Getting Started at NERSC

Richard Gerber

NERSC User Services

September 13, 2011

Berkeley Lab Oakland Scientific Facility

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

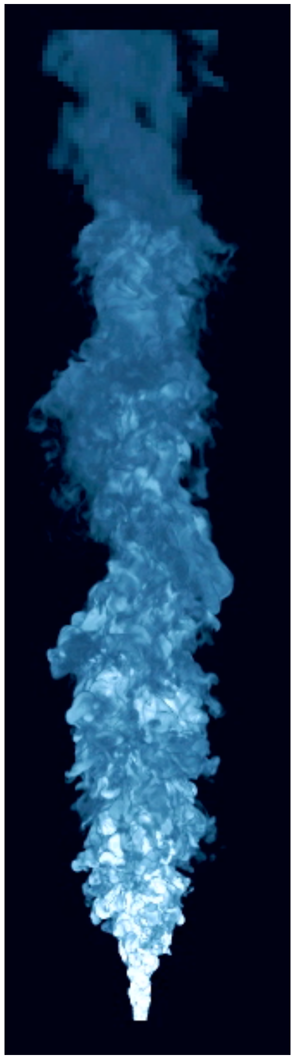
BERKELEY LAB Lawrence Berkeley National Laboratory

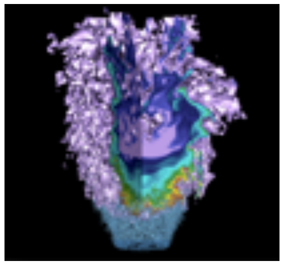
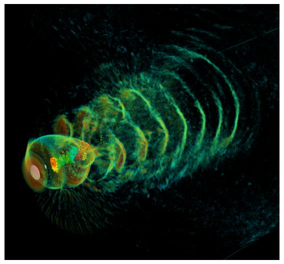
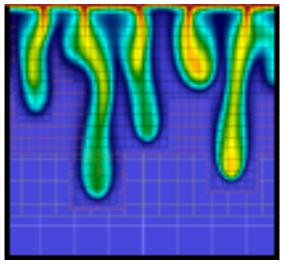
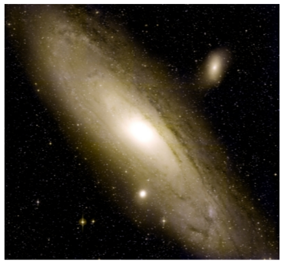
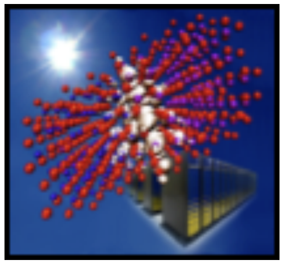
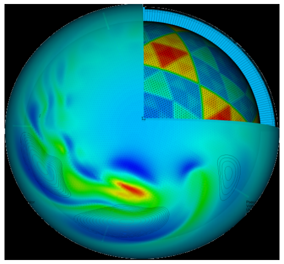
Purpose

- **This presentation will help you use NERSC and its facilities**
  - Practical information
  - Introduction to terms
  - Help you get things done efficiently
- **This is not a programming tutorial**
  - But you will learn how to get help and what kind of help is available
  - We can give presentations on programming languages and parallel libraries – just ask

Outline

- What is NERSC?
- Computing Resources
- How to Get Help
- Accounts and Allocations
- Connecting to NERSC
- Storage Resources
- Computing Environment
- Compiling Code
- Running Jobs



What is NERSC?

National Energy Research Scientific Computing Center

NERSC Facility Leads DOE in Scientific Computing Productivity

NERSC computing for science

- 4000 users, 500 projects
- From 48 states; 65% from universities
- Hundreds of users each day
- ***1500 publications per year***

Systems designed for science

- 1.3PF Petaflop Cray system, Hopper
  - 2nd Fastest computer in US
  - Fastest open Cray XE6 system
  - Additional .5 PF in Franklin system and smaller clusters

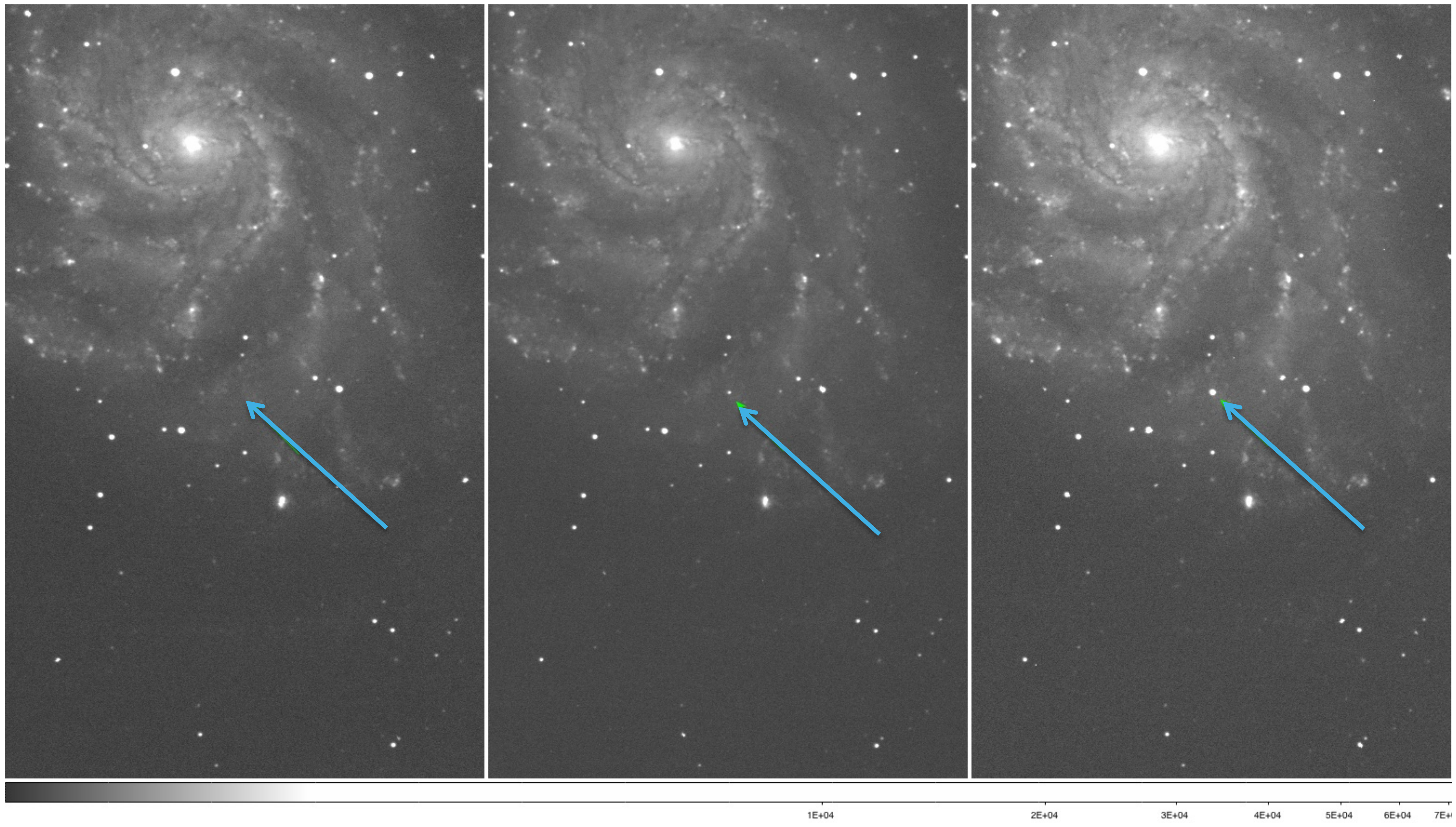
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

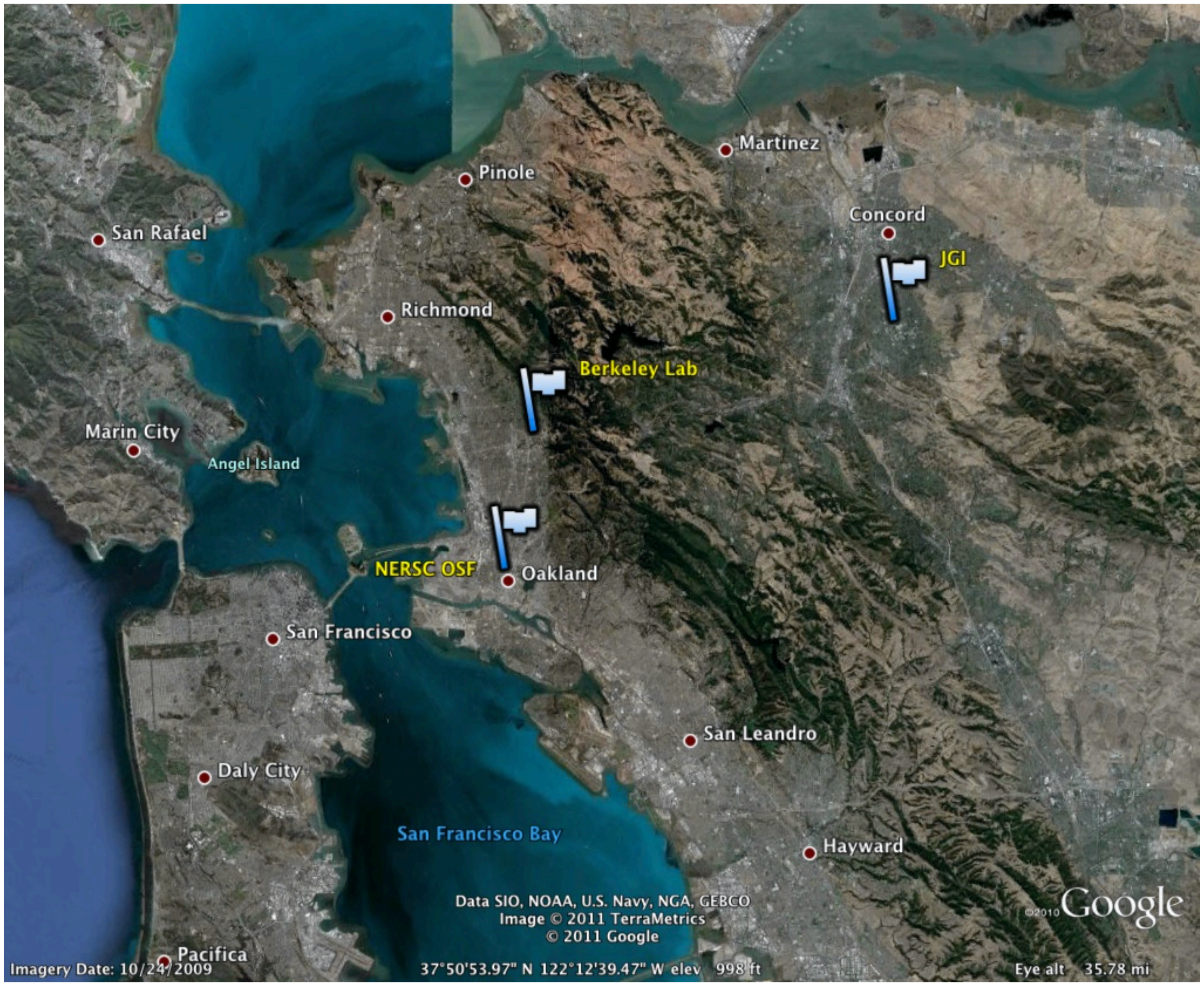
Science at NERSC

Supernova Caught in the Act

Earliest-ever Detection Made Possible by Computing, Networks



Location



BERKELEY LAB
Lawrence Berkeley National Laboratory

NERSC is a DOE Office of Science National Center located at Berkley Lab

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

DOE Office of Advanced Scientific Computing Facilities

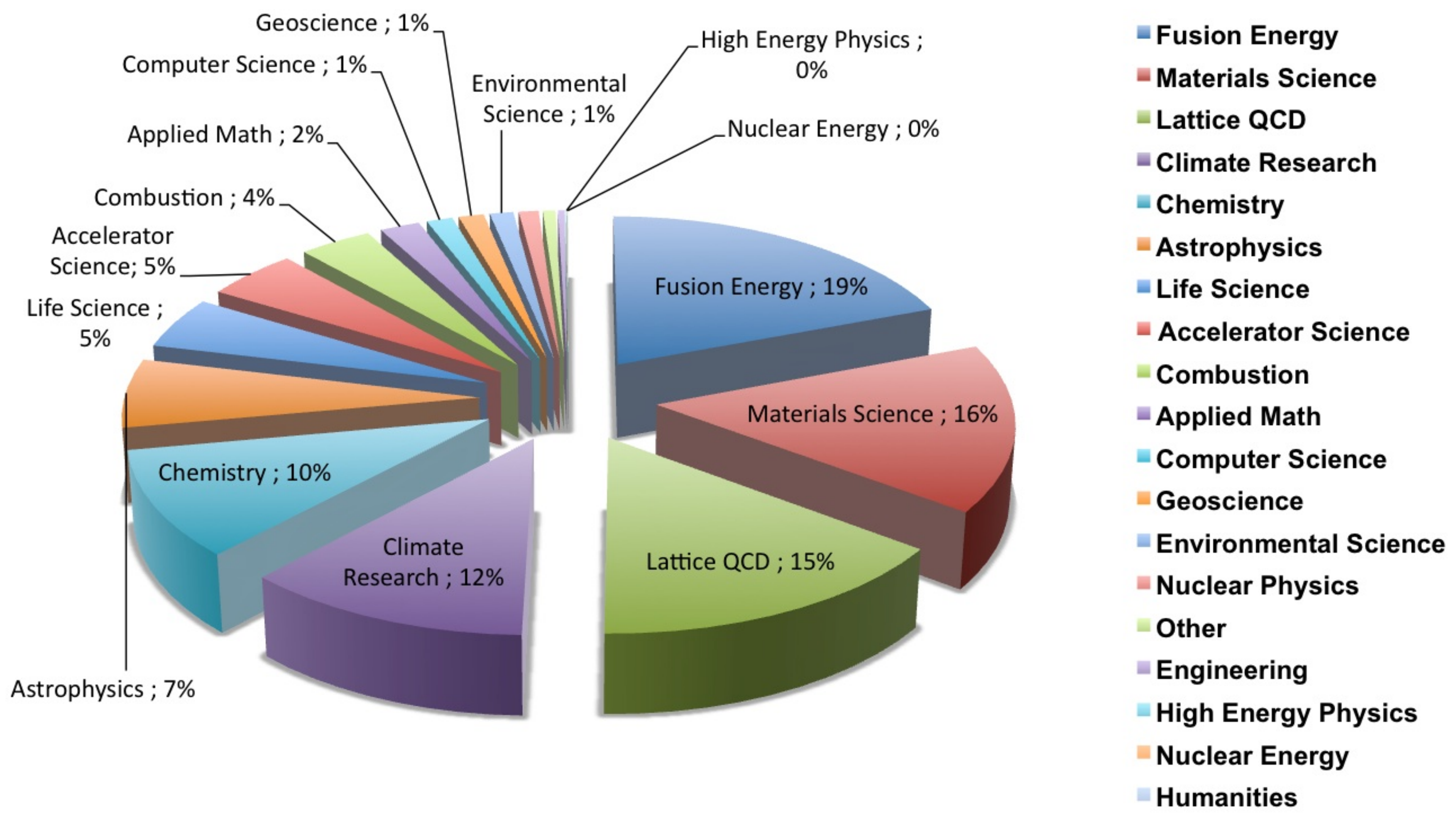
NERSC at LBNL

- **1000s users,100s projects**
- **Allocations:**
  - 80% DOE program managers
  - 10% ASCR Leadership Computing Challenge
  - 10% NERSC reserve
- **Science includes all of DOE Office of Science**
- **Machines procured competitively**

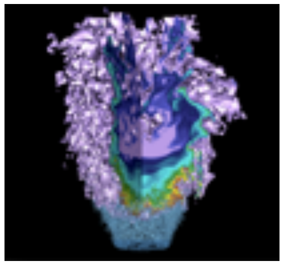
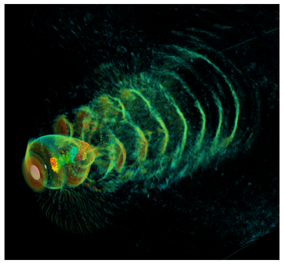
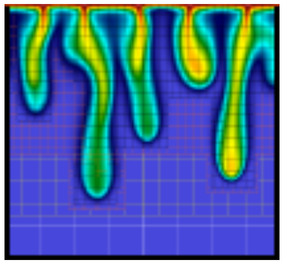
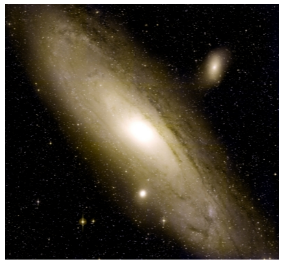
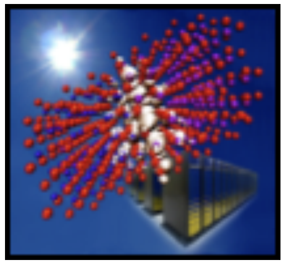
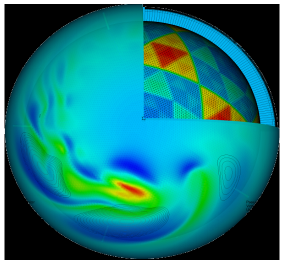
"Leadership Facilities" at Oak Ridge & Argonne

- **100s users 10s projects**
- **Allocations:**
  - 60% ANL/ORNL managed INCITE process
  - 30% ACSR Leadership Computing Challenge*
  - 10% LCF reserve
- **Science limited to largest scale; no commitment to DOE/SC offices**
- **Machines procured through partnerships**

NERSC Workload



NERSC 2011 Allocations By Science Area

Computing Resources

NERSC Systems

Large-Scale Computing Systems

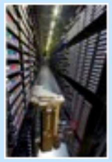
Hopper (NERSC-6): Cray XE6
6,384 compute nodes, 153,216 cores
110 Tflop/s on applications; 1.27 Pflop/s peak

Franklin (NERSC-5): Cray XT4
9,532 compute nodes; 38,128 cores
~25 Tflop/s on applications; 356 Tflop/s peak

Clusters
140 Tflops total
Carver
IBM iDataplex cluster
PDSF (HEP/NP)
~1K core cluster
Magellan Cloud testbed
IBM iDataplex cluster
GenePool (JGI)
~5K core cluster

NERSC Global File system (NGF)
Uses IBM's GPFS
1.5 PB capacity
5.5 GB/s of bandwidth

HPSS Archival Storage
40 PB capacity
4 Tape libraries
150 TB disk cache



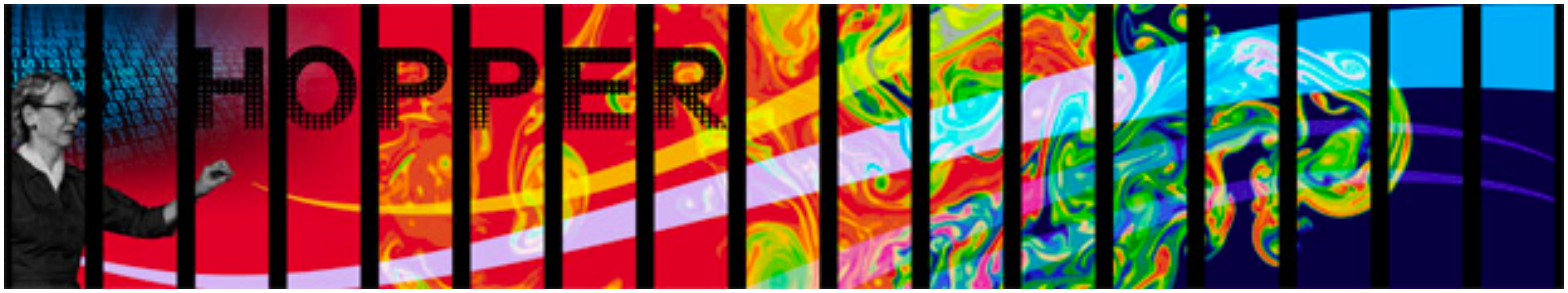
Analytics

Euclid
512 GB shared mem
Dirac
GPU testbed
48 nodes

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

Hopper - Cray XE6



- 1.2 GB memory / core (2.5 GB / core on "fat" nodes) for applications
- /scratch disk quota of 5 TB
- 2 PB of /scratch disk
- Choice of full Linux operating system or optimized Linux OS (Cray Linux)
- PGI, Cray, Pathscale, GNU, Intel compilers

- 153,408 cores, 6,392 nodes
- "Gemini" interconnect
- 2 12-core AMD 'MagnyCours' 2.1 GHz processors per node
- 24 processor cores per node
- 32 GB of memory per node (384 "fat" nodes with 64 GB)
- 216 TB of aggregate memory

Use Hopper for your biggest, most computationally challenging problems.

Franklin - Cray XT4

38,288 compute cores

9,572 compute nodes

One quad-core AMD 2.3 GHz Opteron processors (Budapest) per node

4 processor cores per node

8 GB of memory per node

78 TB of aggregate memory

1.8 GB memory / core for applications

/scratch disk default quota of 750 GB

Light-weight Cray Linux operating system

No runtime dynamic, shared-object libs

PGI, Cray, Pathscale, GNU compilers

Use Franklin for all your computing jobs, except those that need a full Linux operating system.

Carver - IBM iDataPlex

3,200 compute cores

400 compute nodes

2 quad-core Intel Nehalem 2.67 GHz processors per node

8 processor cores per node

24 GB of memory per node (48 GB on 80 "fat" nodes)

2.5 GB / core for applications (5.5 GB / core on "fat" nodes)

InfiniBand 4X QDR

NERSC global /scratch directory quota of 20 TB

Full Linux operating system

PGI, GNU, Intel compilers

Use Carver for jobs that use up to 512 cores, need a fast CPU, need a standard Linux configuration, or need up to 48 GB of memory on a node.

Dirac – GPU Computing Testbed

50 GPUs

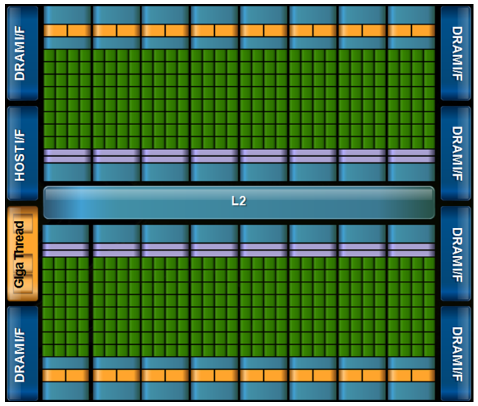
50 compute nodes

- 2 quad-core Intel Nehalem 2.67 GHz processors
- 24 GB DRAM memory

44 nodes: 1 NVIDIA Tesla C2050 (Fermi) GPU with 3GB of memory and 448 cores

1 node: 4 NVIDIA Tesla C2050 (Fermi) GPU's, each with 3GB of memory and 448 processor cores.

InfiniBand 4X QDR



CUDA 4.0, OpenCL, PGI and HMPP directives

DDT CUDA-enabled debugger

PGI, GNU, Intel compilers

Use Dirac for developing and testing GPU codes.

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

Hands-On

- **SSH to hopper.nersc.gov**
  - UNIX/Mac: ssh –Y –A hopper.nersc.gov
  - PC/Other: use your SSH application
- **On Hopper:**

```
% module load training
% cd $SCRATCH
% cp -rp $EXAMPLES .
% cd NewUser
% cd flip
```

Hands-On

- On Hopper:

Fortran Compiler

C Compiler

Submit

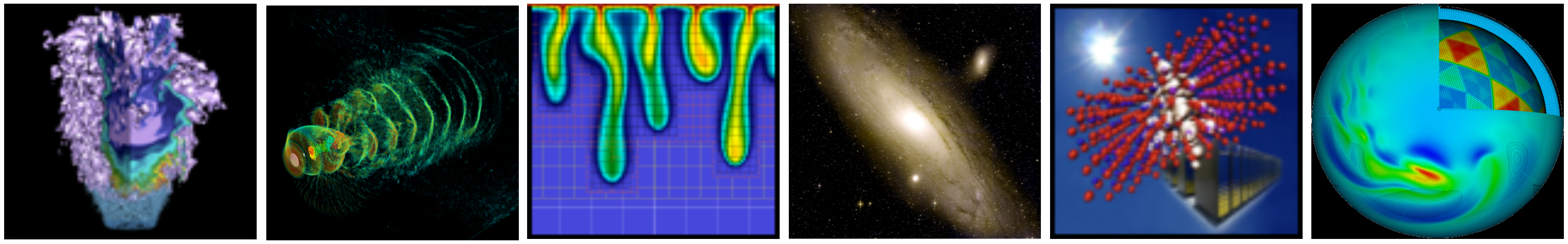
Check Status

```
% make
ftn  -c  flip.f90
cc   -c -o printit.o printit.c
ftn  -o flip  flip.o printit.o
% qsub flip.pbs
% qstat –u <your_user_name>
```

```
Job ID     Username Queue   Jobname        SessID NDS  TSK Memory Time  S Time
801493.sdb ragerber debug   flip              --    --   --     -- 00:10 Q   --
```

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory



How to Get Help

NERSC Services

- **NERSC's emphasis is on enabling scientific discovery**
- **User-oriented systems and services sets NERSC apart from other centers**
- **Help Desk / Consulting**
  - Immediate direct access to consulting staff that includes 7 Ph.Ds
- **User group (NUG) has tremendous influence**
  - Monthly teleconferences & yearly meetings
  - Open to all users
- **Requirement-gathering workshops with top scientists**
  - One each for the six DOE Program Offices in the Office of Science
  - http://www.nersc.gov/science/requirements-workshops/
- **Ask, and we'll do whatever we can to fulfill your request**

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

How to Get Help

http://www.nersc.gov/

1-800-666-3772 (or 1-510-486-8600)

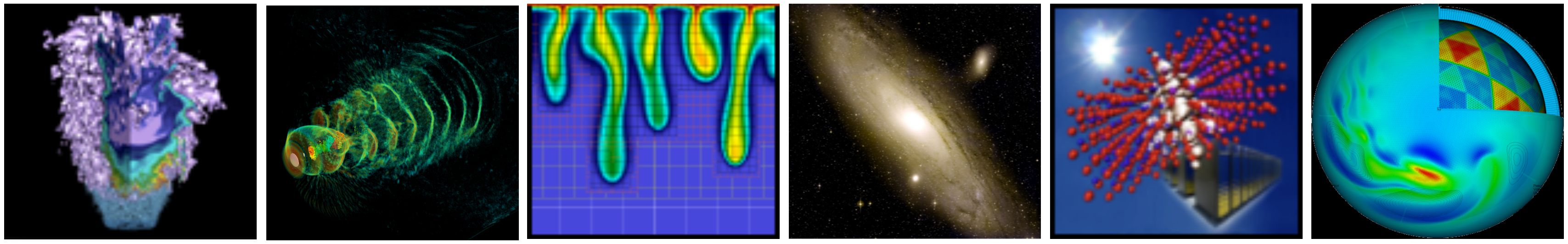
Computer Operations* = menu option 1 (24/7)

Account Support (passwords) = menu option 2,
accounts@nersc.gov

HPC Consulting = menu option 3, or consult@nersc.gov
(8-5, M-F Pacific time)

Online Help Desk = https://help.nersc.gov/

* Passwords during non-business hours



Accounts & Allocations

Accounts

There are two types of "accounts" at NERSC. It is important to differentiate between them.

1. Your personal, private account
   - Associated with your "login" or "user name”
   - Identifies you to our systems and is used when logging into NERSC systems and web services.
   - Your PI requests an account for you.
2. An allocation account, or "repository” (aka “repo”)
   - Like a bank account you use to "pay" for computer time.
   - PIs request allocations of time and/or storage
   - An individual user may belong to one or many repositories.

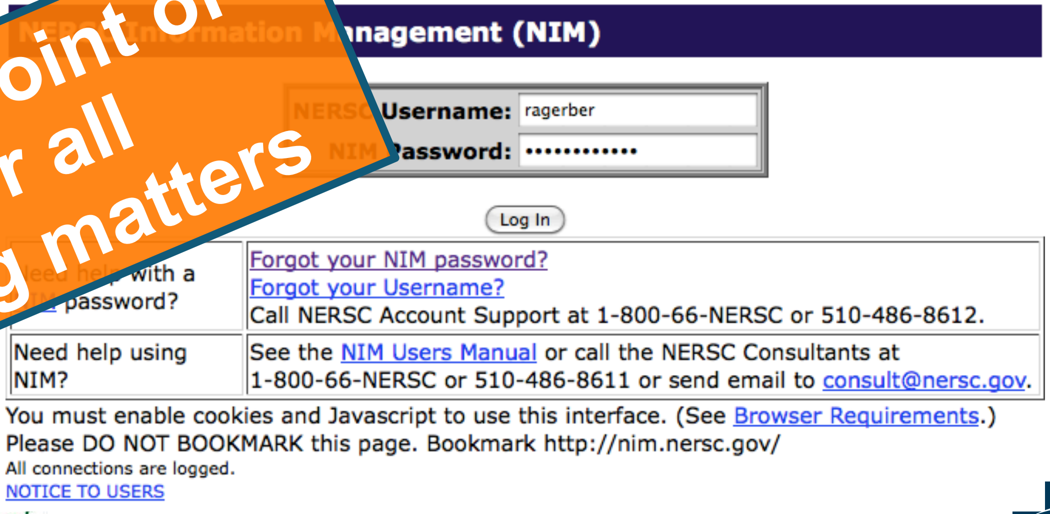
To apply for either type of account, see the NERSC web site at http://www.nersc.gov/.

Allocations

- **Computer time and storage allocations are awarded by DOE**
- **Project PIs apply through the ERCAP process**
  - ERCAP proposals for 2012 due Sep. 23, 2011
- **Most allocations are awarded in the fall**
  - Allocation year starts in January
  - Small startup allocations are awarded throughout the year
  - Additional time available through NISE and ALCC
- **You must have access to an allocation of time to run jobs at NERSC (be a member of a "repo")**
- **If your repo runs out of time, you can request more through your project's DOE program manager who handles NERSC allocations (list available on NERSC web site)**

Accounting Web Interface (NIM)

- Log into the NERSC NIM web site at https://nim.nersc.gov/ to manage your NERSC accounts.
- In NIM you can check your daily allocation balances, change your password, run reports, update your contact information, change your login shell, etc.



NERSC Information Management (NIM)

NERSC Username: ragerber
NIM Password: ••••••••••••

Log In

Need help with a NIM password?	Forgot your NIM password? Forgot your Username? Call NERSC Account Support at 1-800-66-NERSC or 510-486-8612.
Need help using NIM?	See the NIM Users Manual or call the NERSC Consultants at 1-800-66-NERSC or 510-486-8611 or send email to consult@nersc.gov.

You must enable cookies and Javascript to use this interface. (See Browser Requirements.)
Please DO NOT BOOKMARK this page. Bookmark http://nim.nersc.gov/
All connections are logged.
NOTICE TO USERS

NIM is the point of truth for all accounting matters

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

Hands-On

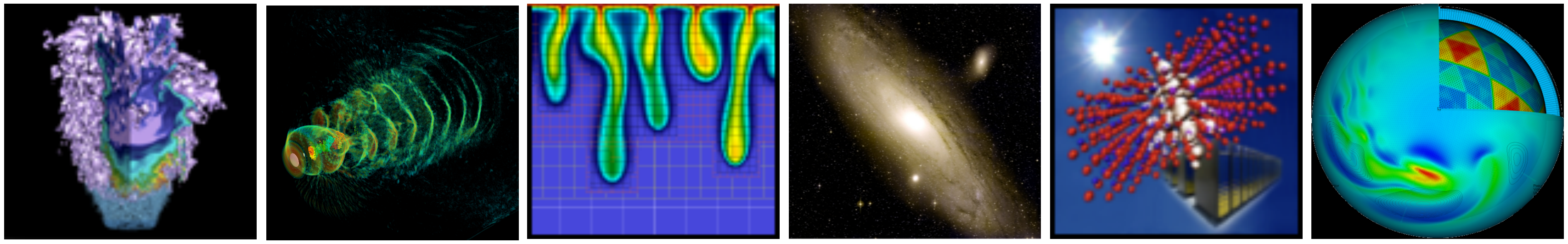
- **Look at the batch script you submitted**

```
% cat flip.pbs
```

- **Note: no charging info**
  - You have a "default repo" that is charged
  - If you have more than one repo, and you want to charge against a secondary repo, use

```
% qsub -A <repo_name> flip.pbs
```

- **Note: charges are applied daily in the wee hours of the morning**
- **If you are out of time, you can't submit new jobs**



Connecting to NERSC

Yushu Yao

(Separate Presentation)

U.S. DEPARTMENT OF ENERGY | Office of Science

National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

What is Globus Online?

- **Reliable file transfer.**
  - Easy “fire and forget” file transfers
  - Automatic fault recovery
  - High performance
  - Across multiple security domains
- **No IT required.**
  - No client software installation
  - New features automatically available
  - Consolidated support and troubleshooting
  - Works with existing GridFTP servers
  - Globus Connect solves “last mile problem”

“Fantastic! I have started using globus connect to transfer data, and it only took me 5 minutes to set up. Thank you!”

– *NERSC user*

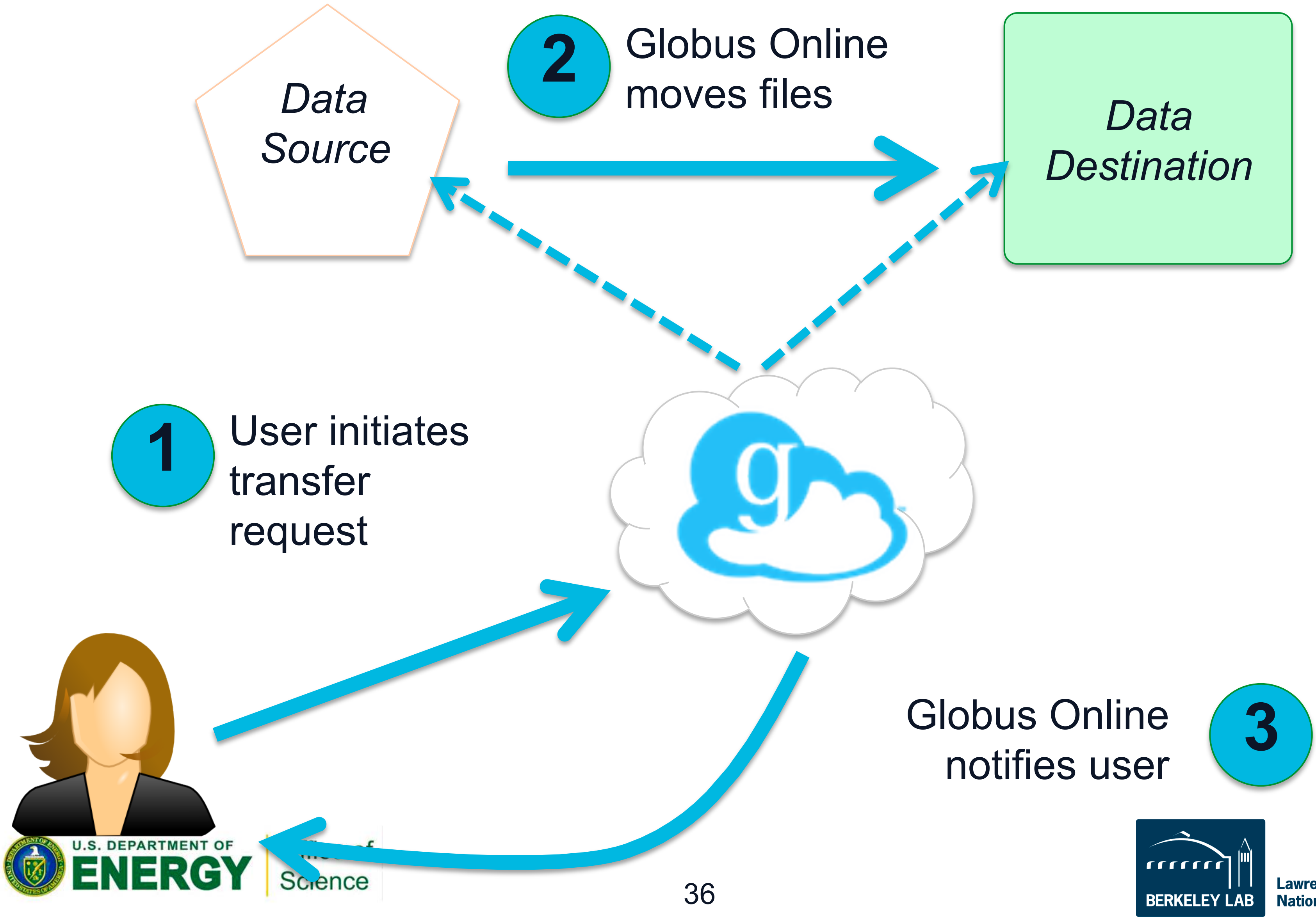
“The service is reliable and easy to use.”

– *National Center for Supercomputing Applications*

“I moved 400 GB of files and didn’t even have to think about it.”

– *Lawrence Berkeley National Lab*

How It Works


Data Source
2
Globus Online moves files
Data Destination
1
User initiates transfer request
Globus Online notifies user
3

File Movement Options

We strive to make Globus Online broadly accessible…

- Researchers with no IT background can just move files using the **Web GUI**
- Developers who want to automate workflows can use the **Command Line Interface (CLI)**
- System builders who don't want to re-engineer file transfer solutions can use the **REST API**

For More Information

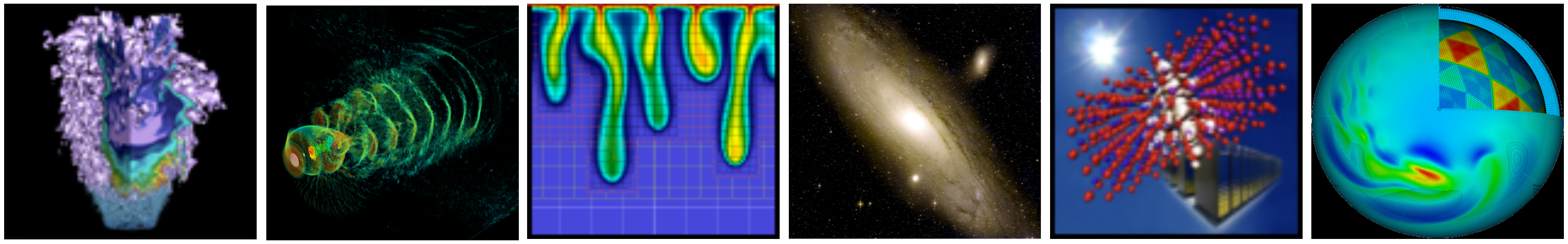
- **Visit https://www.globusonline.org/signup to:**
  - Get a free account and start moving files
- **Visit www.globusonline.org for:**
  - Tutorials
  - FAQs
  - Pro Tips
  - Troubleshooting
- **Contact support@globusonline.org for:**
  - Help getting started
  - Help using the service

Hands-On

- **Check your job's status**

```
% qstat –u <your_username>
(you should see nothing!)
% cat flip.o.<your_jobid>.sdb
```

- **This is the Standout Output (STDOUT) file**
  - Text from write *, print *, printf()
  - Info from the batch system



Data Storage Resources

Data Storage Types

- **“Spinning Disk”**
  - Interactive access
  - I/O from compute jobs
  - “Home”, “Project”, “Scratch”
  - Note: No on-node direct-attach disk at NERSC
- **Archival Storage**
  - Permanent, Long-Term Storage
  - Tapes, fronted by disk cache
  - “HPSS” (High Performance Storage System)

Local vs. Global File Systems

- **“Local” file systems**
  - Accessible only from one host
  - /scratch and /scratch2 on Hopper & Franklin
  - Usually highest performance
- **“Global” file systems**
  - Accessible from many (or all) NERSC systems
  - Home, /project, “global /scratch”
  - Good (& improving) performance

Home Directory

- When you log in to a NERSC computer you are in your "Home" directory.
- Permanent storage
  - No automatic backups
- The full UNIX pathname is stored in the environment variable $HOME

```
hopper04% echo $HOME
/global/homes/r/ragerber
```

- $HOME is a global file system
  - You see all the same directories and files when you log in to any NERSC computer.
- Your quota in $HOME is 40 GB and 1M inodes (files and directories).
  - Use “myquota” command to check your usage and quota

Use $HOME to store source code, small files you want to save permanently

Scratch Directories

- Each system has a large, high-performance ”scratch” file system.
- Each user has a personal directory referenced by $SCRATCH (and maybe $SCRATCH2).
- Data in $SCRATCH is purged (12 weeks from last access)
- Always save data you want to keep to HPSS (see below)
- $SCRATCH is local on Franklin and Hopper, but Carver and future systems use a global scratch file system.
- Data in $SCRATCH is not backed up and could be lost if a file system fails.

Use $SCRATCH for large, high-performance I/O.

Project Directories

- All NERSC systems mount the NERSC global "Project" file system.
- "Project directories" are created upon request for projects (groups of researchers) to store and share data.
- The default quota in /project is 4 TB.
- While data can be written and read from a parallel job on all system, performance ~~will~~ *may* not be as good as on $SCRATCH.
- Data in /project is not purged, but there are no automatic user backups either.

Use /project for sharing data and codes among a group of researchers and/or to share data between NERSC systems.

File System Summary

File System	Path	Type	Default Quota	Backups	Purge Policy
Global Home	$HOME	GPFS	40GB 1,000,000 Inodes	No	Not Purged
Global Scratch	$GSCRATCH	GPFS	20TB 2,000,000 Inodes	No	Files not Accessed for 12 Weeks are deleted
Global Project	/project/projectdirs/projectname	GPFS	4TB 4,000,000 Inodes	No	Not Purged
Hopper Local Scratch	$SCRATCH and $SCRATCH2	Lustre	5TB 5,000,000 Inodes (Combined)	No	Coming Soon
Franklin Local Scratch	$SCRATCH and $SCRATCH2	Lustre	750GB 1,000,000 Inodes (Combined)	No	Files not Accessed for 12 Weeks are deleted
HPSS	See HPSS Pages	TAPE	See HPSS Pages	No	Not Purged

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

IO Tips

- Use $SCRATCH for good IO performance from a production compute job
- Write large chunks of data (MBs or more) at a time from your code
- Use a parallel IO library (e.g. HDF5)
- Read/write to as few files as practical from your code (try to avoid 1 file per MPI task)
- Use $HOME to compile unless you have too many source files or intermediate (*.o) files
- Do not put more than a few 1,000s of files in a single directory
- Save any and everything important to HPSS

Navigating NERSC File Systems

A NERSC Training Event

NERSC Oakland Facility & Web Broadcast

http://www.nersc.gov/users/training/events/nersc-file-systems/

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Archival Storage (HPSS)

- For permanent, archival storage
- Front-ending the tape subsystem is 150TB fast-access disk
- Permanent storage is magnetic tape, disk cache is transient
  - 15PB data in 100M files written to 26k cartridges
  - Cartridges are loaded/ unloaded into tape drives by sophisticated library robotics

Hostname: archive.nersc.gov

Data increasing by 1.7X per year

120 M files stored

44,000 tape slots

44 PB maximum capacity today

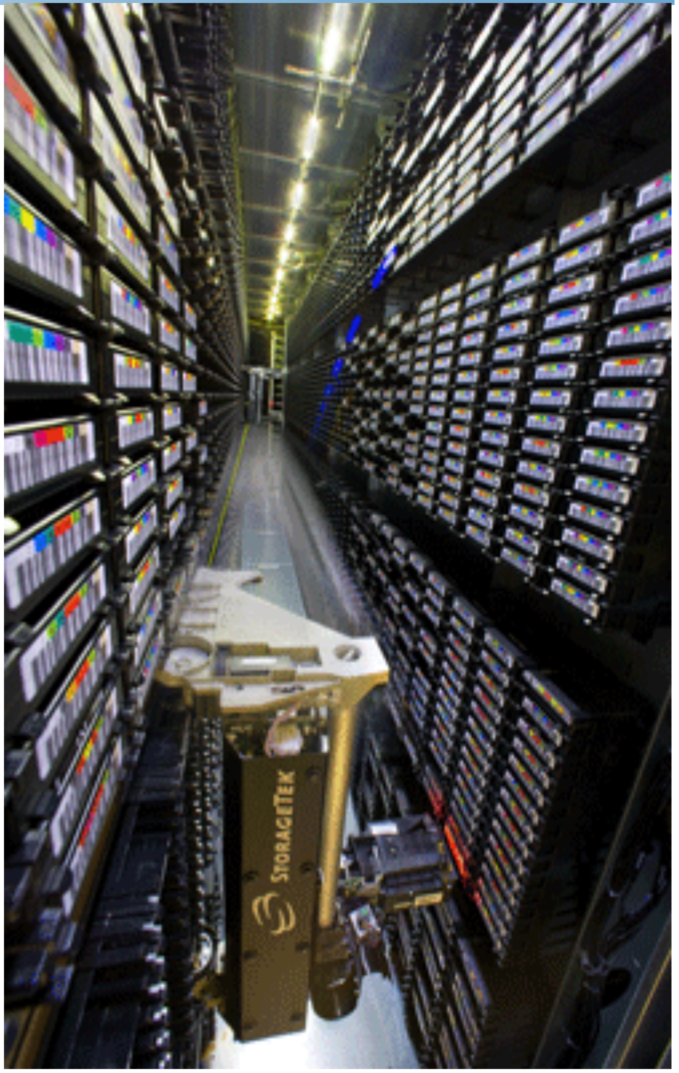
Average data xfer rate: 100 MB/sec

Authentication

- **NERSC storage uses a token-based authentication method for HPSS; no ssh access**
  - User places encrypted authentication token in ~/.netrc file at the top level of the home directory on the compute platform
- **Authentication tokens can be generated in 2 ways:**
  - Automatic – NERSC auth service:
    - Log into any NERSC compute platform; Type “hsi”; Enter NERSC password
  - Manual – https://nim.nersc.gov/ website
    - Under “Actions” dropdown, select “Generate HPSS Token”; Copy/paste content into ~/.netrc; chmod 600 ~/.netrc
- **Tokens are username and IP specific—must use NIM to generate a different token for use offsite**

HPSS Clients

- **Parallel, threaded, high performance:**
  - HSI
    - Unix shell-like interface
  - HTAR
    - Like Unix tar, for aggregation of small files
  - PFTP
    - Parallel FTP
- **Non-parallel:**
  - FTP
    - Ubiquitous, many free scripting utilities
- **GridFTP interface (garchive)**
  - Connect to other grid-enabled storage systems



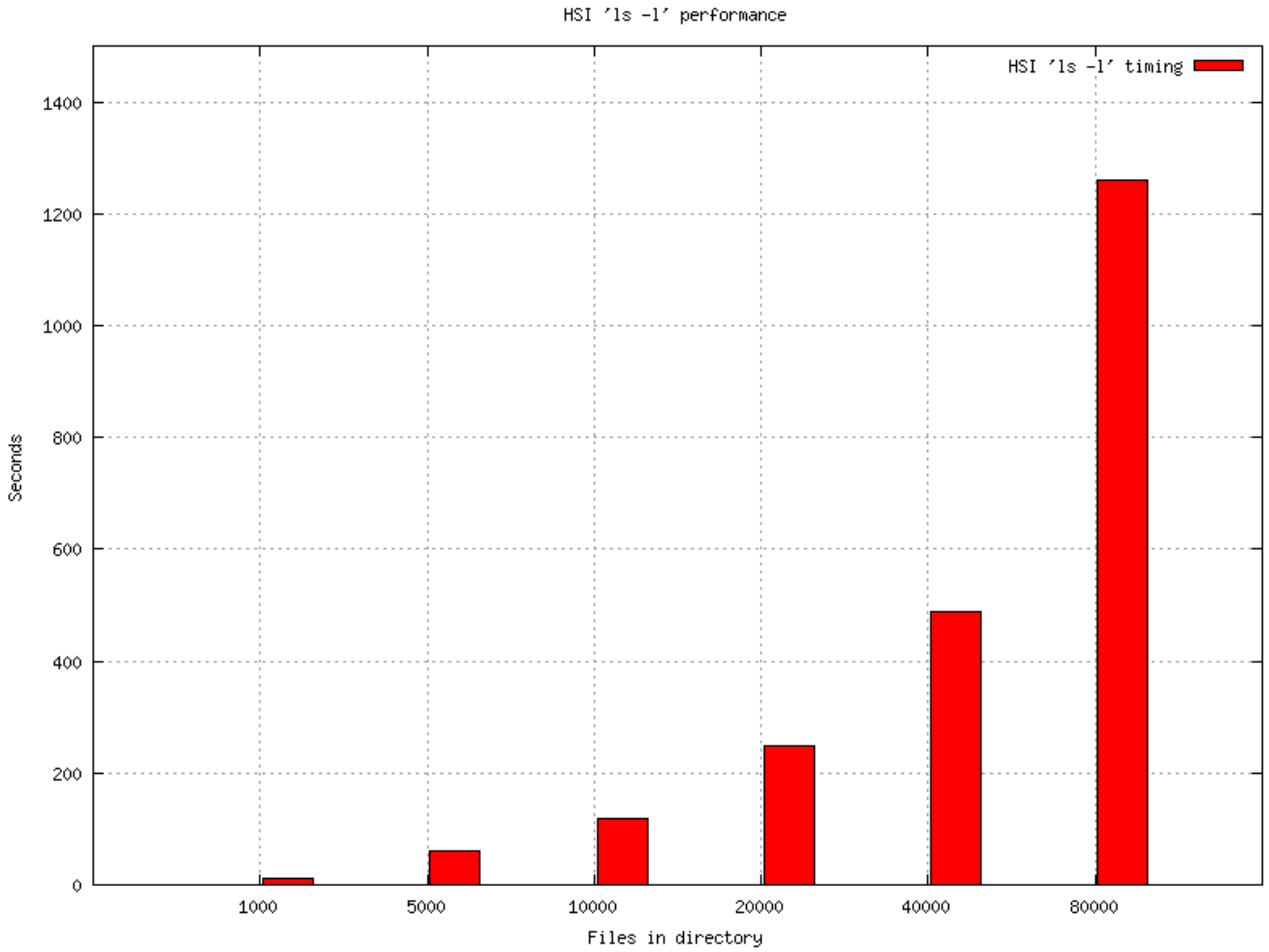
Archive Technologies, Continued…

- **HPSS clients can emulate file system qualities**
  - FTP-like interfaces can be deceiving: the archive is backed by tape, robotics, and a single SQL database instance for metadata
  - Operations that would be slow on a file system, e.g. lots of random IO, can be impractical on the archive
  - It's important to know how to store and retrieve data efficiently

- **HPSS does not stop you from making mistakes**
  - It is possible to store data in such a way as to make it difficult to retrieve
  - The archive has no batch system. Inefficient use affects others.

Avoid Common Mistakes

- **Don't store many small files**
  - Make a tar archive first, or use htar
- **Don't use recursively store or retrieve large directory trees**
- **Don't stream data via UNIX pipes**
  - HPSS can't optimize transfers of unknown size
- **Don't pre-stage data to disk cache**
  - May evict efficiently stored existing cache data
- **Avoid directories with many files**
  - Stresses HPSS database
- **Long-running transfers**
  - Can be error-prone
  - Keep to under 24 hours
- **Use as few concurrent sessions as required**
  - Limit of 15 in place


HSI 'ls -l' performance
HSI 'ls -l' timing
Seconds
0
200
400
600
800
1000
1200
1400
1000
5000
10000
20000
40000
80000
Files in directory

Hands-On

- Use htar to save a directory tree to HPSS

```
%cd $SCRATCH (or where you put the NewUser directory)

%htar -cvf NewUser.tar NewUser
```

- See if the command worked

```
%hsi ls -l NewUser.tar
-rw-r-----   1 ragerber  ccc   6232064 Sep 12 16:46 NewUser.tar
```

Data Transfer and Archiving

A NERSC Training Event

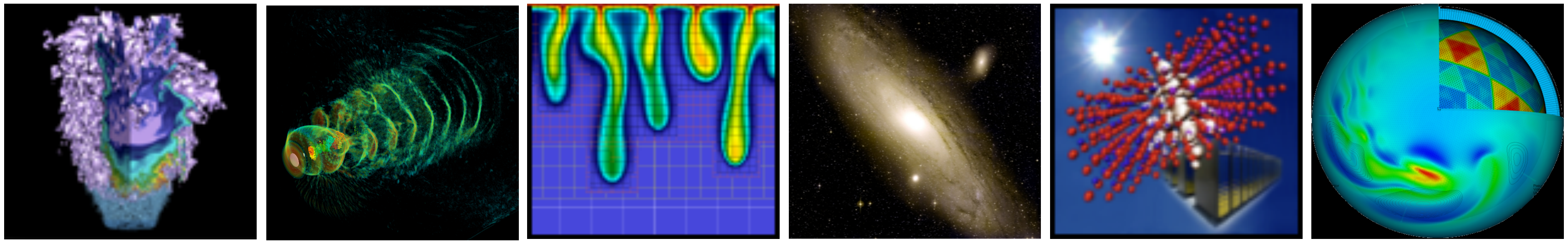
NERSC Oakland Facility & Web Broadcast

http://www.nersc.gov/users/training/events/data-transfer-and-archiving/

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory



Computing Environment

Shell Initialization Files

- NERSC installs dot-files in your home directory (e.g. .login, .profile)
  - Commands in dot-files are executed when you log in (or start a new shell)
  - Do not modify these or your jobs and compiles will not work correctly.
- Each dot-file sources an additional file with the same name, but with an .ext extension.
  - Put your local modifications in these .ext files (e.g. .login.ext, .profile.ext)

Modules

- Easy access to NERSC's extensive software collection is controlled by the modules utility.
- With modules, you manipulate your computing environment to use applications and programming libraries.
- In many cases, you can ignore modules because NERSC has already loaded a rich set of modules for you when you first log in.
- If you want to change that environment you "load," "unload," and "swap" modules.
- A small set of module commands can do most of what you'll want to do

module list

- Shows you your currently loaded modules.
- When you first log in, you have a number of modules loaded for you. Here is an example from Hopper.

```
hopper03% module list
Currently Loaded Modulefiles:
  1) modules/3.2.6.6                          11) xpmem/0.1-2.0301.25333.20.2.gem
  2) xtpe-network-gemini                      12) xe-sysroot/3.1.61
  3) pgi/10.9.0                               13) xt-asyncpe/4.9
  4) xt-libsci/10.5.01                        14) atp/1.1.2
  5) xt-mpich2/5.2.1                          15) PrgEnv-pgi/3.1.61
  6) udreg/2.2-1.0301.2966.16.2.gem           16) eswrap/1.0.8
  7) ugni/2.1-1.0301.2967.10.23.gem           17) xtpe-mc12
  8) pmi/2.1.1-1.0000.8296.10.8.gem           18) xt-shmem/5.2.1
  9) dmapp/3.0-1.0301.2968.22.24.gem          19) torque/2.4.8-snap.201004261413
 10) gni-headers/2.1-1.0301.2931.19.1.gem     20) moab/5.3.6-s14846
```

- The most important module is called "PrgEnv-pgi", which lets you know that the environment is set up to use the Portland Group compiler suite.

module avail

- The "module avail" command will list all the available modules. It's a very long list, so I won't list it here
- You can use the module's name stem to do a useful search

```
nid00163% module avail PrgEnv

PrgEnv-cray/1.0.1(default)   PrgEnv-pathscale/2.2.48B(default)
PrgEnv-gnu/2.2.48B(default) PrgEnv-pgi/2.2.48B(default)
```

- Here you see that four programming environments are available using the Cray, GNU, Pathscale, and PGI compilers.
- The word "default" is confusing here; it does not refer to the default computing environment, but rather the default version of each specific PrgEnv module. (It just happens that in this case, there is only one version available of each.)

module swap

Let's say you want to use the Cray compiler instead of PGI.

```
%module swap PrgEnv-pgi PrgEnv-cray
```

Now you are using the Cray compiler suite. That's all you have to do.

You don't have to change your makefiles, or anything else in your build script unless they contain PGI or Cray-specific options or features.

Hands-On

```
% module list
% ftn -V
pgf90 11.3-0 64-bit target on x86-64 Linux -tp shanghai
% module swap PrgEnv-pgi PrgEnv-cray
% ftn -V
Cray Fortran : Version 7.4.0  Mon Sep 12, 2011  17:01:32
```

module load

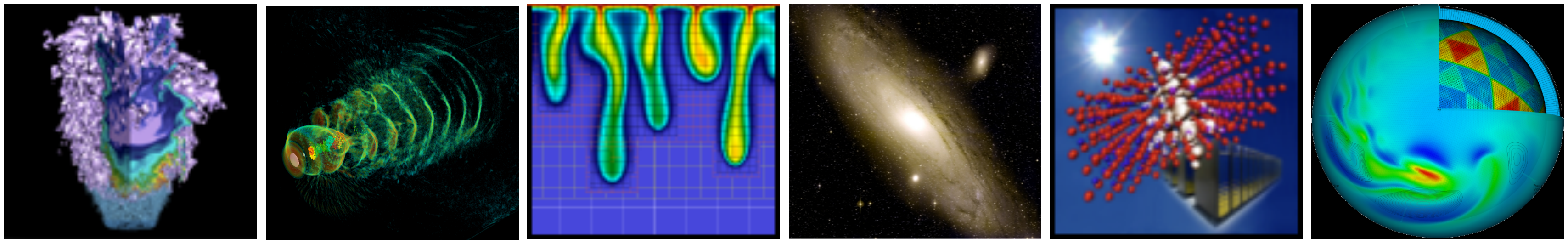
- There is plenty of software that is not loaded by default.
- You can consult the NERSC web pages to see a list, or you can use the "module avail" command to see what modules are available
- For example, if you want to use the NAMD molecular dynamics application. Try "module avail namd".

```
nid00163% module avail namd
namd/2.6(default) namd/2.7b1_plumed namd/cvs
namd/2.7b1        namd/2.7b2
```

- The default version is 2.6, but say you'd rather use some features available only in version 2.7b2. In that case, just load that module.

```
nid00163% module load namd/2.7b2
```

- The “namd2” binary for version 2.7b2 is now in your UNIX search path.



Compiling Code

Invoking the Compilers

- **Let's assume that you’re compiling**
  - a parallel application
  - using MPI and the code is
  - written in Fortran, C, or C++
- **Then compiling is easy**
  - You will use standard compiler wrapper
  - All the include file and library paths are set
  - Linker options are set

Parallel Compilers

Platform	Fortran	C	C++
Cray	ftn	cc	CC
Others	mpif90	mpicc	mpiCC

```
!Filename hello.f90
program hello

    implicit none
    include "mpif.h"

    integer:: myRank
    integer:: ierror

    call mpi_init(ierror)

    call mpi_comm_rank(MPI_COMM_WORLD,myRank)

    print *, "MPI Rank ",myRank," checking in!"

    call mpi_finalize(ierror)
```

```
% ftn –o hello.x hello.f90
```

That's it!

No `-I/path/to/mpi/include` or `-L/path/to/mpi/lib`

It's all taken care of for you.

“Serial” compilers

- **You can use serial compilers as you would on a typical Linux cluster**
  - gcc, gfortran, pgf90, etc.
  - Won’t run on compute nodes on Crays
  - You need to supply all the compiler and linker options
  - May have to load a module to access a given compiler (e.g. module load pgi/11.2.0)

Using Programming Libraries (Cray)

All you have to do is load the appropriate module and compile.

Let's compile an example code that uses the HDF5 I/O library.
First let's try it in the default environment.

```
nid00195% cc -o hd_copy.x hd_copy.c
INFO: linux target is being used
Can't find include file hdf5.h (hd_copy.c: 39)
```

The compiler doesn't know where to find the include file.
Now let's load the hdf5 module and try again.

```
nid00195% module load hdf5
nid00195% cc -o hd_copy.x hd_copy.c
```

We're all done and ready to run the program! No need to manually add the path to HDF5; it's all taken care of by the scripts.

Using Programming Libraries (non-Cray)

```
% mpicc -o hd_copy.x hd_copy.c
  Can't find file hdf5.h (hd_copy.c: 39)
  PGC/x86-64 10.8-0: compilation aborted
% module load hdf5
% mpicc -o hd_copy.x hd_copy.c
 Can't find file hdf5.h (hd_copy.c: 39)
 PGC/x86-64 10.8-0: compilation aborted
```

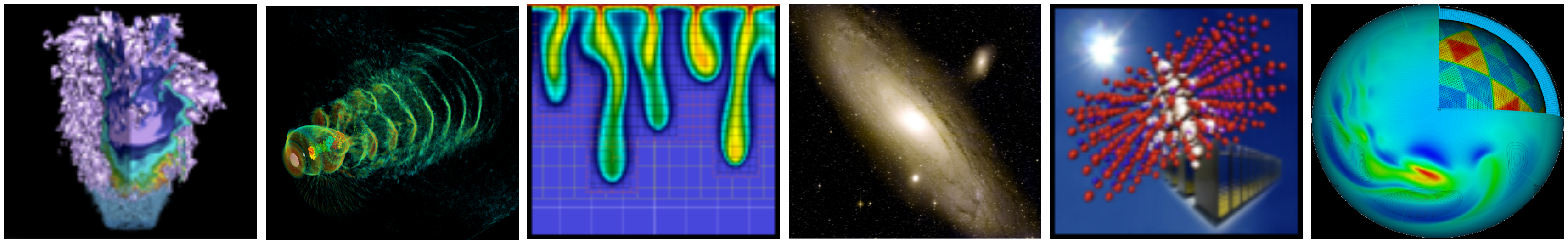
Even with the module loaded, the compiler doesn't know where to find the HDF5 files.

Using Programming Libraries (non-Cray)

We have to use environment variables defined in the module (use "module show" to see them).

```
% mpicc -o hd_copy.x hd_copy.c $HDF5
% module show hdf5
-------------------------------------------------------------------
/usr/common/usg/Modules/modulefiles/hdf5/1.8.3:

conflict          hdf5-parallel
module            load szip
module            load zlib
setenv            HDF5_DIR /usr/common/usg/hdf5/1.8.3/serial
setenv            HDF5 -L/usr/common/usg/hdf5/1.8.3/serial/lib -
lhdf5_cpp -lhdf5_fortran -lhdf5_hl -lhdf5 -L/usr/common/usg/zlib/
default/lib -lz -L/usr/common/usg/szip/default/lib -lsz -I/usr/
common/usg/hdf5/1.8.3/serial/include -I/usr/common/usg/
hdf5/1.8.3/serial/lib -I/usr/common/usg/zlib/default/include -I/
usr/common/usg/szip/default/include
setenv            HDF5_INCLUDE -I/usr/common/usg/hdf5/1.8.3/
serial/include
prepend-path      PATH /usr/common/usg/hdf5/1.8.3/serial/bin
prepend-path      LD_LIBRARY_PATH /usr/common/usg/hdf5/1.8.3/
serial/lib
```



Running Jobs

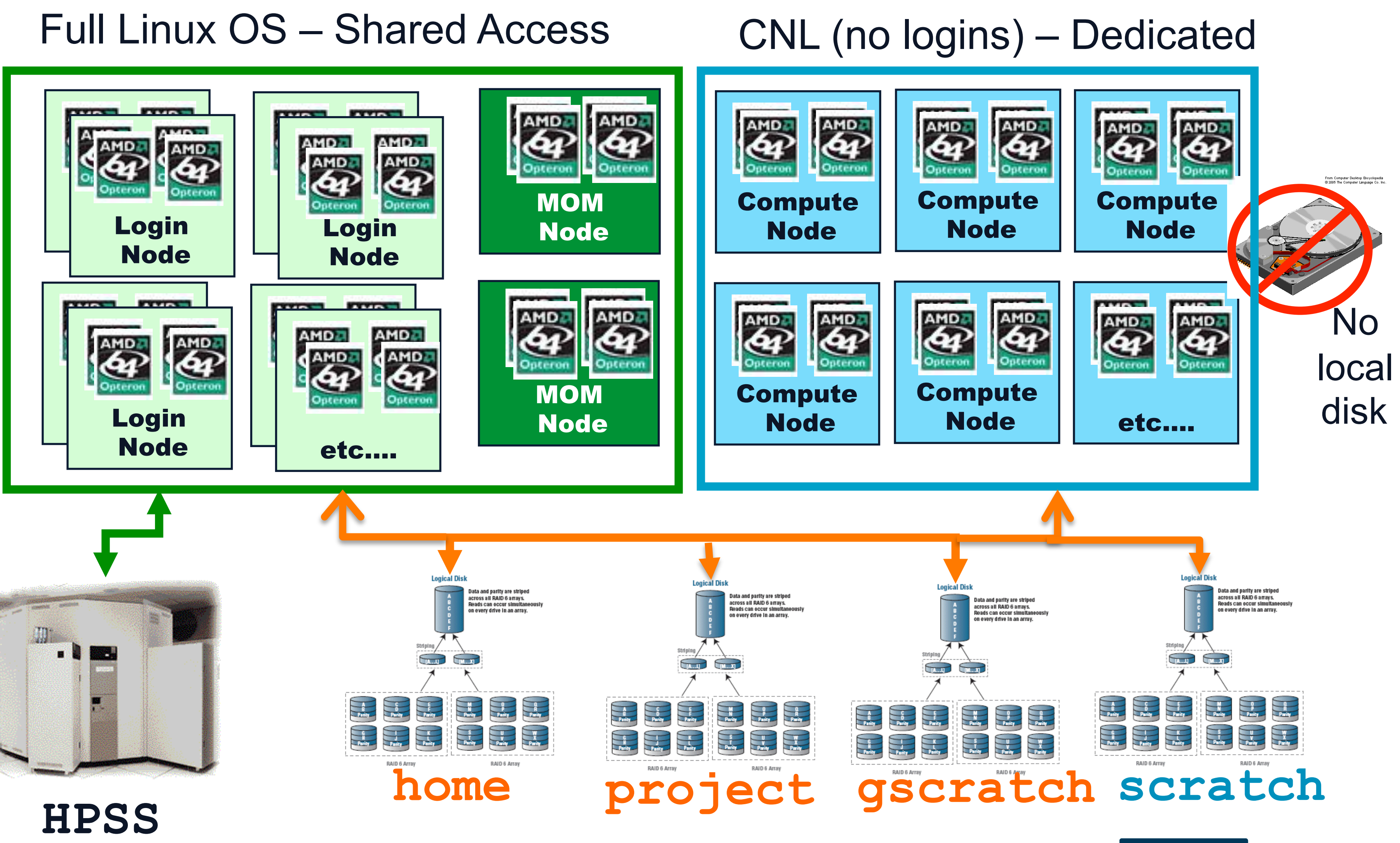
Jobs at NERSC

- Most jobs are parallel, using 10s to 100,000+ cores
- Production runs execute in batch mode
- Interactive and debug jobs are supported for up to 30 minutes
- Typically run times are a few to 10s of hours.
  - Each machine has different limits.
  - Limits are necessary because of MTBF and the need to accommodate 4,000 users' jobs
- Many jobs "package" lower concurrency runs into one job
  - Can increase throughput
  - Possible to run many "serial jobs" as one parallel job
  - Load balance may be an issue

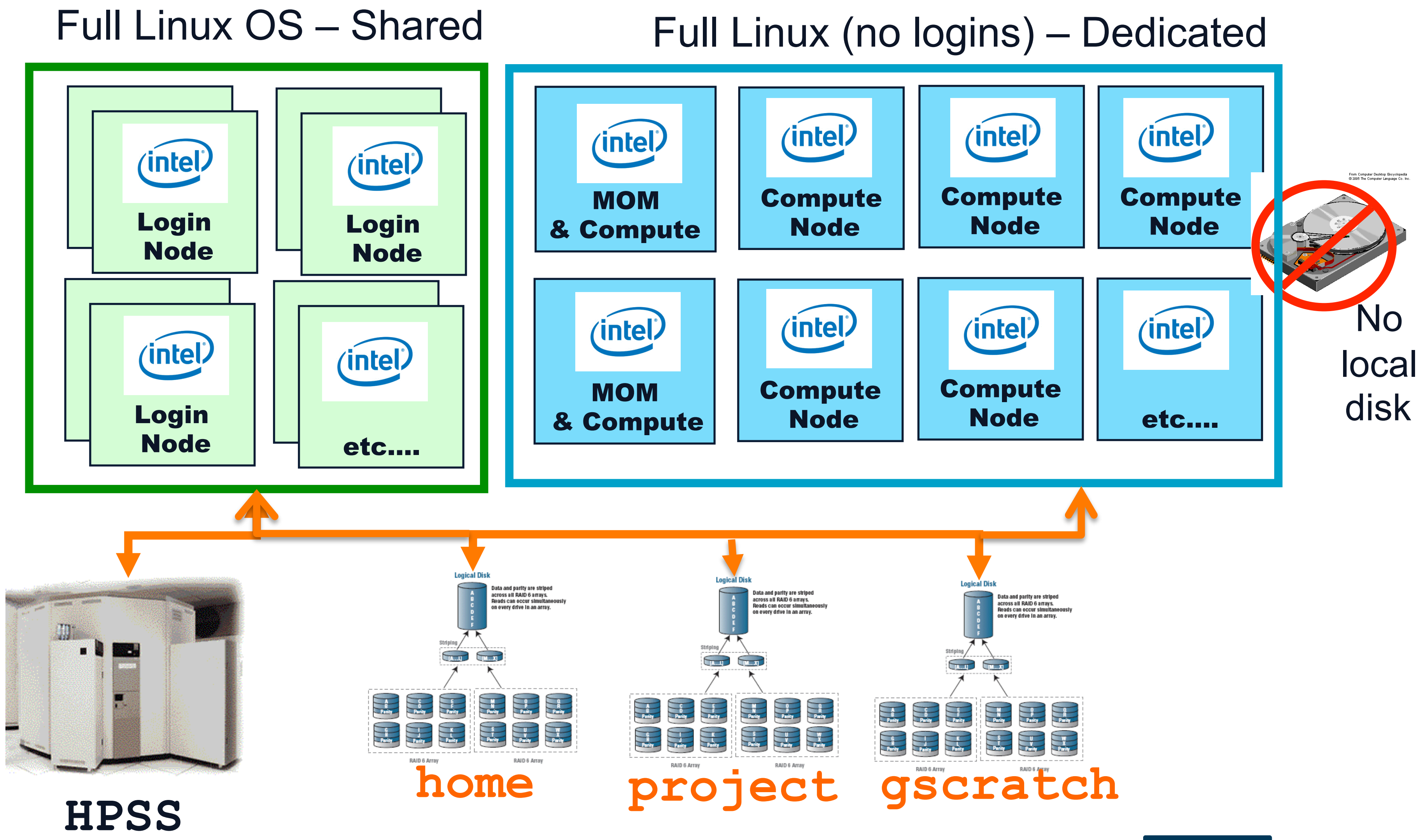
Types of Nodes

- Each supercomputer has 3 types of nodes that you will use directly
  - Login nodes
  - Compute nodes
  - “MOM” nodes
- Login nodes
  - Edit files, compile codes, run UNIX commands
  - Submit batch jobs
  - Run short, small utilities and applications
- Compute nodes
  - Execute your application; dedicated to your job
  - No direct login access
- “MOM” nodes
  - Execute your batch script commands
  - Carver: “head” compute node; Cray: shared “service” node

Cray Systems



Carver / Dirac



Launching Parallel Jobs

- A "job launcher" application executes your code
  - Distributes your executables to all your nodes
  - Starts concurrent execution of N instances of your program
  - Manages execution of your application
  - On Crays: the job launcher is called "aprun"
  - On Carver: "mpirun"
- Only the job launcher can start your job on compute nodes
- You can't run the job launcher from login nodes

Submitting Jobs

- To run a job on the compute nodes you must write a “batch script,” which contains
  - Batch directives to allow the system to schedule your job
  - An aprun or mpirun command that launches your parallel executable
- Submit the job to the queuing system with the qsub command
  - `%qsub my_batch_script`

Sample Hopper Batch Script

```
#PBS -q debug
#PBS -l mppwidth=96
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS –A one_of_my_valid_repos
#PBS -V

cd $PBS_O_WORKDIR
aprun -n 96./my_executable
```

Options can also be specified on the qsub command line (e.g. qsub –A <batch_script_name>

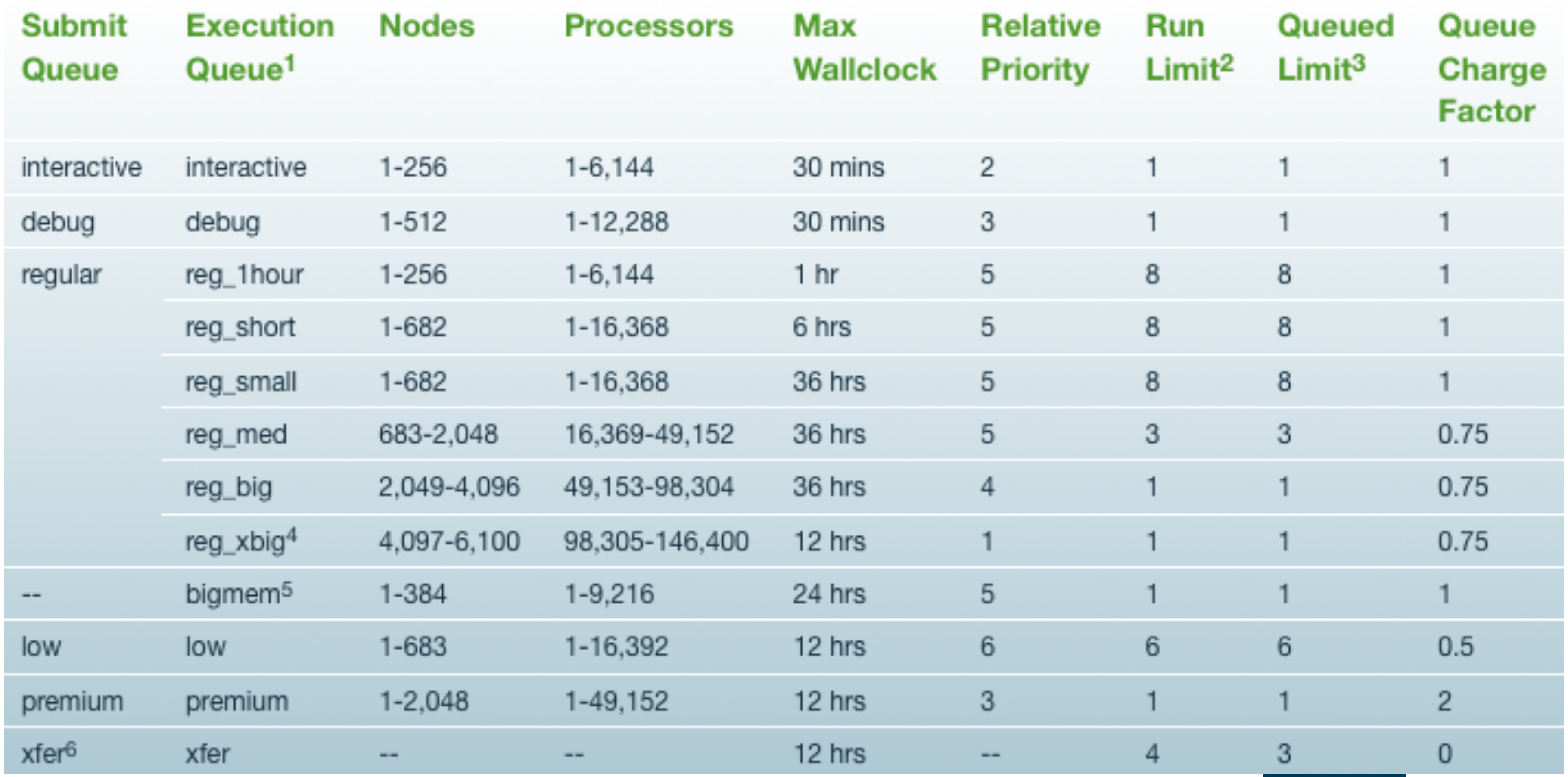
The PBS directives required for each system are different, so consult the NERSC web site for details.

Monitoring Your Job

- Once your job is submitted, it will start when resources are available
- Monitor it with
  - qstat –a
  - qstat –u username
  - showq
  - qs
  - NERSC web site “Queue Look”

Job Limits

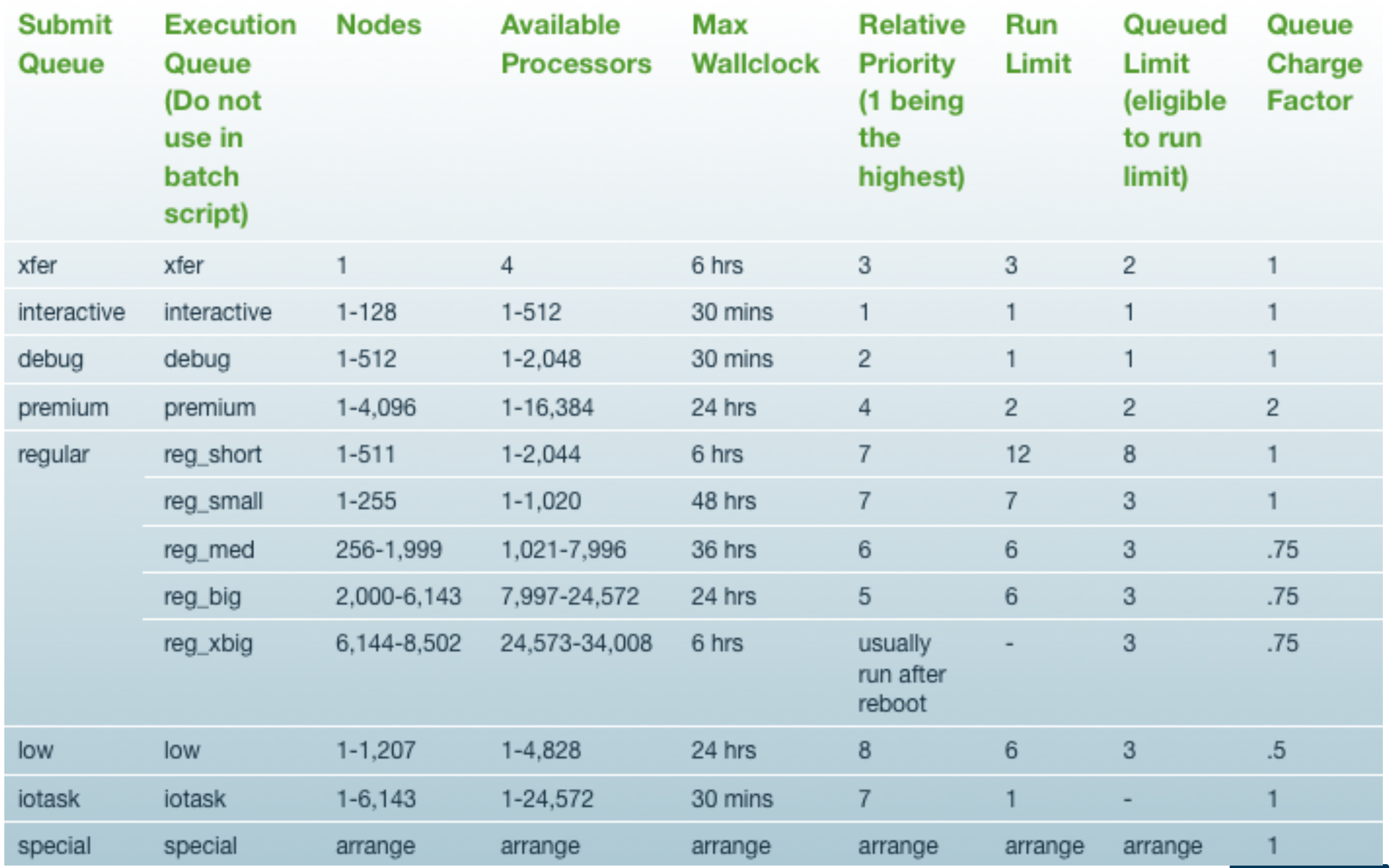
There are per user, per machine job limits. See the NERSC web site for details. Here are the limits on Hopper as of Sep 12, 2011.



Submit Queue	Execution Queue[1]	Nodes	Processors	Max Wallclock	Relative Priority	Run Limit[2]	Queued Limit[3]	Queue Charge Factor
interactive	interactive	1-256	1-6,144	30 mins	2	1	1	1
debug	debug	1-512	1-12,288	30 mins	3	1	1	1
regular	reg_1hour	1-256	1-6,144	1 hr	5	8	8	1
	reg_short	1-682	1-16,368	6 hrs	5	8	8	1
	reg_small	1-682	1-16,368	36 hrs	5	8	8	1
	reg_med	683-2,048	16,369-49,152	36 hrs	5	3	3	0.75
	reg_big	2,049-4,096	49,153-98,304	36 hrs	4	1	1	0.75
	reg_xbig[4]	4,097-6,100	98,305-146,400	12 hrs	1	1	1	0.75
--	bigmem[5]	1-384	1-9,216	24 hrs	5	1	1	1
low	low	1-683	1-16,392	12 hrs	6	6	6	0.5
premium	premium	1-2,048	1-49,152	12 hrs	3	1	1	2
xfer[6]	xfer	--	--	12 hrs	--	4	3	0

Job Limits

There are per user, per machine job limits. See the NERSC web site for details. Here are the limits on Franklin as of Sep 12, 2011.



Submit Queue	Execution Queue (Do not use in batch script)	Nodes	Available Processors	Max Wallclock	Relative Priority (1 being the highest)	Run Limit	Queued Limit (eligible to run limit)	Queue Charge Factor
xfer	xfer	1	4	6 hrs	3	3	2	1
interactive	interactive	1-128	1-512	30 mins	1	1	1	1
debug	debug	1-512	1-2,048	30 mins	2	1	1	1
premium	premium	1-4,096	1-16,384	24 hrs	4	2	2	2
regular	reg_short	1-511	1-2,044	6 hrs	7	12	8	1
	reg_small	1-255	1-1,020	48 hrs	7	7	3	1
	reg_med	256-1,999	1,021-7,996	36 hrs	6	6	3	.75
	reg_big	2,000-6,143	7,997-24,572	24 hrs	5	6	3	.75
	reg_xbig	6,144-8,502	24,573-34,008	6 hrs	usually run after reboot	-	3	.75
low	low	1-1,207	1-4,828	24 hrs	8	6	3	.5
iotask	iotask	1-6,143	1-24,572	30 mins	7	1	-	1
special	special	arrange	arrange	arrange	arrange	arrange	arrange	1

Job Limits

There are per user, per machine job limits. See the NERSC web site for details. Here are the limits on Carver as of Sep 12, 2011.

Submit Queue	Execution Queue	Nodes	Available Cores	Max Wallclock	Relative Priority	Run Limit	Eligible Limit	Charge Factor
interactive	interactive	1-8	1-64	30 mins	1	2	1	1.0
debug	debug	1-32	1-256	30 mins	2	2	1	1.0
regular	reg_short	1-16	1-128	4 hrs	3	8	4	1.0
	reg_small	1-16	1-128	48 hrs	3	6	3	1.0
	reg_med	17-32	129-256	36 hrs	3	5	2	1.0
	reg_big	33-64	257-512	24 hrs	3	3	1	1.0
	reg_long	1-4	1-32	168 hrs	3	2	1	1.0
	reg_xlong	1-4	1-32	504 hrs	3	2	1	1.0
low	low	1-32	1-256	12 hrs	4	5	3	0.5
magellan	mag_short	1-16	1-128	4 hrs	3	8	4	1.0
	mag_small	1-16	1-128	48 hrs	3	6	3	1.0
	mag_med	17-32	129-256	36 hrs	3	5	2	1.0
	mag_big	33-64	257-512	24 hrs	3	3	1	1.0
	mag_long	1-4	1-32	168 hrs	3	2	1	1.0
mag_serial	mag_serial	1	1	24 hrs	3	40	20	1.0
mag_xlmem	mag_xlmem	1	32	48 hrs	3	2	1	1.0

Interactive Parallel Jobs

- You can run small parallel jobs interactively for up to 30 minutes

```
% qsub –I –V –lmppwidth=32
[wait for job to start]
% cd $PBS_O_WORKDIR
% aprun –n 32 ./mycode.x
```

How Your Jobs Are Charged

- **Your repository account is charged for each core your job was allocated for the entire length of your job.**
  - The minimum allocatable unit is a node. Hopper has 24 cores/node, so your minimum charge on Hopper is 24*`walltime`.
  - e.g., `mppwidth=96` for 1 hour of run time is charged 96*1 hour = 96 MPP Hours (assuming the default setting of `mppnppn=24`)
  - You are charged for your actual run time, not the value of `walltime` in your batch script.
- **If you have access to multiple repos, pick which one to charge in your batch script**
  - `#PBS -A` *`repo_name`*

Charge Factors & Discounts

- **Each machine has a "machine charge factor" (mcf) that multiplies the "raw hours" used**
  - Hopper and Franklin have mcf=1.0
  - Carver has mcf=1.5
- **Queues have "priority charge factors" (pcf) and corresponding relative scheduling priorities**
  - Premium pcf=2.0
  - Low pcf=0.5
  - Everything else pcf=1.0
- **On Hopper only:**
  - reg_med, reg_big, reg_xbig jobs get a 25% discount
- **Storage and bandwidth are allocated and charged for HPSS**
  - Exhausting an HPSS allocation is rare
  - See the NERSC web site for details

NERSC

National Energy Research Scientific Computing Center

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory