Introduction to Minerva

Minerva Scientific Computing Environment

https://labs.icahn.mssm.edu/minervalab

Patricia Kovatch
Eugene Fluder, PhD
Hyung Min Cho, PhD
Lili Gai, PhD
Dansha Jiang, PhD

March 11, 2020

Mount Sinai

Outlines

- Minerva compute and storage resources
- Minerva account and logging in
- User software environment
- File transfer, web server and data archive
- Preview on basic LSF commands

Minerva cluster @ Mount Sinai

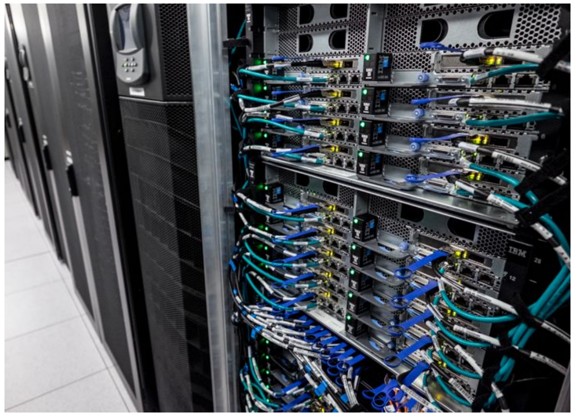
Chimera Computes:

- 4x login nodes - Intel Skylake 8168 24C, **2.7GHz** - **384 GB** memory
- 274 compute nodes - Intel 8168 24C, **2.7GHz**
  - 13,152 cores (48 per node (2 sockets/node) - **192 GB**/node)
- 4x high memory nodes - Intel 8168 24C, 2.7GHz - **1.5 TB** memory
- 48 V100 GPUs in 12 nodes -

  Intel 6142 16C, 2.6GHz - 384 GB memory - 4x V100-16 GB GPU

BODE2 Computes:

- $2M S10 BODE2 awarded by NIH (Kovatch PI)
- 78 compute nodes - Intel 8268, **2.9 GHz**
  - 3,744 cores (48 cores per node - **192 GB**/node)
  - **Open to all NIH funded projects**



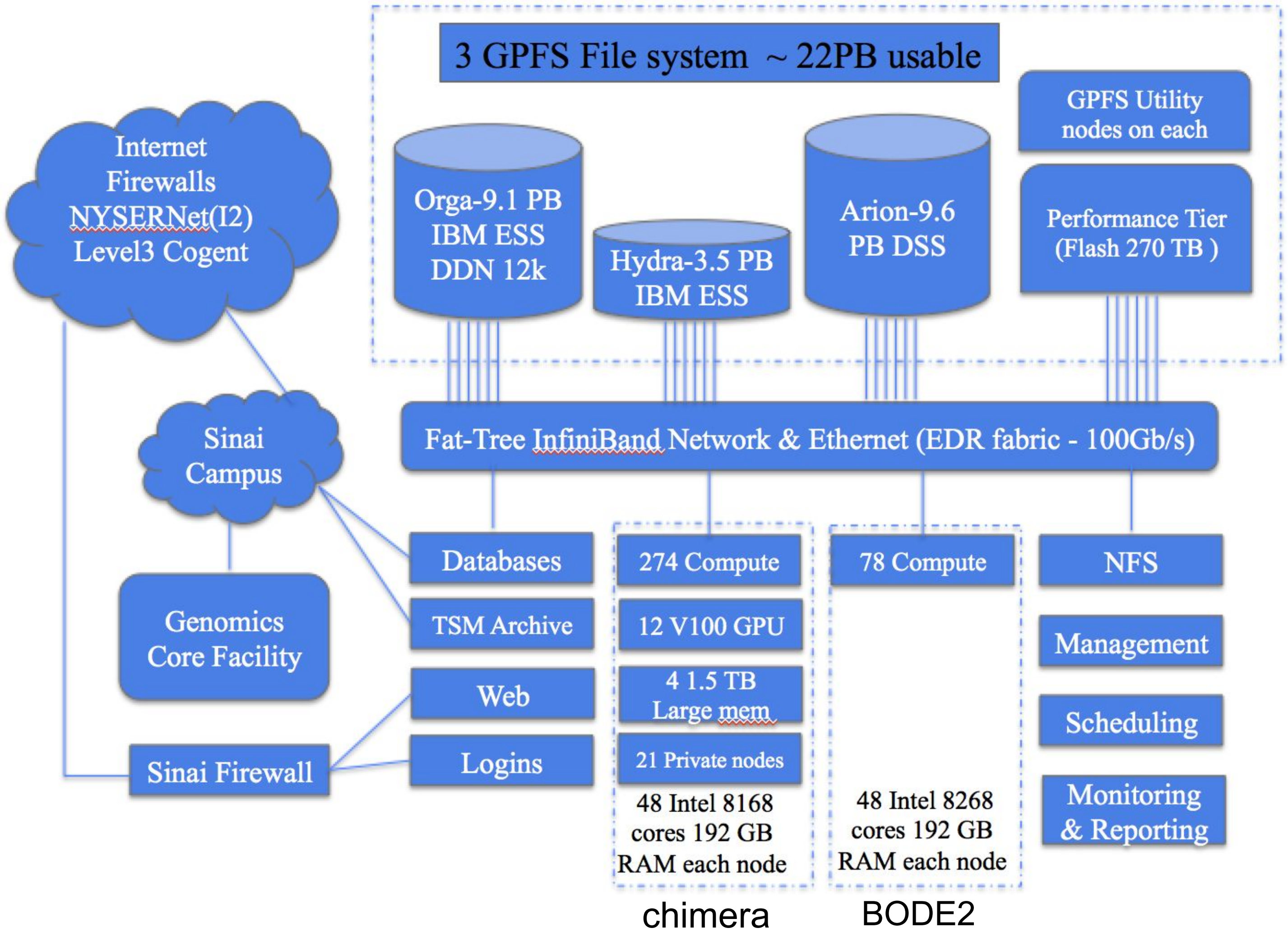
Storage:

Total number of storage space: ~22PB usable in total

- **/sc/hydra** : new file system on Minerva as primary storage
  - Have the same folders as /sc/orga (work, projects, scratch).
  - Use the system path environment variable in scripts
- /sc/orga is still mounted on Minerva
  - Will be merged to hydra.
- **/sc/arion** will be available soon **(10 petabytes of usable storage from BODE2)**

Minerva

- **Minerva nodes and infrastructure fully refreshed 2019**



3 GPFS File system ~ 22PB usable
GPFS Utility nodes on each
Internet Firewalls NYSERNet(I2) Level3 Cogent
Orga-9.1 PB IBM ESS DDN 12k
Hydra-3.5 PB IBM ESS
Arion-9.6 PB DSS
Performance Tier (Flash 270 TB)
Sinai Campus
Fat-Tree InfiniBand Network & Ethernet (EDR fabric - 100Gb/s)
Genomics Core Facility
Databases
TSM Archive
Web
Logins
Sinai Firewall
274 Compute
12 V100 GPU
4 1.5 TB Large mem
21 Private nodes
48 Intel 8168 cores 192 GB RAM each node
78 Compute
48 Intel 8268 cores 192 GB RAM each node
NFS
Management
Scheduling
Monitoring & Reporting
chimera
BODE2

Logging in - General

Minerva is a Linux machine with Centos 7.6

- Linux is command line based, not GUI
- Linux was developed using TTY devices. Commands are short and many times cryptic, but there is usually a good reason

Logging in require an SSH client installed on your machine, a username, a memorized password, and a one-time use code obtained from a physical/software token

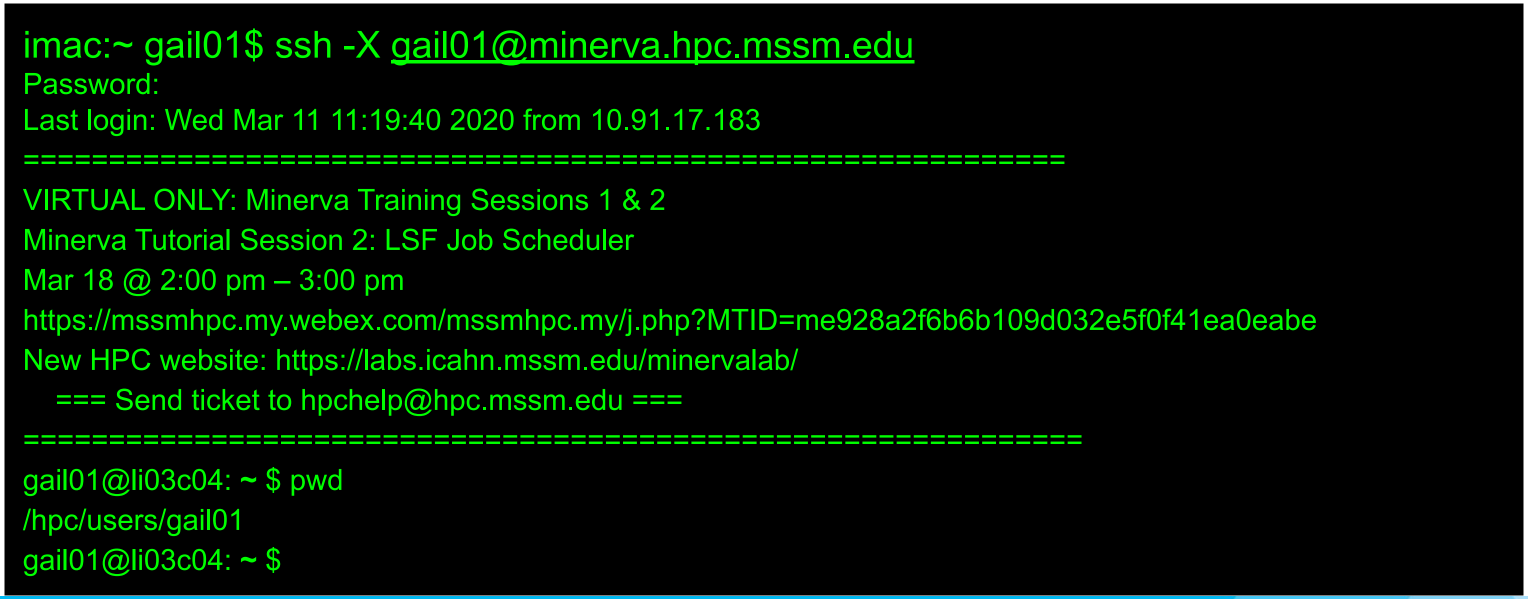
- SSH client: terminal (Mac), Putty or MobaXterm (Windows)
- Apply for an account at https://acctreq.hpc.mssm.edu/
- Register your token at the Self Service Portal (https://register4vip.mssm.edu/vipssp/)
- Logging info at https://labs.icahn.mssm.edu/minervalab/logging-in/

Logging in - Linux / Mac

Connect to Minerva via ssh:

- ssh *your_userID*@minerva.hpc.mssm.edu
- To display graphics remotely on your screen, pass the “-X” or “-Y” flag:
  - ssh -X *your_userID*@minerva.hpc.mssm.edu
  - Mac: Install XQuartz on your mac first

- Open a terminal window on your workstation(Linux/Mac)
  - Landed on one of the login nodes, and at your home directory
    - Never run jobs on login nodes
    - For file management, coding, compilation, etc., purposes only



```
imac:~ gail01$ ssh -X gail01@minerva.hpc.mssm.edu
Password:
Last login: Wed Mar 11 11:19:40 2020 from 10.91.17.183
=====================================================================
VIRTUAL ONLY: Minerva Training Sessions 1 & 2
Minerva Tutorial Session 2: LSF Job Scheduler
Mar 18 @ 2:00 pm – 3:00 pm
https://mssmhpc.my.webex.com/mssmhpc.my/j.php?MTID=me928a2f6b6b109d032e5f0f41ea0eabe
New HPC website: https://labs.icahn.mssm.edu/minervalab/
  === Send ticket to hpchelp@hpc.mssm.edu ===
=====================================================================
gail01@li03c04: ~ $ pwd
/hpc/users/gail01
gail01@li03c04: ~ $
```

Logging in - Windows

- **Install MobaXterm from** https://mobaxterm.mobatek.net/
    - Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more

OR

- **Install PuTTY from www.putty.org**
    - Google it. It will be the first hit.
    - https://www.youtube.com/watch?v=ma6Ln30iP08

- **If you are going to be using GUI's**
    - In Putty: Connection > SSH > X11
        - Ensure "Enable X11 forwarding" is selected
    - On Windows box install Xming
        - Google; Download; Follow bouncing ball
    - Test by logging into Minerva and run the command: xclock
        - Should see a clock

Minerva Login summary

4 new login nodes: **minerva[11-14]**, which points to the login node **li03c[01-04]**

- **minerva[11-12] (or li03c[01-02]) are external login nodes**
  - Public login nodes when you are off-campus
- **minerva[13-14] (or li03c[03-04]) are internal login nodes**
  - only available within campus-network

Users	Login method	Login servers	Password Components
Sinai users	user1	**@minerva.hpc.mssm.edu** @minerva11.hpc.mssm.edu @minerva12.hpc.mssm.edu	Sinai Password + 6 Digit Symantec VIP token code
External users	user1+**yldap**		HPC Password + YubiKey Button Push (Will migrate to school VIP tokens)

Note: Load balancer **Round-robin** is configured for **minerva.hpc.mssm.edu.** It will distribute client connections to the nearest across a group of login nodes.

Minerva Storage

- Storage is in folders and subfolders. In linux, subfolders are separated by "/"
- 4-ish folders you can have (Possibly multiple project folders)

Home	/hpc/users/<userid>	• 20GB quota. • Slow. Use for "config" files, executables…NOT DATA • NOT purged and is backed up
Work	/sc/hydra/work/<userid>	• 100GB quota • Fast, keep your personal data here • NOT purged but is NOT backed up
Scratch	/sc/hydra/scratch/<userid>	• Free for all, shared wild west • Current size is about 100TB • Purge every 14 days and limit per user is 5TB
Project	/sc/hydra/projects/<projectid> `$df -h /sc/hydra/projects/<projectid>`	• PI's can request project storage. • Need to submit an allocation request and get approval from allocation committee https://hpc.mssm.edu/hpc-admin/forms/allocation-request (under upgrade) • Not backed up • Incurs charges **$107**/TiB/yr

User Software Environment

<u>OS:</u> Centos 7.6 with **glibc-2.17(GNU C library) available.**

<u>Some key packages:</u>

GCC: system default /usr/bin/gcc is gcc 4.8.5

$ module load gcc (default is 8.3.0)

Python: default version 3.7.3

$ module load python (it will load python and all available python packages)

R: default version 3.5.3 (will update to 3.6.3 soon)

$ module load R (it will load R and all available R packages)

Perl: default system version 5.16.3

$module load CPAN

Anaconda3: default version 2018-12

$module load anaconda3

schrodinger: 2019-1

$module load schrodinger

Matlab access: *$module load matlab*

- The cost for the license is $100.00 per activation, and request form at https://mountsinai.formstack.com/forms/mathworksacademiclicense

User Software Environment: Lmod

Lmod Software Environment Module system implemented:

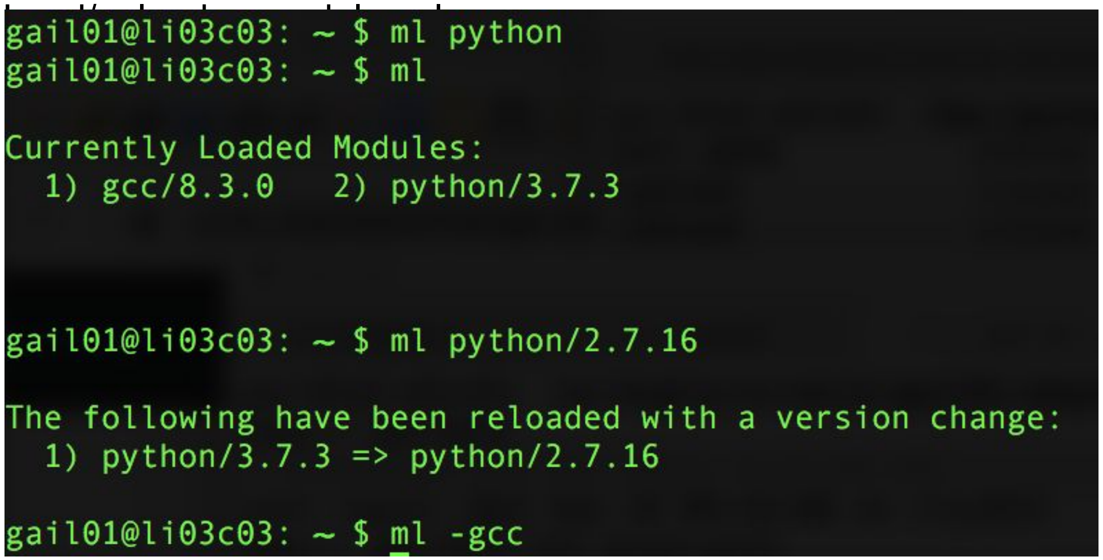
- Written in lua, but reads the TCL module files, and module command will all work
- Search for all possible module: $ module avail or $ module spider

  Check all available R versions

  $ ml spider R

  …….R/3.3.1, R/3.4.0-beta, R/3.4.0, R/3.4.1, R/3.4.3_p, R/3.4.3, R/3.5.0, R/3.5.1_p, R/3.5.1, R/3.5.2, R/3.5.3

- [illegible]



```
gail01@li03c03: ~ $ ml python
gail01@li03c03: ~ $ ml

Currently Loaded Modules:
  1) gcc/8.3.0   2) python/3.7.3



gail01@li03c03: ~ $ ml python/2.7.16

The following have been reloaded with a version change:
  1) python/3.7.3 => python/2.7.16

gail01@li03c03: ~ $ ml -gcc
```

- Autocompletion with tab
- **module save**: Lmod provides a simple way to store the currently loaded modules and restore them later through named collections

User Software Environment: Lmod

Example

```
> ml  python  bedtools  gnuplot  fftw
> ml  R  python/2.7.16  -fftw
> ml  R/3.6.0
> ml
Currently Loaded Modules:
  1) zlib/1.2.8        4) gcc/8.3.0                      7) gsl/2.5         10) R/3.5.3
  2) bedtools/2.29.0   5) intel/parallel_studio_xe_2019   8) libpng/12       11) python/2.7.16
  3) gnuplot/5.2.6     6) hdf5/1.10.5                     9) java/1.8.0_211
> ml  save  myenv1
> ml  purge
> ml
No modules loaded
> ml  restore  myenv1
> ml  savelist
Named collection list :
  1) default  2) myenv1
> ml  disable  myenv1
```

https://lmod.readthedocs.io/en/latest/010_user.html

User Software Environment

Anaconda3:

- Support minimal conda environments (such as tensorflow, pytorch, qiime)

  e.g., tensorflow (both in CPU and GPU)

  ```
  $module load anaconda3 ( or anaconda2 )
  $module load cuda
  $source activate tfGPU
  ```

- User should install their own envs locally,
  - ➔ Use option -p PATH, --prefix PATH Full path to environment location (i.e. prefix).

    ```
    $conda create python=3.x -p /sc/orga/hydra/gail01/conda/envs/myenv
    ```

  - ➔ Set envs_dirs and pkgs_dirs in .condarc file, specify directories in which environments and packages are located

    ```
    $conda create -n myenv python=3.x
    ```

```
$ cat ~/.condarc file
envs_dirs:
- /sc/hydra/work/gail01/conda/envs
pkgs_dirs:
- /sc/hydra/work/gail01/conda/pkgs
```

- Set conda base auto-activation false

  conda config --set auto_activate_base false

User Software Environment - some config

- You can load modules in your **.bashrc** script to load them on startup
- You can create your own modules and modify MODULEPATH so they can be found

  ```
  export MODULEPATH=/hpc/users/fludee01/mymodules:$MODULEPATH
  ```

  or

  ```
  module use /hpc/users/fludee01/mymodules
  ```

Run applications by Containers: Singularity

Singularity tool is supported, instead of docker (Security concern)

- Docker gives superuser privilege, thus is better at applications on VM or cloud infrastructure

To use singularity:

```
$ module load singularity
```

To pull a singularity image:

```
$ singularity pull --name hello.simg shub://vsoch/hello-world
```

To pull a docker image:

```
$singularity pull docker://ubuntu:latest
```

To run a singularity image:

```
$ singularity run hello.simg          # or,        $ ./hello.simg
```

Note: /tmp and user home directory is automatically mounted into the singularity image. If you would like to **get a shell with hydra mounted** in the image, use command:

```
$ singularity run -B /sc/hydra/project/xxx hello.simg
```

To build a new image from recipe files: use Singularity Hub or your local workstation

- Singularity build is not fully supported due to the sudo privileges for users
- After registering an account on Singularity Hub, you can pull or upload your recipe, trigger the singularity build and download the image after built.
- Convert docker recipe files to singularity recipe files:

```
$ml python
$spython recipe Dockerfile Singularity
```

Web server at https://users.hpc.mssm.edu/

New web server is up and under **test** currently

- Setting up python environment
- Address is https://users.hpc.mssm.edu/~userid/, for example: https://users.hpc.mssm.edu/~gail01/
- Will send announcement and more documentation on this

Step 1:

If this folder does not exist in your home directory, you should create it. $ mkdir ~/www

Step 2:

1) Place content in the www folder. $ cat > ~/www/index.html <<EOF

Hello World from my website.

EOF

2) put files or create symlink (from hydra) under the ~/www

WARNING WARNING WARNING: Be careful! Content, executables, scripts, symlinks, applications, etc. within the www/ folder may be (or are) publicly accessible. Scripts and applications launched via Apache in that folder run as your user! They can access any data (including your groups' /project data), delete data, archive data, submit jobs, cancel jobs, email people, etc., as your user. **You are responsible for any actions taken on your behalf!**

File Transfer

- **On Minerva: use login nodes (33h) or interactive nodes (12h).**

  Data transfer node will be available soon.
- **Globus online (Preferred, when available):**
  - Minerva Endpoint: mssm#minerva
  - More information at http://www.globusonline.org
  - Globus Connect Personal to make your laptop an endpoint
- **SCP, SFTP, rsync:**
  - Good for relatively small files, not hundreds of TB's
  - *Some scp apps for Windows/Mac use cached password. This feature must be turned off.*
- **Physical Transport:**
  - We do support the copying of physical hard drives on the behalf of users

Archiving Data: TSM Overview

- Keep for 6 years with two copies
- Can be accessed via either a GUI or the command line

```
$ module load java
$ dsmj -se=userid
```

or

```
$ dsmc -se= userid
```

- Works only on internal login nodes, i.e., **minerva13, minerva14**; NOT on external login node (i.e., minerva11, minerva12)
- Large transfers can take a while. Use a *screen* session and disconnect to prevent time-outs
- Full more details at https://labs.icahn.mssm.edu/minervalab/archiving-data/
- Collaboration account:
  - If your group is in need of a collaboration account for group related tasks like archiving a project directory or managing group website, contact us at hpchelp@mssm.edu

    For more info. see https://labs.icahn.mssm.edu/minervalab/collaboration-account/

Load Sharing Facility(LSF)

A Distributed Resource Management System

Submit Batch Jobs via LSF on Minerva - bsub

- LSF job scripts are very much like bash shell scripts
- bsub options can be entered on command line and/or by placing #BSUB “cookies” in the submitted script

bsub [options] my_batch_job

This will submit the command script “my_batch_job” using the options on the command line. This will NOT interpret the #BSUB cookies in the script.

bsub [options] < my_batch_job, if the job script contains #BSUB cookies:

This will interpret the #BSUB cookies in the script. Options on the command line override what is in the script.

LSF: job submission examples

Interactive session:

```
# interactive session
$ bsub -P acc_hpcstaff -q interactive -n 1 -W 00:10 -Is /bin/bash
# interactive GPU nodes, flag “-R v100” is required
$ bsub -P acc_hpcstaff -q interactive -n 1 -R v100 -R rusage[ngpus_excl_p=1] -W 01:00  -Is /bin/bash
```

Batch jobs submission:

```
# simple standard job submission
$ bsub -P acc_hpcstaff -q premium-n 1 -W 00:10 echo “Hello World”
# GPU job submission if you don’t mind the GPU card model
$ bsub -P acc_hpcstaff -q gpu  -n 1 -R rusage[ngpus_excl_p=1] -W 00:10 echo “Hello World”
# himem job submission, flag “-R himem” is required
$ bsub -P acc_hpcstaff -q premium  -n 1 -R himem -W 00:10 echo “Hello World”
```

To see the list of accessible project accounts:

```
$ mybalance
 User_ID      Project_name        BODE
-------       ---------------     -------
choh07        acc_hpcstaff        Yes
choh07        acc_DGXTrial        No
 ...
```

LSF: batch job submission using a script

```
$ cat  star.lsf
```

```
#!/bin/bash
#BSUB -J mySTARjob                    # Job name
#BSUB -P acc_hpcstaff                 # allocation account
#BSUB -q premium                      # queue
#BSUB -n 8                            # number of compute cores
#BSUB -W 4:00                         # walltime in HH:MM
#BSUB -R rusage[mem=4000]             # 32 GB of memory (4 GB per core)
#BSUB -R span[hosts=1]                # all cores from the same node
#BSUB -o %J.stdout                    # output log (%J : JobID)
#BSUB -eo %J.stderr                   # error log
#BSUB -L /bin/bash                    # Initialize the execution environment

module load star
WRKDIR=/sc/hydra/projects/hpcstaff/benchmark_star
STAR --genomeDir $WRKDIR/star-genome  --readFilesIn Experiment1.fastq  --runThreadN 8
--outFileNamePrefix Experiment1Star
```

```
$ bsub -q express  <  star.lsf
Job <2937037> is submitted to queue <express>.
```

LSF: Queue structure in Chimera (bqueues)

Queue structure in Chimera		
Queue	**Wall time limit**	**available resources**
interactive (Dedicated to interactive jobs)	12 hours	4 nodes+1 GPU node
premium	6 days	270 nodes + 4 himem nodes
express	12 hours	274 nodes (incl. 4 dedicated nodes)
long	2 weeks	4 dedicated (192 cores)
gpu	6 days	44 V100
private	unlimited	private nodes

***default memory : 3000MB / per core**

Useful LSF Commands (see man page for details)

`bsub`

- submits a job interactively or in batch using LSF batch scheduling and queue layer of the LSF suite

`bjobs <job ID# >`

- displays information about jobs in queue or a recently run job. You can use the ***-l*** option to view a more detailed accounting

`bkill <job ID# >`

- kill the job with job ID number of #

`bkill 0`

- kill all your jobs

`bhist -l <job ID# >`

- displays historical information about jobs. A “-a” flag can displays information about both finished and unfinished jobs

Last but not Least

Got a problem? Need a program installed? Send an email to:

hpchelp@hpc.mssm.edu