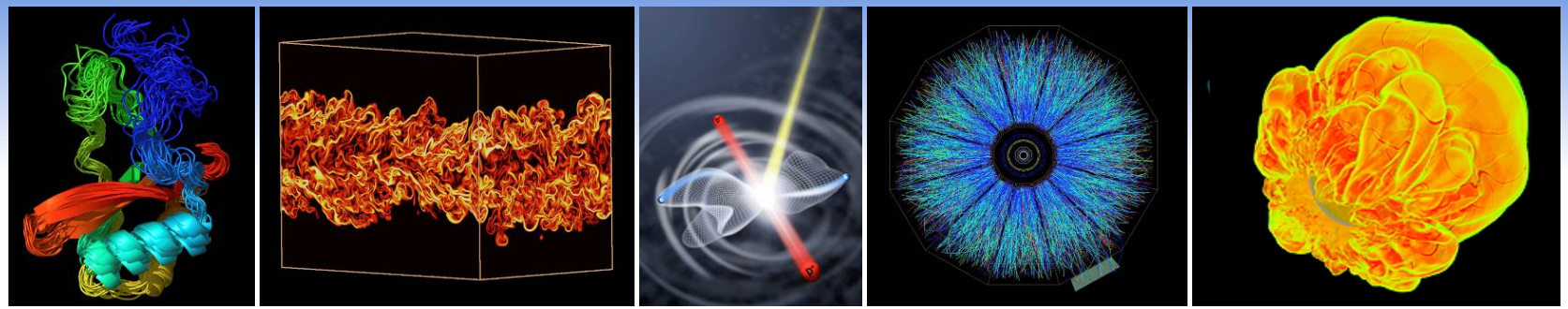


Introduction to Computing at NERSC

Richard Gerber

NERSC User Services

NERSC User Group Training

October 18, 2010

NERSC Oakland Scientific Facility

U.S. DEPARTMENT OF ENERGY

NERSC

BERKELEY LAB

Outline

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

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

What is NERSC?

OVERVIEW

U.S. DEPARTMENT OF ENERGY

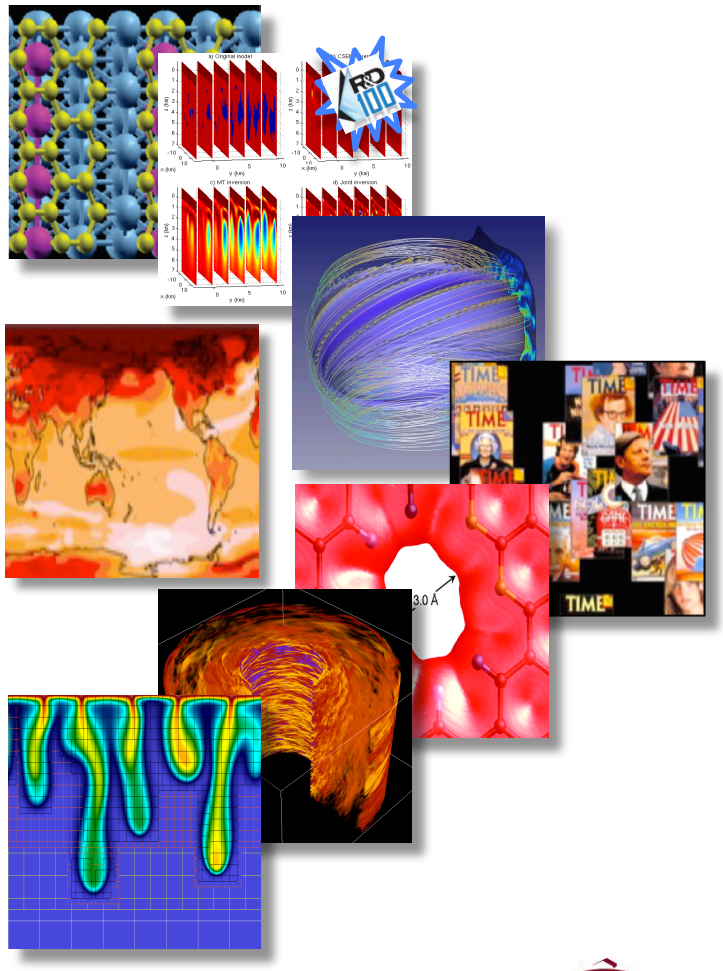
BERKELEY LAB

NERSC Mission

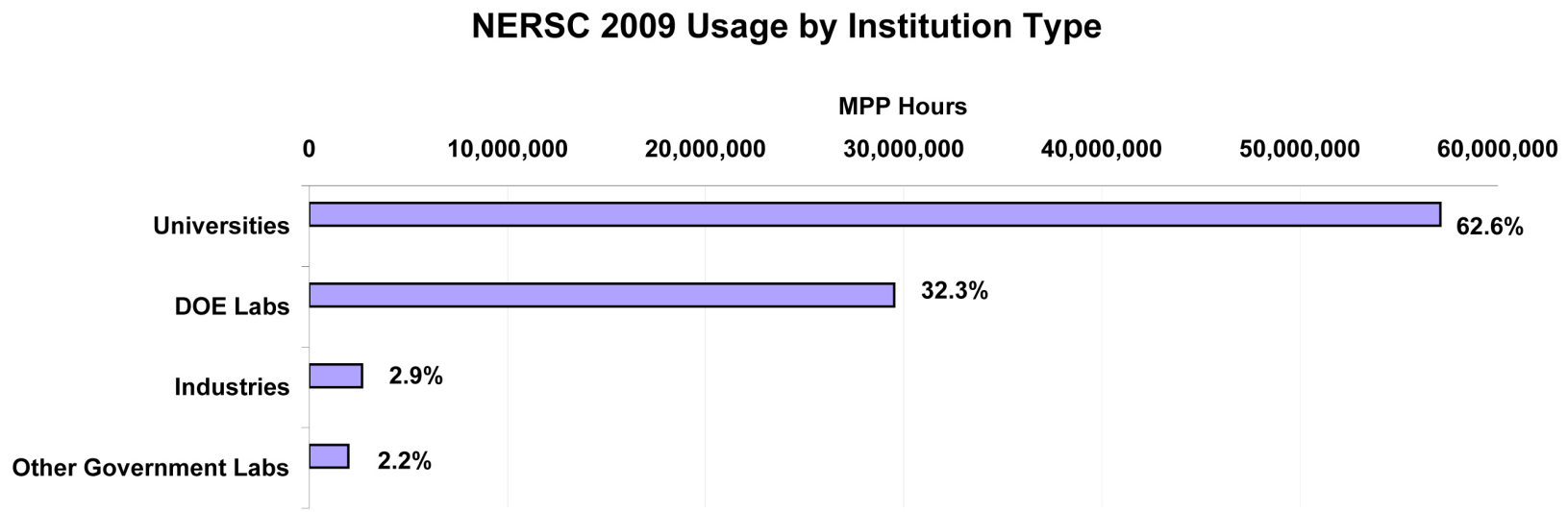
NERSC's mission is to *accelerate the pace of scientific discovery* by providing high-performance computing, information, data, and communications services to the DOE Office of Science community.

NERSC is the Production Facility for DOE Office of Science

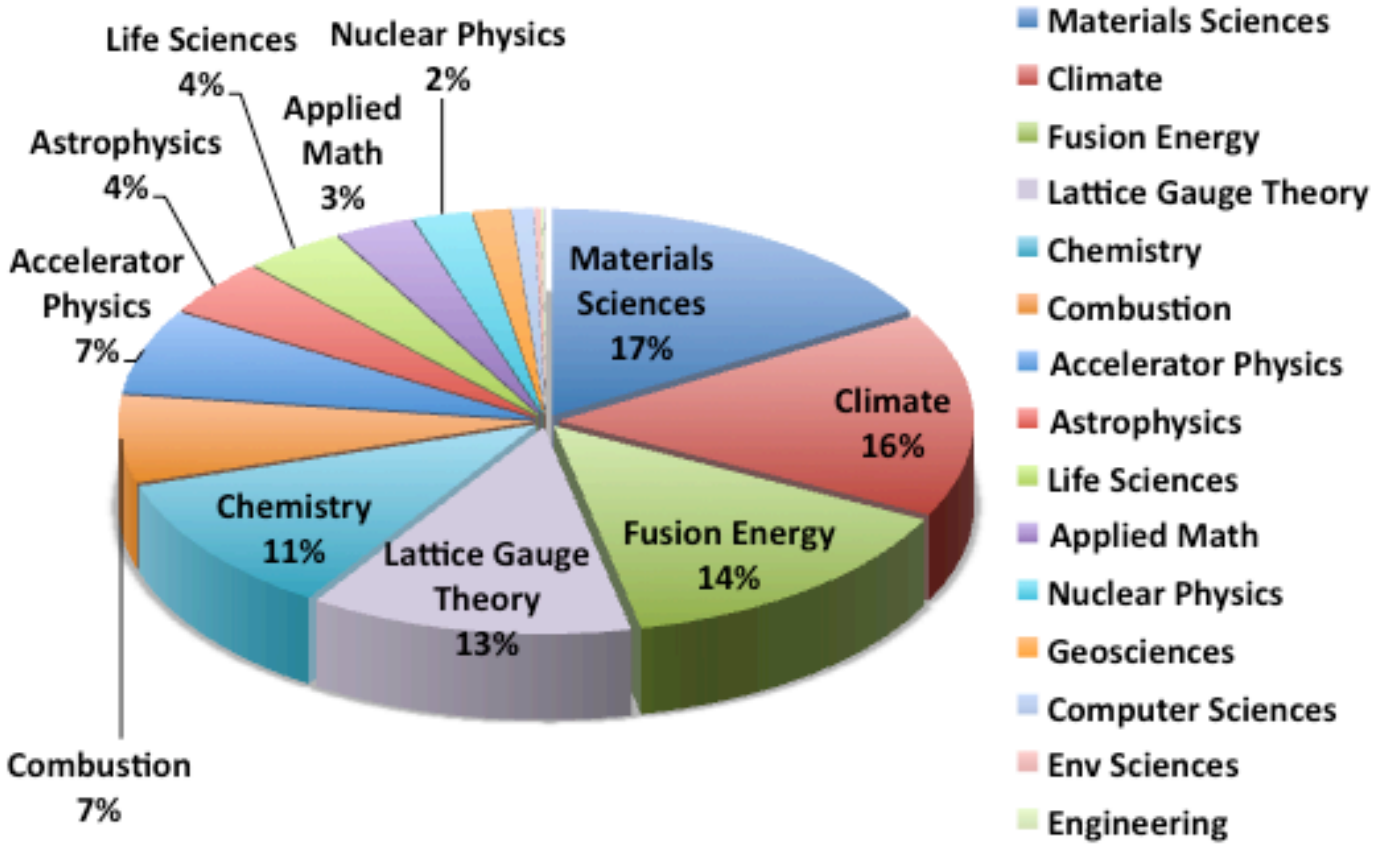
- **NERSC serves a diverse workload**
  - **3,000 users, 400 projects,**
  - **500 codes**
- **Allocations controlled primarily by DOE**
  - **80% Annual Production awards:**
    - **From 10K hour to 5M hour**
    - **Proposal-based; DOE chooses**
  - **10% DOE ASCR Leadership Computing Challenge**
  - **10% NERSC reserve ("NISE")**



NERSC User Demographics



Science View of Workload



NERSC 2008 Allocations By Science Area

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

What is NERSC?

COMPUTING RESOURCES

NERSC Systems for Science

Large-Scale Computing System

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

Hopper (NERSC-6): Cray XT
- Phase 1: Cray XT5, 668 nodes, 5344 cores
- Phase 2: > 1 Pflop/s peak (late 2010 delivery)

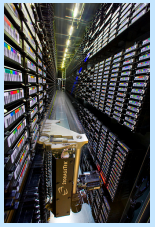
Clusters
105 Tflops combined

Carver
- IBM iDataplex cluster

PDSF (HEP/NP)
- Linux cluster (~1K cores)

Cloud testbed
- IBM iDataplex cluster

NERSC Global Filesystem (NGF)
Uses IBM's GPFS
1.5 PB; 5.5 GB/s



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

Analytics

- Euclid (512 GB shared memory)
- GPU testbed (48 nodes)

Hopper (Phase II) - Cray XE6

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

Use Hopper II for your biggest, most computationally challenging problems.

Hopper (Phase I) – Cray XT5

- 5,312 compute cores
- Cray Seastar interconnect
- 664 compute nodes
- Two quad-core AMD 2.4 GHz Opteron processors (Istanbul) per node
- 8 processor cores per node
- 16 GB of memory per node
- 10.6 TB of aggregate memory
- 1.8 GB memory / core for applications

- 2 PB of /scratch disk
- Linux operating system (Cray Linux)
- PGI, Cray, Pathscale, GNU compilers
- /scratch disk quota of 2 TB

Use Hopper for jobs of moderate parallel concurrency (512-1,024 cores) and for codes developed to run on Cray systems.

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.

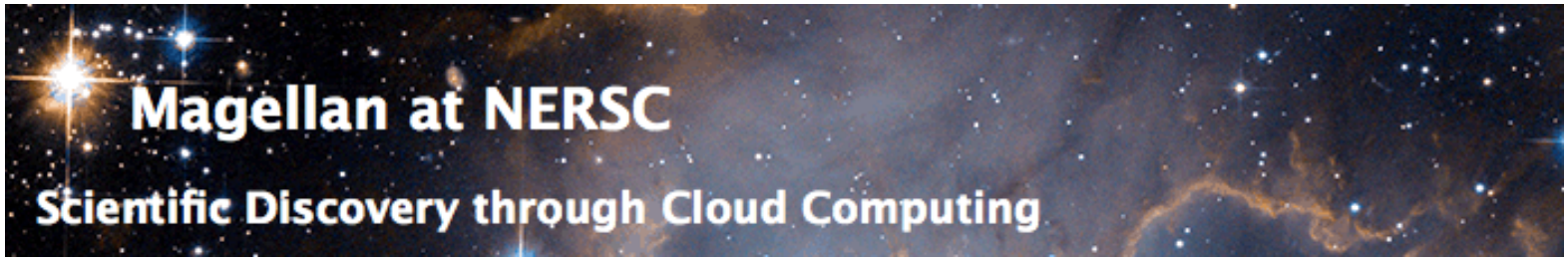
Carver - IBM iDataPlex

- 3,200 compute cores
- 400 compute nodes
- Two 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 using up to 512 cores, need a fast CPU, need a standard Linux configuration, or need up to 48 GB of memory on a node.

Magellan - IBM IDataPlex



- Dedicated to HPC Cloud Computing research
- 4,480 compute cores
- 560 compute nodes
- Two quad-core Intel Nehalem 2.67 GHz processors per node
- 8 processor cores per node
- 24 GB of memory per node (48 GB on 160 "fat" nodes)
- 2.5 GB / core for applications (5.5 GB / corre on "fat" nodes)
- NERSC global /scratch directory quota of 20 TB
- Full Linux operating system
- PGI, GNU, Intel compilers

PDSF

PDSF is a special-purpose Linux cluster used exclusively by High Energy Physics and Nuclear Physics projects.

You do not have access to PDSF unless you are part of one of those projects.

See NERSC consultants for more information about PDSF.

What is NERSC?

STORAGE RESOURCES

Data Storage Types

- "Spinning Disk"
  - Interactive access
  - I/O from compute jobs
  - "Home", "Project", "Scratch"
- Archival Storage
  - Permanent, Long-Term Storage
  - Tapes, fronted by disk cache
  - "HPSS" (High Performance Storage System)

Home Directory

- When you log in you are in your "Home" directory.
- The full UNIX pathname is stored in the environment variable $HOME.
- $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 500,000 inodes (files and directories).
  - Use “myquota” command to check your usage and quota
- Permanent storage, but no automatic user backups

Scratch Directories

- Each system has a large, high-performance ”scratch” directory.
- Each user has a personal directory referenced by $SCRATCH (and maybe $SCRATCH2).
- All I/O from your compute jobs should be directed to $SCRATCH
- 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 will use a global file system.
- Data in $SCRATCH is not backed up and could be lost if a file system fails.

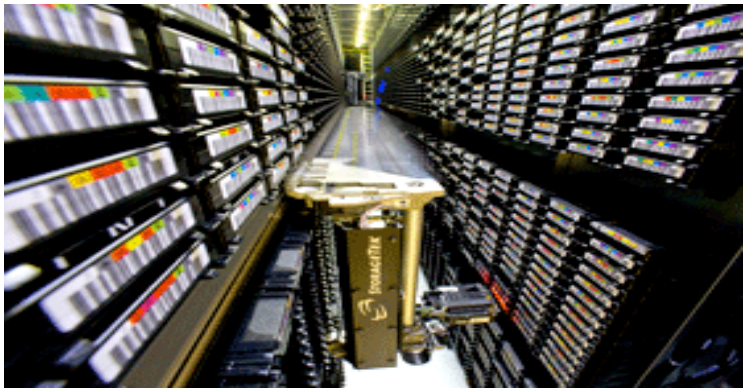
Project Directories

- All NERSC systems mount the NERSC global "Project" file system.
- "Project directories" are created upon request for projects (a group of researchers) to store and share data.
- The default quota in /project is 1 TB.
- While data can be written and read from a parallel job on all system, performance will not be as good as on $SCRATCH.
- Data in /project is not purged, but there are no automatic user backups either.

Archival Storage (HPSS)

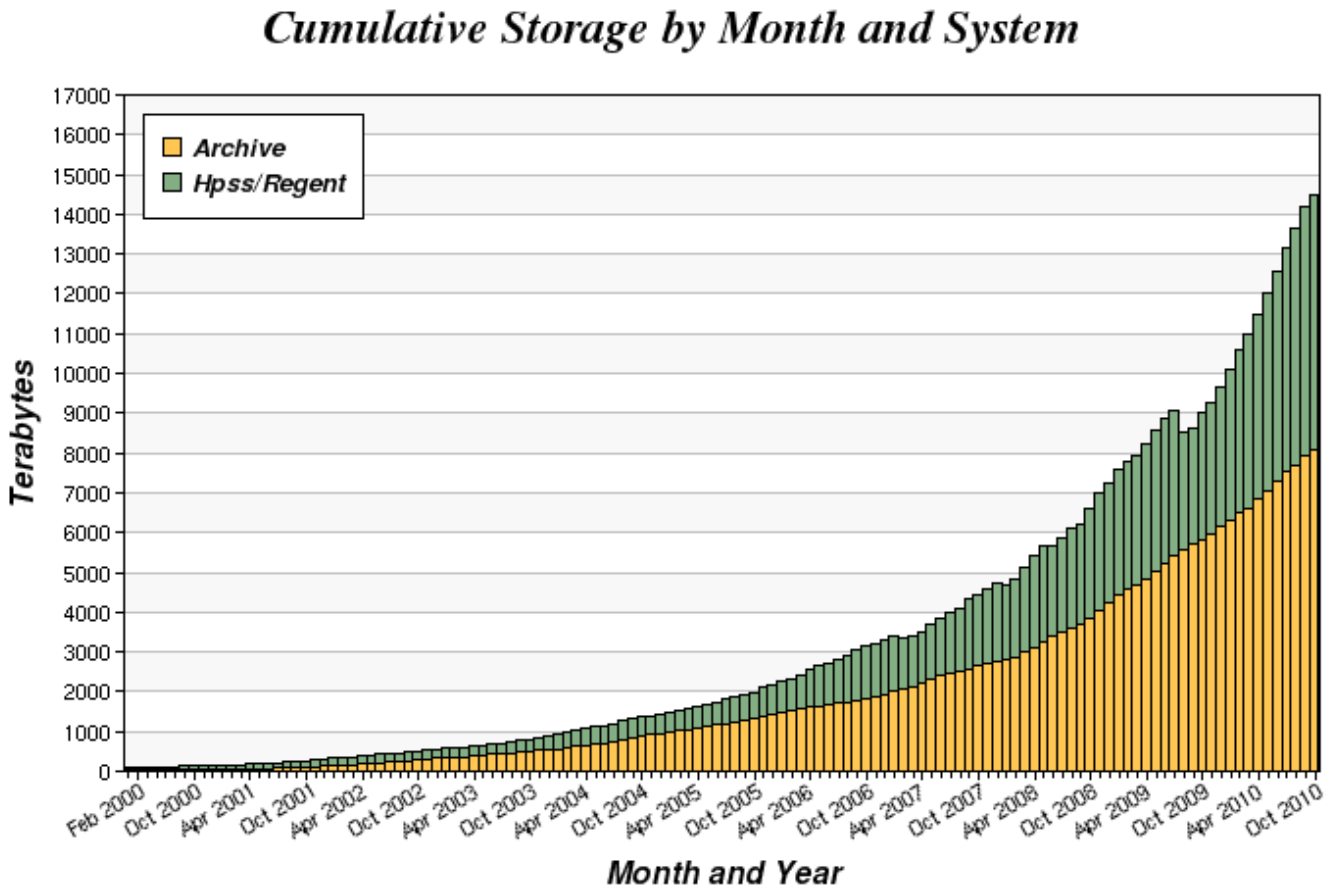
- For permanent, archival storage
- You transfer files to and from HPSS using one of ftp, pftp, or the HPSS hsi client.
- You do not log in to HPSS using your NERSC password directly. You must generate a token in NIM.
- This token is placed in a file named $HOME/.netrc or use a long encrypted string as your password.
- An example token:

```
machine archive.nersc.gov
login fredflintstone
password
02SYMz5aA3USZyYWdlabJlcgDC3LvjZZTCmT9p
L81Ln+zbgwyMEQ==
```



- **Hostname: archive.nersc.gov**
- **Almost 15 Petabyes of data stored**
- **Data increasing by 1.7X per year**
- **120 M files stored**
- **100 TB disk cache**
- **8 STK robots**
- **44,000 tape slots**
- **44 PB maximum capacity today**
- **Average data xfer rate: 100 MB/sec**

HPSS Data


Cumulative Storage by Month and System
Archive
Hpss/Regent
Terabytes
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Feb 2000
Oct 2000
Apr 2001
Oct 2001
Apr 2002
Oct 2002
Apr 2003
Oct 2003
Apr 2004
Oct 2004
Apr 2005
Oct 2005
Apr 2006
Oct 2006
Apr 2007
Oct 2007
Apr 2008
Oct 2008
Apr 2009
Oct 2009
Apr 2010
Oct 2010
Month and Year

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

USING NERSC

How to Get Help

http://www.nersc.gov

Technical questions, account support, passwords, computer operations

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

Computer Operations = menu option 1 (24/7)

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

HPC Consulting = menu option 3, or consult@nersc.gov

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

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Using NERSC

ACCOUNTS & ALLOCATIONS

Accounts

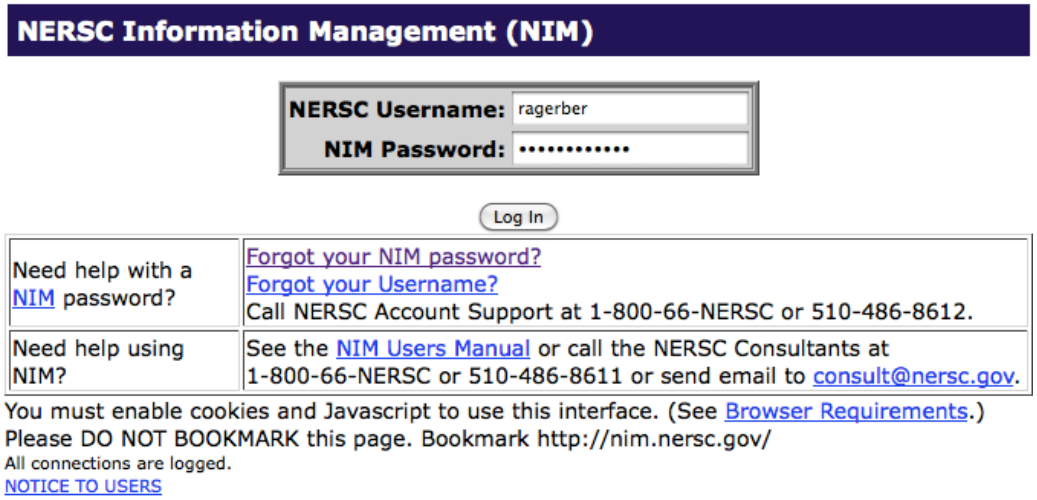
- There are two types of "accounts" at NERSC. It is important to differentiate between the two kinds.
  1. Your personal, private account
     - Associated with your "login" or "user name"
     - Identifies you to our systems and is used when logging into NERSC systems and web services.
  2. An allocation account, or "repository" (aka "repo")
     - Like a bank account you use to "pay" for computer time.
     - 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

- You must have an allocation of time to run jobs at NERSC (be a member of a “repo”)
- Project PIs apply through the ERCAP process
- Computer time and storage allocations are awarded by DOE
- Most allocations are awarded in the fall
- Allocation year starts in January
- Small startup allocations are awarded throughout the year
- There is additional time available through NISE and ALCC

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
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Using NERSC

CONNECTING TO NERSC

Logins and Passwords

- Your "user name" or "login name" is your unique identifier.
  - You will receive your user name from the NERSC Account Support office when your account is created.
- Each real person has a single password associated with their login account.
  - As a new user, you must get your initial password by talking to the NERSC Account Support office at 1-866-NERSC, menu option 2.
  - You can change or reset your password at https://nim.nersc.gov/ or by calling the Account Support Office during business hours or Computer Operations 24x7.
- If you repeatedly fail to type your correct password when accessing a NERSC system, your account on that system will be locked.
  - You can call 1-866-NERSC 24x7 or send email to support@nersc.gov during business hours to get your account unlocked.

Logging In

- You connect to NERSC using "ssh" from UNIX-like systems or by using an SSH-compatible application
- Login in with your NERSC user name and NERSC password (aka "NIM password" or "NERSC LDAP password").
- It is convenient to use the options -Y (forward X11 authentication) and -A (forward SSH credentials) so you can transparently display X Windows applications and authenticate when using SSH-based applications on the destination host.
- `% ssh -A -Y -l user franklin.nersc.gov`
- Your can use grid-based tools (e.g. gridftp) also; please ask the NERSC consultants for details.

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Using NERSC

COMPUTING ENVIRONMENT

Computing Environment

- When you log in to any NERSC computer (not HPSS), you are in your global $HOME directory.
- You initially land in the same place no matter which machine you connect to: franklin, hopper, carver - they are all the same.
- This means that if you have files or binary executables that are specific to a certain system, you need to manage keeping them separate.
- Many people make subdirectories for each system in their home directory. Here is a listing of my home directory.

```
nid00163% ls
bassi/    datatran/  hopper/    silence/   turing/
bin/      davinci/   jacquard/  software@  web@
carver/   franklin/  project@   tesla/     www@
common/   grace/     rohan/     training@  zwicky/
```

Shell Initialization Files

- NERSC installs dot-files in your home directory (e.g. .login, .profile)
  - 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 an .ext extension.
  - Put your local modifications in these .ext files (e.g. .login.ext, .profile.ext)

Modules

- Easy access to software 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 module 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 Franklin.

```
nid00163% module list
Currently Loaded Modulefiles:
  1) modules/3.1.6.5                          12) cray/csa/3.0.0-1_2.0202.19602.75.1
  2) moab/5.3.6                               13) cray/account/1.0.0-2.0202.19482.49.3
  3) torque/2.4.7                             14) cray/projdb/1.0.0-1.0202.19483.52.1
  4) xt-asyncpe/4.0                           15) Base-opts/2.2.48B
  5) xtpe-barcelona                           16) pgi/10.5.0
  6) xtpe-target-cnl                          17) xt-libsci/10.4.3
  7) xt-service/2.2.48B                       18) pmi/1.0-1.0000.7901.22.1.ss
  8) xt-os/2.2.48B                            19) xt-mpt/5.0.0
  9) xt-boot/2.2.48B                          20) xt-pe/2.2.48B
 10) xt-lustre-ss/2.2.48B_1.6.5               21) PrgEnv-pgi/2.2.48B
 11) cray/job/1.5.5-0.1_2.0202.19481.53.6     22) cray/MySQL/5.0.64-1.0202.2899.21.1
```

- The most important module is called "PrgEnv-pgi", which let 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 computing environment. 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.

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 is now in your UNIX search path.

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Using NERSC

COMPILING CODE

Invoking the Compilers

- Let's assume that you're compiling code that will run as a parallel application using MPI and the code is written in Fortran, C, or C++.
- Then compiling is easy because you will use standard compiler wrapper scripts that bring in all the include file and library paths and set linker options that you'll need.

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 need for -I/path/to/mpi/include or –L/path/to/mpi/lib

It's all taken care of for you.

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

```
% mpicc -o hd_copy.x hd_copy.c $HDF5
Take a look at the module to see env variables
% 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
```

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Using NERSC

RUNNING JOBS

Login Nodes and Compute Nodes

- Each supercomputer has two types of nodes that you will use directly
  - Login nodes
  - Compute nodes
- Login nodes
  - Edit files and compile codes
  - Issue interactive UNIX commands
  - Submit batch jobs
  - Don't support compute jobs (30 min CPU limit)
- Compute nodes
  - Run your parallel jobs
  - Do not allow you direct login access

Launching Parallel Jobs

- A “job launcher” distributes your code to all the nodes in your parallel job, starts them, and manages their execution.
- On Cray the job launcher is called “aprun” and on other systems it is “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=128
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS -e my_job.$PBS_JOBID.err
#PBS -o my_job.$PBS_JOBID.out
#PBS -V

cd $PBS_O_WORKDIR
aprun -n 128 ./my_executable
```

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
  - qs
  - NERSC web site “Queue Look”

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
```