformatics applications – AUGUSTUS, BamTools, BCFTools, BLAST, Canu, Cutadapt, Cufflinks, Tophat, Trinity, etc
- Others:
  - Compilers – GNU, INTEL, PGI
  - Programming – Anaconda, Java, Perl, Python, Matlab, etc
  - Chemistry, Engineering, Graphics, Statistics (R), etc

Job Submission Workflow

- Log on to Login node using MyID and password, and two-factor authentication with Archpass Duo: **ssh MyID@sapelo2.gacrc.uga.edu**
- On Login node, change directory to global scratch : **cd /lustre1/MyID**
- Create a working subdirectory for a job : **mkdir ./workDir**
- Change directory to workDir : **cd ./workDir**
- Transfer data from local computer to workDir : use **scp** or **SSH File Transfer** to connect Transfer node
  - Transfer data on cluster to workDir : log on to Transfer node and then use **cp** or **mv**
- Make a job submission script in workDir : **nano ./sub.sh**
- Submit a job from workDir : **qsub ./sub.sh**
- Check job status : **qstat_me** or Cancel a job : **qdel JobID**

Example: Job Submission Script

Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

```
  GNU nano 2.3.1                File: sub.sh

#PBS -S /bin/bash
#PBS -q batch
#PBS -N bowtie2_test
#PBS -l nodes=1:ppn=1
#PBS -l mem=2gb
#PBS -l walltime=1:00:00

#PBS -M pakala@uga.edu
#PBS -m ae

cd $PBS_O_WORKDIR

module load Bowtie2/2.3.3-foss-2016b

time bowtie2 -p 1 -x ./index/lambda_virus -U ./myreads.fq -S output.sam

^G Get Help    ^O WriteOut    ^R Read File    ^Y Prev Page    ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is     ^V Next Page    ^U UnCut Text  ^T To Spell
```

→ Linux shell (bash)
→ Queue Name (batch)
→ Name of the job (bowtie2_test)
→ Number of nodes (1), Number of cores/node (1)
→ Max amount of physical memory used by the job (2GB)
→ Max wall clock time (1 hr) for the job

→ Email notifications

→ Use the directory from which the job is submitted as the working directory

→ Load the module of Bowtie2, version 2.3.3

Submit a job using qsub

```
pakala@sapelo2-sub2 workdir$ pwd
/lustre1/pakala/workdir
pakala@sapelo2-sub2 workdir$ ls
index  myreads.fq  sub.sh
pakala@sapelo2-sub2 workdir$ qsub sub.sh
11743.sapelo2
```

sub.sh is job submission script to

1. specify computing resources:
2. load software using **ml load**
3. run any Linux commands you want to run
4. run the software

Check job status using qstat_me

```
pakala@sapelo2-sub2 workdir$ qstat_me
Job ID              Name             User      Time Use S Queue
------------------- ---------------- --------- -------- - -----
11743.sapelo2       bowtie2_test     pakala    00:12:40 C batch
11744.sapelo2       bowtie2_test     pakala    00:05:17 R batch
11746.sapelo2       bowtie2_test     pakala    00:02:51 R batch
11747.sapelo2       bowtie2_test     pakala           0 Q batch
```

R : job is running
C : job completed (or canceled or crashed) and is no longer running. (This status is displayed for 24 hours)
Q : job is pending, waiting for resources to become available

Note: “Time Use” is the CPU time, instead of the wall-clock time of your job staying on cluster!

Cancel job using qdel

```
pakala@sapelo2-sub2 workdir$ qdel 11746
pakala@sapelo2-sub2 workdir$ qstat_me
Job ID              Name             User      Time Use S Queue
------------------ ---------------- --------- -------- - -----
11743.sapelo2      bowtie2_test     pakala    00:12:40 C batch
11744.sapelo2      bowtie2_test     pakala    00:05:17 R batch
11746.sapelo2      bowtie2_test     pakala    00:03:15 C batch
11747.sapelo2      bowtie2_test     pakala           0 Q batch
```

job 11746 status is changed from R to C

C status will stay in list for 24 hour

- How to request Sapelo2 User Account
- Resources available on Sapelo2

Request Sapelo2 User Account

Sapelo2 cluster user account: MyID@sapelo2.gacrc.uga.edu

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

1. The UGA PI uses the GACRC online form http://help.gacrc.uga.edu/userAcct.php to request a user account for a group member.
2. Once we received the request, we will verify it with the PI.
3. After verification by the PI, the new user will be required to attend a training session.
4. After the user attended training, we will provision a Sapelo account for the user.
5. A welcome letter is sent to the user once user account is ready.

Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

Resources on Sapelo2 - GACRC Wiki

Main Page: http://wiki.gacrc.uga.edu

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

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

Transfer Files: https://wiki.gacrc.uga.edu/wiki/Transferring_Files

Linux Commands: https://wiki.gacrc.uga.edu/wiki/Command_List

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

User Account Request: https://wiki.gacrc.uga.edu/wiki/User_Accounts

Support: https://wiki.gacrc.uga.edu/wiki/Getting_Help

Thank You!