Introduction to NERSC for the Joint Genome Institute
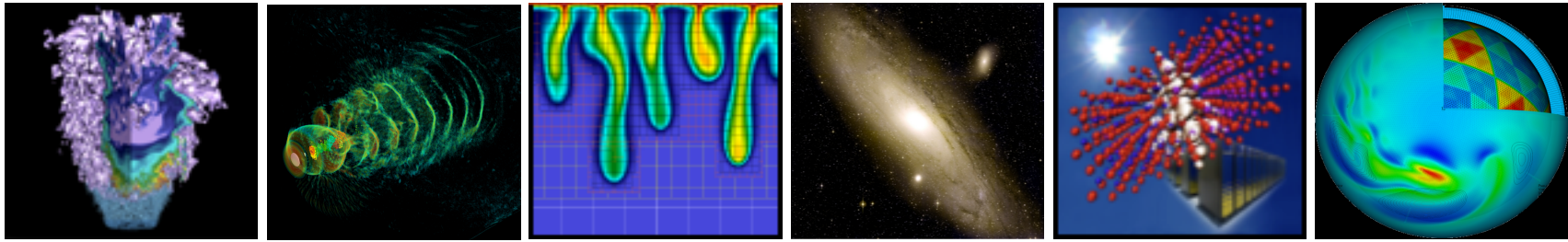
Richard Gerber

NERSC User Services

May 2, 2011

Joint Genome Institute, Walnut Creek, CA

Outline

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

What is NERSC?

NERSC is the Production Facility for DOE Office of Science

- **NERSC serves a large population**
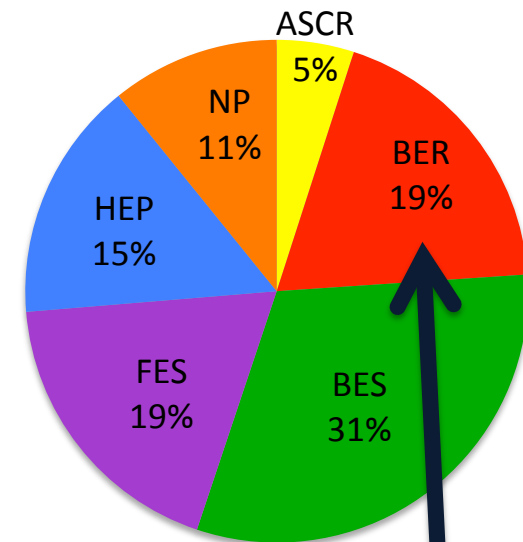
  Over 3000 users, 400 projects, 500 code instances

- **Focus on "unique" resources**
  - Expert consulting and other services
  - High end computing systems
  - High end storage systems
  - Interface to high speed networking

- **Science-driven**
  - Machines procured competitively using application benchmarks from DOE/SC
  - Allocations controlled by DOE/SC Program Offices to couple with funding decisions

2010 Allocations

ASCR 5%
BER 19%
NP 11%
HEP 15%
FES 19%
BES 31%

Biological and Environmental Research

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB | Lawrence Berkeley National Laboratory

ASCR's Computing Facilities

NERSC at LBNL

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

LCFs at ORNL & ANL

- **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**

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB — Lawrence Berkeley National Laboratory
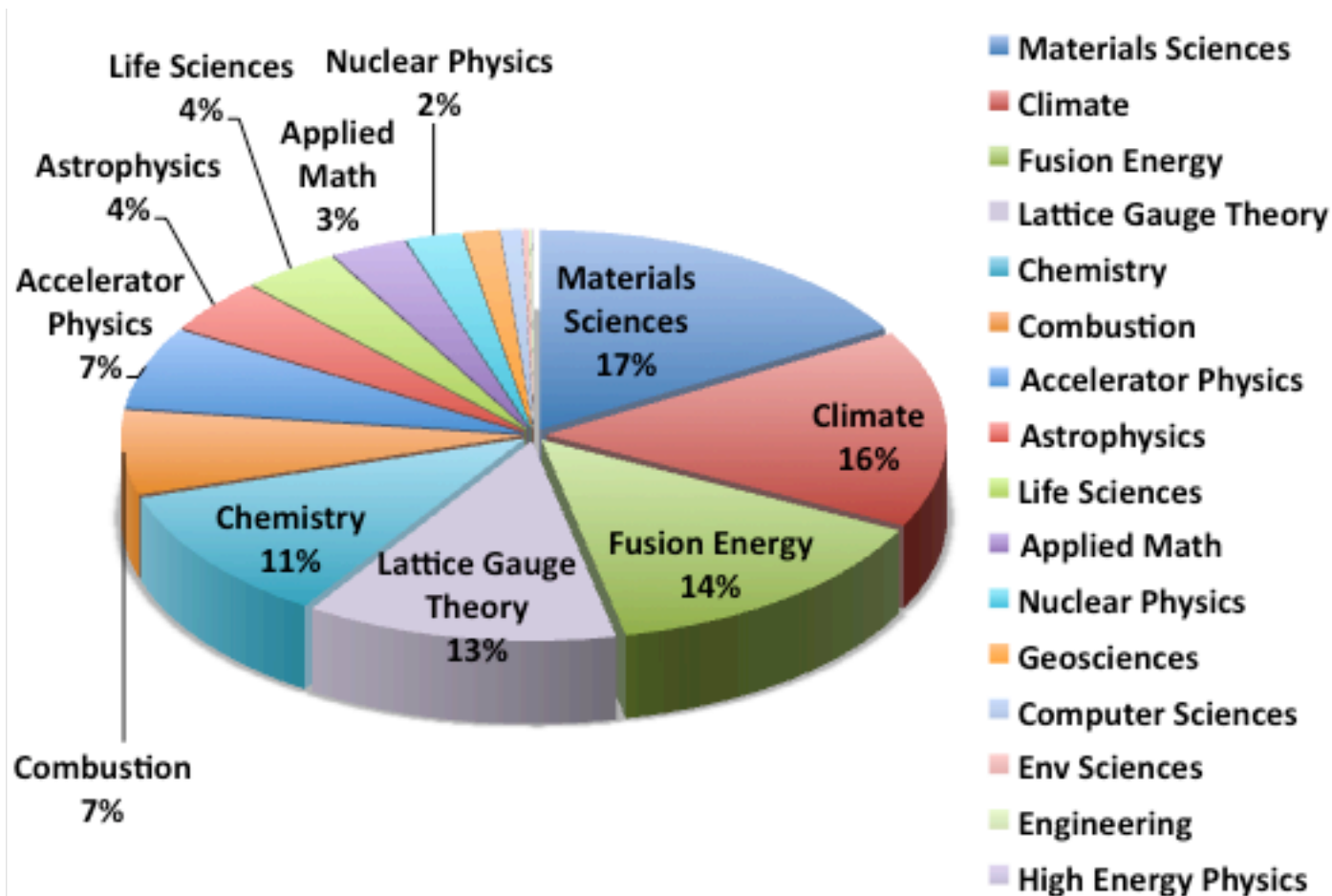
NERSC 2008 Allocations
By Science Area

Who's Who?

Senior Management

Kathy Yelick, Director
Howard Walter, Deputy

Jeff Broughton, Systems
Francesca Verdier, Services

Systems

Computational Systems
  J. Broughton (Acting) [10]
Computer Operations
  S. Lowe [9]
Genomic Systems
  B. Draney [6]
Network & Security
  B. Draney [9]
Storage Systems
  J. Hick [9]
Technology Integration
  S. Canon [3]

Services

Advanced Technologies
  J. Shalf [6]
Analytics
  P. Nugent [8]
Open Software & Programming
  D. Skinner [12]
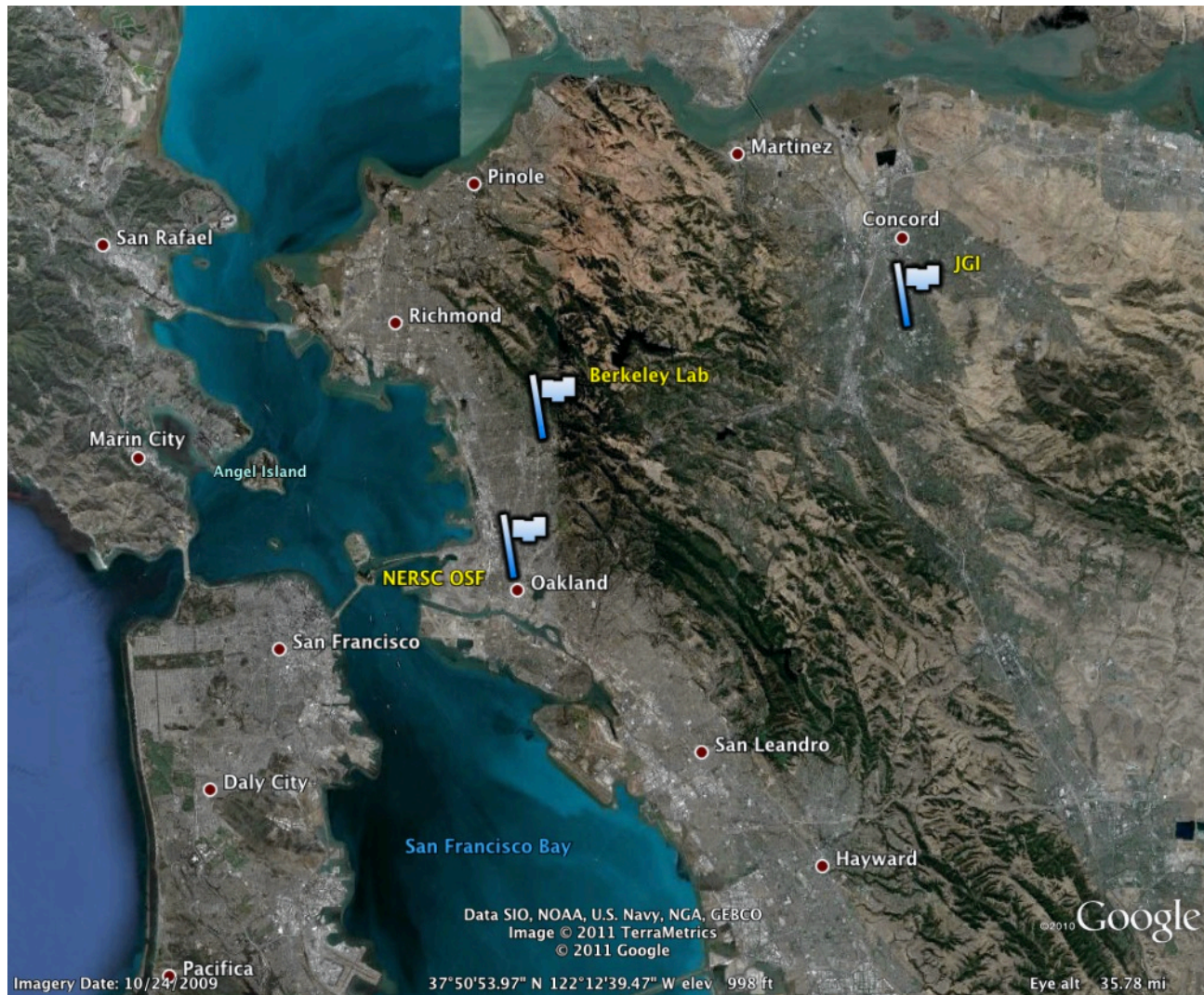**Petascale Postdocs**
  [6]
User Services
  K. Antypas [9]
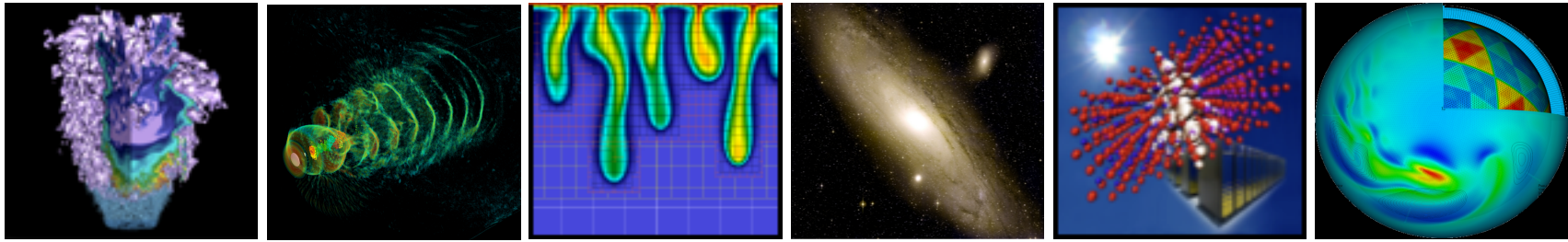
Computing Resources

NERSC Systems

Large-Scale Computing Systems

Hopper (NERSC-6): Cray XE6

  6,384 compute nodes, 153,216 cores

  136 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

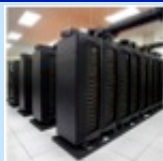 IBM iDataplex cluster

PDSF (HEP/NP)

  ~1K core cluster

Magellan Cloud testbed

  IBM iDataplex cluster

GenePool (JGI)

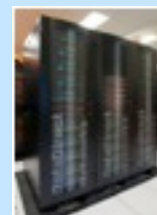 ~5K core cluster

**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

Major Compute Systems

- **Hopper**
  - Hopper is NERSC's flagship computer for running high-performance parallel scientific codes.
- **Franklin**
  - Franklin, an earlier-generation Cray computer, augments the Hopper system.
- **Carver**
  - Carver provides a generic full Linux environment for codes that need operating system features that are not available on the Cray systems or don't demand massive parallelism.

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 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.

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB | Lawrence Berkeley National Laboratory

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)

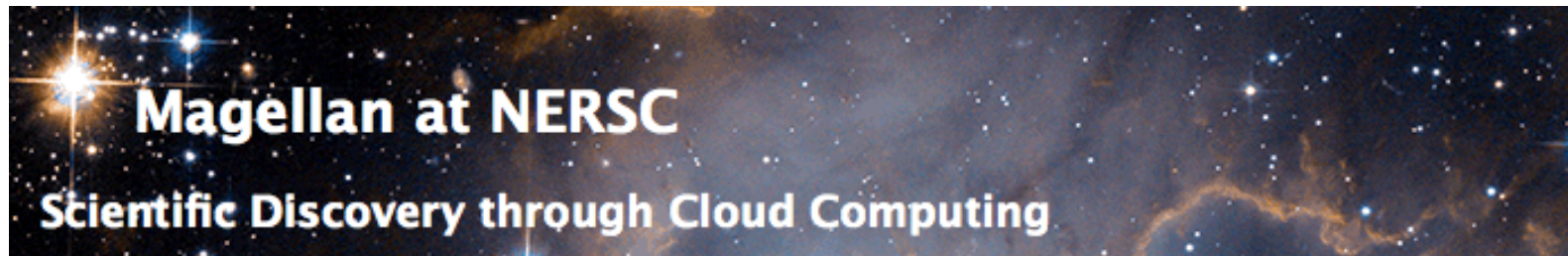2.5 GB / core for applications (5.5 GB / core on "fat" nodes)

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.

Magellan - IBM IDataPlex

Magellan at NERSC
Scientific Discovery through Cloud Computing

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 / core on "fat" nodes)

NERSC global /scratch directory quota of 20 TB

Full Linux operating system

PGI, GNU, Intel compilers

Aside: Why Do You Care About Parallelism?
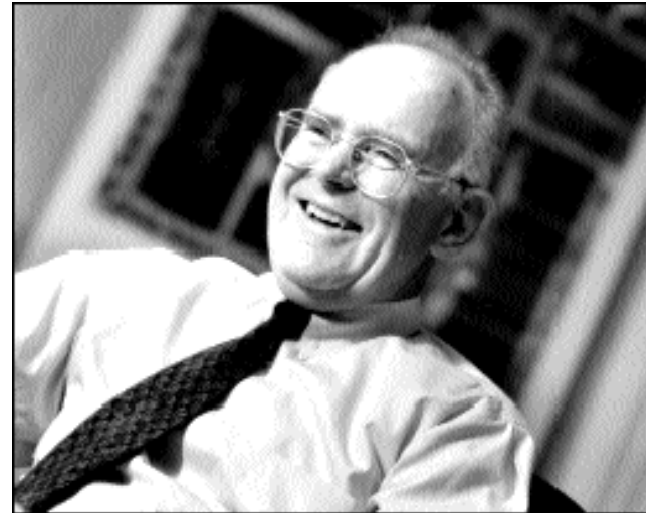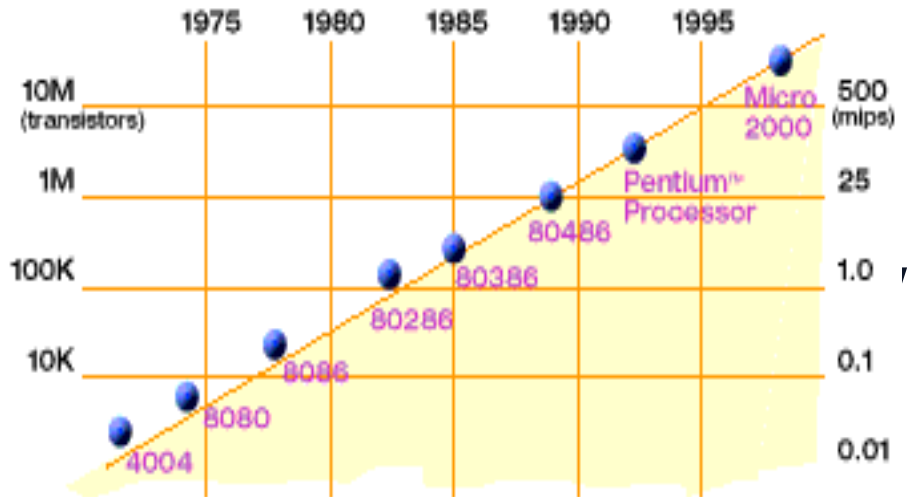
Moore's Law

2X transistors/Chip Every 1.5 years
Called "Moore's Law"
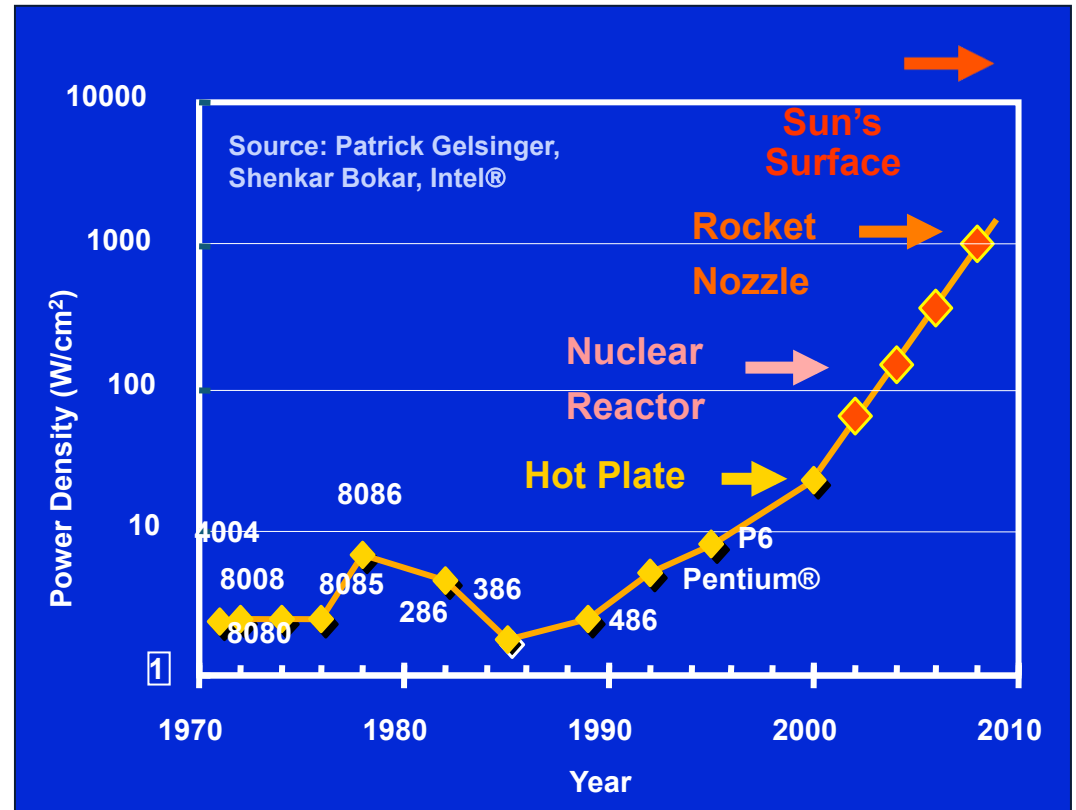 Microprocessors have become smaller, denser, and more powerful.

Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Slide source: Jack Dongarra

18

Power Density Limits Serial Performance

- Concurrent systems are more power efficient
  - Dynamic power is proportional to $V^2 fC$
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly
  - Save power by lowering clock speed

Source: Patrick Gelsinger, Shenkar Bokar, Intel®

- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- More transistors, but not faster serial processors
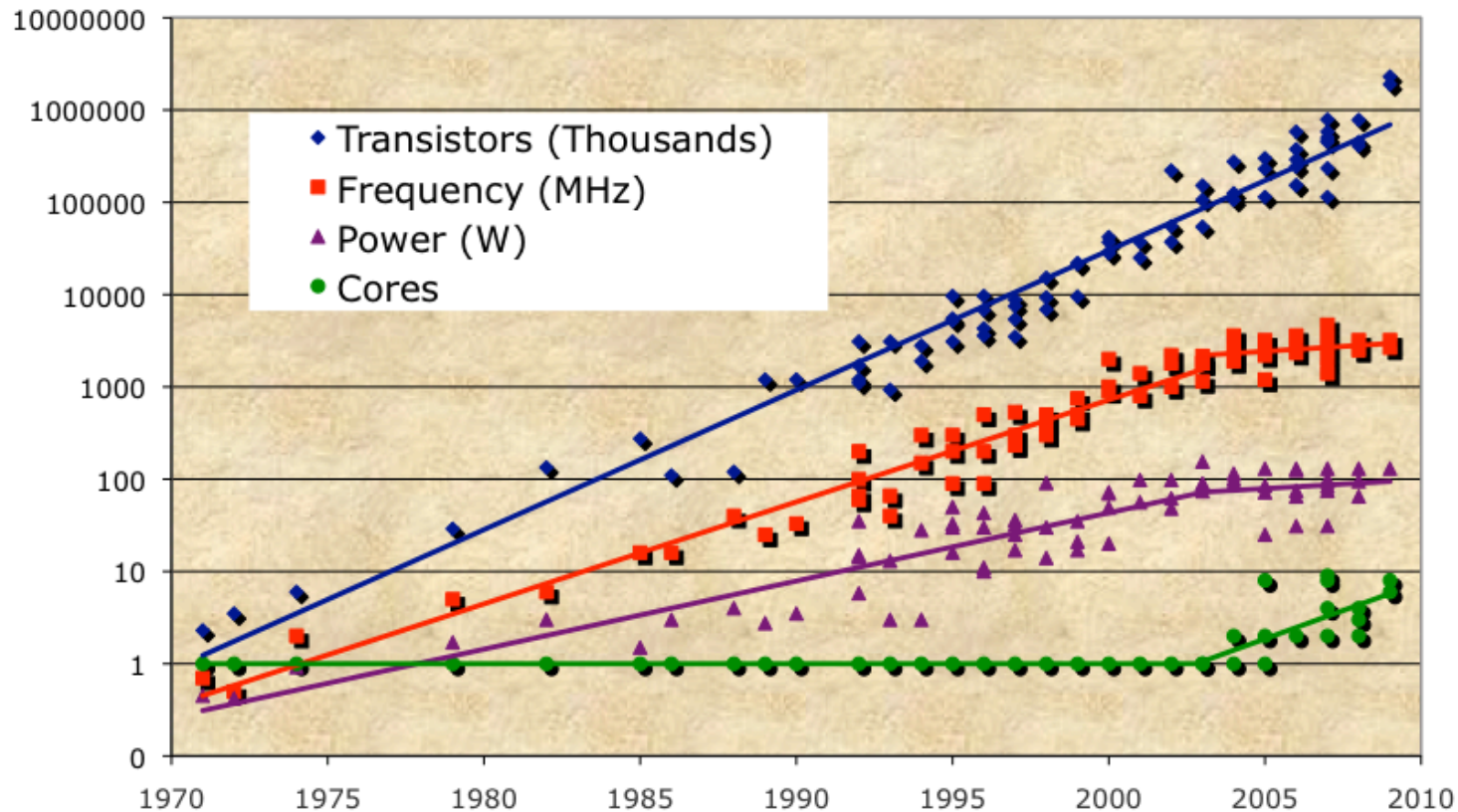
- **Chip density is continuing increase ~2x every 2 years**
- **Clock speed is not**
- **Number of processor cores may double instead**
- **Power is under control, no longer growing**

- **Number of cores per chip will double every two years**

- **Clock speed will not increase (possibly decrease)**

- **Need to deal with systems with millions of concurrent threads**

- **Need to deal with inter-chip parallelism as well as intra-chip parallelism**

- **Your take-away:**

  - *Future performance increases in computing are going to come from exploiting parallelism in applications*
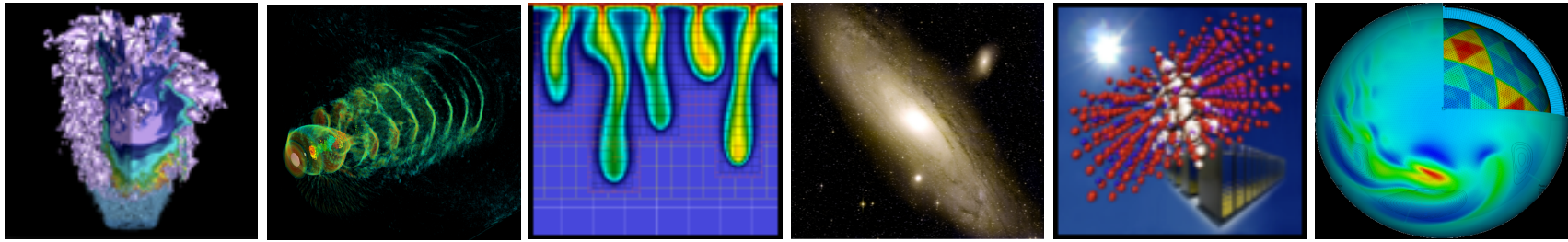
Data Storage Resources

- **"Spinning Disk"**
  - Interactive access
  - I/O from compute jobs
  - "Home", "Project", "Scratch"
  - Note: No on-node disk at NERSC

- **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 1M 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" file system.

- Each user has a personal directory referenced by $SCRATCH (and maybe $SCRATCH2).

- Significant 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 use a global scratch 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 (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.
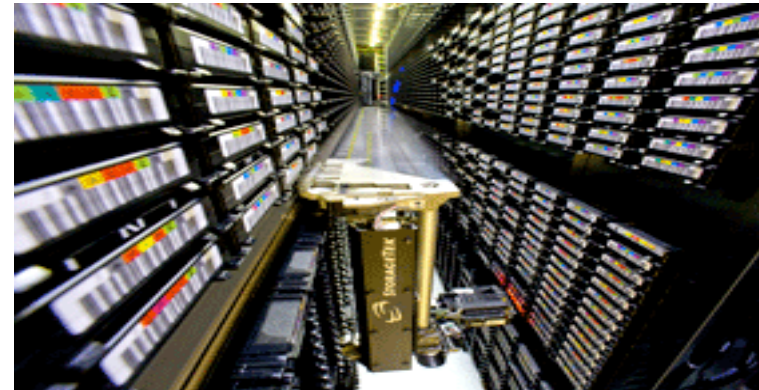
Archival Storage (HPSS)

For permanent, archival storage

You transfer files to and from HPSS using one of ftp, pftp, or the HPSS hsi client.

For offsite access 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.
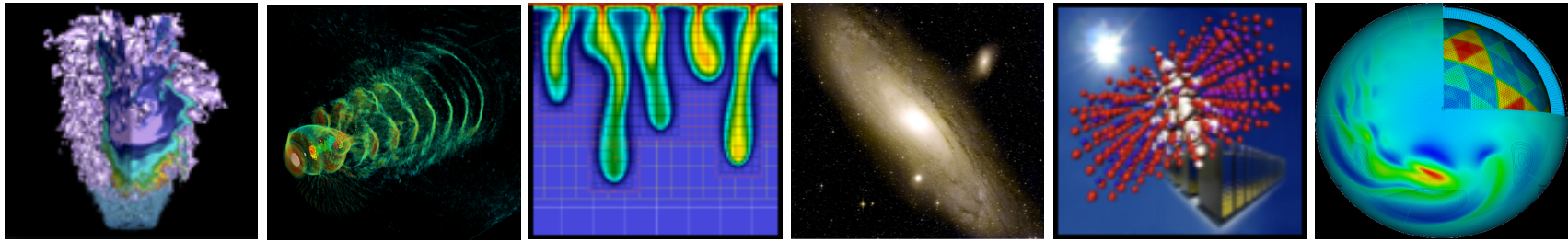
Hostname: archive.nersc.gov

Over 15 Petabyes of data stored

Data increasing by 1.7X per year

120 M files stored

150 TB disk cache

8 STK robots

44,000 tape slots

44 PB maximum capacity today

Average data xfer rate: 100 MB/sec

Navigating NERSC File Systems

A NERSC Training Event
NERSC Oakland Facility & Web Broadcast
Tuesday, May 3, 2011
10:00 a.m. PST

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

How to Get Help

NERSC Services

- **NERSC's emphasis is on enabling scientific discovery**

- **User-oriented systems and services**
  - This is what 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

- **Requirement-gathering workshops with top scientists**
  - Completed five, including BER
  - ***Microbial Genome and Metagenome Data Processing and Analysis with the IMG Family of Systems***; Victor M. Markowitz, LBNL; Natalia N. Ivanova and Nikos C. Kyrpides, Genome Biology Program, JGI

- **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/

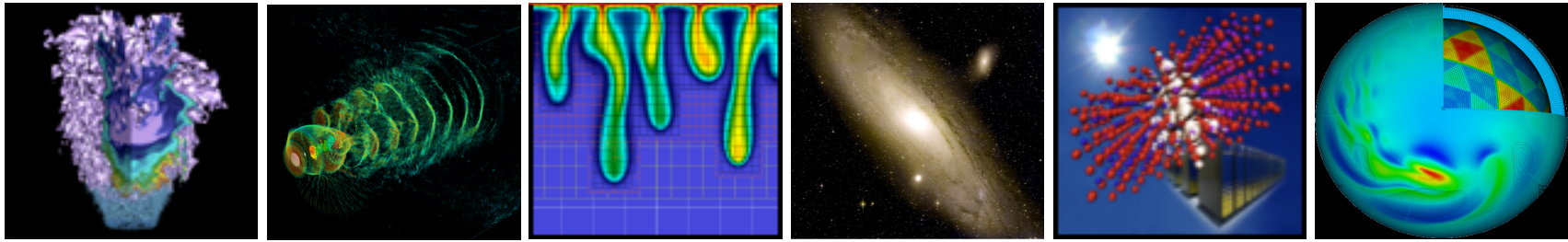## 1-800-666-3772 (or 1-510-486-8600)

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

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

- 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 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

Passwords and Resets

Passwords

New users must call Account Support for an initial temporary password

Change it at https://nim.nersc.gov

Answer security questions, then you can reset it at https://nim.nersc.gov if you forget it

Login Failures

3 or more consecutive login failures will disable your ability to log in

Send e-mail to accounts@nersc.gov or call 1-800-66-NERSC to reset your failure count
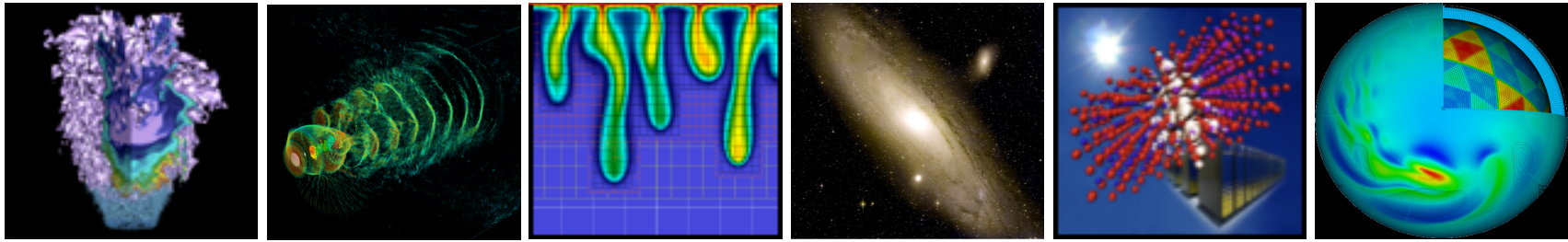
Connecting to NERSC

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 1-800-66-NERSC 24x7 or send email to accounts@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 hopper.nersc.gov`

- Your can use grid-based tools (e.g. gridftp) also; please ask the NERSC consultants for details.

Computing Environment

U.S. DEPARTMENT OF ENERGY | Office of Science

NeRSC — National Energy Research Scientific Computing Center

BERKELEY LAB — Lawrence Berkeley National Laboratory

- 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 carefully manage your files.

- 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/
```

- 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)

- 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 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.)

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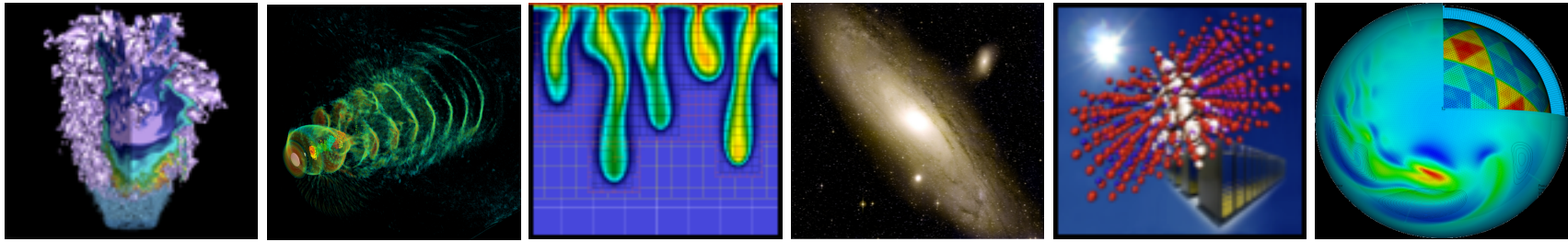
- The "namd2" binary for version 2.7b2 is now in your UNIX search path.

Hands On

Connecting and Environment

https://www.nersc.gov/users/training/events/jgi/hands-on/

Compiling Code

- **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.

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

- **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)

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.

```
% 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
```

Compiling Code

https://www.nersc.gov/users/training/events/jgi/hands-on/

Running Jobs

Jobs at NERSC

- **Most jobs are parallel, using 10s to 100,000+ cores.**

- **Many use custom codes; others use pre-installed applications**

- **Typically run a few hours, up to 48. Longer runs can be accommodated if needed and logistically possible.**

- **Many jobs "package" lower concurrency runs into one job**

 - Even many "serial jobs"

  - Load balance may be an issue

Login Nodes and Compute 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
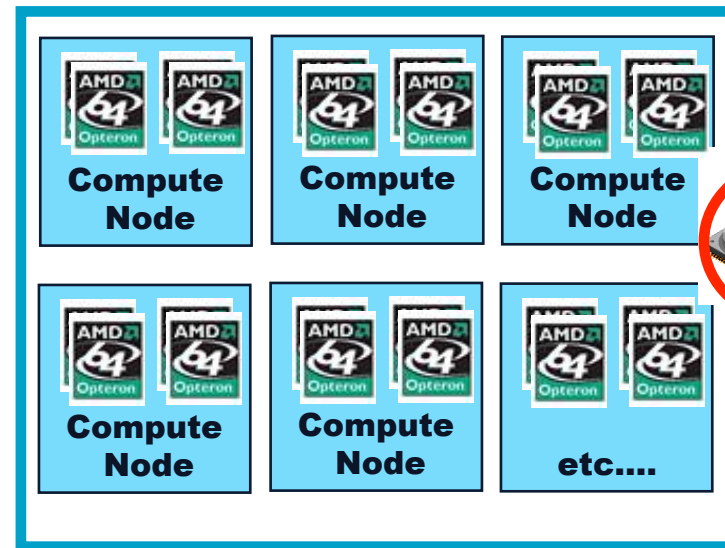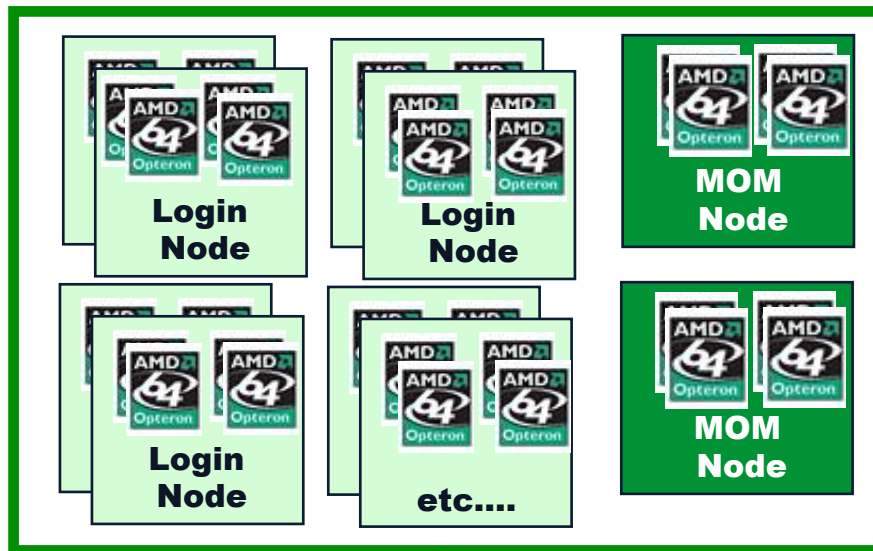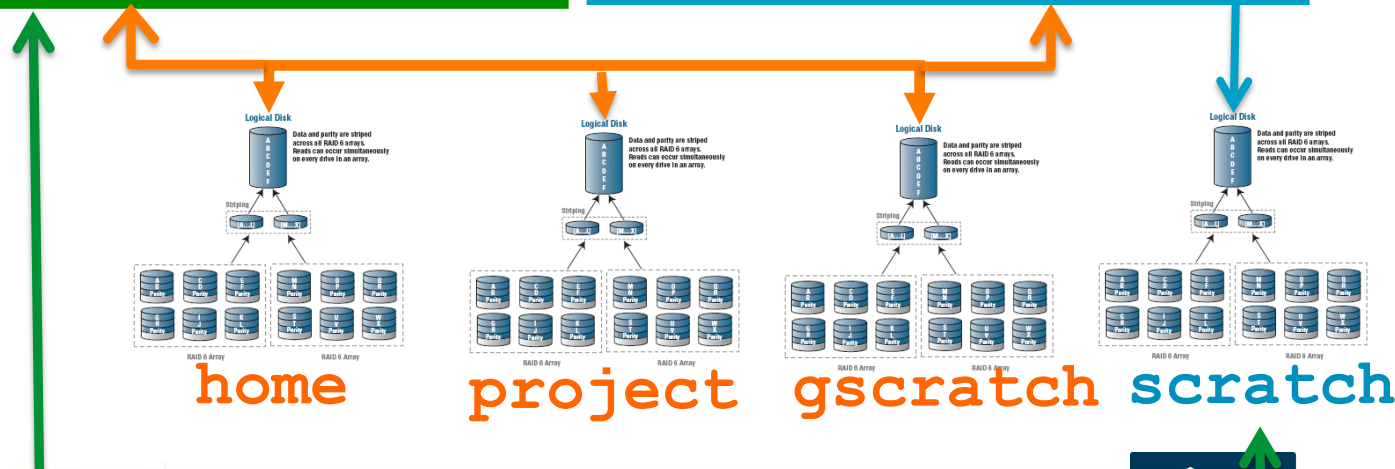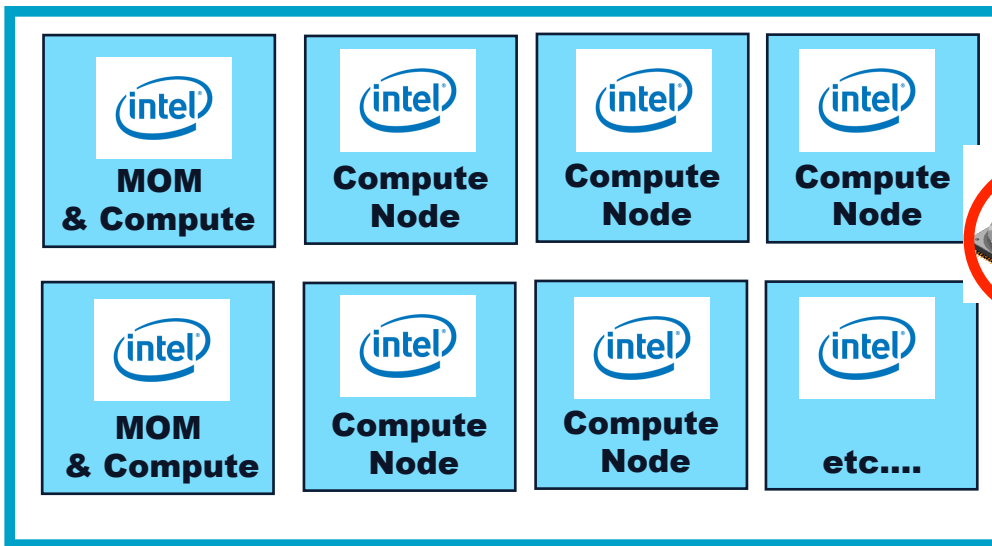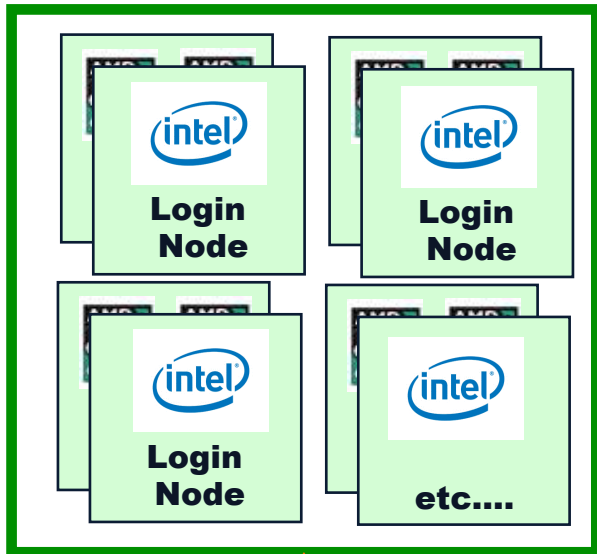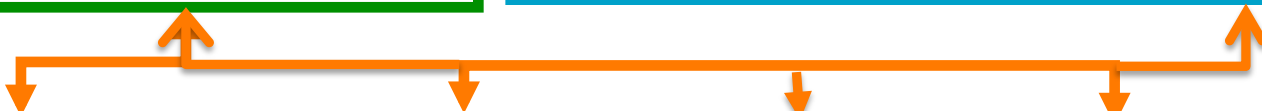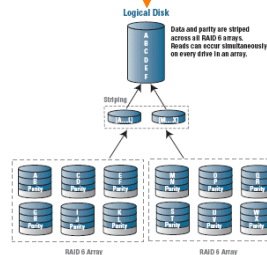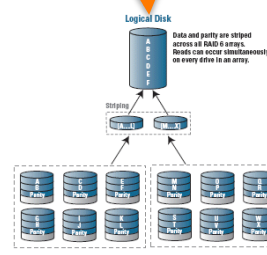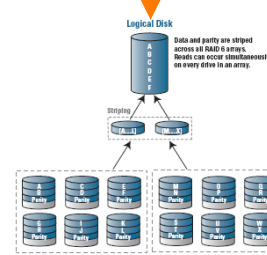
- A "job launcher" distributes your code to all the nodes in your parallel job, starts them, and manages their execution.
- On Cray the job launcher is called "aprun" and on other systems it is "mpirun".
- Only the job launcher can start your job on compute nodes
- You can't run the job launcher from login nodes

- 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`

```
#PBS -q debug
#PBS -l mppwidth=96
#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 96./my_executable
```

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

- 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"

- 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
```

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB | Lawrence Berkeley National Laboratory

Running Jobs

https://www.nersc.gov/users/training/events/jgi/hands-on/