Lawrencium 101

Wei Feinstein, Ph.D.

High Performance Computing
Lawrence Berkeley National Laboratory

Outline

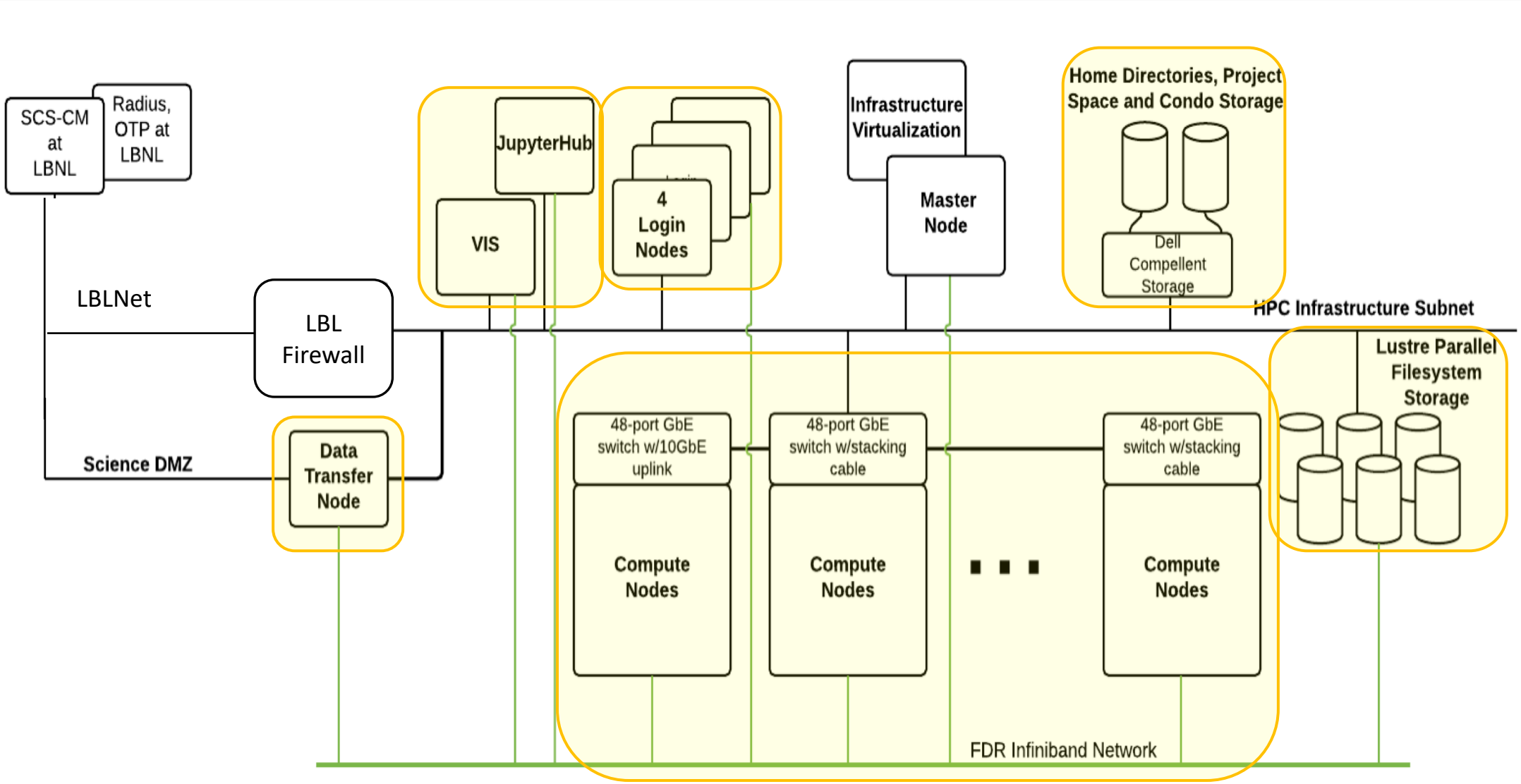
- LBNL Supercluster overview
- Software module farm service
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

LBNL Supercluster

- LBNL Supercluster
  - 15 Departmental clusters: ALSACC, JGI, Nano, Vulcan, Dirac1…
  - Condo cluster: Lawrencium
- Lawrencium is a LBNL Condo Cluster Computing program
  - Significant investment from LBNL
  - Individual PIs purchase nodes and storage
  - Computional cycles are shared among all lawrencium users
- Share the same supercluster infrastructure:

  High speed infiniBand parallel file system, OTP authentication, OS and security updates, software module farm, job scheduler, home/project storage, lustre parallel file system and backend network infrastructure.

Lawrencium Cluster Overview



Lawrencium Hardware Configuration

Partition	Nodes	Node List	CPU	Cores	Memory	Infiniband	Accelerator
lr2 (DECOMMISSIONED 11/1/2019)	198	n0[000-141].lr2 n0[146-153].lr2 n0[161-180].lr2 n0[182-208].lr2	INTEL XEON X5650	12	24GB	QDR	-
		n0181.lr2			96GB		
lr3	337	n0[000-163].lr3 n0[309-336].lr3	INTEL XEON E5-2670	16	64GB	FDR	-
		n0[164-203].lr3 n0[213-308].lr3	INTEL XEON E5-2670 v2	20	64GB		
lr4	141	n0[000-095].lr4 n0[099-110].lr4 n0[112-135].lr4 n0[139-147].lr4	INTEL XEON E5-2670 v3	24	64GB	FDR	-
lr5	188	n0[000-143].lr5 n0[148-191].lr5	INTEL XEON E5-2680 v4	28	64GB	FDR QDR	-
			INTEL XEON ES-2640 v4	20	128GB		
lr6	88	n0[000-087].lr6	INTEL Gold 6130 (Skylake)	24	96GB	FDR	-
				64	128GB		
lr6	156	n0[088-115, 144-271].lr6	INTEL XEON Gold 5218 (Cascade)	60	96GB	FDR	
				96	128GB		
lr_bigmem	2	n0210.lr3	INTEL XEON E5-4620	32	1024GB	QDR	-
		n0211.lr3	INTEL XEON E7-4860 v2	48	1024GB		
es1	24	n[0000-0011].es1	Intel XEON E5-2623	8	64GB	FDR	4X NVIDIA 1080TI
		n[0012-0023].es1			187GB		2X NVIDIA V100
cf1	72	n0[000-071].cf1	INTEL XEON PHI 7210	64	192GB	FDR	
cm1	14	n0[000-013].cm1	AMD EPYC	48	256GB	FDR	

https://sites.google.com/a/lbl.gov/high-performance-computing-services-group/lbnl-supercluster/lawrencium:q

Storage and Backup

Name	Location	Quota	Backup	Allocation	Description
HOME	/global/home/users/$USER	10GB	Yes	Per User	HOME directory for permanent data storage
GROUP-SW	/global/home/groups-sw/$GROUP	200GB	Yes	Per Group	GROUP directory for software and data sharing with backup
GROUP	/global/home/groups/$GROUP	400GB	No	Per Group	GROUP directory for data sharing without backup
SCRATCH	/global/scratch/$USER	none	No	Per User	SCRATCH directory with Lustre high performance parallel file system
CLUSTERFS	/clusterfs/axl/$USER	none	No	Per User	Private storage for AXL condo
CLUSTERFS	/clusterfs/cumulus/$USER	none	No	Per User	Private storage for CUMULUS condo
CLUSTERFS	/clusterfs/esd/$USER	none	No	Per User	Private storage for ESD condos
CLUSTERFS	/clusterfs/geoseq/$USER	none	No	Per User	Private storage for CO2SEQ condo
CLUSTERFS	/clusterfs/nokomis/$USER	none	No	Per User	Private storage for NOKOMIS condo

User Account

- User account request
- User agreement consent

https://sites.google.com/a/lbl.gov/hpc/getting-an-account

Project Accounts

- PI Computing Allowance (PCA) account: free 300K SUs per year (pc_xxx)
- Condo account: PIs purchase and contribute compute nodes to the general condo pool (lr_xxx)
  - Run jobs within their condo contributions for free
- Recharge account: with minimal recharge rate ~ $0.01/SU (ac_xxx)
- $25 per user per month, recha

https://sites.google.com/a/lbl.gov/hpc/getting-an-account

Recharge weights

System	Rate	Compute Node Description
LR6	1 SU per core	Intel Xeon Gold 6130 - 32-core nodes
LR5	0.75 SU per core-hr	Intel Broadwell 28-core nodes
LR4	0.75 SU per core-hr	Intel Haswell 24-core nodes
LR3	0.50 SU per core-hr	Intel SandyBridge 16-core and IvyBridge 20-core nodes
LR2	0.00 SU per core-hr	Intel Westmere 12-core nodes
CF1	0.40 SU per core-hr	Intel Xeon PHI - 64-core nodes
ES1	1 SU per core	Intel Xeon ES - 8-core GPU nodes (1080TI and v100)
CM1	1 SU per core	AMD EPYC - 48-core nodes

Cluster Login

- Login server: lrc-login.lbl.gov (4 login nodes)
  - ssh $USER@lrc-login.lbl.gov
  - OTP authentication

https://sites.google.com/a/lbl.gov/high-performance-computing-services-group/getting-started/login

DO NOT run jobs on login nodes!!

Outline

- LBNL Supercluster overview
- Software module farm
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

Software Module Farm

- Commonly used compiler, software tools provided to all cluster users
- Maintained on a centralized storage device and mounted as read-only NFS file system
- Categories :
  - Compilers: intel, gcc, MPI compilers, Python
  - Tools: matlab, singularity, cuda
  - Applications: machine learning, QChem, MD, cp2k
  - Libraries: fftw, lapack
- module : tool for managing users' software environment
  - Setup users' environment : PATH, LD_LIBRARY_PATH…

```
[wfeinstein@n0003 ~]$ module av

------------------------------------------------------ /global/software/sl-7.x86_64/modfiles/langs -------------------------------------
-----------------
clang/3.9.1              cuda/9.1                intel/2016.4.072(default) julia/1.0.3              r/3.4.2
clang/9                  gcc/4.8.5               intel/2018.1.163          perl/5.20.0              ruby/2.4.0
cuda/10.0                gcc/5.4.0               intel/2018.5.274.par      python/2.7
cuda/7.5                 gcc/6.3.0(default)      java/1.8.0_121            python/3.5
cuda/8.0                 gcc/7.4.0               julia/0.5.0               python/3.6
cuda/9.0(default)        gcc/9.2.0               julia/0.6.4               r/3.2.5

------------------------------------------------------ /global/software/sl-7.x86_64/modfiles/tools -------------------------------------
-----------------
arpack-ng/3.4.0         gflags/2.2.0            leveldb/1.19            octave/4.2.0            snappy/1.1.4            valgrind/3.15.0
bazel/0.11.1            git/2.11.1              libxaw3d/1.6.2          opencv/3.2.0            subversion/1.9.4        vim/7.4
cmake/3.15.0            globus/6.0.1478289945   llvm/3.9.1              openmotif/2.3.3         suitesparse/4.5.4       visit/2.12.0
cmake/3.7.2             glog/0.3.4              lmdb/0.9.19             paraview/5.1.2          swig/3.0.12             xkbcommon/0.7.1
eigen/3.3.3             gnu-parallel/2019.03.22 matlab/r2016b           proj.4/4.9.3            szip/2.1.1              xpdf/3.04
emacs/25.1              gnuplot/5.0.5           matlab/r2017b(default)  protobuf/3.2.0          tcl/6.9.8609.2
fontconfig/2.13.91      grace/5.1.22            matlab/r2019a           qchem/4.2               tcl/8.5.11
freetype2/2.10.0        gv/3.7.4                mercurial/4.1           qchem/5.2               texlive/2016
gdal/2.2.3              imagemagick/7.0.4-7     nano/2.7.4              qt/5.4.2                tmux/2.7
geany/1.32              intltool/0.50.2         nerschpass/4.0.1.3      singularity/3.2.1       totalview/8.10.0-0

------------------------------------------------------ /global/software/sl-7.x86_64/modfiles/apps --------------------------------------
-----------------
bio/blast/2.6.0                 ml/mxnet/0.9.3-py27             ml/tensorflow/1.8.0-cpu-py36    ms/gromacs/5.1.4-mpi
bio/bowtie2/2.3.0               ml/mxnet/0.9.3-py35             ml/tensorflow/1.8.0-py36        ms/lammps/17Nov16-mpi
bio/picard/2.9.0                ml/mxnet/0.9.3-py36             ml/tensorflow/2.0.0b1-py36      ms/lammps/lammps-2019
bio/samtools/1.4                ml/mxnet/0.9.3-r-3.2.5          ml/torch/torch7                 ms/namd/2.12-cuda
bio/vcftools/0.1.15             ml/tensorflow/1.12.0-cpu-py36   ms/cp2k/4.1-cuda                ms/namd/2.12-mpi
env/bce/current                 ml/tensorflow/1.12.0-py36       ms/cp2k/4.1-impi-2018.5.274     ms/nwchem/6.6-mpi
ml/caffe/rc5                    ml/tensorflow/1.7.0-py27        ms/cp2k/4.1-mpi                 ms/proj/5.2-gcc-6.3.0-lapack
ml/cntk/2.0.beta15-py27         ml/tensorflow/1.7.0-py35        ms/geant4/10.4.0                ms/proj/5.2-intel-2018.1.163-mkl
ml/cntk/2.0.beta15-py35         ml/tensorflow/1.7.0-py36        ms/gromacs/5.1.4-cuda           ms/qe/6.1-mpi

------------------------------------------------ /global/software/sl-7.x86_64/modfiles/intel/2016.4.072 ---------------------------------
-----------------
3.10.2-complex                    fftw/3.3.8-intel                nco/4.6.4-intel-p               openmpi/3.0.0-intel
antlr/2.7.7-intel                 gsl/2.3-intel                   nco/4.6.4-intel-s               openmpi/3.0.1-intel
atlas/3.10.3-intel                hdf5/1.8.18-intel-p             nco/4.7.4-intel-p               petsc/3.10.4-complex
berkeley_upc/2.24.0-intel(default) hdf5/1.8.18-intel-s            ncview/2.1.7-intel              slepc/3.10.2-complex
berkeley_upc/2.26.0-intel         hdf5/1.8.20-intel-p             netcdf/4.4.1.1-intel-p          tbb/2016.4.072
boost/1.63.0-intel                ipp/2016.4.072                  netcdf/4.4.1.1-intel-s          udunits/2.2.24-intel
daal/2016.4.072                   lapack/3.8.0-intel              netcdf/4.6.1-intel-p
fftw/2.1.5-intel                  mkl/2016.4.072                  openmpi/2.0.2-intel(default)
fftw/3.3.6-intel(default)         ncl/6.3.0-intel                 openmpi/2.1.3-intel
[wfeinstein@n0003 ~]$
```

Module commands

- **`module purge`**: clear user's work environment
- **`module available`**: check available software packages
- **`module load xxx`**: load a package
- **`module list`**: check currently loaded software
- What to do if a persistent setup is desired?
  ~/.bashrc

https://sites.google.com/a/lbl.gov/high-performance-computing-services-group/getting-started/sl6-module-farm-guide

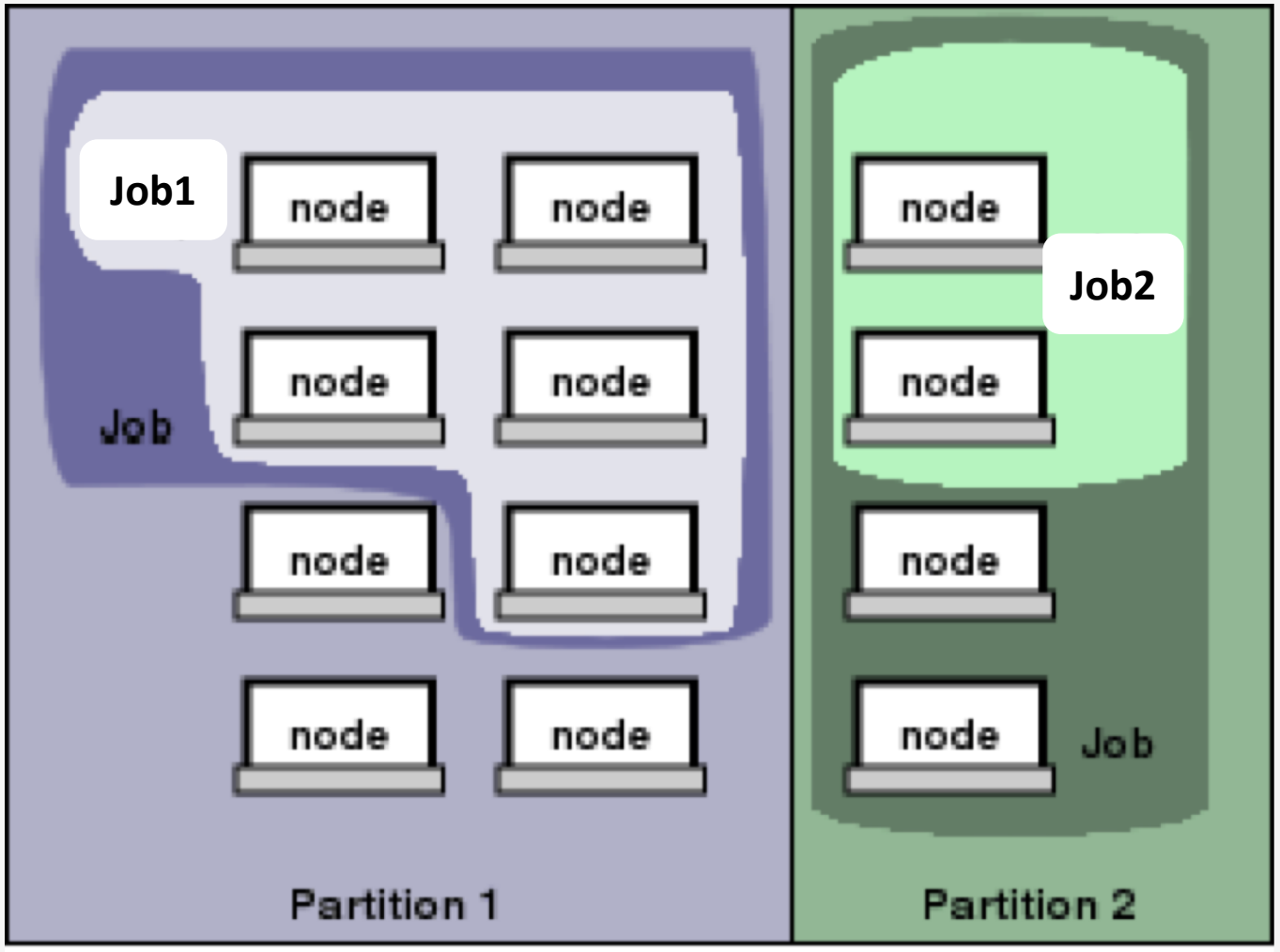
Software Module Farm

- Commonly used compiler, software tools provided to all cluster users
- Maintained on a centralized storage device and mounted as read-only NFS file system
- Components:
  - Compilers: intel, gcc, MPI compilers, Python
  - Tools: matlab, singularity, cuda
  - Applications: machine learning, QChem, MD, cp2k
  - Libraries: fftw, lapack
- Module: framework to manage users' software environment
  - Setup users' environment: PATH, LD_LIBRARY_PATH…
- Users may install their own software

Outline

- LBNL Supercluster overview
- Software module farm
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

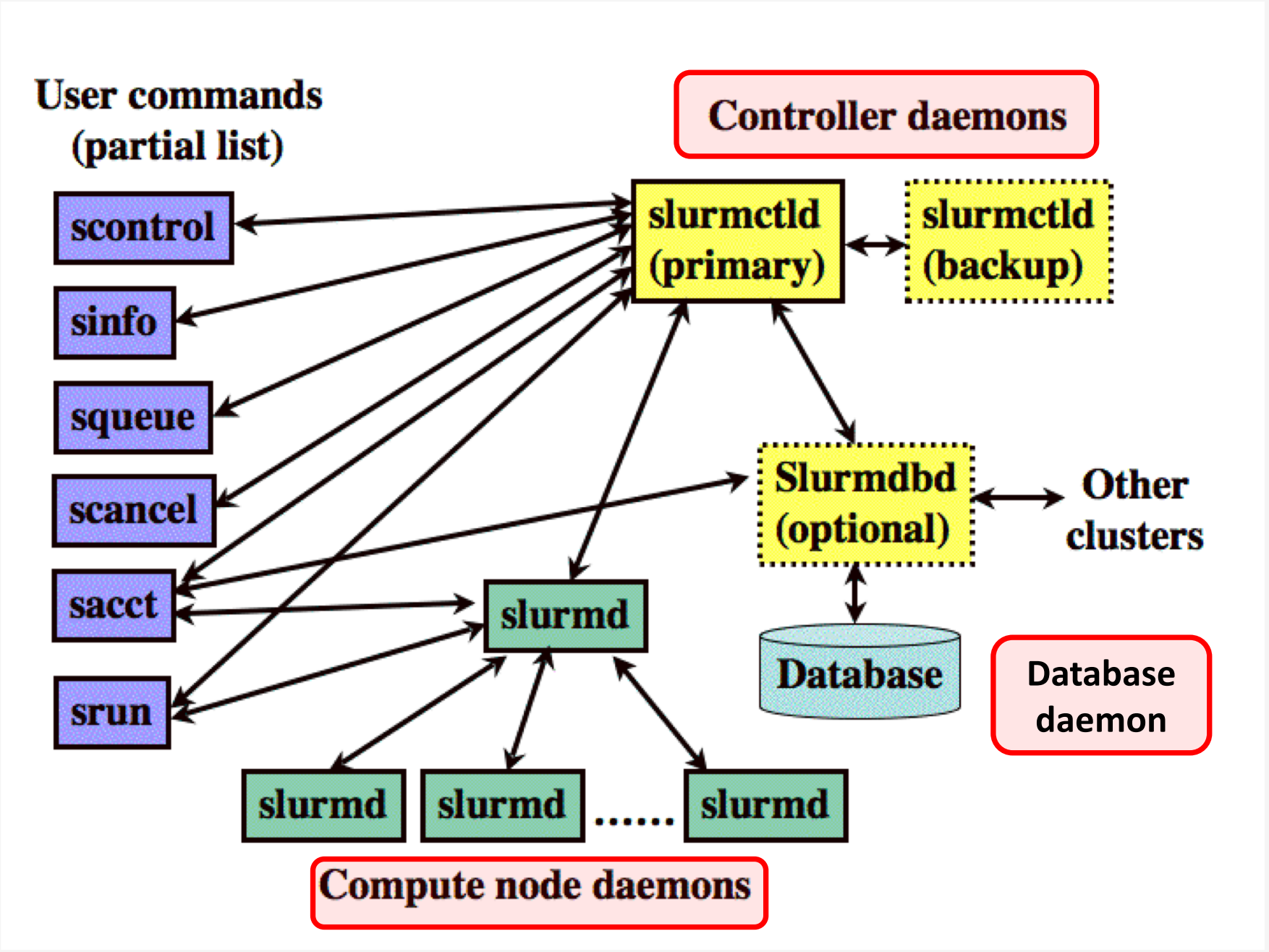
Partitions, nodes and jobs



SLURM

Workload manager
&
job scheduling
system for
HPC clusters

(Simple Linux Utility
for Resource
Management)



What happens when a job is submitted

- Job submission via srun, sbatch, salloc
- Check user’s account association

```
[wfeinstein@n0002 ~]$ sacctmgr show assoc -p|grep davidp
perceus-00|etna|davidp|etna_bigmem|1|||0|||||||||normal|||
perceus-00|etna|davidp|etna-shared|1|||0|||||||||normal|||
perceus-00|etna|davidp|etna_gpu|1|||0|||||||||normal|||
perceus-00|etna|davidp|etna|1|||0|||||||||normal|normal||
perceus-00|vulcan|davidp|vulcan_c20|1|||0|||||||||normal|||
perceus-00|vulcan|davidp|vulcan_gpu|1|||0|||||||||vulcan_gpu|||
perceus-00|vulcan|davidp|vulcan|1|||0|||||||||normal,vulcan_debug|||
perceus-00|ac_nanotheory|davidp|cm1|1|||||||||||cm1_debug,cm1_normal|||
perceus-00|ac_nanotheory|davidp|lr6|1|||||||||||lr_debug,lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|es1|1|||||||||||es_debug,es_lowprio,es_normal|||
perceus-00|ac_nanotheory|davidp|cf1|1|||||||||||cf_debug,cf_lowprio,cf_normal|||
perceus-00|ac_nanotheory|davidp|lr5|1|||||||||||lr_debug,lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|lr4|1|||||||||||lr_debug,lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|lr_bigmem|1|||||||||||lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|lr3|1|||||||||||lr_debug,lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|lr2|1|||||||||0|||lr_debug,lr_lowprio,lr_normal|||
perceus-00|ac_nanotheory|davidp|mako|1|||||||||||mako_debug,mako_lowprio,mako_normal|||
perceus-00|catamount|davidp|catamount|1|||||||||||cm_debug,cm_long,cm_medium,cm_short|||
perceus-00|hbar|davidp|hbar1|1|||0|||||||normal|normal||
perceus-00|lr_nanotheory|davidp|lr3|1|||||||||||condo_nanotheory|||
```

What happens when a job is submitted

- Job submission via srun, sbatch, salloc
- Check user's account association
- Job validation very 5 minutes
  - Check requested resources (memory, time and nodes) against available resources
  - Scan the queue to see if any small jobs fit into reserved nodes preventing starving small jobs
    Backfill
  - Assign nodes to jobs
  - Kill jobs upon completion

Job priority

- Weights of job priority factors

```
[wfeinstein@n0003 ~]$ sprio -w
  JOBID PARTITION   PRIORITY    AGE  FAIRSHARE  PARTITION    QOS
  Weights                       1000   100000      10000       1000000
```

- QOS has the heaviest weight (why?)

```
[wfeinstein@n0003 ~]$ sacctmgr show qos -p |grep condo_mp_es1
condo_mp_es1|1000000|00:00:00|lr_lowprio|cluster|||1.000000|node=1|||||||||||
||||cpu=1|

[wfeinstein@n0003 ~]$ sacctmgr show qos -p |grep lr_normal
lr_normal|1000|00:00:00|lr_lowprio|cluster|||1.000000|||||||node=64|||
3-00:00:00|||||||cpu=1|
```

QOS (Quality of Services)

QOS on Lawrencium: lr_debug, lr_lowprio, lr_normal

[wfeinstein@n0002 ~]$ sacctmgr show qos -p|head -1; **sacctmgr show qos -p |grep lr_normal**

Name|Priority|GraceTime|Preempt|PreemptMode|Flags|UsageThres|UsageFactor|GrpTRES|GrpTRESMins|GrpTRESRunMins|GrpJobs|GrpSubmit|GrpWall|MaxTRES|MaxTRESPerNode|MaxTRESMins|MaxWall|MaxTRESPU|MaxJobsPU|MaxSubmitPU|MaxTRESPA|MaxJobsPA|MaxSubmitPA|MinTRES|

lr_normal|1000|00:00:00|lr_lowprio|cluster|||1.000000|||||||node=64|||3-00:00:00|||||||cpu=1|

[wfeinstein@n0002 ~]$ **sacctmgr show qos -p |grep lr_debug**

lr_debug|10000|00:00:00|lr_lowprio|cluster|||1.000000|||||||node=4|||00:30:00|||||||cpu=1|

[wfeinstein@n0002 ~]$ **sacctmgr show qos -p |grep lr_lowprio**

lr_lowprio|0|00:00:00||requeue|||1.000000|||||||||||||||||

QOS (Quality of Services)

- QOS on Lawrencium: lr_debug, lr_lowprio, lr_normal
  - lr_normal: 72 hours max, no preemption
  - lr_debug: 30 minutes max, high priority
  - lr_lowprio: no charge, but jobs can be preempted when higher priority jobs request resources
- Condo QOS: defined by individual PIs who contributed hardware to the condo pool, no lowprio QOS

Slurm commands

- **`sbatch:`** submit a job to the batch queue system

```
sbatch myjob.sh
```

- **`srun:`** request an interactive node(s) and login automatically

```
srun -A ac_xxx -p lr5 -q lr_normal -t 1:0:0
--pty bash
```

- **`salloc`** : request an interactive node(s)

```
salloc –A pc_xxx –p lr6 –q lr_debug –t 0:30:0
```

Slurm commands

- **sinfo :** view information about partitions and nodes (idle, allocated, drain, down)

  ```
  sinfo –r –p lr6
  ```

- **squeue:** check the current jobs in the batch queue system

  ```
  squeue –u $USER
  ```

- **sacct:** information on jobs

  ```
  sacct -X -o 'jobid,user,partition,nodelist,stat'
  sacct --helpformat (list of fields)
  ```

- **`scancel`** : cancel a job

  ```
  scancel jobID
  ```

Outline

- LBNL Supercluster overview
- Software module farm
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

Interactive job submission

- Run programs interactively
- Mainly for code debugging, testing, visualization

```
srun --account pc_xxx --partition lr6
--qos=lr_normal  --time=1:0:0 --constraint=lr_cas
--nodes=2 --pty bash
```

  - Your account association
    - `sacctmgr show assoc –p| grep $USER`
  - Features/constraints (slurm) of partitions

https://sites.google.com/a/lbl.gov/high-performance-computing-services-group/lbnl-supercluster/lawrencium

Features of partitions

Partition	Account	Nodes	Node List	Node Features	Shared	QoS	QoS Limit
etna	nano	167	n0[005-171].etna0	etna	Exclusive	normal	no limit
etna	nano	3	n0[172-174].etna0	etna etna_phi	Exclusive	normal	no limit
etna_gpu	nano	11	n0[175-185].etna0	etna_gpu etna_k80 etna_v100	Exclusive	normal	no limit
etna-shared	nano	5	n0[000-004].etna0	etna_share	Shared	normal	no limit
etna_bigmem	nano	52	n0[186-237].etna0	etna_bigmem	Exclusive	normal	no limit

https://sites.google.com/a/lbl.gov/high-performance-computing-services-group/lbnl-supercluster/lawrencium

Batch job submission

- Job executed without user intervention

```
sbatch myjob.sh
```

Batch job submission script

```
#!/bin/bash -l

#SBATCH --job-name=hello                    (-J)
#SBATCH --partition=lr6                     (-p)
#SBATCH --account=ac_xxx                    (-A)
#SBATCH --qos=lr_normal                     (-q)
#SBATCH --nodes=4                           (-N)
#SBATCH --ntasks=2                          (-n)
#SBATCH --ntasks-per-node=8
#SBATCH --cpus-per-task=1
#SBATCH --time=1-2:0:0                      (-t)
#SBATCH --mem=96G
#SBATCH --constraint=lr6_sky                (-C)
#SBATCH --output=hello_%j.log               (-o)
#SBATCH --error=hello.err_%j.err            (-e)
#SBATCH --mail-user=wfeinstein@lbl.gov
#SBATCH --mail-type=ALL

module load openmpi-gcc/3.1.0
cd $SLURM_SUBMIT_DIR
mpirun –np 64 app-name > log 2>&1
```

SLURM Environment Variables

$SLURM_JOB_ID	Job ID
$SLURM_JOBID	Deprecated. Same as SLURM_JOB_ID
$SLURM_JOB_NAME	Job Name
$SLURM_SUBMIT_DIR	Submit Directory
$SLURM_JOB_NODELIST	Nodes assigned to job
$SLURM_SUBMIT_HOST	Host submitted from
$SLURM_JOB_NUM_NODES	Number of nodes allocated to job
$SLURM_CPUS_ON_NODE	Number of cores/node
$SLURM_NTASKS	Total number of cores for job???
$SLURM_NODEID	Index to node running on relative to nodes assigned to job

Job submission to shared partitions

- Most partitions are configured as exclusive: an entire compute node belongs to a single user
- Shared partitions: es1, mhg, etna …
- Multiple users may share the same node(s)
- Oversubscribe=FORCE in setup in slurm.conf
- Slurm flag --mem is critical when applications need an entire node or have large memory footprint

```
srun -A lr_xx -p es1 -q es_normal -t 1:0:0 -N 1
--ntasks=8 --mem=32G --pty bash
```

Job submission to GPU es1 partition

- es1 partition has 24 GPU nodes
- Either 4x1080Ti or 2xv-100 GPUs attached to each node
- es1 partition is configured as shared
- Slurm flags --gres and --ntasks (1:2 ratio) required

```
srun -A  pc_xxx -p es1 –q es_normal -t 1:0:0
-C es1_v100 --gres=gpu:2 –ntasks=4 -N 1 -pty bash

srun --account=ac_xxx --partition es1 --qos=es_debug
--time=0:30:0 --constraint=es1_1080ti --gres=gpu:4
--ntasks=8 --nodes=1 --pty bash
```

Outline

- LBNL Supercluster overview
- Software module farm
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

Data Transfer Services

- Dedicated data transfer server (DTN): lrc-xfer.lbl.gov

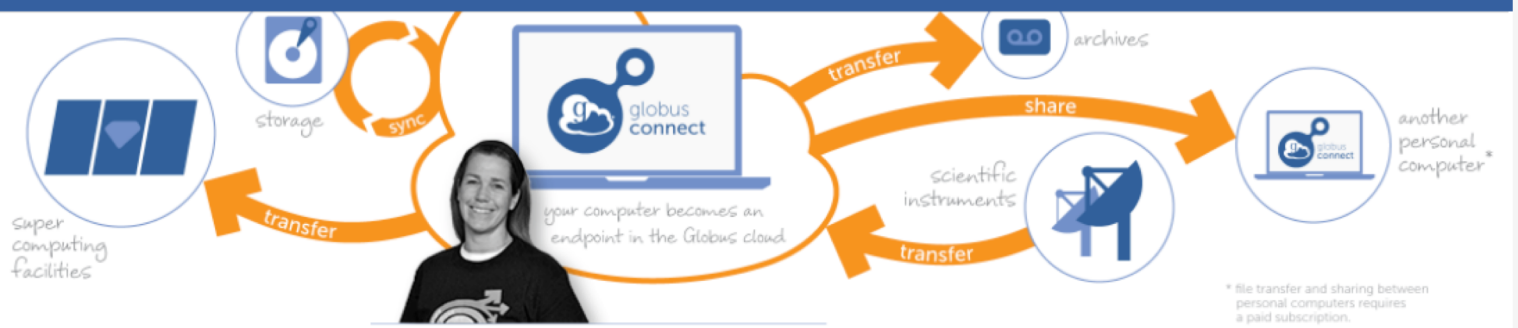
```
scp  -r  your/source/file  uid@lrc-xder.lbl.gov:/cluster/path
rsync -avzh   your/source/file  uid@lrc-xfer.lbl.gov:/cluster/path
```

- Globus Online provide secured unified interface for data transfer: endpoint lbn#lrc + Globus Connect

https://commons.lbl.gov/display/itdivision/Globus+for+Data+Transfers

Globus Connect

Globus Connect Personal



Globus Connect Personal turns your laptop or other personal computer into a Globus endpoint with just a few clicks. With Globus Connect Personal you can share and transfer files to/from a local machine—campus server, desktop computer or laptop—even if it's behind a firewall and you don't have administrator privileges.

Globus Connect Personal puts the power of Globus on your computer.

- Dramatically increases data transfer speeds over scp and other transfer tools.
- Automatically suspends transfers when computer sleeps and resumes when turned on.
- Installs in seconds using native operating system install packages.
- Works with firewalls that block incoming connections, and behind most NATs.
- Uses proven Globus infrastructure for security and authentication.

Globus Connect Personal is available for all major operating systems. Please click on the links below for installation instructions.

Install Globus Connect Personal

Globus Connect Personal for Mac for Mac OS X 10.7 or higher (Intel only)

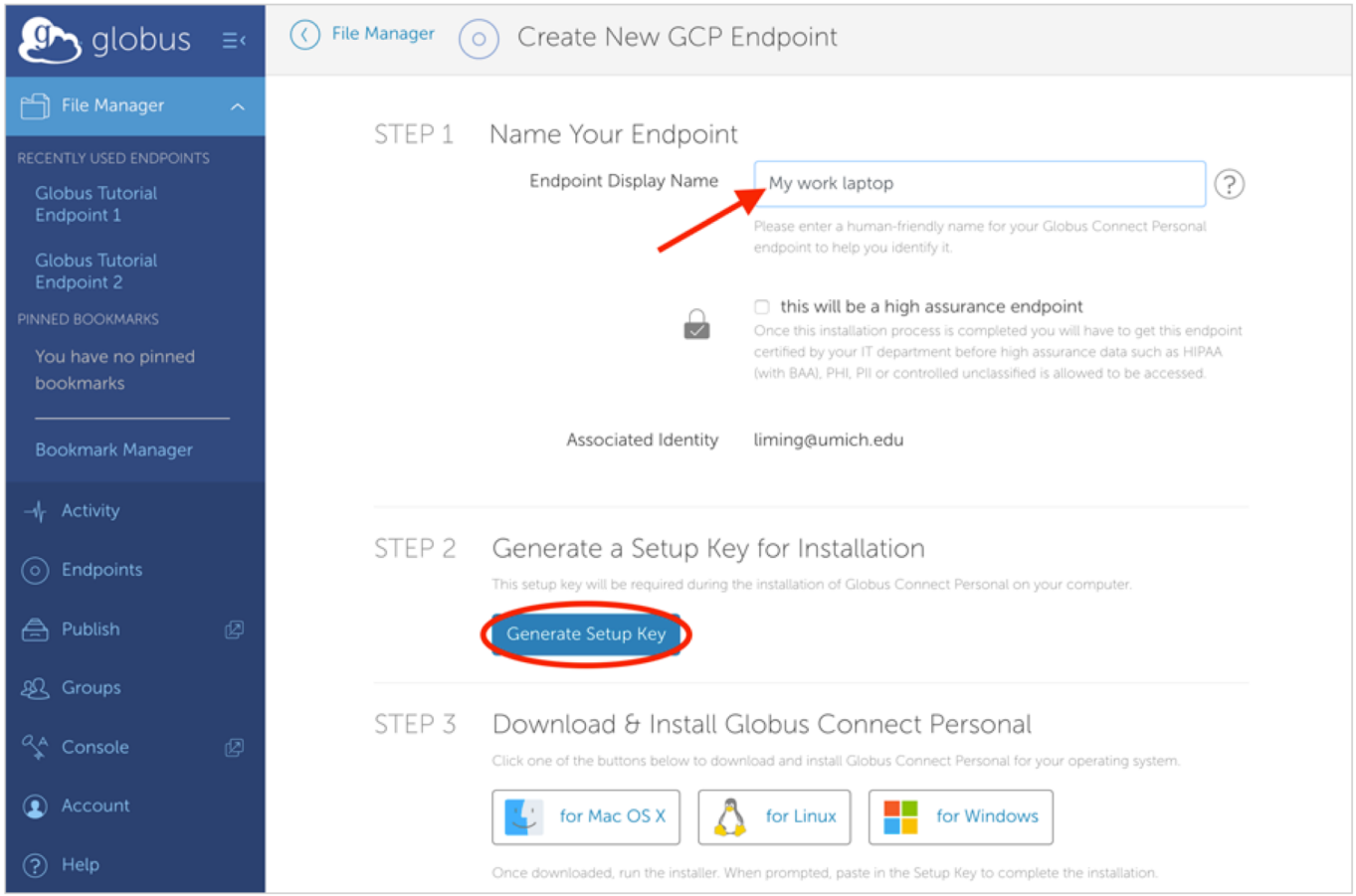
Globus Connect Personal for Linux for common x86-based distributions

Globus Connect Personal for Windows for recent Windows versions

https://www.globus.org/globus-connect-personal

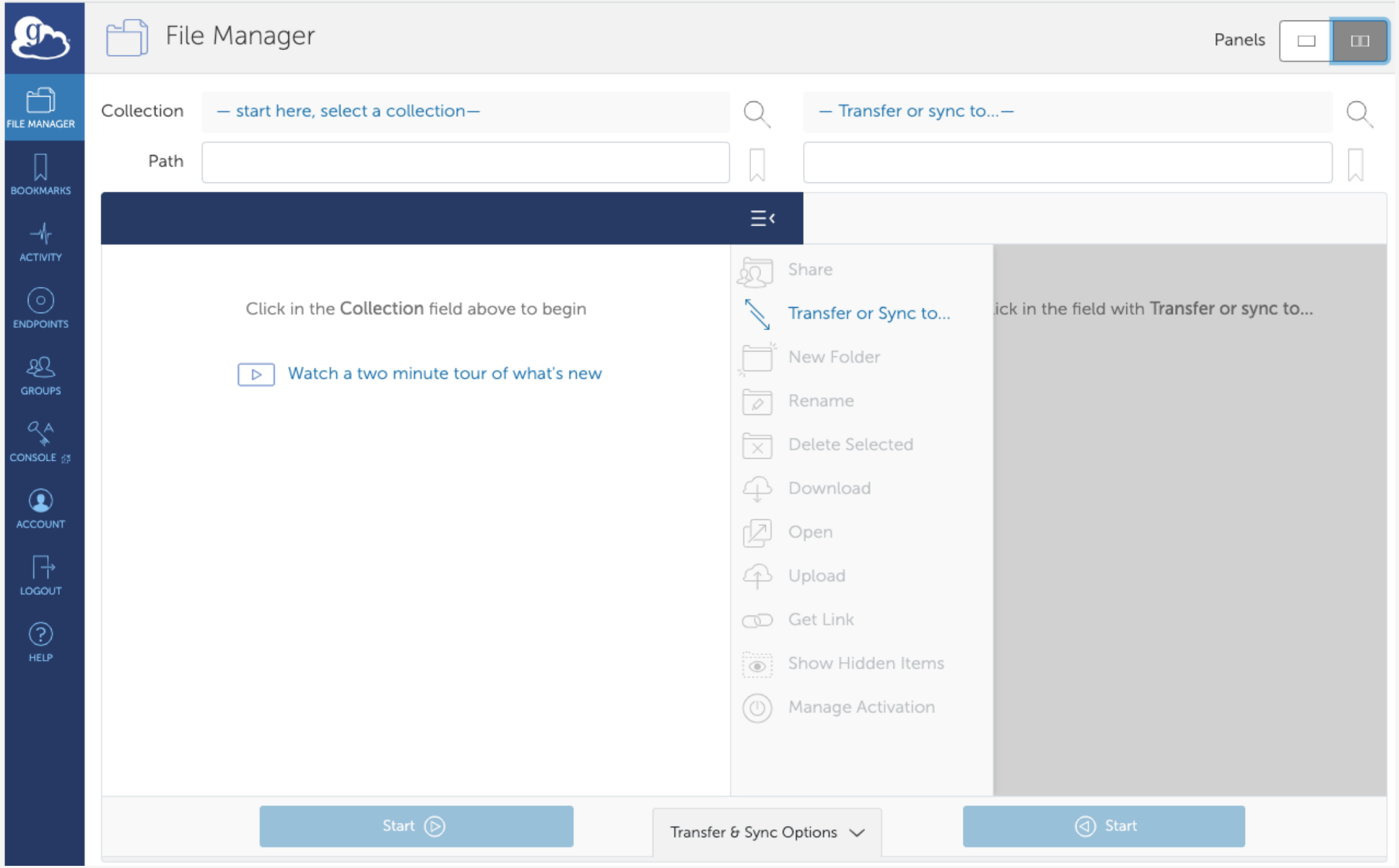
Setup Globus personal end point

1. Type a name for your endpoint. This name can be anything you choose, as long as you haven't previously created another endpoint with the same name.
2. Click "Generate Setup Key".

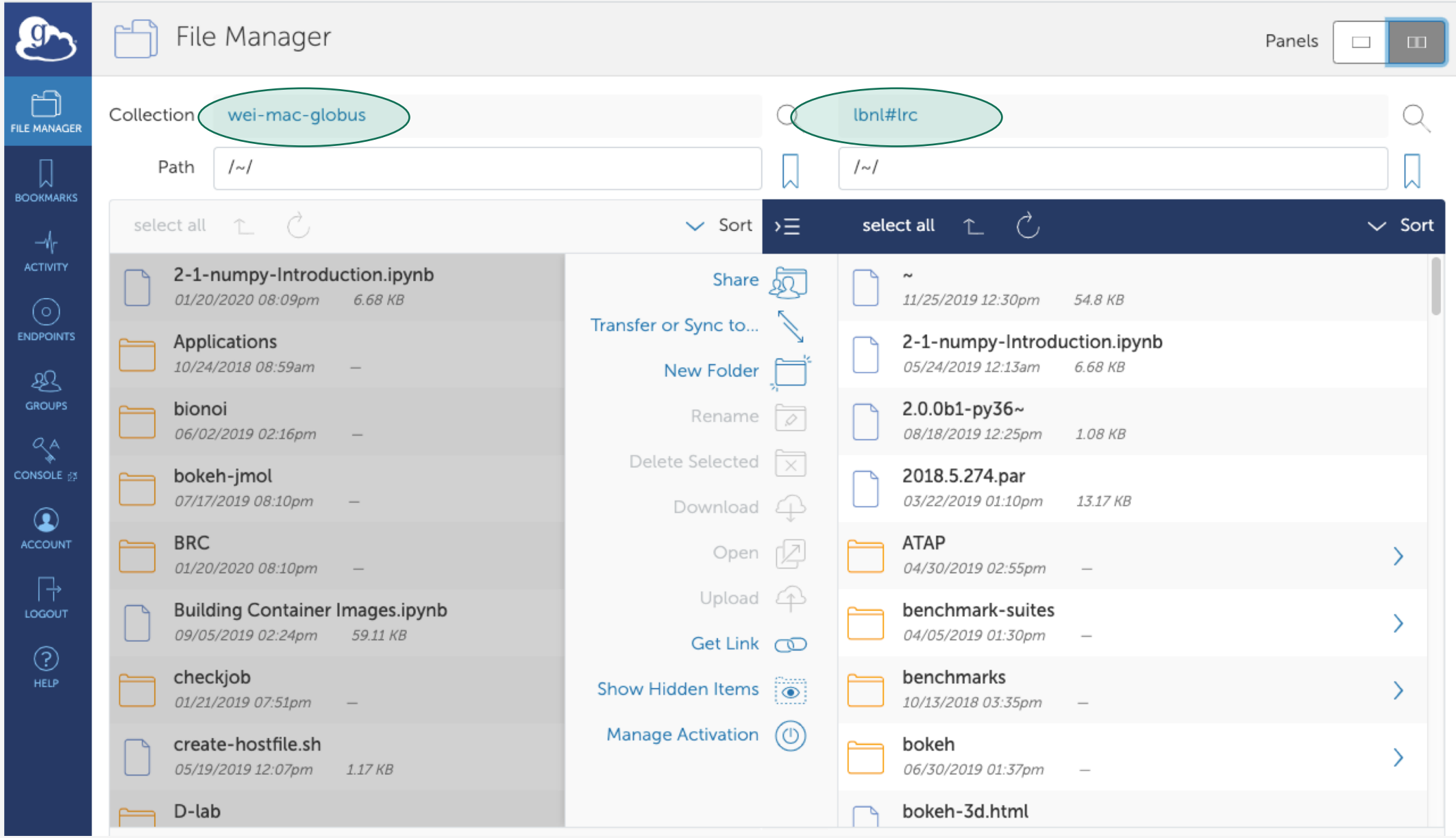


3. Click the clipboard icon next to the generated setup key to copy it to the clipboard. (Alternatively, you may manually select the key and copy it to the clipboard.)
4. Click "for Mac OS X" to download the Mac OS X installer.

Pairing Globus end points



Globus personal end point + lbnl#lrc



Outline

- LBNL Supercluster overview
- Software module farm
- Job scheduler SLURM
- Job submission
- Data transfer
- Jupyter Notebooks
- Remote desktop/visualization
- Resources

Jupyter Notebook

- Jupyterhub: start a Jupyter notebook via https://lrc-jupyter.lbl.gov
- Traditional way of launching a Jupyter notebook
  - All compute nodes are behind the LBL firewall
  - SSH tunneling is not allowed
- Alternative: visualization node (viz) to interactively launch a Jupyter notebook

Remote desktop

- Allow users to run a real desktop within the cluster environment
- Allow applications with a GUI, commercial applications, debugger or visualization applications to render results.
- Allows users to disconnect/resume from anywhere without losing the work.
- RealVNC is provided as the remote desktop service, steps:
  - Login to viz node (lrc-viz.lbl.gov)
  - Start VNC service on viz node
  - Start applications: Firefox, Jupyter notebooks, paraview ...
- Shut it down properly to release resource for other users (logout). Simply close the VNC Viewer does not clean the resource on the server

```
[wfeinstein@viz ~]$  vncserver
VNC(R) Server 6.2.0 (r29523) x64 (Aug 3 2017 17:27:11)
Copyright (C) 2002-2017 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protected by UK patent 2481870; US patent 8760366; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgements see:
https://www.realvnc.com/docs/6/foss.html

\On some distributions (in particular Red Hat), you may get a better experience
by running vncserver-virtual in conjunction with the system Xorg server, rather
than the old version built-in to Xvnc. More desktop environments and
applications will likely be compatible. For more information on this alternative
implementation, please see: https://www.realvnc.com/doclink/kb-546

Running applications in /etc/vnc/xstartup.custom

VNC Server catchphrase: "Ladder happy Belgium. Janet gravity data."
             signature: 6f-12-97-0a-b0-62-0c-d1

Log file is /global/home/users/wfeinstein/.vnc/viz.scs00:12.log
New desktop is viz.scs00:12 (128.3.7.35:12)
```

Launch a Jupyter Notebook via VIZ node

1. Request an interactive node
   - srun -N 1 -p lr4 -A xx -t 1:0:0 -q lr_normal --pty bash
   - Load modules
     - $ module load python/3.6 or ml/tensorflow/1.12.0-py36
     - $ start_jupyter.py

       Your Jupyter Notebook instance is online at:
       http://n0013.lr4:8889/?token=499f9a35ed32d223f43f36a210e210a737e6362915f54033
2. Start a VNC session on viz node
   - vncserver
     - New desktop is viz.scs00 (128.3.7.35)

3. Start VNC Viewer on your laptop and connect the new desktop IP
   - firefox
   - Start a Jypyter notebook by pasting URL above

More

- Parallel computing
- Embarrassing parallel computing
- Python
- Jupyter notebook
- Container technology
- Slurm
- …

Getting help

- Office Hours:
  - Time: 10:30am - noon (Wednesdays)
  - Location: 50B-3209
- Sending us tickets at hpcshelp@lbl.gov
- More information, documents, tips of how to use LBNL Supercluster http://scs.lbl.gov/
- DLab consulting: https://dlab.berkeley.edu/consulting

Q&A