HPCC Cluster Intro

Charles Forsyth
System Administrator
High Performance Computing Center
University of California, Riverside
forsythc@ucr.edu

Agenda

- Cluster Introduction
- Accessing the Cluster
- Storage
- Module System
- Environment Variables
- Submitting Cluster Jobs
- Scripting
- Monitoring the Cluster Jobs
- Helpful Resources

Cluster Introduction

- Purpose
  - Provide Cluster Computing and Bigdata Resources to support and enable research at UCR.
- Access
  - Subscription based membership.
  - Accessible both on and off campus.
  - Organized by Lab and Department Affiliation.
- Restructured under the Research and Economic Development department.

HPCC Website

Website Address: hpcc.ucr.edu

High-Performance Computing Center @ UCR | Nav | Related Websites | Feedback | Google Custom Search

HPCC

- Home
  - Introduction
  - Activity report
  - Slides
- News
- Access & rates
- Documents
- Contacts
- Events
- Hardware
- Software
- Manuals

High-Performance Computing Center (HPCC)

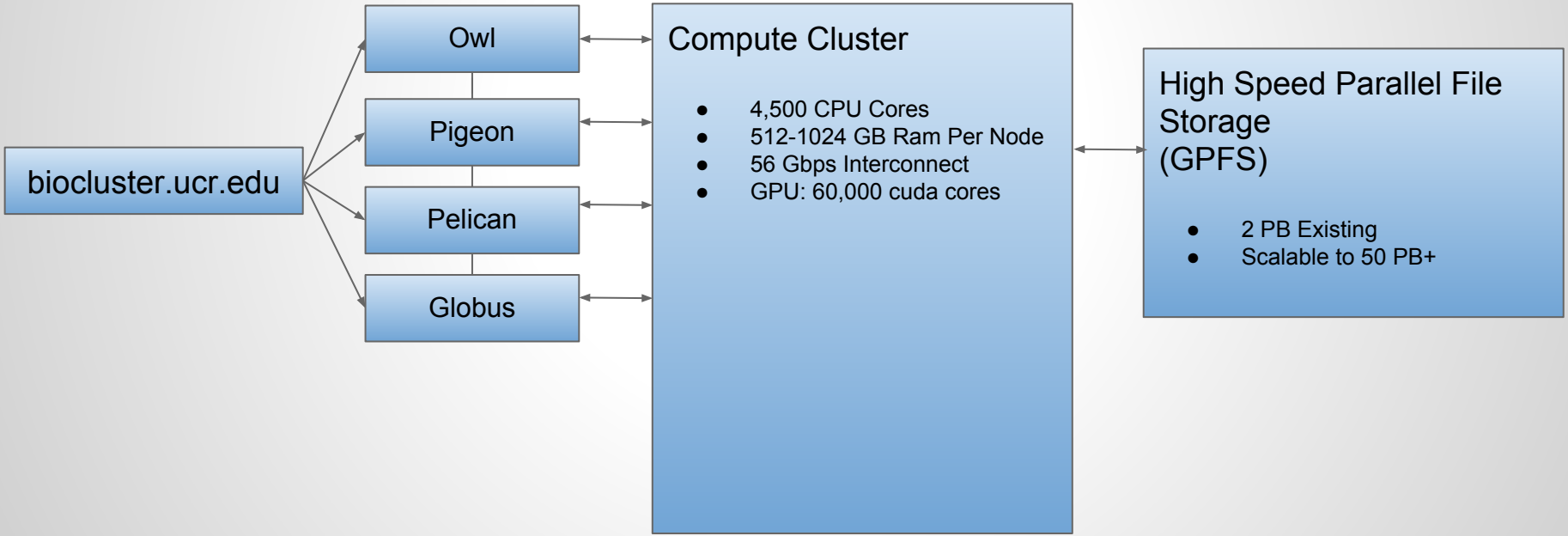
Summary: welcome to the website of the High-Performance Computing Center (HPCC) at UC Riverside. This site gives an overview of the HPC resources and services provided by our center.

Table of Contents

Mission and Services

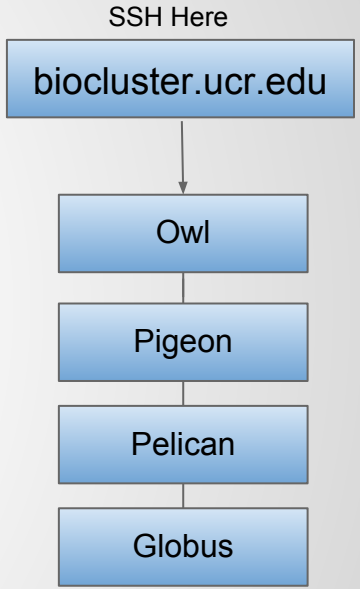
The High-Performance Computing Center (HPCC) provides state-of-the-art research computing infrastructure and training accessible to all UCR researchers and affiliates at low cost. This includes access to the shared HPC resources and services summarized below. The main advantage of a shared research computing environment is access to a much larger HPC infrastructure (with thousands of CPUs/GPUs and many PBs of directly attached storage) than what smaller clusters of individual research groups could afford, while also providing a long-term sustainability plan and professional systems administrative support.

HPCC Cluster Overview



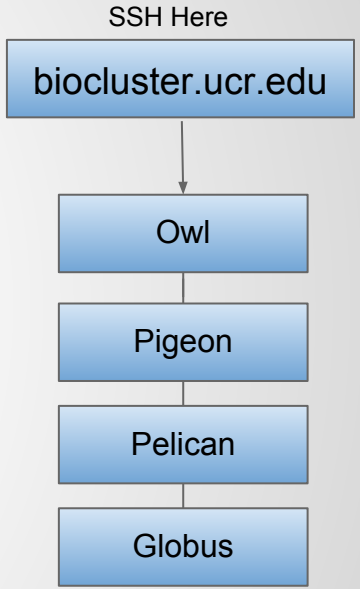
Accessing the Cluster - SSH

- SSH is the main protocol used to interface with the HPCC Cluster.
- Start by ssh'ing from your system to biocluster.ucr.edu
  - Windows
    - Putty or Mobaxterm
      - ssh -XY biocluster.ucr.edu
  - Mac
    - Terminal
      - ssh -XY biocluster.ucr.edu
  - Linux
    - Terminal
      - ssh -XY biocluster.ucr.edu
  - The "-XY" allows graphical output to be passed back to your system.
- Biocluster.ucr.edu is a special DNS name that load balances incoming connections to one of the five login nodes.



Accessing the Cluster - SSH

```
forsythc@chuck-ws: ~
forsythc@chuck-ws:~$ ssh -XY biocluster.ucr.edu
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Tue Jun 27 13:22:51 2017 from 68.190.201.5

--------------------------------------------------------------------------------
              University of California, Riverside - Bioinformatics
--------------------------------------------------------------------------------

More information about the facility and how to use the resources provided can
be found at http://manuals.bioinformatics.ucr.edu/home/hpc

Please send all questions and support requests to support@biocluster.ucr.edu

Note: The default version of R is now 3.3.0
--------------------------------------------------------------------------------

forsythc@pelican:~$
```



Storage - Directory Structure

Home is used for scripting, debugging and small files. (20 GB Quota)

/rhome/username

Bigdata is used for parallel jobs, high read/write operations and big files. (100 GB for 100/yr, 10 TB for 1000/yr)

/bigdata/labname/username

/bigdata/labname/shared

Note: All lab members share the same bigdata pool.

Total = /bigdata/labname/shared + /bigdata/labname/labmember1 + /bigdata/labname/labmember2 + /bigdata/labname/labmemberX

How much data am I using currently: https://dashboard.bioinfo.ucr.edu

Scratch is used for quick access and only temporary (30 days max).

Storage - Directory Structure

```
forsythc@chuck-ws: ~
drwxr-xr-x 800 root     root    262144 Aug 29 19:39 ..
drwxr-xr-x   3 forsythc bioinfo   4096 Aug 28 15:54 slurm-scripts
drwxr-xr-x   2 forsythc bioinfo   4096 Aug 23 16:24 .interproscan-5
drwxr-xr-x   2 forsythc bioinfo   4096 Aug 23 12:51 tmp
drwxr-xr-x   8 forsythc bioinfo   4096 Aug 22 11:03 repos
drwxr-xr-x   2 forsythc bioinfo   4096 Aug 22 10:53 vpn-keys
-rw-r--r--   1 forsythc bioinfo    149 Aug 22 10:26 .bashrc
drwxr-xr-x   4 forsythc bioinfo   4096 Aug 22 10:25 bin
drwx------   3 forsythc bioinfo   4096 Aug 15 23:29 .gnupg
-rw-r--r--   1 forsythc bioinfo   3108 Aug 15 23:24 IVXX-pub-gpg-key.asc
-rw-------   1 forsythc bioinfo     60 Aug 15 21:17 .lesshst
drwxr-xr-x   2 forsythc bioinfo   4096 Aug 11 16:54 .html
-rw-------   1 forsythc bioinfo      7 Jul 27 13:55 .mysql_history
drwxr-xr-x   5 forsythc bioinfo   4096 Jul 27 11:45 .puppetlabs
drwxr-xr-x   2 forsythc bioinfo   4096 Jul 20 09:32 gcc-test
drwx------   3 forsythc bioinfo   4096 Jul 18 14:04 .cache
-rw-r--r--   1 forsythc bioinfo  12113 Jul  6 11:09 .vimrc
drwxr-xr-x  13 forsythc bioinfo   4096 Jun 30 14:52 .rstudio
drwxr-xr-x   3 forsythc bioinfo   4096 Jun 30 10:48 R
drwx------   3 forsythc bioinfo   4096 Jun 27 13:32 .local
-rw-r--r--   1 forsythc bioinfo    536 Jun 22 12:47 compile-r-notes.txt
drwxr-xr-x   2 forsythc bioinfo   4096 Jun 20 09:53 .gstreamer-0.10
drwx------   4 forsythc bioinfo   4096 Jun 20 09:50 .config
drwx------   3 forsythc bioinfo   4096 Jun 20 09:50 .dbus
drwx------   2 forsythc bioinfo   4096 Jun 20 09:50 .gnome2
drwxr-xr-x   3 forsythc bioinfo   4096 Jun 20 09:50 .java
drwxr-xr-x   3 forsythc bioinfo   4096 Jun 20 09:50 .subversion
drwxr-xr-x   3 forsythc bioinfo   4096 Jun 20 09:42 .matlab
drwxr-xr-x   7 forsythc bioinfo   4096 Jun 13 13:53 augustus-config
drwxr-----   3 forsythc bioinfo   4096 Jun  8 14:14 .pki
lrwxrwxrwx   1 forsythc bioinfo     23 Jun  5 15:35 shared -> /bigdata/bioinfo/shared
lrwxrwxrwx   1 forsythc bioinfo     25 Jun  5 15:35 bigdata -> /bigdata/bioinfo/forsythc
drwxr-xr-x  18 forsythc bioinfo   4096 Nov  1  2016 .vim
-rw-r--r--   1 forsythc bioinfo   4882 Oct 31  2016 .tmux.conf
-rw-r--r--   1 forsythc bioinfo    269 Oct 31  2016 .Rprofile
-rw-r--r--   1 forsythc bioinfo    677 Apr  8  2013 .profile
forsythc@owl:~$ ls
IVXX-pub-gpg-key.asc  augustus-config  bin                  gcc-test  shared         tmp
R                     bigdata          compile-r-notes.txt  repos     slurm-scripts  vpn-keys
forsythc@owl:~$ pwd
/rhome/forsythc
forsythc@owl:~$
```

Storage - Uploading/Downloading

- Moving files to or from the Cluster.
  - SCP or SFTP are supported protocols.
- SCP
  - ***scp user@remote_host:file.name .*** # Copies file from server to local machine (typed from local
    # machine prompt). The '.' copies to pwd, you can specify
    # here any directory, use wildcards to copy many files.

    scp file.name user@remote_host:~/dir/newfile.name
    # Copies file from local machine to server.

    scp -r user@remote_host:directory/ ~/dir
    # Copies entire directory from server to local machine.
- SFTP
  - FileZilla is a graphical SFTP client available free for Linux, Mac and Windows platforms.
    - https://filezilla-project.org/
  - The SFTP protocol is also available from the command line terminal in Linux and Mac.

Software Module System

Print available modules

```
module avail
```

Print available modules starting with R

```
module avail R
```

Load default module R

```
module load R
```

Load specific module R version

```
module load R/3.2.0
```

Print list of loaded modules

```
module list
```

Unload module R

```
module unload R
```

Unload specific module R

```
module unload R/3.2.0
```

Environment Variables

- The HPCC cluster uses bash as the default shell environment.
- Within this environment, variables can be set and reused.

```
MYVAR='Something'

export MYVAR='Something'

echo $MYVAR
```

- Some software utilizes this feature and requires that specific environment variables be set.
  - `$HOME #Contains your home path`
  - `$USER #Contains your username`
  - `$PATH #Contains paths of executables`
  - `$LD_LIBRARY_PATH #Contains paths of dependencies`
- Slurm (Cluster Scheduler)

```
$SLURM_SUBMIT_DIR #And many more
```

Environment Variables

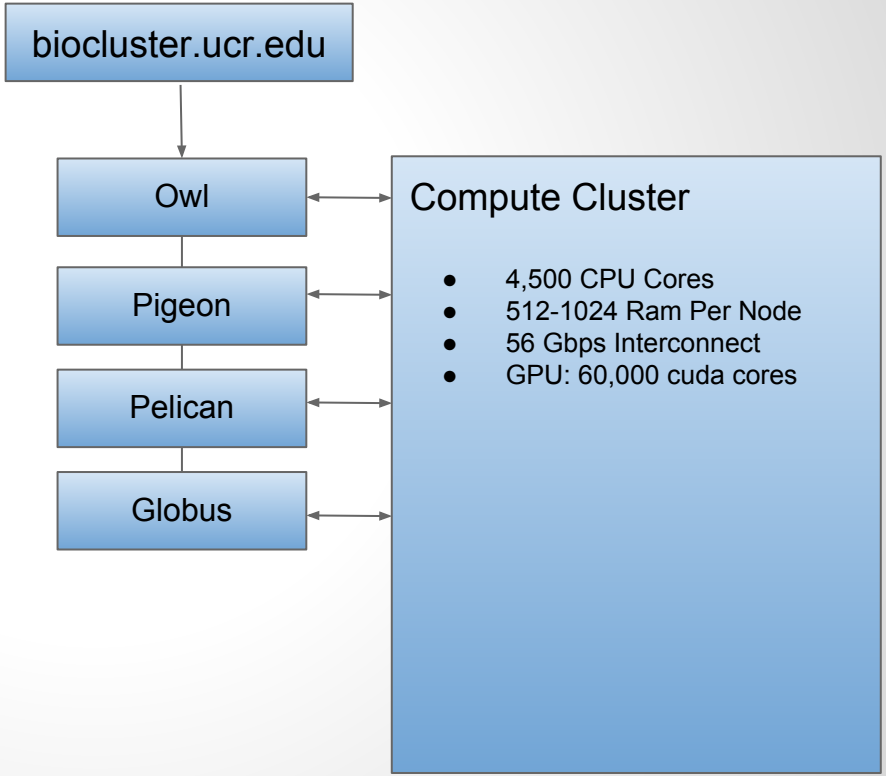
Slurm Job Environment variables (small sample)

```
$SLURM_SUBMIT_DIR   (directory where you ran the script)
$SLURM_JOB_ID       (the SLURM Job ID of the current Job)
$SLURM_NTASKS       (number of cpus the job requested)
$SLURM_NODELIST     (list of nodes this job is running on)
```

Queuing System

Slurm

1. Resource Management
2. Job Scheduler
3. Load Balancing
4. Parallel Computing



Queuing System - Partitions

- Short
  - Nodes: i01-i40
  - Cores: Intel, 256 per user
  - RAM: 1 GB default
  - Time (walltime): 2 hours Maximum
- Intel
  - Default partition
  - Nodes: i01-02,i17-i40
  - Cores: Intel, 256 per user
  - RAM: 1 GB default
  - Time (walltime): 168 hours (7 days) default
- batch
  - Nodes: c01-c48
  - Cores: AMD, 256 per user
  - RAM: 1 GB default
  - Time (walltime): 168 hours (7 days) default
- Highmem
  - Nodes: h01-h04
  - Cores: Intel, 32 per user
  - RAM: 100 GB min and 1024 GB max
  - Time (walltime): 48 hours default
- Gpu
  - Nodes: gpu01-gpu02
  - Cores: Intel, 16 per user
  - RAM: 128 GB default
  - Time (walltime): 100 hours default

Queuing System - Job Limits

Base limit of 512 Cores per group running at anyone time.

Base limit of 256 Cores per person running at anyone time.

See your limits:

```
sacctmgr show account $(id -gn) format=Account,User,Partition,GrpCPUs,GrpMem,GrpNodes --ass | grep $USER
```

View your group limits:

```
sacctmgr show account $(id -gn) format=Account,User,Partition,GrpCPUs,GrpMem,GrpNodes,GrpTRES%30 --ass |
head -3
```

Check total number of cores currently used by your group in all partitions:

```
echo $(squeue -A $(id -gn) -o %C | grep -P '^[0-9]' | tr '\n' '+' | sed 's/+$//g') | bc
```

Queuing System - Status

Current status can be viewed from http://hpcc.ucr.edu/snapshot.html.

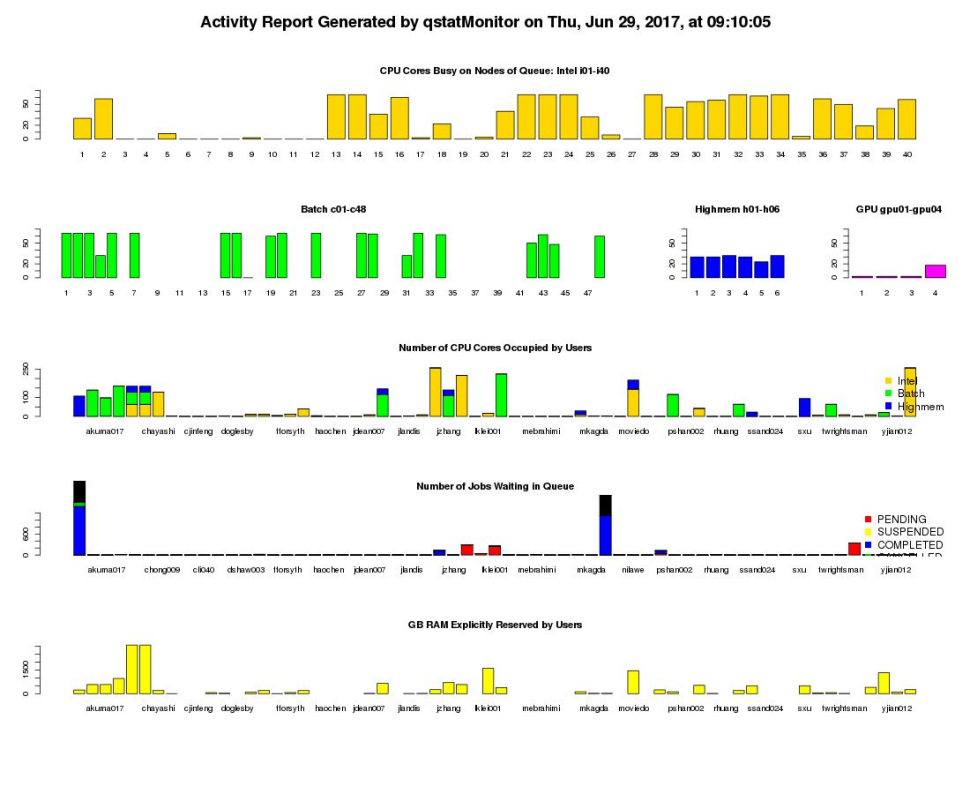
Command line tool:

jobMonitor

Will also give some interesting information about jobs on the Cluster

sinfo

squeue



Queuing System - Submitting a Job

Two major ways to submit on job on the cluster

- **srun**
  - Used for interactive jobs
  - Examples:
    - srun --pty bash -l
    - srun --x11 --mem=1gb --ntasks 1 --time 10:00:00 --pty bash -l
- **sbatch**
  - Used to submit batch(non-interactive) jobs
  - Examples:
    - sbatch sbatch_script.sh
    - sbatch --ntasks=12 --mem=24gb --time=2-10:30:00 sbatch_script.sh

Queuing System - Submitting a Job

Interactive Jobs

srun --x11 --mem=1gb --ntasks 1 --time 10:00:00 --pty bash -l

```
forsythc@chuck-ws: ~
forsythc@owl:~$ srun --pty bash -l
forsythc@i01:~$ pwd
/rhome/forsythc
forsythc@i01:~$ ls
IVXX-pub-gpg-key.asc  augustus-config  bin                  gcc-test  shared         tmp
R                     bigdata          compile-r-notes.txt  repos     slurm-scripts  vpn-keys
forsythc@i01:~$ printenv | grep SLURM | tail -10
SLURM_JOB_CPUS_PER_NODE=2
SLURM_CLUSTER_NAME=biocluster
SLURM_GTIDS=0
SLURM_SUBMIT_HOST=owl
SLURM_JOB_PARTITION=intel
SLURM_STEP_NUM_TASKS=1
SLURM_JOB_ACCOUNT=bioinfo
SLURM_JOB_NUM_NODES=1
SLURM_STEP_TASKS_PER_NODE=1
SLURM_STEP_NODELIST=i01
forsythc@i01:~$
```

Queuing System - Submitting a Job

Scripting

Converting code into a script is useful.

- Easy to run
- Easy to maintain
- Easy to distribute
- Easy to automate

Queuing System - Submitting a Job

Scripting

1. #! (SheBang) - First line in file which defines the interpreter

   ```
   #!/bin/bash
   ```

2. Permissions - At least the owner of the file requires execute permissions

   ```
   chmod u+x myscript.sh
   ```

3. Pass arguments via command line - Makes script reusable

   ```
   myscript.sh prot.fasta 2
   ```

4. Add to PATH - Makes script callable by name and without path

   ```
   export PATH=~/bin:$PATH #add to .bashrc
   ```

Queuing System - Submitting a Job

Scripting

Bash commands:

```
cd ~/mywork
module load ncbi-blast
module load db-ncbi
blastp -query prot.faa -db $NCBI_DB/nr -out prot.txt -num_threads 2
```

Queuing System - Submitting a Job

sbatch

- Used to submit batch(non-interactive) jobs
- Examples:
  - sbatch sbatch_script.sh
  - sbatch --ntasks=12 --mem=24gb --time=2-10:30:00 sbatch_script.sh

```
forsythc@chuck-ws: ~
forsythc@owl:~/slurm-scripts$ sbatch sleep-60.sh
Submitted batch job 1003897
forsythc@owl:~/slurm-scripts$ sbatch -p batch sleep-60.sh
Submitted batch job 1003899
forsythc@owl:~/slurm-scripts$ sbatch --mem=20gb sleep-60.sh
Submitted batch job 1003900
forsythc@owl:~/slurm-scripts$
```

Queuing System - Submitting a Job

sleep-60.sh

```
#!/bin/bash -l

#SBATCH --ntasks=1
#SBATCH --time=00:00:60

# Print current date
date

# sleep for 60 seconds
/bin/sleep 60

# Print name of node
hostname
```

Queuing System - Submitting a Job

multinode.slurm

```
#!/bin/bash -l

#SBATCH --nodes=4-4
#SBATCH --ntasks=4
#SBATCH --time=00:00:30

echo "--------"
srun hostname
echo "--------"
```

Queuing System - Submitting a Job

SBATCH_SCRIPT.sh

```
#!/bin/bash -l
#SBATCH --ntasks=10
#SBATCH --mem=1G
#SBATCH --time=1-00:15:00     # 1 day and 15 minutes
#SBATCH --output=my.stdout
#SBATCH --mail-user=useremail@address.com
#SBATCH --mail-type=ALL
#SBATCH --job-name="just_a_test"
#SBATCH -p intel # This is the default partition, you can use any of the following; intel, batch, highmem, gpu

# Print current date
date

# Load samtools
module load samtools

# Change directory to where you submitted the job from, so that relative paths resolve properly
cd $SLURM_SUBMIT_DIR

# Concatenate BAMs
samtools cat -h header.sam -o out.bam in1.bam in2.bam

# Print name of node
hostname
```

Queuing System - Array Jobs

SBATCH_SCRIPT.sh

```
#!/bin/bash -l
#SBATCH --ntasks=10
#SBATCH --mem=1G
#SBATCH --time=1-00:15:00     # 1 day and 15 minutes
#SBATCH --output=my.stdout
#SBATCH --mail-user=useremail@address.com
#SBATCH --mail-type=ALL
#SBATCH --job-name="just_a_test"
#SBATCH -p intel # This is the default partition, you can use any of the following; intel, batch, highmem, gpu

# Print current date
date

# Load samtools
module load samtools

# Change directory to where you submitted the job from, so that relative paths resolve properly
cd $SLURM_SUBMIT_DIR

# Concatenate BAMs
samtools cat -h header.sam -o out.bam in1.bam in2.bam

# Print name of node
hostname
```

Queuing System - Submitting a Job

Dependency submission (man sbatch)

```
sbatch --dependency=after:JOBID myscript.sh
sbatch --dependency=afterany:JOBID myscript.sh
sbatch --dependency=afterok:JOBID myscript.sh
sbatch --dependency=afternotok:JOBID myscript.sh
```

```
forsythc@chuck-ws: ~
forsythc@owl:~/slurm-scripts$ sbatch sleep-60.sh
Submitted batch job 1003907
forsythc@owl:~/slurm-scripts$ sbatch --dependency=afterok:1003907 sleep-60.sh
Submitted batch job 1003908
forsythc@owl:~/slurm-scripts$ squeue -l -u forsythc
Mon Jul 17 11:35:54 2017
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
           1003907     intel  sleep60 forsythc  PENDING       0:00      1:00      1 (Priority)
           1003908     intel  sleep60 forsythc  PENDING       0:00      1:00      1 (Dependency)
```

Queuing System - job info

Check how busy the cluster is overall (shows all job in all states):

```
squeue -l
```

Check how many jobs overall are in the running state:

```
squeue -l -t R
```

Check how many jobs overall are in the pending state:

```
squeue -l -t PD
```

Check the state of your jobs:

```
squeue -l -u <user_username>
```

```
forsythc@chuck-ws: ~
forsythc@owl:~/slurm-scripts$ squeue -l -u forsythc
Fri Jul  7 14:12:34 2017
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
            960445     batch  sleep60 forsythc  RUNNING       0:39      1:00      1 c18
```

Queuing System - job info detail

Detailed info about a specific job:

```
scontrol show job <JOBID>
```

```
forsythc@chuck-ws: ~
Submitted batch job 969201
forsythc@pigeon:~/slurm-scripts$ scontrol show job 969201
JobId=969201 JobName=sleep60
   UserId=forsythc(3365) GroupId=bioinfo(1054) MCS_label=N/A
   Priority=4294207521 Nice=0 Account=bioinfo QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:06 TimeLimit=00:01:00 TimeMin=N/A
   SubmitTime=2017-07-10T15:57:02 EligibleTime=2017-07-10T15:57:02
   StartTime=2017-07-10T15:57:14 EndTime=2017-07-10T15:58:14 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=intel AllocNode:Sid=pigeon:19625
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=i02
   BatchHost=i02
   NumNodes=1 NumCPUs=2 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=2,mem=2G,node=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryCPU=1024M MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/rhome/forsythc/slurm-scripts/sleep-60.sh
   WorkDir=/rhome/forsythc/slurm-scripts
   StdErr=/rhome/forsythc/slurm-scripts/slurm-969201.out
   StdIn=/dev/null
   StdOut=/rhome/forsythc/slurm-scripts/slurm-969201.out
   Power=
```

Queuing System - job control

Cancel your job:

```
scancel <JOBID>
```

Cancel multiple jobs:

```
scancel <JOBID1> <JOBID2> <JOBID3>
```

Cancel ALL your job (caution this will kill all running and queued jobs):

```
squeue --user $USER --noheader --format '%i' | xargs scancel
```

Sharing Files on the Web

Simply move the files into your html directory when you want to share them.

For example, log into the cluster and do the following:

1. Go to your web directory
   ```
   cd ~/.html/
   ```
2. Create a default test file
   ```
   echo '<h1>Hello!</h1>' > index.html
   ```

Now, test it out by pointing your web-browser to http://biocluster.ucr.edu/~username/

Be sure to replace 'username' with your actual user name, and also check permissions on shared directories and parent directories.

For password protecting html content follow these instructions:
http://manuals.bioinformatics.ucr.edu/home/hpc#TOC-Password-Protect-Web-Pages

In the works

Investigating HPCC cloud integration

- Amazon AWS
- Google Cloud
- Microsoft Azure

Helpful Resources

1. Command Line
   --help, -h, man
2. Online Manual
   http://hpcc.ucr.edu
3. Storage Usage
   https://dashboard.bioinfo.ucr.edu
4. Announcements
   http://hpcc.ucr.edu/news.html