# HPC On-boarding

Sean Cleveland Ph.D, Ron Merrill Ph.D,
David Schanzenbach M.S.

Information Technology Services
Department of Cyberinfrastructure
University of Hawai'i

https://www.hawaii.edu/its/ci/
***uh-hpc-help@lists.hawaii.edu***

January 18, 2016

# Outline

# Outline

# Parallel Computing

- High Performance Compute
  - Each separate process can send and receive data amongst other processes (MPI & OpenMP)
  - If processes must communicate for the overall program to proceed, high-speed networking is needed (only MPI)
- High Throughput Compute
  - *Pleasantly Parallel* – Processes are independent and no communication is necessary

# Outline

# Cray CS300 – History

- Brought online Fall 2014 – Spring 2015
- Initial investment of 1.8 Million by the University of Hawai‘i (UH)
- As of October 2015, more than 200 users have been granted access to the cluster

# Cray CS3000 – Compute Nodes

- 3,800 total cores – **Spring upgrade** $\rightarrow$ 5,400+
- 178 standard nodes – **Spring upgrade** $\rightarrow$ 270
  - Two 10 core Intel® processors (*20 cores total*)
  - Diskless – Some RAM is used for the Operating System
  - $\approx$ 128GB of useable RAM
- 6 large memory nodes
  - Four 10 core Intel® processors (*40 cores total*)
  - Diskless – Some RAM is used for the Operating System
  - $\approx$ 1TB of useable RAM
- CentOS Linux

# Cray CS300 – Storage

Two storage options are currently available on the Cray CS300

1. Lustre®
2. ValueStorage

# Cray CS300 – Storage → Lustre®

- Lustre® is a high performance parallel filesystem
- The Cray CS300 has ≈ 582TB of storage space
- Shared between all compute nodes and login nodes
- Primarily used as scratch space for jobs (Input and Output)
- User do not have a usage quota (soft or hard)
- Certain directories are subject to a 90 day purge policy
- **Data is not backed up! Users are responsible for their own data**

# Cray CS300 – Storage → ValueStorage

- 500TB of scale out storage
- Currently only available for purchase by cluster users and only accessible via the login nodes
- ValueStorage will eventually be available for purchase by everyone
- ValueStorage owners will eventually be able to mount as a network drive (CIFS) on laptop, workstations, servers
- Purchased in 0.5TB increments

**ValueStorage Pricing**

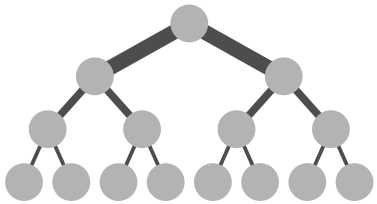
| Product | Annual Cost | Product | Annual Cost |
|---|---|---|---|
| 0.5TB | $65.00 | 0.5TB + Replication | $130.00 |

More Information

*All prices are subject to change*

# Cray CS300 – Network

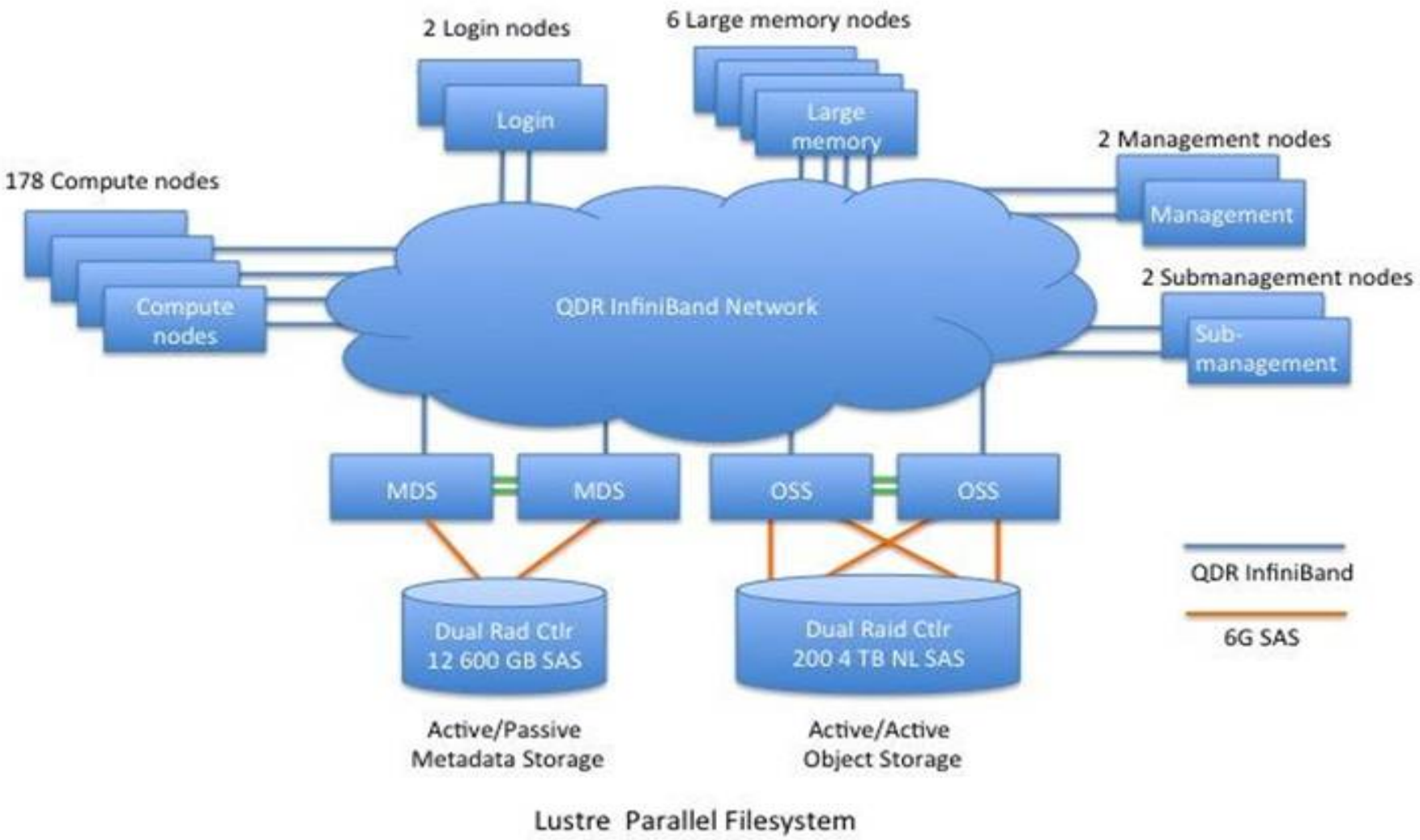
- 40Gb Infiniband inter-connects (QDR)
  - High speed inter-connect between compute nodes, Lustre® storage and Login nodes
  - Utilizes the *fat tree network topology*



Source: https://en.wikipedia.org/wiki/Fat_tree

- 10Gb login node internet connectivity
  - Speed test from UH to CERN clocked transfer speeds up to 2+ Gb/s

# Cray CS300 – Layout



# Outline

# Community Resources

- The initial investment in the cluster provides resources for all Faculty, Staff, and Students affiliate with the University of Hawai'i
- All users can run on the publicly accessible partitions: *community.q, lm.q, sb.q, kill.q, htc.q*
- For some users, the publicly available resources may not be enough . . .

# Condo Model

- The Condo model allows users to buy nodes (*condos*) to incorporate into the cluster
- Node owners are provided with priority access to their purchased hardware
- All nodes have a *5 year warranty*
  - Once a nodes warranty has expired, it will be removed from the cluster
- Condo owners are given early access to purchased nodes
  - Access is granted as soon as the funds land in our accounts
  - The 5 year warranty will not begin until newly ordered nodes are installed
- Condo owners are also given the option to purchase 1TB of Lustre® storage per node purchased

# Service Units

- In some cases, users will not require owning a node, but still need priority access
- An alternative to purchasing a node, is to purchase service units ($SU$)
- SUs come in two variates:
  - Standard node units
    - 20 core hours, with access to 128GB of ram on a standard node
  - Large memory node units
    - 40 core hours, with access to 1TB of ram on a large memory node
- A minimum order totaling $500 is required

# Price Card

## Condo Price Card

| Product | Cost | Product | Cost |
|---|---|---|---|
| Standard | $6,600.00 | Standard + 2 GPUs (Nvidia® K40) | $13,600.00 |
| Large Memory | $33,900.00 | 1 TB Lustre® Storage for 5 years | $600.00 |

## Service Unit Price Card

| Product | Cost | Minimum Order |
|---|---|---|
| Standard node | $0.50 per SU | 1,000 SU ($500.00) |
| Large memory node | $2.00 per SU | 250 SU ($500.00) |

*All prices are subject to change*

# Outline

# Overview – Cluster Interaction

- Connecting to a cluster @ UH
  - Login to the cluster
  - Verify user permissions
- User directories
- Transferring files
  - Globus
- Software
  - Modules
  - Acquiring software
  - Compiliers
- Managing user jobs
  - Job scheduler
  - Using SLURM
  - Partitions
  - Submitting jobs (Examples)

# Connecting to a cluster @ UH

- To connect to the cluster, we utilize a client which communicates using the Secure Shell (*SSH*) protocol
- Linux and MacOSX, typically have a SSH client already installed
- Windows typically does not come with an SSH client installed
- Windows 10 may come pre-installed with a SSH client, but it might not be stable
- Suggested SSH clients for Windows include:
  - SSH Secure Shell (SSH 3.2.9)
  - Putty

- The Cray CS300 has two login nodes:
  - uhhpc1.its.hawaii.edu
  - uhhpc2.its.hawaii.edu

Let's attempt to login!

# Login to the Cray CS300

## Windows

- If SSH 3.2.9 installed (Lab PCs have it installed)
- Open the start menu, and type "SSH" and you should see a program called "SSH Secure File Terminal Client"
- Click "Quick Connect" and enter the following information:

  **Host Name:** uhhpc1.its.hawaii.edu –OR– uhhpc2.its.hawaii.edu
  **User Name:** Your UH User name e.g., user99
  **Port:** 22

- Press "Connect"
- Enter your UH user password when prompted and press the return key

# Login to the Cray CS300

## Mac & Linux

- Open a terminal window
- Enter one of the following:
  - ssh <UH User name>@uhhpc1.its.hawaii.edu
  - ssh <UH User name>@uhhpc2.its.hawaii.edu
  - **Example:** ssh user99@uhhpc1.its.hawaii.edu
- Enter your UH user password when prompted and press the return key

# On Initial Login . . .

Validate that all system permissions are correct for your user

1. Test that you can list files in your home: '`ls -la`'
2. Test making a file in your home: '`touch test.txt`'
3. Go into ~/lus: '`cd ~/lus/`'
4. Test making a file in your lus directory: '`touch test.txt`'
5. Go into ~/apps: '`cd ~/apps/`'
6. Test making a file in your apps directory: '`touch test.txt`'

**Result**

Did you get any errors? Let us know if you did

Notes:

- On login you are placed in /home/<username>/
- By default, ~ is equivalent to /home/<username>/

# Overview – Cluster Interaction

- Connecting to a cluster @ UH
  - Login to the cluster
  - Verify user permissions
- User directories
- Transferring files
  - Globus
- Software
  - Modules
  - Acquiring software
  - Compilers
- Managing user jobs
  - Job scheduler
  - Using SLURM
  - Partitions
  - Submitting jobs (Examples)

# User Directories

## Home

```
[user99@login ~]$ ls -l
total 0
lrwxrwxrwx 1 user99 user99 23 Jan 15 20:38 apps -> /lus/scratch/usr/user99
lrwxrwxrwx 1 user99 user99 19 Jan 15 20:38 lus -> /lus/scratch/user99
lrwxrwxrwx 1 root   root   37 Jan 15 20:41 purge -> /lus/scratch/log/purge/current/user99
```

- ~/ is not on the Lustre® filesystem and **should not be used for job data!**
- ~/lus/ is a symlink to the Lustre® scratch
  - This is where all your job data files should live
  - Items in this directory ***are*** subject to our 90 day purge policy
- ~/apps/ is a symlink to where programs should be stored
  - Items in this directory ***are not*** subject to our 90 day purge policy
  - Directory is monitored for abuse
- ~/purge/ is typically a dead symlink
  - Symlink becomes active if the user has files that are part of the next automatic purge
  - When ~/purge/ is active, the directory containing two files – *purge_list.txt* & *totals.txt*
  - An email is sent to users if they have files that will be purged
  - Email notification is sent out 14 days before the purge takes place

# User Directories

## Filesystems

```
[user99@login ~]$ df -h
Filesystem                                  Size  Used Avail Use% Mounted on
10.10.0.3:/ha_cluster/home                  1.8T  888G  851G  52% /home
10.12.0.51@o2ib:10.12.0.52@o2ib:/scratch    582T  429T  125T  78% /lus/scratch
```

- /home/<username> exists on a NFS mounted filesystem
  - Only has 1.8TB of useable space
  - Using all this space may cause problems for the entire cluster
  - Not a high performance filesystem and small in size
- /lus/scratch/ is the Lustre® filesystem
  - Has 582TB of useable space
  - ~/apps/, ~/lus/, and ~/purge/ all point to directories on this filesystem
  - High performance and a lot more space for users to use
  - No hard or soft quotas are in place
  - Utilization is managed through the 90 day purge policy

# Overview – Cluster Interaction

- Connecting to a cluster @ UH
  - Login to the cluster
  - Verify user permissions
- User directories
- Transferring files
  - Globus
- Software
  - Modules
  - Acquiring software
  - Compilers
- Managing user jobs
  - Job scheduler
  - Using SLURM
  - Partitions
  - Submitting jobs (Examples)

# Available File Transfer Protocols

- The cluster has the following options for transferring files:
  - scp (RCP+SSH protocol)
  - rsync (rsync protocol with SSH transport)
  - SFTP (SSH FTP protocol)
  - Globus (Grid FTP protocol)
- All options are widely used, and have clients that can be found for on most major operating systems

SFTP, scp, and rsync are fairly common on Linux systems,
but Globus is not as common . . .

# Globus

## What is Globus?

The Globus transfer service provides high-performance, secure, file transfer and synchronization between endpoints.

Globus handles all the difficult aspects of data transfer, allowing application users to easily start and manage transfers between endpoints, while automatically tuning parameters to maximize bandwidth usage, managing security configurations, providing automatic fault recovery, and notifying users of completion and problems.

## Definition

An ***endpoint*** is one of the two file transfer locations – either the source or the destination – between which files can move. Once a resource (such as a server, cluster, storage system, laptop, or other system) is defined as an endpoint, it will be available to authorized users who can transfer files to or from this endpoint.

---

https://www.globus.org/file-transfer

# Globus

If you already use Globus, the endpoints for the Cray CS300 are:

- hawaii#UHHPC1
- hawaii#UHHPC1

For those that have not used Globus, and wish to try it out, you can learn how to use Globus by visiting the Cyberinfrastructure website for more information:
http://www.hawaii.edu/its/ci/hpc-resources/hpc-tutorials/globus-quick-start-guide/

# Overview – Cluster Interaction

- Connecting to a cluster @ UH
  - Login to the cluster
  - Verify user permissions
- User directories
- Transferring files
  - Globus
- Software
  - Modules
  - Acquiring software
  - Compilers
- Managing user jobs
  - Job scheduler
  - Using SLURM
  - Partitions
  - Submitting jobs (Examples)

## Modules

A tool to help users manage their Unix or Linux shell environment, by allowing groups of related environment-variable settings to be made or removed dynamically.[1]

- We globally install frequently requested software packages and create modules for all users to access
- Access to modules is via the **module** command
  - 'module avail' – list installed modules
  - 'module load <module name>' – Loads the named module
  - 'module unload <module name>' – Remove named module
  - 'module purge' – Unload all loaded modules
- Installing software in your ~/apps directory is suggested to prevent us from being a bottleneck

[1] https://en.wikipedia.org/wiki/Environment_Modules_(software)

# Acquiring Software – Binaries and/or Source

- You can transfer software source, binaries or scripts into your ~/apps directory on the Cray CS300
  - Binaries compiled as x86_64 (64-bit) for CentOS 6.5 or RHEL6.5 should work
- You may also download tar or zipped software/source code directly from the login nodes using tools like **wget** & **curl**
- You may also clone source repositories using the correct software revision tool: **git**, **svn**, **hg**, **cvs**, etc.

# Compilers

- We have the Intel®, GNU (gcc, g++), Cray® & PGI® compilers
- Compiling must take place on a compute node
  - Interactive sessions are useful for compiling software
  - Sandbox nodes mirror the environment the compute nodes provide and are ideal for compilation
  - Login nodes **do not** load all the software and libraries found on the compute nodes
- Intel® compilers are recommended for best performance
  - Intel® 2013 compilers:
    - module load intel/ics – Loads Intel® compilers: **icc**, **ifort**, **icpc**
    - module load intel/impi – Loads Intel® MPI wrapper: **mpiicc**, **mpiifort**, **mpiicpc**
  - Intel® 2016 compilers:
    - We have 2 floating seats for Intel® 2016 compiler
    - intel_2016/ics
    - intel_2016/impi

# Overview – Cluster Interaction

- Connecting to a cluster @ UH
  - Login to the cluster
  - Verify user permissions
- User directories
- Transferring files
  - Globus
- Software
  - Modules
  - Acquiring software
  - Compilers
- Managing user jobs
  - Job scheduler
  - Using SLURM
  - Partitions
  - Submitting jobs (Examples)

# Managing User Jobs

User jobs all come in different shapes and sizes:

- Require multiple nodes working in concert towards a common goal (MPI)
- Require a single node, in which they use multiple threads work together (OpenMP, pthreads)
- Require a lot of cores to process a lot of data in an identical manner, yet none of the inputs have dependencies on another (HTC)

The Cray CS300 is capable of handling many different types of jobs, but with so many users in a multi-user environment, how do we turn this chaos into organized chaos?

This looks like a job for a ***job scheduler***!

# What is a Job Scheduler?

## Definition

A job scheduler is a computer application for controlling unattended background program execution (commonly called batch processing)

Basic features expected of a job scheduler include:

- Interfaces which help to define workflows and/or job dependencies
- Automatic submission of executions
- Interfaces to monitor the executions
- Priorities and/or queues to control the execution order of unrelated jobs

For the Cray CS300, we utilize the **S**imple **L**inux **U**tility **R**esource **M**anager or simply known as the *SLURM scheduler*

---

https://en.wikipedia.org/wiki/Job_scheduler

https://en.wikipedia.org/wiki/Slurm_Workload_Manager

http://slurm.schedmd.com/slurm.html

# SLURM

## How are jobs scheduled?

User submitted jobs are assigned a priority using a fairshare algorithm. Factors such as: runtime, resource usage request, age of job, and amount of compute time a user has used in recent history are all factored in to assigning a priority to a give job

# SLURM commands

SLURM has a series of commands, each of which allow users to interact with the job scheduler

- ***srun*** – Used to submit a job for execution or initiate job steps in real time
- ***srun.x11*** – Used to submit an interactive job with x11 support
- ***sbatch*** – Used to submit a job script for later execution. The script could contain one or more srun commands
- ***scancel*** – Used to cancel a pending or running job or job step. It can also be used to send an arbitrary signal to all processes associated with a running job or job step
- ***sinfo*** – Reports the state of partitions and nodes managed by Slurm. It has a wide variety of filtering, sorting, and formatting options
- ***squeue*** – Reports the state of jobs or job steps
- ***sacct*** – Used to report job or job step accounting information about active or completed jobs
- ***scontrol*** – The administrative tool used to view and/or modify Slurm state. Note that many scontrol commands can only be executed as user root

– Examples usage of the SLURM commands can be seen on schedmd's quickstart

– Each command should have a 'man' page, or displays help when the -h flag is used

---

http://slurm.schedmd.com/quickstart.html

# Partitions

## What is a partition?

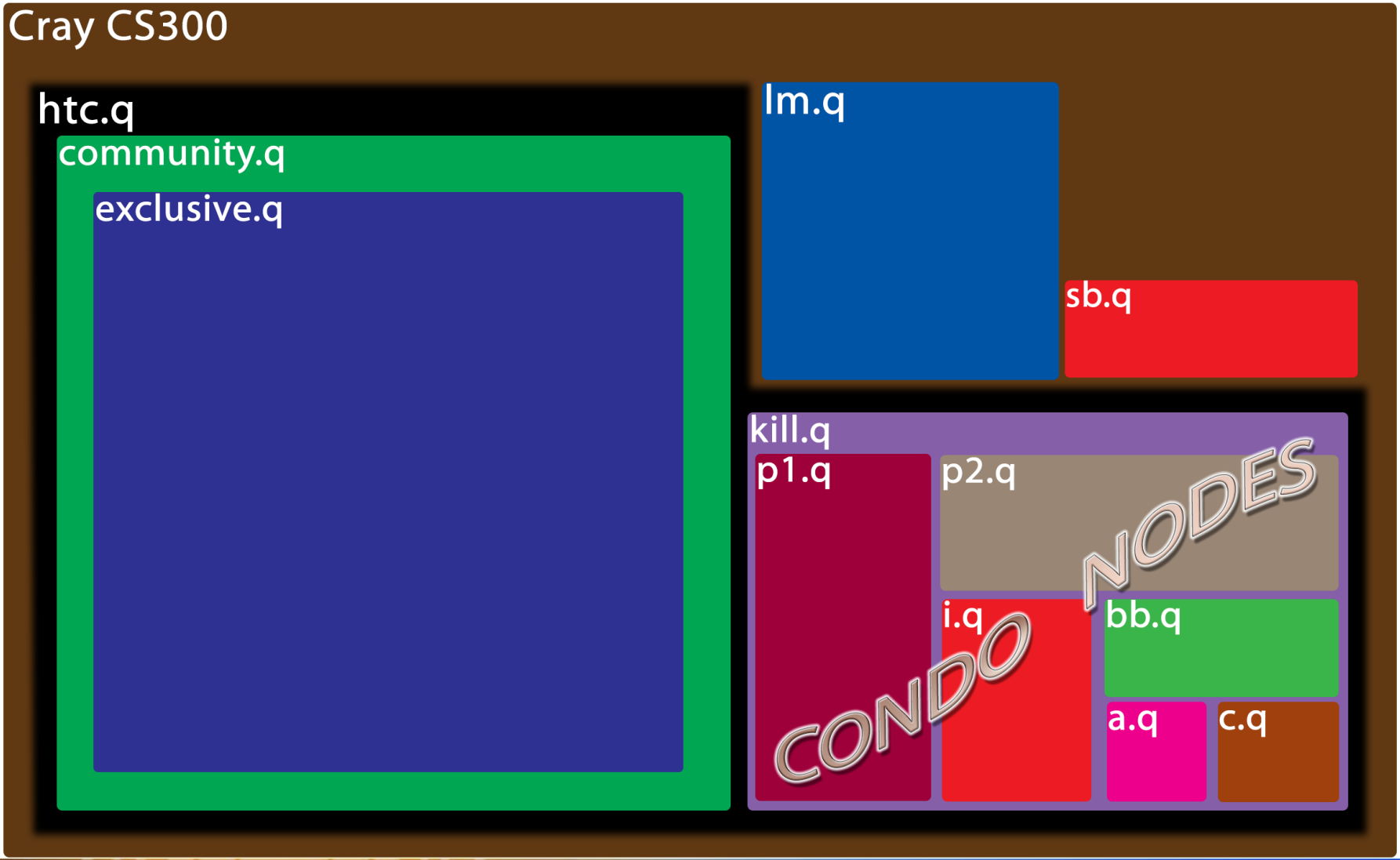
A partition can be thought of as a group of nodes/resources divided into possibly overlapping sets. Each partition can be considered as a job queue, each of which has an assortment of constraints such as job size limit, job time limit, users permitted to use it, etc. Priority-ordered jobs are allocated nodes within a partition until the resources (nodes, processors, memory, etc.) within that partition are exhausted.[2]

- The Cray CS300 currently has six public partitions:
  **lm.q**, **community.q**, **exclusive.q**, **sb.q**, **kill.q**, **htc.q**
- Jobs submitted to kill.q and htc.q can be preempted by jobs in other partitions
- More details on partition constraints can be viewed by visiting the Cyberinfrastructure website or by using the following command

  `[login ~]$ scontrol show partition <partition name>`
  Example: `[login ~]$ scontrol show partition community.q`

[2] http://slurm.schedmd.com/quickstart.html

# Partitions


Cray CS300
htc.q
community.q
exclusive.q
lm.q
sb.q
kill.q
p1.q
p2.q
i.q
bb.q
a.q
c.q
CONDO
NODES

# Interactive Job with SLURM

## Interactive session (no X11)

```
[login ~]$ srun --immediate --partition sb.q --nodes 1 --cpus-per-task 5 --tasks-per-node 1 --time 0-01:00 --pty /bin/bash
```

## Interactive session (with X11)

1. Connect via SSH using the -Y option, X11 forwarding enabled
2. run srun.x11 to start a session on a node

```
[local ~]$ ssh -Y user99@uhhpc1.its.hawaii.edu
[login ~]$ srun.x11 --immediate --partition sb.q --nodes 1 --cpus-per-task 5 --tasks-per-node 1 --time 0-01:00
[compute-0001 ~]$ xterm
```

# SLURM sbatch – Submission Script File (MPI Job)

```
[login lus]$ cat mpi.slurm

#!/bin/sh
#SBATCH --job-name=MPI_example
#SBATCH --partition=exclusive.q
## 3 day max run time for community.q, kill.q, exclusive.q, and htc.q.  1 Hour max run time for sb.q
#SBATCH --time=3-00:00:00
## task-per-node x cpus-per-task should not typically exceed core count on an individual node
#SBATCH --nodes=4
#SBATCH --tasks-per-node=20
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=6400 ## max of 6400 for standard nodes, max of 26214 for large memory nodes
#SBATCH --error=hello-%A_%a.err ## %A - filled with jobid. %a - filled with job arrayid
#SBATCH --output=hello-%A_%a.out ## %A - filled with jobid. %a - filled with job arrayid
## Useful for remote notification
#SBATCH --mail-type=BEGIN,END,FAIL,REQUEUE,TIME_LIMIT_80
#SBATCH --mail-user=user@test.org

## All options and environment variables found on schedMD site: http://slurm.schedmd.com/sbatch.html
## Intel MPI manual: https://software.intel.com/en-us/mpi-refman-lin-html
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}
export I_MPI_FABRICS=tmi
export I_MPI_PMI_LIBRARY=/opt/local/slurm/default/lib64/libpmi.so

cd $SLURM_SUBMIT_DIR
srun  -n ${SLURM_NTASKS}  ./hello_mpi.intel
```

# SLURM sbatch – Submission Script File (Non-MPI Job)

```
[login lus]$ cat hello_world.slurm

#!/bin/sh
#SBATCH --job-name=example
#SBATCH --partition=community.q
## 3 day max run time for community.q, kill.q, exclusive.q, and htc.q.  1 Hour max run time for sb.q
#SBATCH --time=3-00:00:00
## task-per-node x cpus-per-task should not typically exceed core count on an individual node
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=5
#SBATCH --mem-per-cpu=6400 ## max of 6400 for standard nodes, max of 26214 for large memory nodes
#SBATCH --error=hello-%A_%a.err ## %A - filled with jobid. %a - filled with job arrayid
#SBATCH --output=hello-%A_%a.out ## %A - filled with jobid. %a - filled with job arrayid
## Useful for remote notification
#SBATCH --mail-type=BEGIN,END,FAIL,REQUEUE,TIME_LIMIT_80
#SBATCH --mail-user=user@test.org

## All options and environment variables found on schedMD site: http://slurm.schedmd.com/sbatch.html

export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}

cd $SLURM_SUBMIT_DIR
./hello_world
```

# SLURM sbatch – Submission Script File (Job Array)

```
[login lus]$ cat job_array.slurm

#!/bin/sh
#SBATCH --job-name=example
#SBATCH --partition=community.q
## 3 day max run time for community.q, kill.q, exclusive.q, and htc.q.  1 Hour max run time for sb.q
#SBATCH --time=3-00:00:00
## task-per-node x cpus-per-task should not typically exceed core count on an individual node
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --cpus-per-task=5
#SBATCH --mem-per-cpu=6400 ## max of 6400 for standard nodes, max of 26214 for large memory nodes
#SBATCH --error=buzz-%A_%a.err ## %A - filled with jobid. %a - filled with job arrayid
#SBATCH --output=buzz-%A_%a.out ## %A - filled with jobid. %a - filled with job arrayid
## Useful for remote notification
#SBATCH --mail-type=BEGIN,END,FAIL,REQUEUE,TIME_LIMIT_80
#SBATCH --mail-user=user@test.org

## All options and environment variables found on schedMD site: http://slurm.schedmd.com/sbatch.html

export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}

cd $SLURM_SUBMIT_DIR
./worker_bee -i input_${SLURM_ARRAY_TASK_ID}.flower -o output_${SLURM_ARRAY_TASK_ID}.honey
```

# Outline

# Cluster Etiquette

# Outline



# Overview

- Login node usage policy
- Scratch filesystem purge policy

# Login Node Usage Policy

***Login node usage policy is subject to change***
***Users will be notified via email prior to changes taking effect***

# Lustre® Filesystem Purge Policies

Due to users not having any quotas on the Lustre® filesystem, we need some policy in place which removes older files from the system. To accomplish this, we have currently implemented a purge policy, for any file that is older than 90 days.

- Which of my directories are subject to the purge policy?
  - **Answer:** All files and folders found in ~/lus/ are subject to the 90 day purge policy. Symlinks are excluded from the purge, and are not followed by the purge bot.
- How frequently are files for purging identified?
  - **Answer:** Current frequency is every other month, but the frequency many increase or decrease based on the fill rate of the Lustre® filesystem

***Purge policy is subject to change***
***Users will be notified via email prior to changes taking effect***

# Outline

# FAQ

- Are files on the cluster backed up?
  - **Answer: NO!** User files on the cluster ***are not backed up***. It is up to you as the user to validate and maintain your own backups. The Cyberinfrastructure team takes no responsibility for any data that is lost due to human or mechanical error.

# Questions?