Introduction to Chimera

Minerva Scientific Computing Environment

https://hpc.mssm.edu

Patricia Kovatch
Eugene Fluder, PhD
Hyung Min Cho, PhD
Lili Gai, PhD
Bhupender Thakur, PhD
Francesca Tartaglione, MS
Dansha Jiang, PhD

April 9, 2019

Mount Sinai

Outline & Highlights

- **Chimera resources: Compute and Storage**
  - Faster CPU, more GPU nodes
  - /sc/orga -> /sc/hydra
- **Chimera Login**
- **User Software Environment**
  - Centos 7.6
  - Lmod
- **LSF 10.1: Jobs and Queues**
  - Long queue with wall time of 2 weeks
  - Interactive queue: internet access, GPU available
  - Gold is not implemented

Chimera: Compute and Storage

Chimera Computes:

- 4x login nodes - Intel Skylake 8168 24C, **2.7GHz** - **384 GB** memory
- 280 compute nodes - Intel 8168 24C, **2.7GHz**
  - 13,440 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

Total number of cores (computes + GPU + high mem) = 14,304 cores

Chimera Storage:

- Created new file system **/sc/hydra** on Chimera as primary storage
  - GPFS 5.0.2
  - Have the same structure as /sc/orga (work, projects, scratch directories)
  - Use the system path environment variable in scripts
- /sc/orga is still mounted on Chimera
  - Will migrate orga directories to hydra
  - Will remove orga (ETA: end of 2019)

Suggestion to users during migration

Use environment variable instead of absolute path:

In your job submission script, add

PROJECT=/sc/orga/projects/my_project

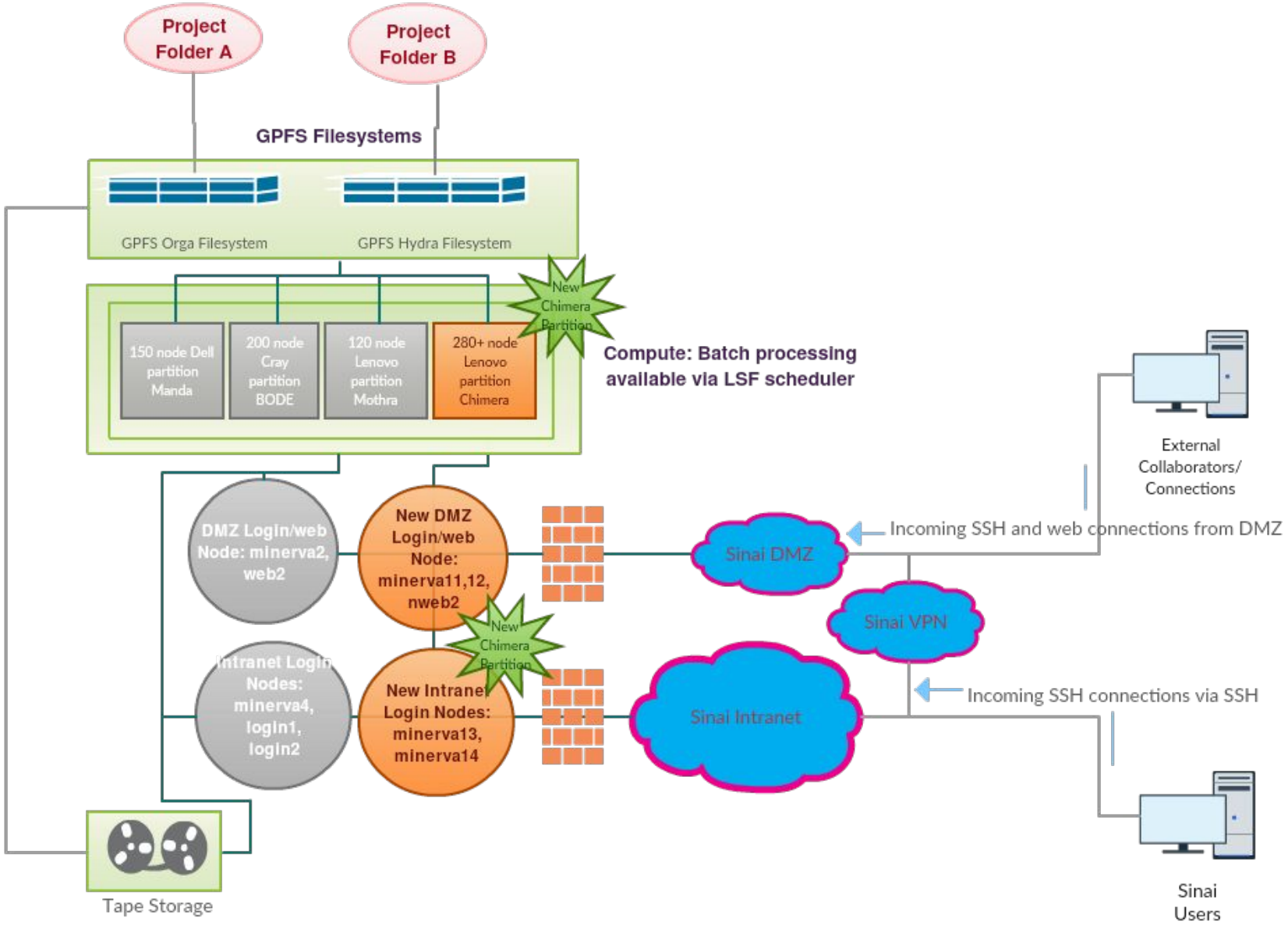
and modify all "/sc/orga/projects/my_project" to "$PROJECT".

Once your project dir is moved to hydra, change to

PROJECT=/sc/hydra/projects/my_project

You can do the same for work and scratch directory if needed.

Minerva Architecture



Chimera Login: New login nodes

New set of login nodes:

- 4 new login nodes: **minerva[11-14]**, which points to the login node **li03c[01-04]**.
  - **minerva[13-14] (or li03c[03-04]) are internal login nodes**

    - currently available via Minerva and MSSM campus.
  - **minerva[11-12] (or li03c[01-02]) are external login nodes**

    - will be available when DMZ network is setup.

- Data transfer nodes: **data2, data4**.
  - will be included in Chimera partition.
- Other login nodes, minerva2&4 and login1&2, will be retired along with their compute partition.

Chimera Login: Login method

During migration period, you can assess Chimera by:

1. All users: hop directly from Minerva internal login nodes (passwordless):

   [jiangd03@minerva4 ~]$ ssh li03c03

   [jiangd03@minerva4 ~]$ ssh li03c04

2. Login from campus (two factor authentication), please choose any of the following combination:

Users	Login method	Login servers	Password Components
Sinai users	user1 user1+vkrb	@chimera.hpc.mssm.edu @minerva13.hpc.mssm.edu @minerva14.hpc.mssm.edu	Sinai Password + 6 Digit Symantec VIP token code
External users	user1+**yipa**		HPC Password + YubiKey Button Push

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

Two password sets: external users

Different password sets used in Minerva			
	MSSM password	**HPC password - Minerva**	**HPC password - Chimera**
Users	Sinai users	external users (everyone)	external users (everyone)
Where it is applicable	Sinai (email, wifi) Old Minerva + Chimera	Old Minerva partition	Chimera partition
Server @Realm	Kerberos server @mssmcampus.mssm.edu	Ldap server	freeIPA server @hpc.mssm.edu
Password reset	via Mount Sinai Password Manager https://passwordreset.mountsinai.org/	via shell command (ldap-passwd) in Minerva: **$ hpcpasswd**	via shell command (ipa passwd) in Chimera partition: **$ hpcpasswd**
Two factor login method	+vkrb (Symantec token, default)	+yldap (yubi key)	+yipa (yubi key)

External users: change HPC password

Your initial password is random generated string stored at your home directory .credential.
After login to Chimera via hopping, use unmunge command to decode:

[hpctrn01@li03c03 ~]$ cat .credential

MUNGE:AwQDAAAdxUrBi1znWTBV28wOJZP3XouClwxG/ubzooIBvGIS8jTH/p/I+2ru6ovfpZXPub3H/sCww7eP6pVeCKt
mhu1BehLnWWFYoj0jMEwOJcGWnT83z3rIAKBbs1HJ/bYyhg5zugaX:

[hpctrn01@li03c03 ~]$ cat .credential | unmunge

…[some info]...

*4Ml;vID4|$IIK7@%_s*9}<*

To change password, run following 2 commands. New password has to be longer than 8 characters:

[hpctrn01@li03c01 ~]$ kinit

*Password for hpctrn01@HPC.MSSM.EDU: 4Ml;vID4|$IIK7@%_s*9}<*

Note:

- No output if correct password was entered.

- Error message "*kinit: Password incorrect while getting initial credentials*" if wrong password was entered.

- Same command can be used to check your new HPC password 10 mins after changed (cache expire).

[hpctrn01@li03c03 ~]$ hpcpasswd

*Current Password: 4Ml;vID4|$IIK7@%_s*9}<*

New Password:

Enter New Password again to verify:

Changed password for "hpctrn01@HPC.MSSM.EDU"

Home Directory: /hpc/user

New NFS storage (/hpc):

- User quota for home directory is increased to 20 GB.

To facilitate the transition and move data over the new storage:

- The old /hpc is mounted on the login nodes as **/hpc-old/users in READ-ONLY mode**.
- A link (~/userid-old-home) is created in user home dir pointing to their old home directory.

```
[jiangd03@li03c03 ~]$ ls -l
lrwxrwxrwx 1 root root 23 Mar 26 10:53 jiangd03-old-home -> /hpc-old/users/jiangd03
```

- **Note:** /hpc-old is not mounted on compute nodes, so job will fail if it uses ~/userid-old-home.

We urge users to move their data over the new storage as the old NFS will be out of support in July 2019.

User Software Environment

OS: **Centos 7.6** was deployed in Chimera

- **Glibc-2.17 available.** If higher version needed, go with a container
- Key packages of latest version are being built under centos7.6, and set as default
  Such as gcc/8.3.0, openmpi/4.0.1, intel/2019, Python/3.7.3, R/3.5.3, Rstudio/1.1.463

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

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

Anaconda3: default version 2018-12

$module load anaconda3

User Software Environment

Anaconda3:

- Support minimal conda environments (such as tensorflow, pytorch, quime)
- User should install their own envs locally,
  - ➔ Use option -p PATH, --prefix PATH Full path to environment location (i.e. prefix).

    $conda create python=3.6 -p /sc/orga/work/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=3.6

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

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
- Load/uload a module - ml

```
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
```

  $ 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

Lmod: module save

```
gail01@li03c03: ~ $ ml

Currently Loaded Modules:
  1) openmpi/4.0.1   2) boost/1.69.0   3) python/3.7.3   4) gcc/8.3.0   5) intel/parallel_studio_xe_2019   6) R/3.5.3   7) rstudio/1.1.463   8) selfsched/0.34

gail01@li03c03: ~ $ ml save testsave
Saved current collection of modules to: "testsave"

gail01@li03c03: ~ $ ml savelist
Named collection list :
  1) testsave

gail01@li03c03: ~ $ ml describe testsave
Collection "testsave" contains:
   1) openmpi    2) boost    3) python    4) gcc    5) intel    6) R    7) rstudio    8) selfsched
gail01@li03c03: ~ $ ml purge
gail01@li03c03: ~ $ ml
No modules loaded

gail01@li03c03: ~ $ ml restore testsave
Restoring modules from user's testsave
gail01@li03c03: ~ $ ml

Currently Loaded Modules:
  1) openmpi/4.0.1   2) boost/1.69.0   3) python/3.7.3   4) gcc/8.3.0   5) intel/parallel_studio_xe_2019   6) R/3.5.3   7) rstudio/1.1.463   8) selfsched/0.34
```

LSF: Job scheduling system

New Job scheduling system in Chimera:

- New job scheduling server with LSF upgrade to v10.1.
- Job temporary dir configured to /local/JOBS instead of /tmp.

Interactive sessions:

- No interactive nodes is configured in Chimera partition to avoid abusive usage. Interactive sessions is available via job scheduler in the **interactive queue**.
- Interactive1&2 and Interactive5&6 will be retired along with their compute partition.
- Nodes in interactive queues will have outside network access, i.e., **data transfer** will be available in the interactive sessions.
- **Interactive GPU** is available for job testing.

LSF: Queue structure in Chimera

Queue structure in Chimera			
Queue	**priority/APS**	**Wall time limit**	**available resources**
interactive		12 hours	4 nodes+1 GPU node
normal	100/APS	3 days	77 nodes
premium	200/APS	6 days	200 nodes + 2 high-mem
express		12 hours	280 nodes
long	100/APS	2 weeks	2 dedicated highmem nodes (96 cores)
GPU	100/APS	6 days	48 V100
private		unlimited	private nodes

LSF: Job submission examples

Interactive session:

 # interactive session

$ bsub -P hpcstaff -q interactive -n 1 -W 00:10 -Is /bin/bash

 # interactive GPU nodes, flag "-R v100" is required

$ bsub -P hpcstaff -q interactive -n 1 -R v100 -R rusage[ngpus_excl_p=1] -W 01:00 -Is /bin/bash

Batch jobs submission:

 # standard job submission

$ bsub -P hpcstaff -q normal -n 1 -W 00:10 echo "Hello World"

 # GPU job submission if you don't mind the GPU card model

$ bsub -P hpcstaff -q gpu -n 1 -R rusage[ngpus_excl_p=1] -W 00:10 echo "Hello World"

 # flag "-R v100" is required if you want to use certain GPU card (v100/p100)

$ bsub -P hpcstaff -q gpu -n 1 -R v100 -R rusage[ngpus_excl_p=1] -W 00:10 echo "Hello World"

 # himem job submission, flag "-R himem" is required

$ bsub -P hpcstaff -q premium -n 1 -R himem -W 00:10 echo "Hello World"

Job submission script example in test.lsf

```
#!/bin/bash
#BSUB -J myjob                    # Job name
#BSUB -P hpcstaff                 # allocation account or Unix group
#BSUB -q express                  # queue
#BSUB -n 1                        # number of compute cores
#BSUB -W  6:00                    # walltime in HH:MM
#BSUB -R rusage[mem=4000]         # 4 GB of memory requested
#BSUB -o %J.stdout                # output log (%J : JobID)
#BSUB -eo %J.stderr               # error log
#BSUB -L /bin/bash                # Initialize the execution environment

module load gcc
which gcc
echo “Hello Chimera”
```

Submit the script with the **bsub** command:

```
bsub < test.lsf
```

Containers: Singularity

Singularity is installed on

- login node
- compute nodes in the interactive queue

  $ bsub -P hpcstaff -q interactive -n 1 -W 00:10 -R singularity -Is /bin/bash
- selected compute nodes in the premium queue

  $ bsub -P hpcstaff -q premium -n 1 -W 00:10 -R singularity singularity run hello.simg

To pull a singularity image:

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

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 but not /sc/orga. If you would like to get a shell with orga mounted in the image, use command:

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

Note: Singularity build is not fully supported due to the sudo privileges for normal users. If you would like to build a new image, you can use Singularity Hub. After registering an account on Singularity Hub, you can pull or upload your recipe, trigger the singularity build and download the image after built.

Summary of user migration effort

1. **For external user** or users who use HPC password:
   change **HPC password** after login and validate **ipa** login method.
2. Copy files from old **home** dir to new home.
3. Copy files from **work** and **scratch** directory from /sc/orga to /sc/hydra.
4. HPC admins will copy files for **project directory** (no effort from users is needed), users/PI will be contacted during the migration.
5. Modify your **path to the project directory** when you start using /sc/hydra.
6. Change your **job submission script** with new queue structure.
7. Test **new packages** are working.

All updates will be announced on our HPC website as well as the weekly newsletter.

Follow us by visiting https://hpc.mssm.edu/, weekly update and twitter

Important dates:

2019 Chimera partition installation plan

- **Apr 01, 2019:** Chimera in production
- **Jul 01, 2019:** Retire Manda, Mothra, BODE
- **Sep 01, 2019:** Chimera **HIPAA compliant cluster**
- **Dec 31, 2019:** /sc/orga unavailable

Last but not Least

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

hpchelp@hpc.mssm.edu