Introduction to

High-Performance Computing (HPC)

HPC Cluster

- multi-user, shared resource
- lots of nodes = lots of processing capacity + lots of memory
- a system like this requires constant maintenance and upkeep, and there is an associated cost

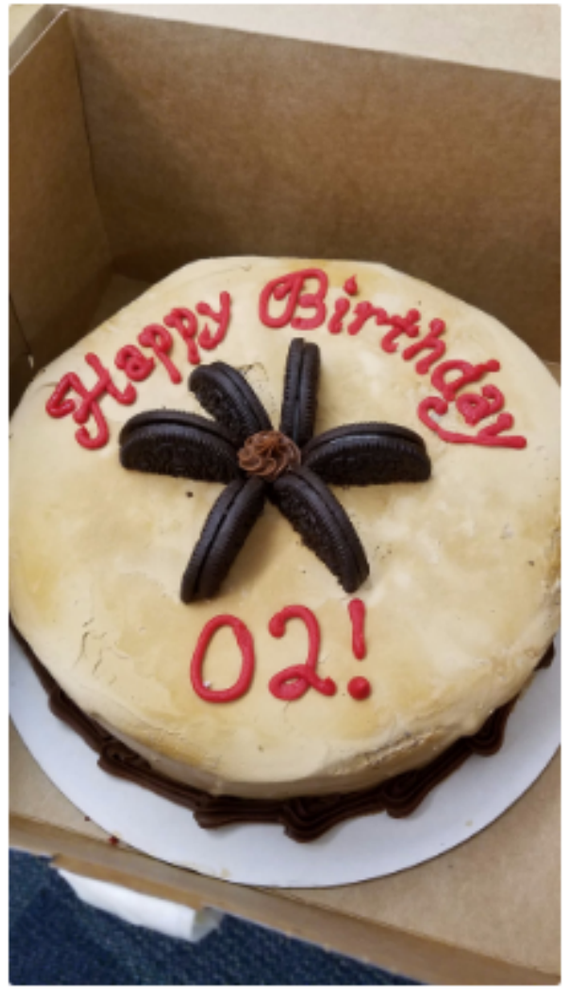
Wiki page:

https://wiki.rc.hms.harvard.edu/display/O2

Orchestra
HIGH PERFORMANCE COMPUTING CLUSTER

Tweets by @hms_rc

HMSResearchComputing @hms_rc
It's cake time for RC -- happy birthday to O2 !!



Sep 12, 2017

HMSResearchComputing @hms_rc
O2 is officially being launched today as the new RC production HPC cluster!! (thnx beta testers!) Get started at: hmsrc.me/O2docs

Sep 12, 2017

Introduction to High Performance Computing and O2 for New Users

HMS Research Computing

(Slides courtesy of Kris Holton & Kathleen Keating at HMS-RC)

Welcome to O2!

- HMS Research Computing's newest High-Performance Compute cluster to enhance the compute capacity available to HMS Researchers
- Heterogeneous environment of newer, faster cores with high memory allocation to facilitate multi-core and parallelized workflows
- SLURM scheduler to efficiently dispatch jobs

O2 Tech Specs

- 12200+ cores
- 32, 28, or 20 cores per node
- 256-160GB RAM (memory) per node (8-9GB/core)
- 9 756GB RAM and 1 1TB highmem nodes
- 133 GPU cards
  - (71 GPUs available to Quad researchers only)
- CentOS 7 Linux
- SLURM job scheduler

2-Factor Authentication

- For logins using WiFi networks other than HMS Private/ Harvard Secure
- Easiest: download Duo app to phone
- Similar to the setup for Harvard Key logins
- Setup details at: https://wiki.rc.hms.harvard.edu:8443/display/O2/Two+Factor+Authentication+on+O2

Using O2!

1. Logging in to remote machines (securely)

- When logging in we used the “ssh” command,
  ssh stands for Secure SHell

- **ssh** is a protocol for data transfer that is secure, i.e the data is encrypted as it travels between your computer and the cluster (remote computer)

- Commonly used commands that use the **ssh** protocol for data transfer are, **scp** and **sftp**

Logging Into O2

- Open a terminal

  `ssh yourEcommons@o2.hms.harvard.edu`

- If outside of “approved” internet sources (HMS Private/Harvard Secure):

  Type 1/2/3 for DUO push/sms/phone

Welcome to O2!

Where are you in O2?

```
mfk8@login01:~$
```

You are logged into a “**shell login server**”, **login01-05**.These are not meant for heavy lifting!

```
mfk8@login01:~$ pwd
```

You are in your home directory.

Interactive Sessions

- The login servers are not designed to handle intensive processes, and CPU usage is throttled.
- Start by entering your first job! This will (usually) log you into a "**compute** node!"

```
mfk8@login01:~$ srun --pty –p interactive –t 0-12:00
                --mem 8G bash
```

"`srun --pty`" is how interactives are started

"`-p interactive`" is the partition

"`-t 0-12:00`" is the time limit (12 hours)

"`--mem 8G`" is the memory requested

```
mfk8@compute-a:~$
```

2. Using & installing software

LMOD: Software Modules

- Most “software” on O2 is installed as an environment module.
- LMOD system adds directory paths of software into $PATH variable, to make sure the program runs without any issues.
- Allows for clean, easy loading, including most dependencies, and switching versions.

LMOD: Software Modules

Most software is compiled against something called "gcc-6.2.0" — so, we need to load that before loading other programs that depend on it.

```
$ module load gcc/6.2.0
$ module avail  #to see software now available to load
$ module spider  #verbose list of all software available
```

Loading/Unloading Modules

Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```

Load the module

```
$ module load bowtie2/2.2.9
$ bowtie2
```

Which module version is loaded (if at all)?

```
$ which bowtie2
$ module list
$ echo $PATH
```

Loading/Unloading Modules

Need help with the module?

```
$ module help bowtie2/2.2.9
```

Unloading modules

```
$ module unload bowtie2/2.2.9
```

Dump all modules

```
$ module purge
```

3. The Job Scheduler, SLURM

Simple Linux Utility for Resource Management (SLURM)

- Fairly allocates access to resources (computer nodes) to users for some duration of time so they can perform work

- Provides a framework for starting, executing, and monitoring batch jobs

- Manages a queue of pending jobs; ensures that no single user or core monopolizes the cluster

Choosing the proper resources for your job with the appropriate `SBATCH` options

Submitting Jobs

In an “interactive session”, programs can be run directly, however your computer will have to remain connected to the cluster for the duration of this run.

```
mfk8@compute-a:~$ bowtie2 -c 4 hg19 file1_1.fq
```

What if you wanted to run the program, close your computer and come back later to check on it?

A script with the required commands can be submitted to O2 (SLURM) using the `sbatch` command.

```
mfk8@compute-a:~$ sbatch mybowtiejob.sh
```

Creating a job submission script

```
#! /bin/sh

#SBATCH -p short
#SBATCH –t 0-03:00
#SBATCH –c 4
#SBATCH --mem=8G
#SBATCH –o %j.out
#SBATCH –e %j.err
#SBATCH -J bowtie2_run1
#SBATCH --mail-type=ALL
#SBATCH ––mail-user=mfk8@med.harvard.edu

module load gcc/6.2.0
module load bowtie2/2.2.9

bowtie -c 4 hg19 file1_1.fq
```

Save script as myJobScript.run and run it as follows:

```
$ sbatch myJobScript.run
```

***O2 will notify you when the job is done, or if there is an error*

Partitions -p

Partition	Priority	Max Runtime	Max Cores	Limits
short	12	12 hours	20	
medium	6	5 days	20	
long	4	30 days	20	
interactive	14	12 hours	20	2 job limit
priority	14	30 days	20	2 job limit
mpi	12	5 days	640	20 core min
highmem	12	5 days	20	750G
gpu, gpu_quad, gpu_requeue		160 GPU hours	34 (total)	420G (total)
transfer		5 days	4	

Runtime: -t

- -t days-hours:minutes
- -t hours:minutes:seconds
- Need to specify how long you estimate your job will run for
- Aim for 125%
- Subject to maximum per partition
- Excessive runlimits (like partition max) take longer to dispatch, and affect fairshare

Cores: -c

- -c X to designate cores: max 20 per job
- -N X to constrain all cores to X nodes
- CPU time: wall time (-t) * (-c) cores used
- Unable to use cores not requested (no overefficient jobs): cgroups constraint
- Adding more cores does not mean jobs will scale linearly with time, and causes longer pend times

Memory: --mem

- Only 1G is allocated by default
- `--mem XG` #total memory over all cores
- `--mem-per-cpu XG` #total memory per CPU requested, use for MPI
- No unit request (G) defaults to Megabytes
  - 8G ~= 8000

Job Priority

- Dynamically assigned
- Factors contributing: Age, Fairshare, Partition, QOS, Nice
- Fairshare: 0-1 scale
- Check your fairshare:
  - `$ sshare -Uu $USER`
- Check job priority values for your pending jobs:
  - `$ sprio -u $USER`

Managing jobs and getting information about submitted/running jobs

Job Monitoring: Current jobs

- `$ O2squeue`
  - JOBID, PARTITION, STATE, TIME_LIMIT, TIME, NODELIST(REASON), ELIGIBLE_TIME, START_TIME, TRES
  - O2squeue documentation

- *Other options*:

```
$ squeue -u eCommons –t RUNNING/PENDING
$ squeue -u eCommons –p Partition
$ squeue -u eCommons –start
```

- *Detailed job info:*

```
$ scontrol show jobid <jobid>
```

Job Information: Past Jobs

- `$ O2sacct`
  - JobID, Partition, State , NodeList, Start, Timelimit, Elapsed, CPUTime , TotalCPU, AllocTRES, MaxRSS
  - Can specify job ID, job status, and/or timeframe to report accounting info
  - O2sacct documentation

- *Other options*:

```
$ sacct –j jobid
$ sacct –r partition
$ squeue –s state
$ sacct --helpformat #get available fields you can specify
```

Cancelling/Pausing Jobs

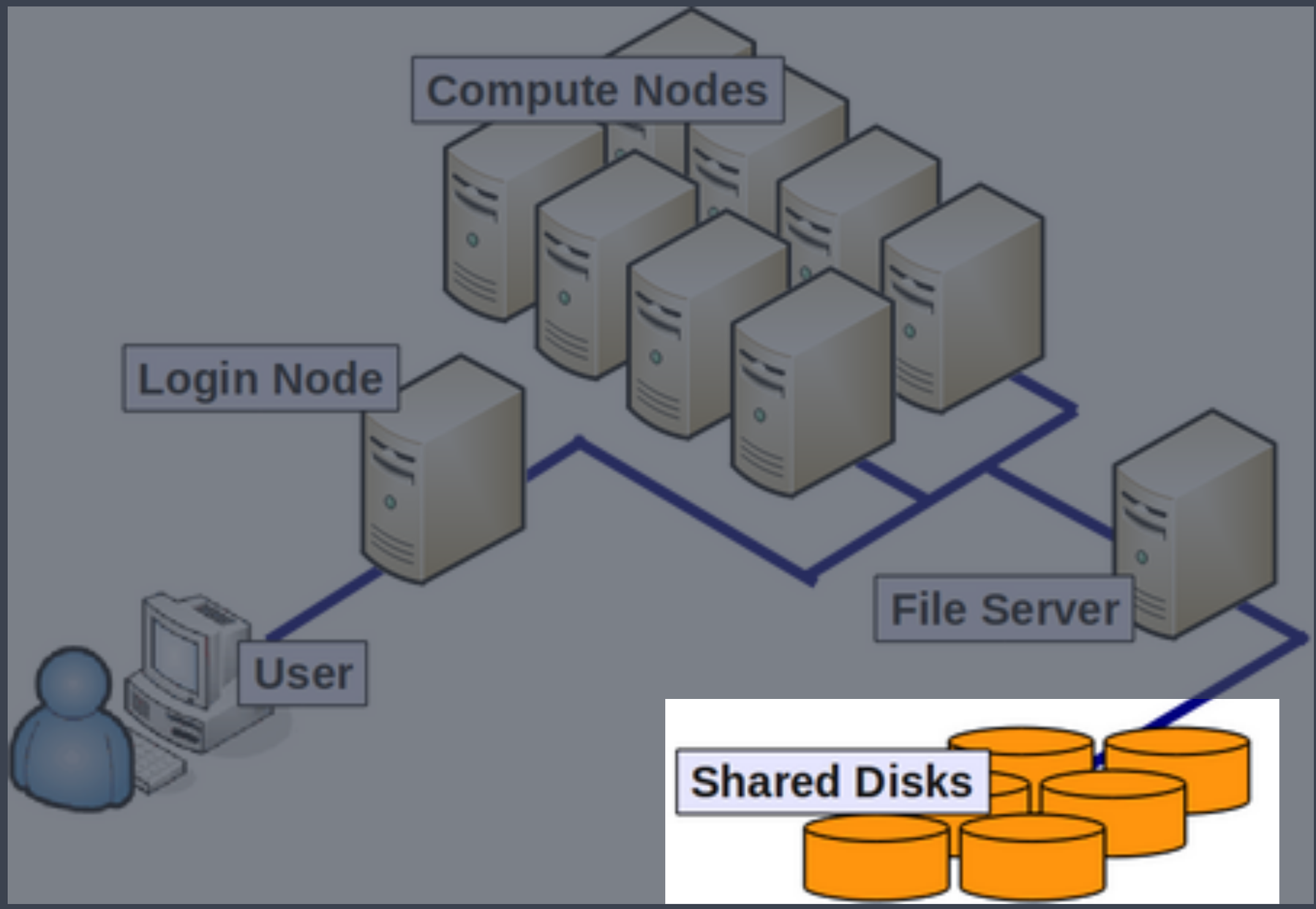
- `$ scancel <jobid>` #Cancels specific job
- `$ scancel –t PENDING` #Cancels pending job
- `$ scancel --name JOBNAME` #Cancels job by name
- `$ scancel jobid_[indices]` #array indices
- `$ scontrol hold <jobid>` #pause pending jobs
- `$ scontrol release <jobid>` #resume

Exercise!

https://tinyurl.com/sbatch-exercise

4. Filesystems and storage

Filesystems and storage



Filesystems and storage

- Storage on HPC systems is organized differently than on your personal machine

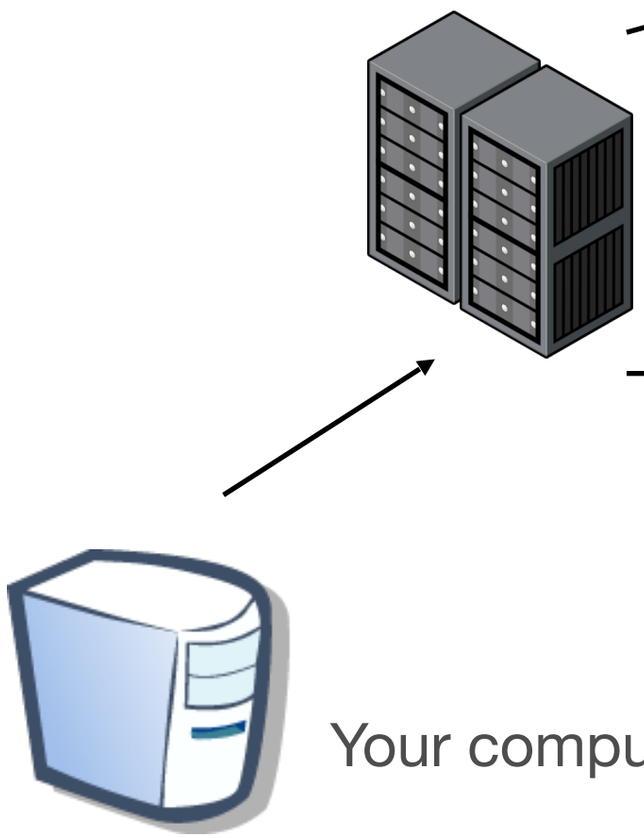
- Physical disks are bundled together into a virtual volume; this volume may represent a single filesystem, or may be divided up, or partitioned, into multiple filesystems

- Filesystems are accessed over the internal network

O2 Primary Storage

O2 Cluster

- 11000+ cores
- SLURM batch system



Your computer

/home

- /home/user_id
- quota: 100GB per user
- Backup: extra copy & snapshots:
- daily to 14 days, weekly up to 60 days

/n/data1, /n/data2, /n/groups

- /n/data1/institution/dept/lab/your_dir
- quota: expandable
- Backup: extra copy & snapshots:
- daily to 14 days, weekly up to 60 days

Temporary “Scratch” storage

- **`/n/scratch3/users/<first_eCommons_char>/<eCommons>`**

  `e.g. /n/scratch3/users/m/mfk8`
- For data only needed temporarily during analyses
- Each user can use up to 10 TB and 1 million files/directories
- Files not accessed for 30 days are automatically purged
- No backups!
- To create your scratch3 directory run from a login node:

  `$ /n/cluster/bin/scratch3_create.sh`
- Scratch3 documentation

Important Note about O2 Storage

- O2 can only be used to store data of Harvard Security Level 3 and below.

- None of the standard filesystems are automatically encrypted, and **cannot** be used for HIPAA-protected or other secure data (Harvard's data security above level 3) unless those data have been de-identified.

HMS Storage Offerings

- ***Active*** – available now, formerly Tier 1
  - Active Compute: O2 group folders, /n/data1, /n/data2, /n/groups
    - e.g., /n/data1/institution/dept/lab
  - Active Collaboration: research.files, /n/files on transfer cluster
  - Research data that is frequently accessed, modified, or computed against.
- ***Standby*** – available now, new and improved alternative to Tier 2
  - Infrequently accessed data, that is directly available for reference, retrieval, or analysis.
  - Accessible as /n/standby/institution/dept/lab on transfer cluster
- ***Cold*** – future offering, formerly Tier 3
  - Rarely accessed data requiring long-term retention, for regulatory or historical purposes

HMS Storage Offerings

- For more detail on all the Storage Offerings, please see the Research Computing Storage Services Website.

- Please see the dedicated Standby Storage page for more information on how to access and utilize Standby.

- New/additional Storage can be requested through the Storage Request Forms through the STAT Service Portal.

For more direction

Email: rchelp@hms.harvard.edu

Website: http://rc.hms.harvard.edu

Wiki: https://wiki.rc.hms.harvard.edu/display/O2/O2

Phone: 617-432-2000 (HMS IT Service Desk, 8a-5p)

Twitter: @hms_rc

Location: Gordon Hall 500, 5th Floor, 25 Shattuck Street

- https://rc.hms.harvard.edu/office-hours/ for Zoom web conferencing during remote work

Office hours: Wednesdays 1-3p for pressing needs, but appointments encouraged.