QIIME2 Workshop: Day 1
Introduction to HCC

Exercises: Running Applications

- In order to do some comparisons, your collaborator Professor Reddy, wants to analyze his newest samples using the same pipeline used to analyze data from mid 2017. His previous pipeline used samtools version 1.4. Would he be able to run this analysis on the Crane cluster?

```
module spider samtools
```

  No, version 1.4 is not available on Crane

- You agree to work with your collaborator to develop a more recent version of his pipeline. What is the most recent version of samtools available?

  From the output above, we can see the newest version is 1.6

Exercises: Running Applications (cont.)

- In addition to the BLAST+ alignment package (the **`blast`** module), HCC also has a parallel version of BLAST in the **`mpi-blast`** module. How many other implementations of the BLAST algorithm are available on HCC clusters?

  `module keyword blast`

  Of the 6 results, 4 use the BLAST algorithm: **`blast`**, **`blast-legacy`**, **`mpiblast`** and **`rmblast`**.

Exercises: Running Applications (cont.)

- Research Assistant Sofia has 10,000 sequences she wants to do a BLAST alignment against the human genome. She decides she wants to run it in parallel using the **`mpiblast`**. What command(s) does she need to use to load the package?

```
module load cuda
module load mpiblast
```

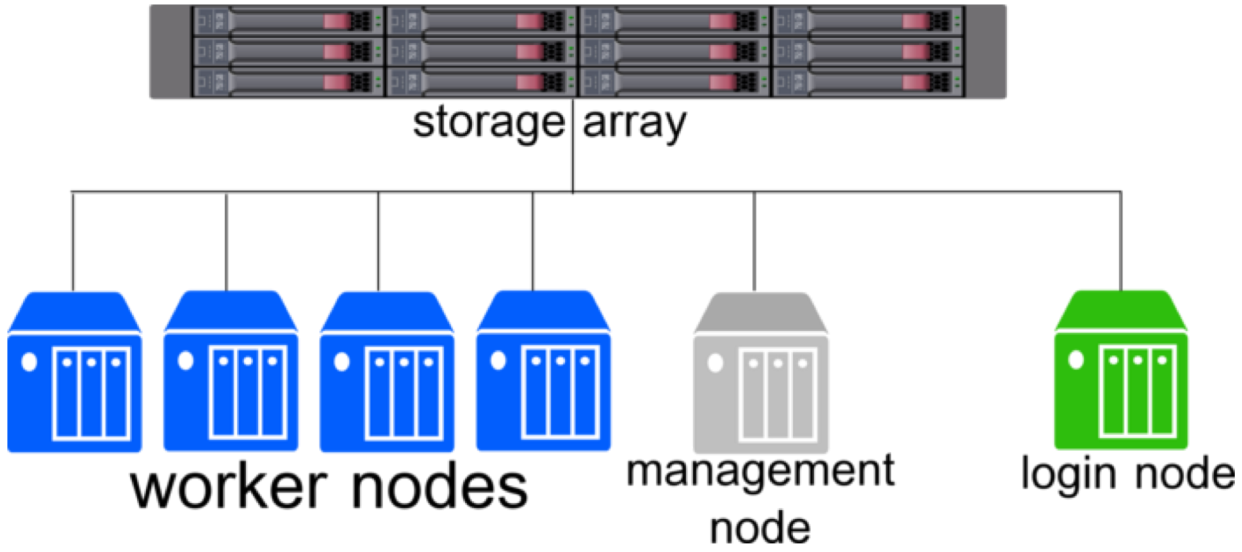
or

```
module load cuda mpiblast
```

Schedule

9:00 – 10:30	Introduction to HPC Connecting to the Clusters Navigating in Bash
10:30 – 10:45	Break
10:45 – 12:00	File Manipulation Wildcards and Pipes
12:00 – 1:00	Lunch
1:00 – 2:15	Writing Reusable Scripts Running Applications on the Clusters
2:15 – 2:30	Break
2:30 – 4:00	**Submitting Jobs** **Transferring Data with Globus**

Running Jobs



- All analysis must be done on the worker nodes
- Processes started on the login node will be killed
  - Limit usage to brief, non-intensive tasks like file management and editing text files
- HCC uses the SLURM scheduler to manage and allocate resources
- Jobs can be run in interactive or batch format

Interactive vs Batch Jobs

Interactive Jobs

- Allows the user to type in commands and run programs **interactively**
- Must remain connected and wait for resources to be allocated
- Job can be interrupted
- Uses the **`srun`** command

Batch Jobs

- Create a **submit script** which contains commands and job information
- Add it to the scheduler queue
- The scheduler holds the job until resources become available
  - User can disconnect and the job will remain queued
- Uses the **`sbatch`** command

Batch Jobs

- To create a batch job, the user must first make a submit script
  - Submit scripts include all job resource information:
    - number of nodes
    - required memory
    - Runtime
  - **If the job exceeds the requested memory or time, it will be killed.**
- Submit script is then added to the job queue using the **`sbatch`** command
- **`squeue`** will show queued and running jobs
- **`sacct`** can be used to find information about completed jobs

Submit Scripts

Commands

Any commands after the SBATCH lines will be executed by the interpreter specified in the shebang – similar to what would happen if you were to type the commands interactively

```
[cathrine98@c3712.crane matlab-tutorial]$ cat invert_single.slurm
#!/bin/sh
#SBATCH --time=0:10:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --mem=10GB
#SBATCH --mem-per-cpu=10GB
#SBATCH --job-name="invert_single"
#SBATCH --error="invert_single.err"
#SBATCH --output="invert_single.out"

module load matlab/r2014b

cd $WORK/matlab-tutorial
mkdir -p /tmp/$SLURM_JOB_ID
matlab -nodisplay -r invertRand
[cathrine98@c3712.crane matlab-tutorial]$
```

SBATCH directives

Immediately after the shebang and before any commands.

Required SBATCH directives are time, nodes and mem.

Common SBATCH Directives

Command	What it does
`--nodes`	Number of nodes requested
`--time`	Maximum walltime for the job – in DD-HHH:MM:SS format maximum of 7 days on batch partition
`--mem`	Real memory (RAM) required per node - can use KB, MB, and GB units – default is MB Request less memory than total available on the node The maximum available on a 512 GB RAM node is 500, for 256 GB RAM node is 250
`--ntasks-per-node`	Number of tasks per node – used to request a specific number of cores
`--mem-per-cpu`	Minimum of memory required per allocated CPU – default is 1 GB
`--output`	Filename where all STDOUT will be directed – default is slurm-<jobid>.out
`--error`	Filename where all STDERR will be directed – default is slurm-<jobid>.out
`--job-name`	How the job will show up in the queue

Exercises: Submitting Jobs

- Navigate into the **`matlab`** directory and locate the **`invertRand.submit`** file. This submit file runs a MATLAB script which randomly generates a 10,000 x 10,000 matrix, inverts it and outputs how long the inversion took.
  - Look at the contents of **`invertRand.submit`** - How many nodes this will run on? How many cores? How much memory and time is requested?
  - Add **`invertRand.submit`** to the scheduler queue. Once the job completes, check the output to see how long it took to invert the matrix.

Exercises: Submitting Jobs (cont.)

- Copy the **`invertRand.submit`** file, renaming it to **`invertRandParallel.submit`**
  - Edit invertRandParallel.submit to use 10 cores
  - Submit the job again and compare the time to your initial run. How much faster or slower is it?
  - Compare times with your neighbor. Did you see the same amount of improvement?

Determining Parameters

How many nodes/memory/time should I request?

- **Short answer:** We don't know.
- **Long answer:** The amount of time and memory required is highly dependent on the application you are using, the input file sizes and the parameters you select.
  - Speak with someone else who has used the software before
  - Trial and error
    - Check the output and utilization of each job will help you determine what parameters you will need in the future.
    - Try different combinations and see what works and what doesn't

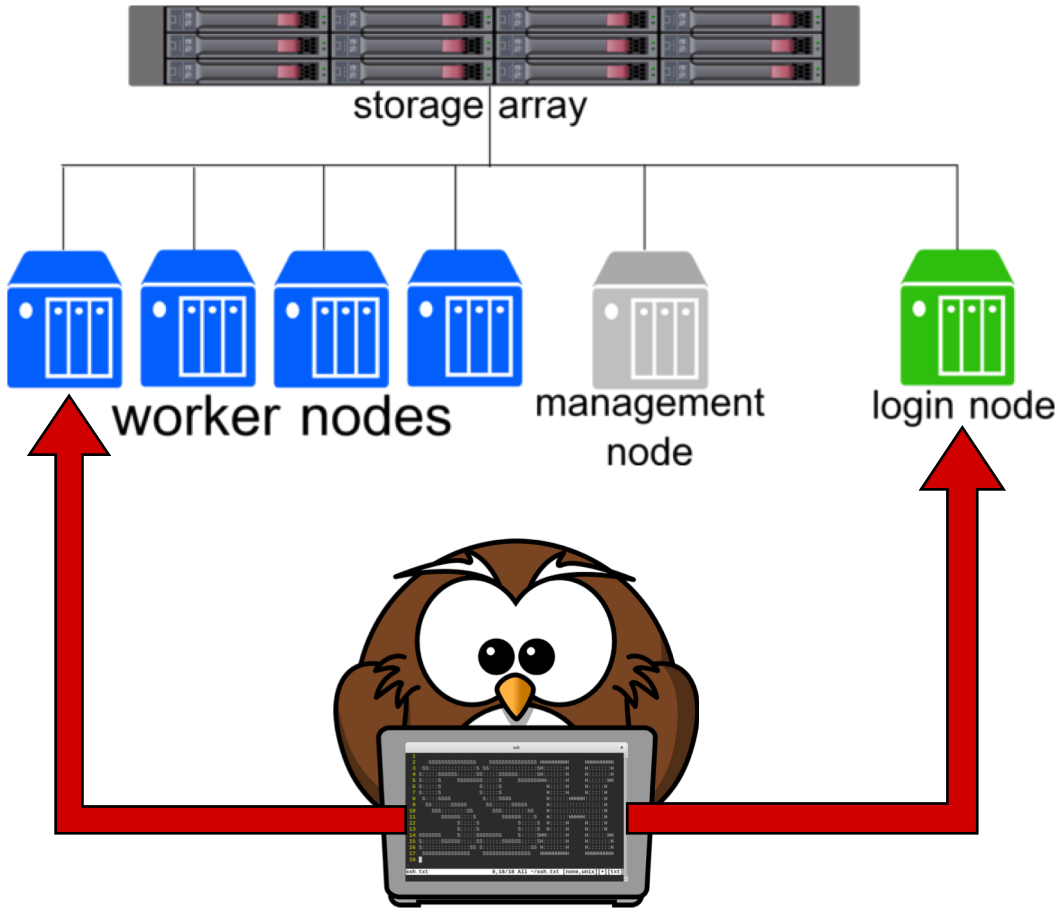
Submit Files Best Practices

- **Put all module loads immediately after SBATCH lines**
  - Quickly locate what modules and versions were used.
- **Specify versions on module loads**
  - Allows you to see what versions were used during the analysis
- **Use a separate submit file for each job**
  - Instead of editing and resubmitting a submit files – copy a previous one and make changes to it
  - Keep a running record of your analysis
- **Redirect output and error to separate files**
  - Allows you to see quickly whether a job completes with errors or not
- **Separate individual workflow steps into individual jobs**
  - Avoid putting too many steps into a single job

Interactive Jobs

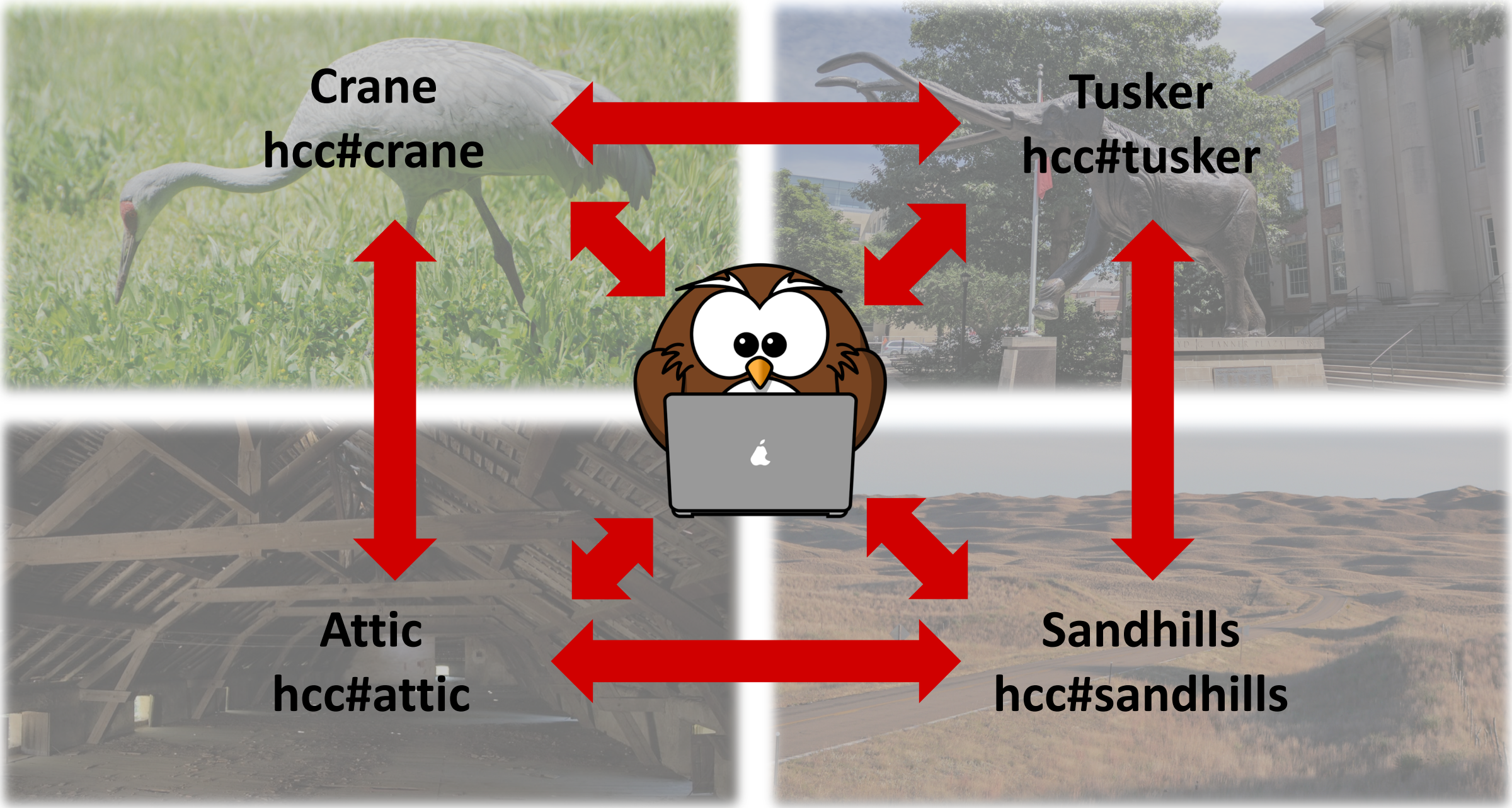
```
[cathrine98@login.crane ~]$ srun --nodes=1 --mem=1gb --pty bash
srun: job 7741317 queued and waiting for resources
srun: job 7741317 has been allocated resources
[cathrine98@c3712.crane ~]$
```

- Interactive jobs work very similarly to batch jobs
- Once resources are allocated, commands can be input interactively
  - All output is directed to the screen
- Once the requested time is up, you will receive a 32 second warning before the job is killed.



Transferring Files with Globus

- **Globus Connect**
  - http://globus.org
  - Fast, secure and robust transfers with user-friendly web interface
  - Uses the High-Speed transfer nodes by default
  - Can transfer directly between clusters, Attic and personal machine


Crane
hcc#crane
Tusker
hcc#tusker
Attic
hcc#attic
Sandhills
hcc#sandhills

Getting Started with Globus

- **Sign up for a Globus account**
  - Individuals from UNL and UNMC can login using their institute credentials
- **Install Globus Connect Personal**
  - Download page: https://www.globus.org/globus-connect-personal
- **Activate HCC endpoint(s)**
  - Instructions: https://hcc-docs.unl.edu/display/HCCDOC/Activating+HCC+Cluster+Endpoints
- **Login and start making transfers**
  - Login link: https://www.globus.org/SignIn
  - Instructions: https://hcc-docs.unl.edu/display/HCCDOC/File+Transfers+Between+Endpoints

Note: $HOME directories are read-only via Globus.

You can transfer from /home but not to /home.

For more information: https://hcc-docs.unl.edu/display/HCCDOC/Globus+Connect

Exercises: Transferring Files with Globus

1. Ensure you have Globus Connect Personal installed and your personal endpoint activated. If you have difficulty with this, please use your red sticky note to alert a helper.

2. Using Globus Connect, copy the **`bio.txt`** file from the Tusker cluster to Crane in your $WORK directory

3. Try copying the file to your $HOME directory. What happens?

4. Download the file onto your personal computer.

Workflow Tips

- **Test/Develop your workflow on a truncated set of data**
  - Use tutorial data if provided
  - Otherwise, make your own
- **Run commands in an interactive job first**
- **Try different combinations of SBATCH options to find what works best**
  - Run multiple jobs on your truncated data to see which run faster
- **Always check output files – even for jobs that produce output**
  - Were they created?
  - What do they contain?
- **Check what resources the job actually used**
  - **`sacct -j <job_number> --format Elapsed,MaxRSS`**

What to do if you're stuck

- **Read the Documentation**
  - http://hcc-docs.unl.edu
    - If the documentation doesn't answer your question, leave a comment or email to let us know!
  - Use **`man`** or **`--help`**
- **Consult Google**
  - Useful for application specific errors or general usage
- **Contact Us**
  - Drop in our offices
    - UNL: Room 118 in the Schorr Center on City Campus
    - UNO: Room 158 in the Peter Kewit Institute on Center Campus
  - Email hcc-support@unl.edu