Setup Instructions

Please complete these steps before the lessons start at 1:00 PM:
Setup instructions: http://hcc.unl.edu/june-workshop-series

If you need help with the setup, please put a red sticky note at the top of your laptop.

When you are done with the setup, please put a green sticky note at the top of your laptop.
June Workshop Series
June 19th: Introductory HCC
University of Nebraska – Lincoln
Holland Computing Center

Jingchao Zhang, Carrie Brown, Adam Carpez,
Emelie Harstad, Natasha Pavlovikj, Caughlin Bohn
June Workshop Series Schedule

**June 5th: Introductory Bash**
Learn how to connect to Holland's High Performance Computing cluster, Crane and how to navigate in the Linux command line with the Bash shell. Topics include the use of pipes, filters, loops, and handling of text files. Lesson materials based on Software Carpentry lessons.

**June 12th: Advanced Bash and Git**
Continue mastering the Bash shell by learning how to write reusable shell scripts to help automate tasks. We will also be looking at how using version control with Git can help streamline collaboration and manage revisions. Lesson materials based on Software Carpentry lessons.

**June 19th: Introductory HCC**
Learn the basics of working with Holland resources. Topics include an overview of HCC offerings, submitting batch jobs, handling data and tips on how to optimize workflow.

**June 26th: Anvil: HCC’s Cloud and the Open Science Grid**
Learn about Anvil, HCC's Open Stack resource which allows researchers greater flexibility by offering cloud-based virtual machines which are ideal for applications that are not well suited to the Linux command line environment. Also, learn about the Open Science Grid (OSG), a nation-wide collection of opportunistic resources that are well suited to large high-throughput applications. Learn what jobs are ideal for OSG and how to begin using this service.
Outline

• Access HCC Clusters

• File Management

• Utilize Modules (Software)

• Job Submission

• HCC Tour
What is a Cluster?
Node

Cluster
**Sandhills**
- 108 node “Condominium” Linux cluster
- 62 nodes with 192 GB RAM
  - 2.3 Ghz and 64 cores/node
- 44 nodes with 128 GB RAM
  - 2.0 Ghz and 32 cores/node
- 2 nodes with 156 GB RAM
  - 1.9 Ghz and 48 cores/node
- 175 TB shared storage
  - 1.5 TB scratch per node

**Crane**
- 548 node Linux cluster
- 452 nodes with 64 GB RAM
  - 2.6 Ghz 2 CPU/16 cores
- 59 nodes with 256 GB RAM
  - 2 CPU/ 36 cores per node
- 37 nodes with 512 GB RAM
  - 2 CPU/ 36 cores per node
- 2 nodes with 156 GB RAM
  - 1.9 Ghz and 48 cores/node
- 175 TB shared storage
  - 1.5 TB scratch per node

**Tusker**
- 82 node Linux cluster
- 2.1 Ghz – 4 CPU/64 cores per node
- 256 GB RAM in most nodes
  - 2 nodes with 512 GB RAM
  - 1 node with 1024 GB RAM
- 500 TB shared storage
  - 500 GB scratch per node

**Red**
- 337 node Linux cluster
- 5888 cores
- 6.67 PB raw storage
- Used by HCC staff scientists to analyze LHC data

**Tusker**
- 82 node Linux cluster
- 2.1 Ghz – 4 CPU/64 cores per node
- 256 GB RAM in most nodes
  - 2 nodes with 512 GB RAM
  - 1 node with 1024 GB RAM
- 500 TB shared storage
  - 500 GB scratch per node.
Anvil

• HCC’s Cloud
• Customizable virtual machines based on the OpenStack Software
• For projects that are not well served by a traditional Linux environment:
  • Windows Applications
  • Other Linux Flavors
  • Web Server
  • Database

Attic

• Near-line data archive
• Backed up in both Lincoln and Omaha for disaster tolerance
• 10 GB/s transfer speed to and from the clusters when using Globus Connect
• Cost per TB is significantly lower than commercial cloud services ($25 / Tb /year)
How to connect

Mac OS / Linux
• Open Terminal
• Type in the following command and press Enter:

```
ssh <user_name>@crane.unl.edu
```

Windows
• Open PuTTY
• Type the Host Name and click Open
• On the second screen, click Yes

Nothing will appear on your screen as you type your password. This is normal.
Two-Factor Authentication

If you are using a cell phone:

• Press 1 to receive a push notification on your device
• Press 2 to receive an automated phone call. Then enter the passcode provided.
• Press 3 to receive a list of passcodes via text message. Enter the passcode which starts with the number indicated

If you are using a hardware token (Yubikey):

• Insert the Yubikey into the computer’s USB slot on your computer. A green light should come on inside the gold circle.
• Press your finger against the gold circle and a passcode will be generated.

The Duo Mobile app can also be used to generate passcodes, even when cell service is unavailable.
Change password

• From the terminal, type the `passwd` command.

• All HCC passwords must meet the following requirements when they are created or changed:
  • at least 8 characters long
  • at least 1 capital letter
  • at least 1 number
  • can not contain an existing dictionary word
  • can not contain information from user's account details (`username`, `email`)

• If you forget your password, send an email to hcc-support@unl.edu.
Exercises

1. Login to 3 clusters using 3 different DUO authentication methods:
   (If you use a demo account and only have a Yubikey, then just login to all clusters using passcodes.)
   - tusker.unl.edu
   - sandhills.unl.edu
   - crane.unl.edu

2. Change your password if you haven’t do so already.

NOTE: We will use crane.unl.edu for today’s tutorial. Make sure you are on Crane at the end of this exercise.

Once you have finished, put up your green sticky note.

If you need help, put up your red sticky note.
Outline

• Access HCC Clusters

• **File Management**

• Utilize Modules (Software)

• Job Submission
Shell Command Review

cd
ls
pwd
mv
rm
cp
nano
touch
wc
sort
echo
cat..
>
>>

change directory
list
print working directory
move or rename
remove
copy
text editor
create empty file
word count
sort lines in file
print to screen
print contents of file(s) to screen
Parent directory
Current directory
Redirect output to file (overwrite)
Redirect output to file (append)

A few tips:

• **Ctrl + C** to escape.
  • E.g. when you type ‘cat’ without a file name.

• Lower case **q** to quit
  • E.g. when you read the command manual. *man ls*

• **Tab** to auto-complete command/filename/directory.
Linux Directory Tree

Week 1 exercise
HCC File System

Example: /home/swanson/jingchao /work/swanson/jingchao /common/swanson/jingchao
# HCC File System

<table>
<thead>
<tr>
<th></th>
<th>home</th>
<th>work</th>
<th>common</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick access:</td>
<td>$HOME</td>
<td>$WORK</td>
<td>$COMMON</td>
</tr>
<tr>
<td>Back up:</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Job IO:</td>
<td>NO</td>
<td>YES</td>
<td>YES (not recommended)</td>
</tr>
<tr>
<td>Purge:</td>
<td>NO</td>
<td>YES (6 months)</td>
<td>NO</td>
</tr>
<tr>
<td>Quota:</td>
<td>20 Gb / user</td>
<td>50 Tb / group</td>
<td>1 Tb / user</td>
</tr>
<tr>
<td>Accessible from all clusters:</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
A few notes

• Put **important files**, such as source code, in $HOME.

• Files in $WORK and $COMMON cannot be recovered.

• Use $WORK for job input/output. It has the **highest IO performance**.

• Use Attic for longer term **archive storage**. ($25 / Tb / year)
Exercises

1. Change to your $COMMON directory on Crane. Create an empty file bio.txt. (hint: touch)

2. From Tusker or Sandhills, find and edit the same bio.txt file to include your name and department. Then go back to Crane and check the contents of this file. What happened? (hint: nano, cat)

3. Change to your $WORK on Crane. Clone the tutorial files by entering the command:
   - `git clone https://github.com/unlhcc/HCCWorkshops.git`

Once you have finished, put up your green sticky note.

If you need help, put up your red sticky note.
Transferring Files

• Transfer files using an SCP client
  • WinSCP (http://winscp.net) – Windows
  • Cyberduck (http://cyberduck.io) – MacOS
  • Filezilla (http://filezilla-project.org) – Windows, MacOS and Linux

• Mac / Linux users: scp
  • Usage: scp user@host:source_file user@host:target_file
    • Example: scp my_file.txt demo01@crane.unl.edu:$WORK/my_file.txt

• 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
Transferring Files (Windows users)

Step 1: WinSCP direct download [link](#). Install the package.

Step 2: Fill in your user information

Step 3: click on Login

Step 4: Use option 1-3 if using Duo mobile app, or use your Yubikey to generate a code.
Transferring Files (Windows users)

Remote files on Crane

Local files on your laptop

Double click this area to change file path manually
Transferring Files (Mac users)

Step 1: Start a new Terminal, change directory to your local Desktop. (cd ~/Desktop)

Step 2: Issue command ‘scp UserName@crane.unl.edu:/path/to/your/file ~/Desktop’

Example:

```
$ cd ~/Desktop/
$ scp jingchao@crane.unl.edu:/home/swanson/jingchao/bio.txt ./
Password:
Duo two-factor login for jingchao

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-0879
2. Phone call to XXX-XXX-0879
3. SMS passcodes to XXX-XXX-0879 (next code starts with: 1)

Passcode or option (1-3): ccccccebrbjikjclruunuifcbvbcblhruvuujbgjhdeh

100% 0 0.0KB/s 00:00
```

Note: Make sure the file you are trying to copy exists on the remote server, otherwise you will receive an error like “scp: /home/swanson/jingchao/bio.txt: No such file or Directory”
Exercises (Using Crane)

1. Copy the ‘bio.txt’ file from Crane $COMMON to your local laptop.
   
   • Windows users: [WinSCP](https://winscp.net/)
   
   • Mac / Linux users: scp
   
   • Optional: Mac users can setup [Cyberduck](https://cyberduck.io/) for file transfer (out of the scope of this tutorial).

Once you have finished, put up your green sticky note.

If you need help, put up your red sticky note.
Getting Started with Globus

• **Sign up for a Globus account**
  • Individuals from UNL, UNK and UNMC can login using their institute credentials
    • [https://www.globus.org/globus-connect-personal](https://www.globus.org/globus-connect-personal)

• **Activate HCC endpoint(s)**
  • Instructions: [https://hcc-docs.unl.edu/display/HCCDOC/Activating+HCC+Cluster+Endpoints](https://hcc-docs.unl.edu/display/HCCDOC/Activating+HCC+Cluster+Endpoints)
  • Activation requires **HCC credentials**

• **Install Globus Connect Personal and activate personal endpoint**

• **Transfer files between endpoints**
  • Instructions: [https://hcc-docs.unl.edu/display/HCCDOC/File+Transfers+Between+Endpoints](https://hcc-docs.unl.edu/display/HCCDOC/File+Transfers+Between+Endpoints)

For more information:
[https://hcc-docs.unl.edu/display/HCCDOC/Globus+Connect](https://hcc-docs.unl.edu/display/HCCDOC/Globus+Connect)
Exercises

1. Using Globus Connect, copy the bio.txt file from Crane $COMMON to your Tusker $WORK directory.

2. Set up a Globus Connect endpoint on your personal computer by following the steps on the slide before.

3. Copy the file HCCWorkshops directory from Crane $WORK to your personal computer.

Once you have finished, put up your green sticky note.

If you need help, put up your red sticky note.
Outline

• Access HCC Clusters

• File Management

• Utilize Modules (Software)

• Job Submission
Running Applications

• All applications installed on HCC clusters are loaded in individual modules implemented with Lmod
  • Lua based module system

• Modules **dynamically change** the user environment
  • $PATH
  • $LD_LIBRARY_PATH

• **Hierarchical** structure
  • If module A is dependent on module B, then module B must be loaded first to load module A

• Typically follows the naming convention <software>/<version>
  • Example: python/2.7

• Load using the **module** command
# Lmod Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>module av</code></td>
<td>Lists all modules available to be loaded</td>
</tr>
<tr>
<td><code>module load &lt;module_name&gt;</code></td>
<td>Load module(s) – can load a list of space delimitated modules</td>
</tr>
<tr>
<td><code>module list</code></td>
<td>Lists all currently loaded modules</td>
</tr>
<tr>
<td><code>module unload &lt;module_name&gt;</code></td>
<td>Unload module(s) – can unload a list of space delimitated modules</td>
</tr>
<tr>
<td><code>module purge</code></td>
<td>Unloads all currently loaded modules</td>
</tr>
<tr>
<td><code>module spider &lt;name&gt;</code></td>
<td>Information about a specific module – can also be used to search</td>
</tr>
</tbody>
</table>

**For more information:**
- `module --help`
- **Available software lists for each cluster:**
  - Crane: [https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Crane](https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Crane)
  - Tusker: [https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Tusker](https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Tusker)
  - Sandhills: [https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Sandhills](https://hcc-docs.unl.edu/display/HCCDOC/Available+Software+for+Sandhills)
A few notes

• Crane/Tusker/Sandhills have different software stacks.

• If you are uncertain whether a software is available, check the webpages mentioned in the previous slide.

• To request a software installation, fill this form.
Exercises

1. Find a list of ‘Core’ modules that do not have any dependences.

2. Load one or more of the modules from the list you just found. Now unload them.

3. Lookup information about the MATLAB module.
   • How many different versions are available?
   • Load a specific version using the `<module_name>/<version>` name

4. Try loading the lammps/10Feb2017 module. Did it work?
   • Now load these modules first
     • compiler/intel/15
     • openmpi/1.10
   • Now try loading lammps/10Feb2017 again. Does it work now? Why?
   • Unload all modules with only one command.
Outline

• Access HCC Clusters

• File Management

• Utilize Modules (Software)

• Job Submission
Running Jobs

• HCC uses the SLURM job scheduler to manage and allocate resources.

• All computations must be done on the worker nodes.

• Job processes started on the login node will be killed.
Batch vs Interactive Jobs

**Batch Jobs**

- The user creates a **submit file** which contains all commands and job information and add it to the queue
- User can disconnect and the job will remain queued
- Uses the `sbatch` command

**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
SLURM Batch Job

Submit jobs to run on the worker nodes

Write a SLURM file
Submit Scripts

Shebang
The shebang tells Slurm what interpreter to use for this file. This one is for the shell (Bash)

```
#!/bin/sh
```

Name of the submit file
This can be anything. Here we are using “serial.slurm” the .slurm makes it easy to recognize that this is a submit file.

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

```
module load matlab/r2014b
mkdir -p /tmp/$SLURM_JOB_ID
matlab -nodisplay -r "invertRand('10^4'), quit"
```

SBATCH options
These must be immediately after the shebang and before any commands.

The only required SBATCH options are time, nodes and mem, but there are many that you can use to fully customize your allocation.
## Common SBATCH Options

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

### For more information:
- `sbatch --help`
- SLURM Documentation: [https://slurm.schedmd.com/sbatch.html](https://slurm.schedmd.com/sbatch.html)
Submit a job

To submit a job to the cluster, use command ‘sbatch FILENAME’

```
[jingchao@login.crane.matlab]$ sbatch invertRand.submit
Submitted batch job 8624967
[jingchao@login.crane.matlab]$  
```

To check job status, use command ‘squeue –u USERNAME’ or ‘squeue –j JOBID’

```
[jingchao@login.crane.matlab]$ squeue -u jingchao
     JOBID PARTITION   NAME      USER ST TIME NODES Nodelist(reason)
   8624986  batch invertRa jingchao R 0:05   1 c1919
   8624985  batch invertRa jingchao R 0:13   1 c1919
[jingchao@login.crane.matlab]$ squeue -j 8624985
     JOBID PARTITION   NAME      USER ST TIME NODES Nodelist(reason)
   8624985  batch invertRa jingchao R 1:02   1 c1919
[jingchao@login.crane.matlab]$  
```
Batch Jobs

• Submit script is added to the job queue using the `sbatch` command.
  • E.g. `sbatch serial.slurm`

• `squeue` will show queued and running jobs.
  • E.g. `squeue –j JOBID; squeue –u USERNAME`

• To scancel a job use `scancel`.
  • E.g. `scancel JOBID`

• `sacct` can be used to find information about completed jobs.
  • E.g. `sacct –j JOBID`
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.
  
  • It requires domain knowledge.
  
  • Ultimately, it comes down to trial and error.
Exercises

1. Navigate into the $WORK/HCCWorkshops/matlab directory of the HCCWorkshops directory and locate the serial.slurm file
   • This submit file runs a MATLAB script which inverts a 10,000 x 10,000 randomly generated matrix and outputs the length of time it took to perform the inversion.
   • Look at the contents of serial.slurm - How many nodes this will run on? How many cores? How much memory and time is requested?
   • Submit the serial.slurm job. Check the output to see how long it took to invert the matrix.

2. Copy the serial.slurm file to a new file using the cp command.
   • Edit the new slurm file to use 4 cores.
     • If you need help, look at the parallel.slurm file which is already configured to run on 4 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?

3. If there’s time, try different combinations of SBATCH commands and see how the running time changes.
SLURM Interactive Job

Switch to worker node
Interactive Jobs

- Interactive jobs start with `srun --pty` command.
- To exit an interactive job, type `exit` or use `Ctrl + d`.
- Once resources are allocated, commands can be typed interactively.
  - All output is directed to the screen.
Exercises

1. Request an interactive job using ‘srun’ command
   • If you can’t think of a setup, try below configurations:
     • 1 node, 1 core with 2 GB RAM each core
     • 2 nodes, 1 core per node with 2 GB RAM each core
     • Do not forget to exit an interactive job first before you start another one

2. Using your interactive job from above, load the module for your favorite programming language (Python, R, MATLAB, etc.)
   • Run the program interactively
     • **Python:**
       1. module load python/3.6
       2. type ‘python’ in the terminal
     • **R:**
       1. module load R/3.4
       2. type ‘R’ in the terminal
     • **Matlab:**
       1. module load matlab/r2014b
       2. type ‘matlab’ in the terminal
A few notes

• We give you credit (CPU hours) if you acknowledge HCC in your work.

• You can buy dedicated resources from HCC. Price listed here.
  
  • For example, a Crane node (16 cores / 64 Gb mem) costs $600 / year.
What to do if you’re stuck

• Read the Documentation
  • http://hcc-docs.unl.edu

• Consult Google
  • Useful for application specific errors or general usage

• Contact Us
  • 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