Using HPCC resource

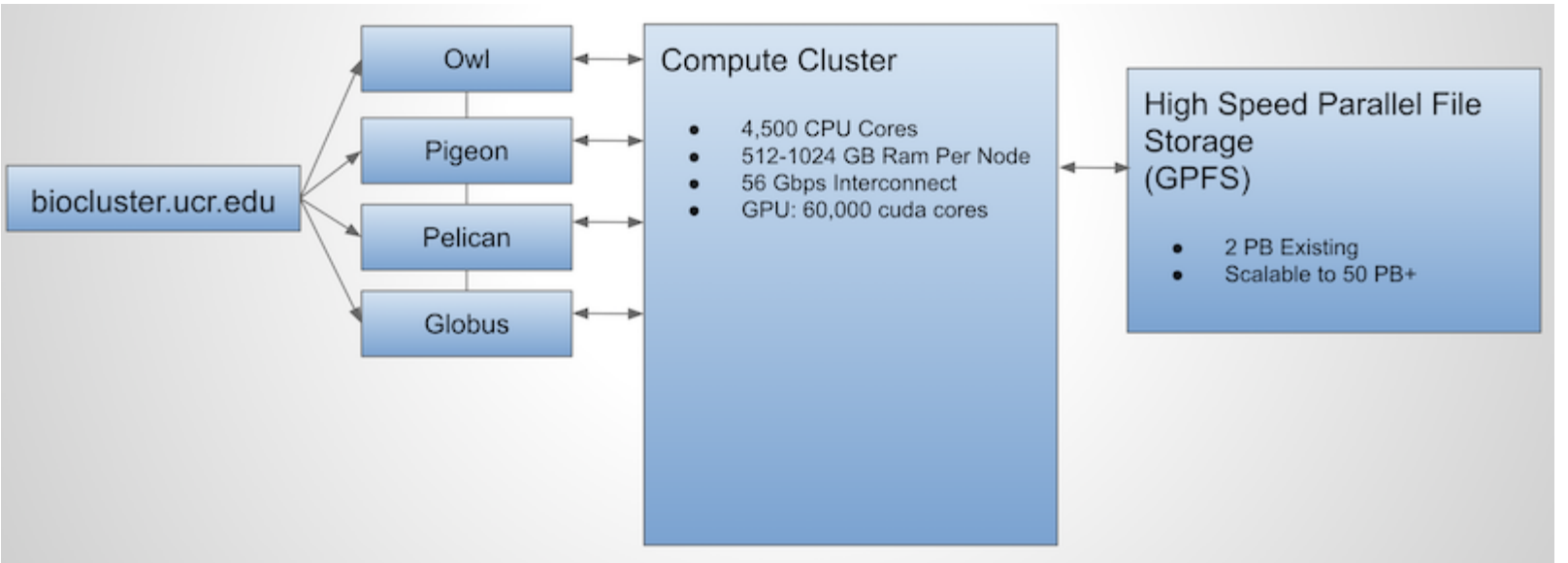
HPCC

High Performance Computing Cluster @UCR http://hpcc.ucr.edu

What is it?

- Collection of computers ~6500 CPU cores
- Includes large memory (1Tb of memory) machines for genome assembly
- Specialized computers with GPU for fast specialized computation
- 2 Petabytes of storage and 2Pb of backup

HPCC


biocluster.ucr.edu
Owl
Pigeon
Pelican
Globus
Compute Cluster
4,500 CPU Cores
512-1024 GB Ram Per Node
56 Gbps Interconnect
GPU: 60,000 cuda cores
High Speed Parallel File Storage (GPFS)
2 PB Existing
Scalable to 50 PB+

logging in

```
$ ssh -XY USERNAME@biocluster.ucr.edu
password:
```

providing -XY makes sure your graphical output can go back and forth from your computer and the cluster

When debugging you can provide the -v flag to help identify all the messages going back and forth

```
$ ssh -v USERNAME@biocluster.ucr.edu
```

Setup your SSH keys to make logging in with your SSH password instead.

file system

On Biocluster there are a couple of folder structures to understand

```
/rhome/USERNAME # your home directory - limited space (20gb)
/bigdata/labname/USERNAME # your 'bigdata' folder (bigger space (100gb+)
/bigdata/labname/shared # shared folder space for your lab
```

Currently everyone is in the the gen220 'lab' during this course so you have access to `/bigdata/gen220/shared` and `/bigdata/gen220/USERNAME`

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

`/scratch` - local space on a cluster node which is FAST disk access but temporary (30 days)

transferring files

Graphical tools:

Filezilla - https://filezilla-project.org/download.php

Command-line:

```
# interactive FTP client
$ sftp USERNAME@biocluster.ucr.edu

# copy a file
$ scp USERNAME@biocluster.ucr.edu:fileoncluster.txt ./file-on-your-machine.txt

# copy a folder, recursively
$ scp -r USERNAME@biocluster.ucr.edu:/bigdata/gen220/shared/simple .

# rsync copies, but can check and only copy changed files
$ rsync -a --progress USERNAME@biocluster.ucr.edu:/bigdata/gen220/shared/simple .

# copy FROM your computer TO the cluster, swap order - here
# copy a folder back to your HOME directory
$ scp -r simple USERNAME@biocluster.ucr.edu:
```

Module system for UNIX programs

Remember your `$PATH` variable in your SHELL sets the available programs you can run.

We have 1000+ packages of tools installed on biocluster to run bioinformatics and cheminformatics and other analyses.

To run these tools we need to modify the PATH to let the SHELL know about them (as all aren't loaded by default).

Use the unix Module system to do this. To list all possible modules:

```
module avail
```

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 a module: `module unload R`

Using the cluster

Currently only shown login to the main "head" node (biocluster.ucr.edu)

To use the 6500 CPUs we need to submit job for running. This is called a job management or queueing system.

We use **SLURM** on the UCR system currently.

http://hpcc.ucr.edu/manuals_linux-cluster_jobs.html

Submitting a job

- Getting an interactive shell (eg get your own CPU to do work on)

```
$ srun --pty bash -l
$ srun --nodes 1 --ntasks 2 --mem 8gb --time 8:00:00 --pty bash -l
```

```
#!/bin/bash
module load ncbi-blast
blastn -num_threads 2 -query file.fa -db db.fa -out result.blastn
```

- Batch/non-interactive job

```
$ sbatch myjob.sh
$ sbatch --ntasks 16 --nodes 1 --mem 12gb --time 72:00:00 myjob.sh
```

Job resources

Requesting job resources

- number of CPUs: `--ntaks N`
- memory: `--mem Xgb`
- runtime: `--time 12:00:00`
- outputfile: `--out results.log`

Can also set these INSIDE the script

```
#!/bin/bash
#SBATCH --nodes 1 --ntasks 2 --mem 2gb --time 3:00:00
module load ncbi-blast
blastn -num_threads 2 -query file.fa -db db.fa -out result.blastn
```

Is my job running?

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`

Cancel a job

```
$ squeue -l
Wed Oct 10 15:31:39 2018
       JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NOD
     2940979     intel zsz4_002  szzhang  PENDING       0:00 41-16:00:00
     3316391     intel    MPI60 agottsch  PENDING       0:00 30-00:00:00
     3371635   highmem anacapa-  erankin  PENDING       0:00 1-16:30:00
     3371167   highmem r-cluste punzu001  PENDING       0:00 2-00:00:00
     2852479   highmem test_hum      yhu  PENDING       0:00 20-00:15:00
```

```
$ scancel JOBID
```