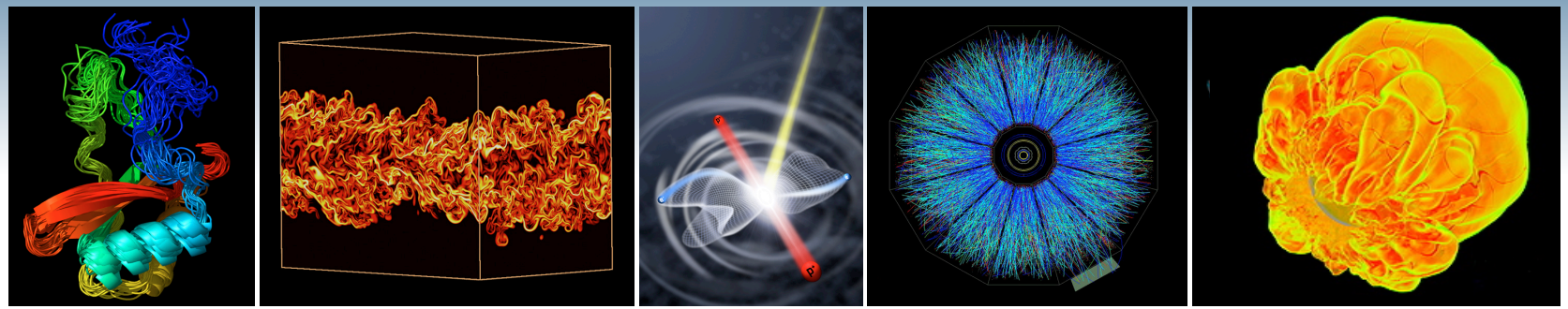


# NERSC Systems and Services Available to You


**Jack Deslippe**
**HPC Consultant, NERSC User Services**


U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

# NERSC is the Primary Computing Center for DOE Office of Science


## NERSC computing for science

- 5000 users, 650 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
- N7 Coming in Next Year
  - Additional smaller clusters

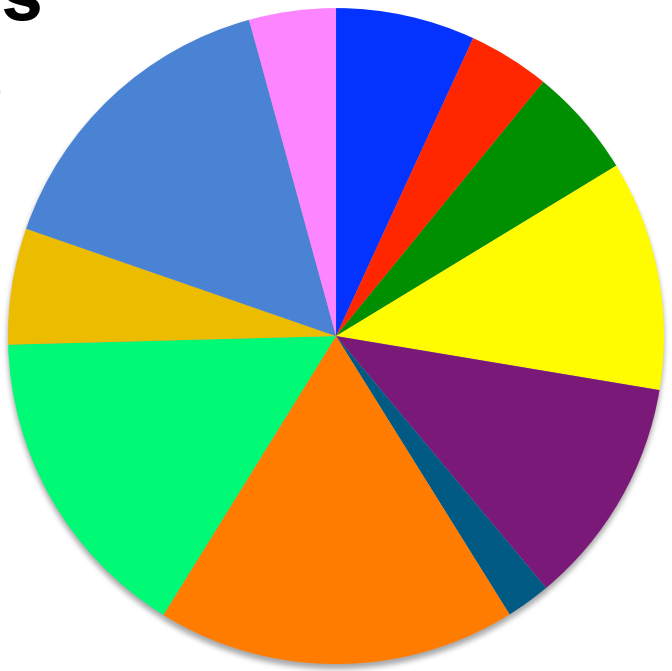

U.S. DEPARTMENT OF ENERGY | Office of Science



BERKELEY LAB
Lawrence Berkeley National Laboratory

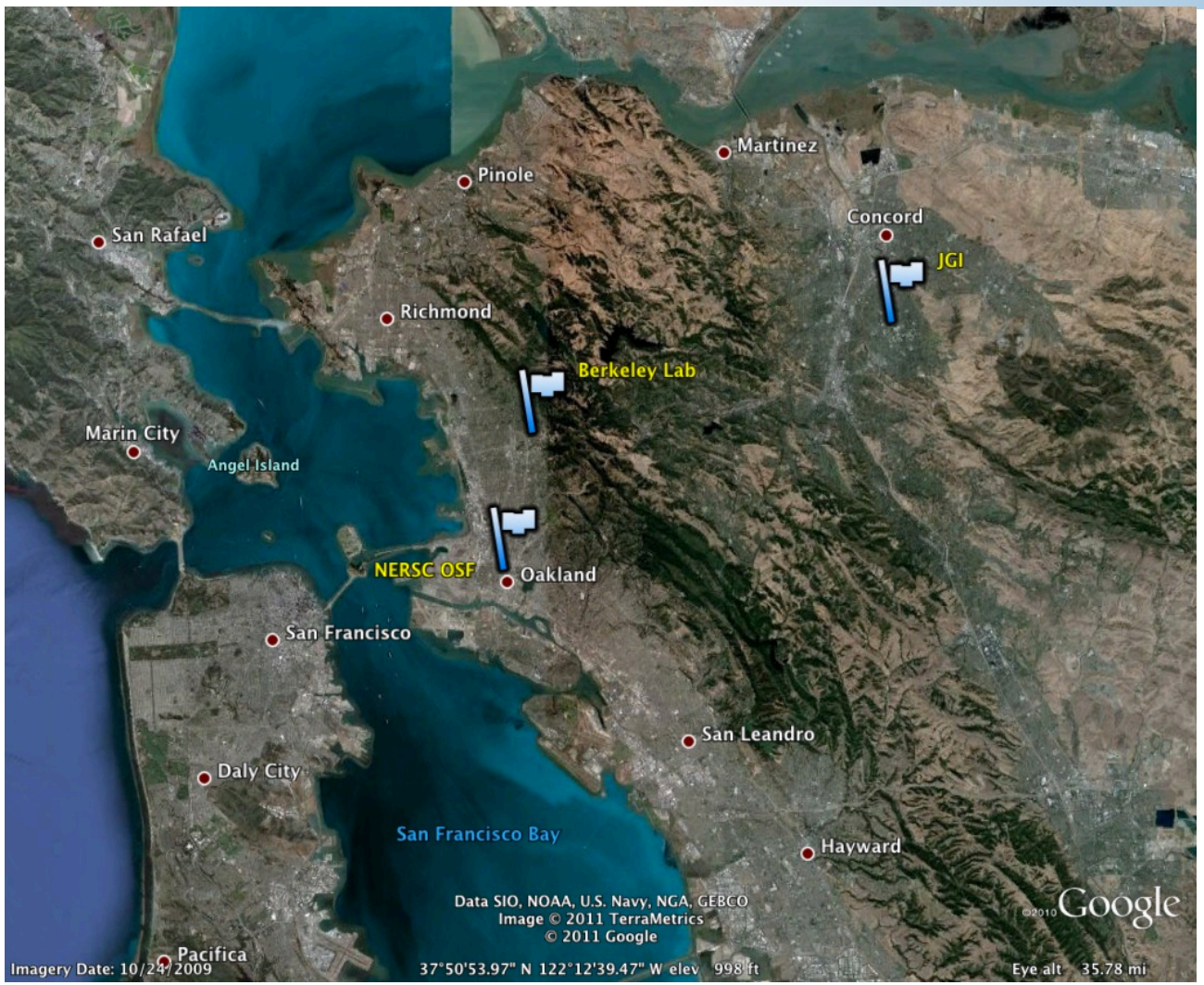
# NERSC is the Primary Computing Center for DOE Office of Science

- NERSC serves a large population
- Focus on "unique" resources
  - Expert consulting and other services
  - High end computing systems
  - High end storage systems




*"NERSC continues to be a gold standard of a scientific High Performance Computational Facility."*
– HPCOA,Review August 2008

# 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



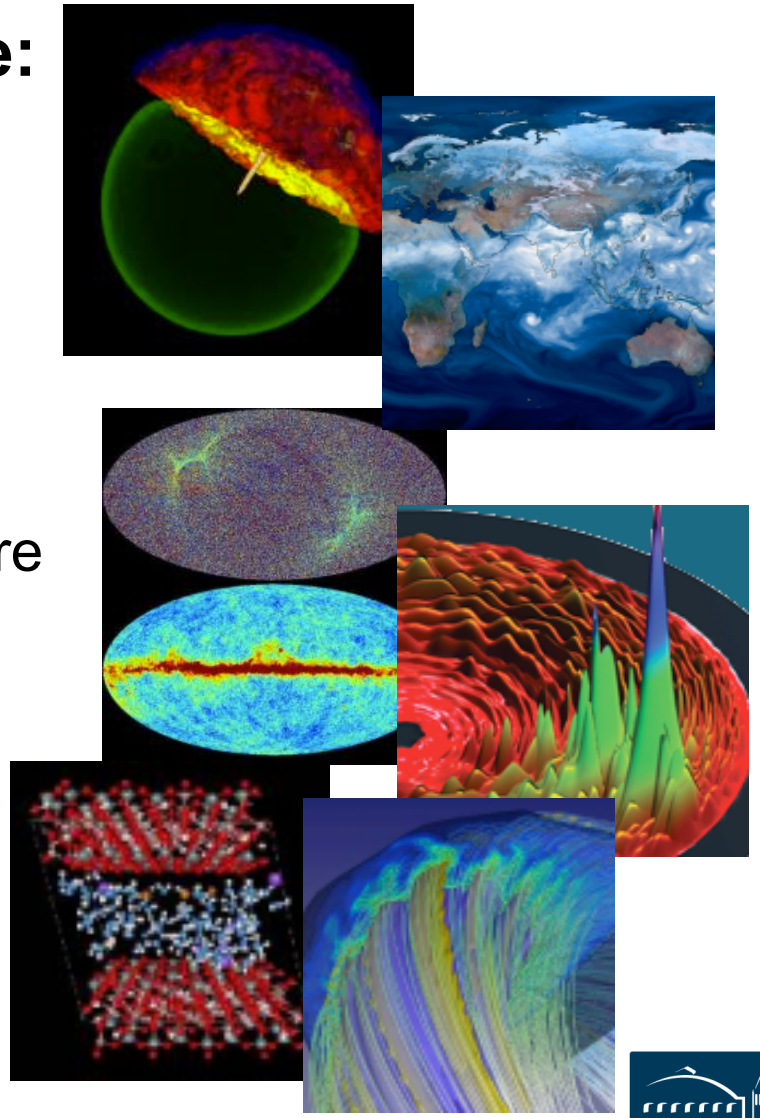
BERKELEY LAB
Lawrence Berkeley National Laboratory

# Location


OSF
Image © 2011 TerraMetrics
© 2011 Europa Technologies
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google
Imagery Date: 9/10/2010
37°48'33.87" N 122°16'05.91" W elev 30 m
Eye alt 391 m

# NERSC Strategy: *Science First*

- **Support computational science:**
  - Provide effective **machines that support fast algorithms**
  - Deploy with **flexible systems software** to run a **broad range** of applications
  - Help users with **expert services**
  - Develop **tools** to make systems more accessible
- **NERSC future priorities are driven by science:**
  - Increase application capability: **"usable Exascale"**
  - **Simulation and data analysis** of simulated and experimental data



# Data Analysis Grows more Automated with the Explosion of Scientific Data Sets

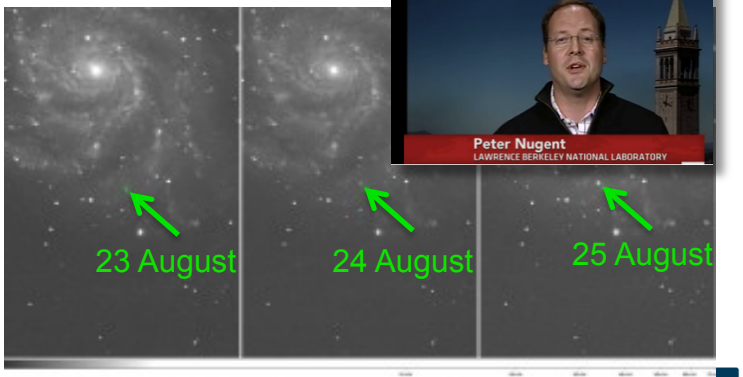
## NERSC used in 2011 Nobel Prize

- Supernovae used as “standard candles” to measure distance.
- Simulations at NERSC in late 90s modeled the appearance from Earth.

## More recently: astrophysics discover early nearby supernova.

- Discovered within hours of its explosion, a rare glimpse at the supernova’s outer layers reveal what kind of star exploded.
- The last such supernova was in 1972. Before that: 1937, 1898 and 1572
- NERSC accepts ~300GB/night and runs machine learning algorithms to process images and detect new transients;

The research shows that the universe is expanding at an accelerating rate. The nature of the dark energy force behind this may be the most important problem in 21st century physics.


U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

# NERSC Systems

## Large-Scale Computing Systems


**Hopper (NERSC-6): Cray XE6**

- 6,384 compute nodes, 153,216 cores
- 1.3 Pflop/s peak

- N7 Coming in 2013

## Clusters

**Carver**

- IBM iDataplex cluster

**PDSF (HEP/NP)**

- ~1K core cluster

**GenePool (JGI)**

- ~5K core cluster

## NERSC Global Filesystem (NGF)

Uses IBM's GPFS

- 1.5 PB capacity
- 10 GB/s of bandwidth

## HPSS Archival Storage

- 40 PB capacity
- 4 Tape libraries
- 150 TB disk cache

## Analytics

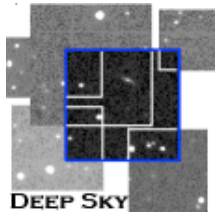
**Euclid**

(512 GB shared memory)

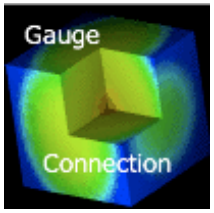
**Dirac** GPU testbed (48 nodes)

# Develop and Provide Science Gateway Infrastructure

- **Goals of Science Gateways**
  - Allow sharing of data on NGF and HPSS
  - Make scientific computing easy
  - Broaden impact/quality of results from experiments and simulations
- **NEWT – NERSC Web Toolkit/API**
  - Building blocks for science on the web
  - newt.nersc.gov

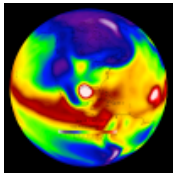



Deep Sky: 450+ Supernovae



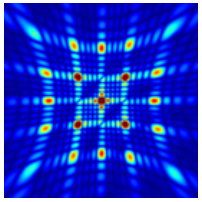

Gauge Connection: QCD


Daya Bay: Real-time processing and monitoring



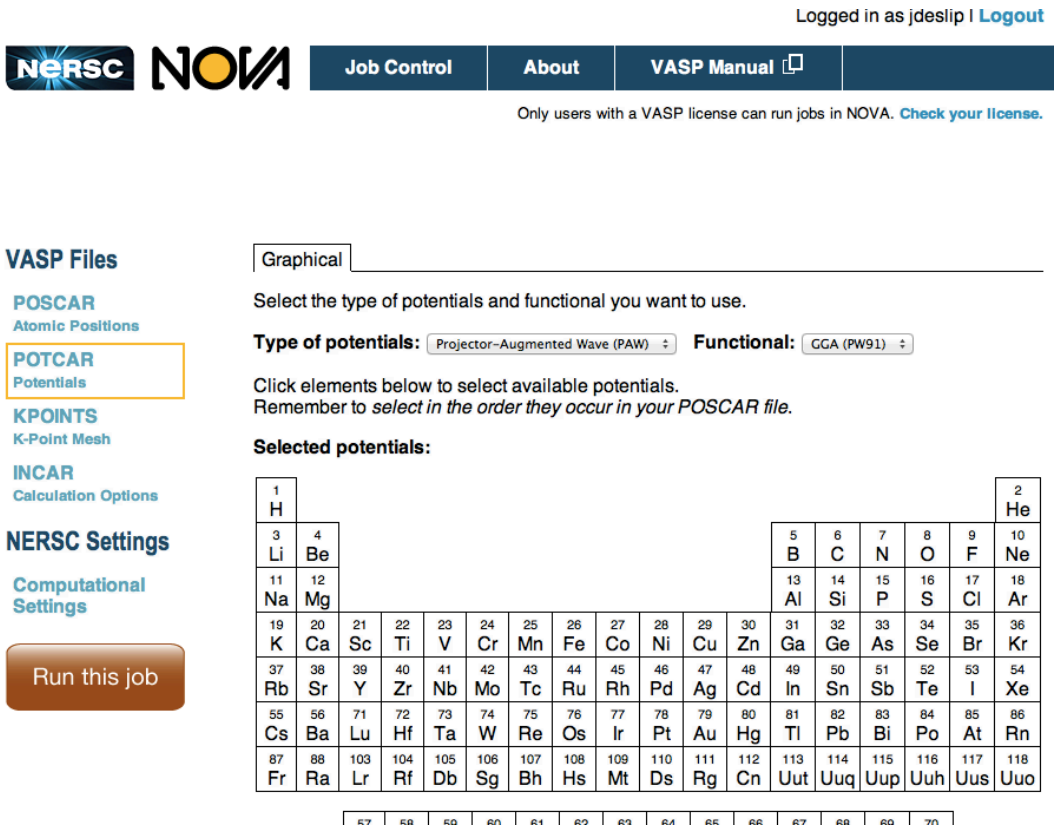
20th Century Reanalysis

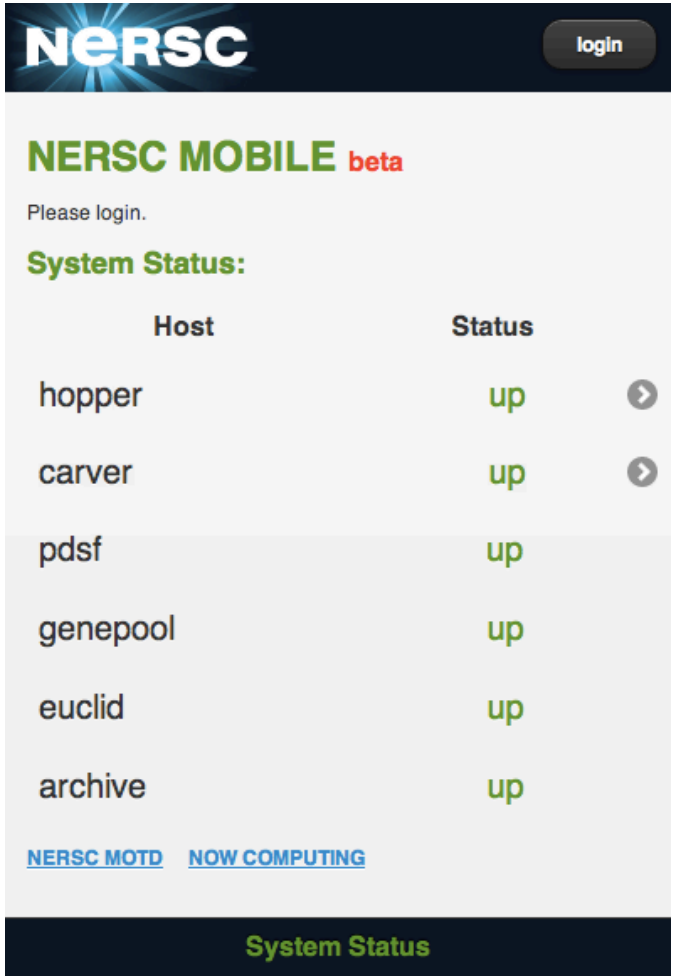

Earth Systems Grid



Coherent X-Ray Imaging Data Bank

# NEWT Apps


Logged in as jdeslip | Logout
NERSC NOVA
Job Control
About
VASP Manual
Only users with a VASP license can run jobs in NOVA. Check your license.
VASP Files
POSCAR
Atomic Positions
POTCAR
Potentials
KPOINTS
K-Point Mesh
INCAR
Calculation Options
NERSC Settings
Computational Settings
Run this job
Graphical
Select the type of potentials and functional you want to use.
Type of potentials: Projector-Augmented Wave (PAW)
Functional: GGA (PW91)
Click elements below to select available potentials.
Remember to select in the order they occur in your POSCAR file.
Selected potentials:
H He
Li Be B C N O F Ne
Na Mg Al Si P S Cl Ar
K Ca Sc Ti V Cr Mn Fe Co Ni Cu Zn Ga Ge As Se Br Kr
Rb Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn Sb Te I Xe
Cs Ba Lu Hf Ta W Re Os Ir Pt Au Hg Tl Pb Bi Po At Rn
Fr Ra Lr Rf Db Sg Bh Hs Mt Ds Rg Cn Uut Uuq Uup Uuh Uus Uuo


NERSC
login
NERSC MOBILE beta
Please login.
System Status:
Host
Status
hopper up
carver up
pdsf up
genepool up
euclid up
archive up
NERSC MOTD
NOW COMPUTING
System Status

# NERSC HPC Machines Overview



11

# NERSC-6
# Grace "Hopper"


**Cray XE6**

1.3 PF Peak

**Processor**

AMD MagnyCours

2.1 GHz 12-core

8.4 GFLOPs/core

24 cores/node

32-64 GB DDR3-1333 per node

**System**

Gemini Interconnect (3D torus)

6384 nodes

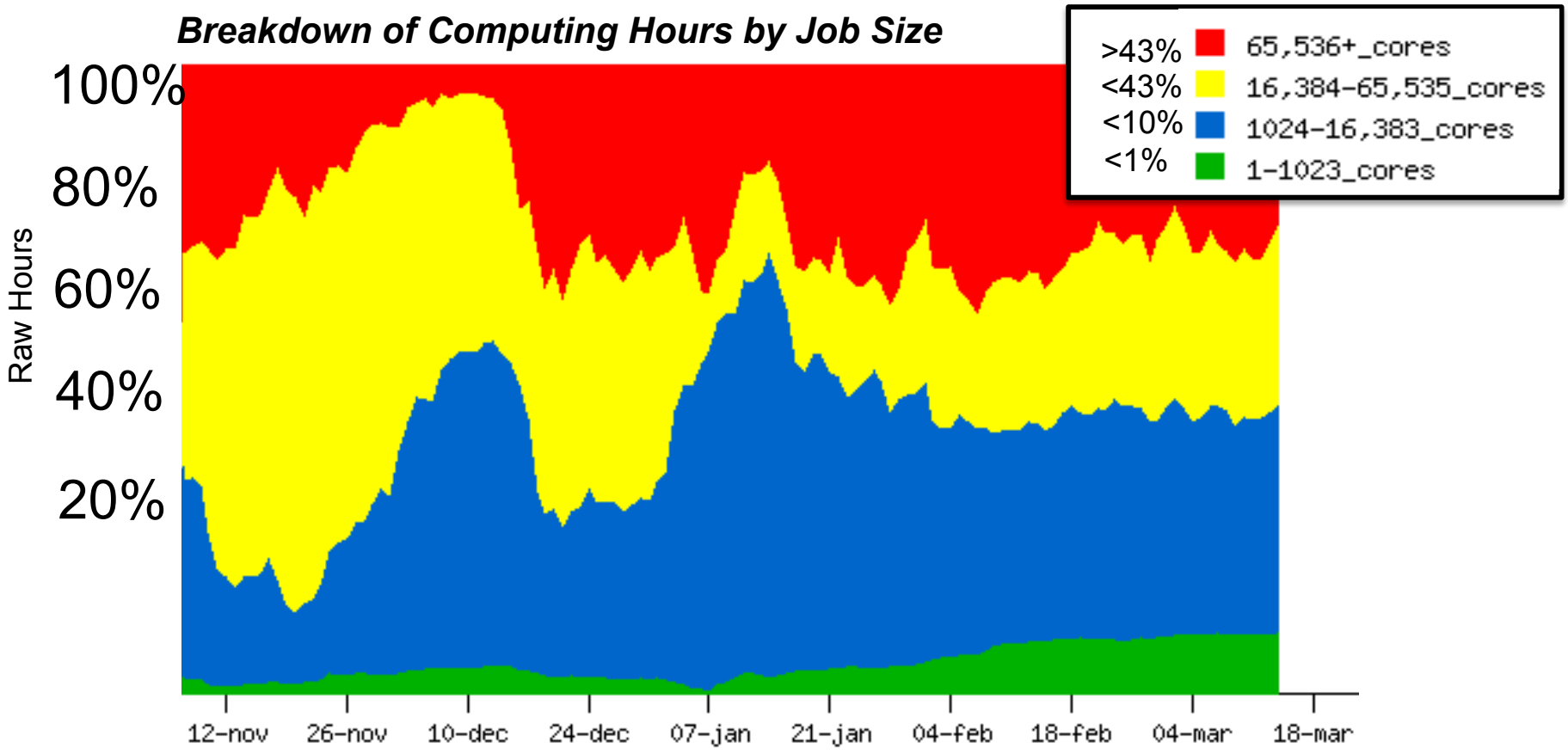
153,216 total cores

**I/O**

2PB disk space

70GB/s peak I/O Bandwidth

# Hopper Job Size Mix



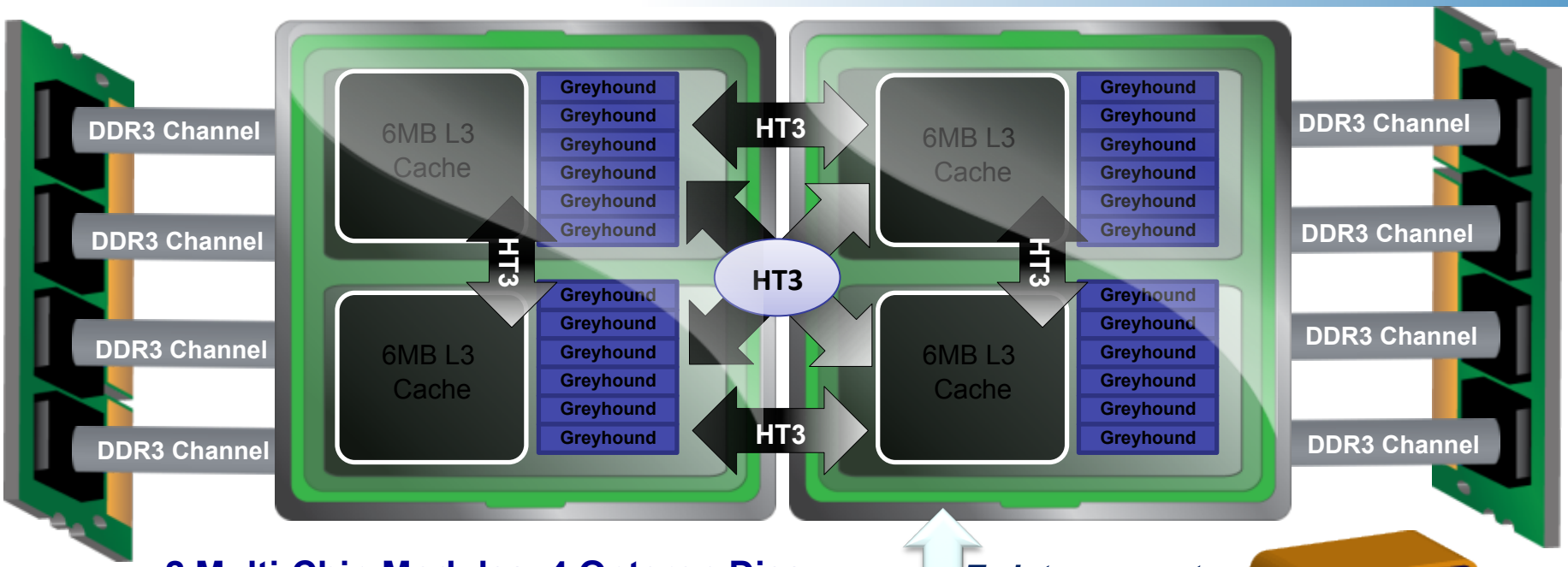

- *Hopper is a 153,216 core system.*

# Efficiently using Hopper and preparing yourself for future trends

- CPU Clock rates are stalled (not getting faster)
  - # nodes is about the same, but # cores is growing exponentially
  - Think about parallelism from node level
  - Consider hybrid programming to tackle intra-node parallelism so you can focus on # of nodes rather than # of cores
- Memory capacity not growing as fast as FLOPs
  - Memory per node is still growing, but per core is diminishing
  - Threading (OpenMP) on node can help conserve memory
- Data locality becomes more essential for performance
  - NUMA effects (memory affinity: must always be sure to access data where it was first touched)




# XE6 Node Details: 24-core Magny Cours




- **2 Multi-Chip Modules, 4 Opteron Dies**
- **8 Channels of DDR3 Bandwidth to 8 DIMMs**
- **24 (or 16) Computational Cores**
  - 64 KB L1 and 512 KB L2 caches for each core
  - 6 MB of shared L3 cache on each die
- **Dies are fully connected with HT3**

Images Courtesy of Cray Inc.

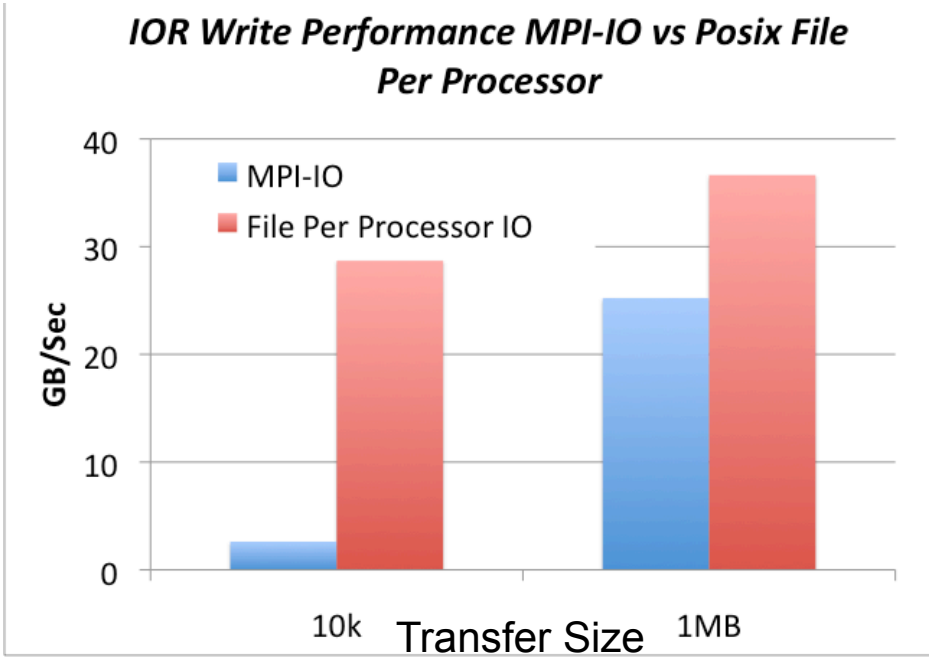
# File Systems

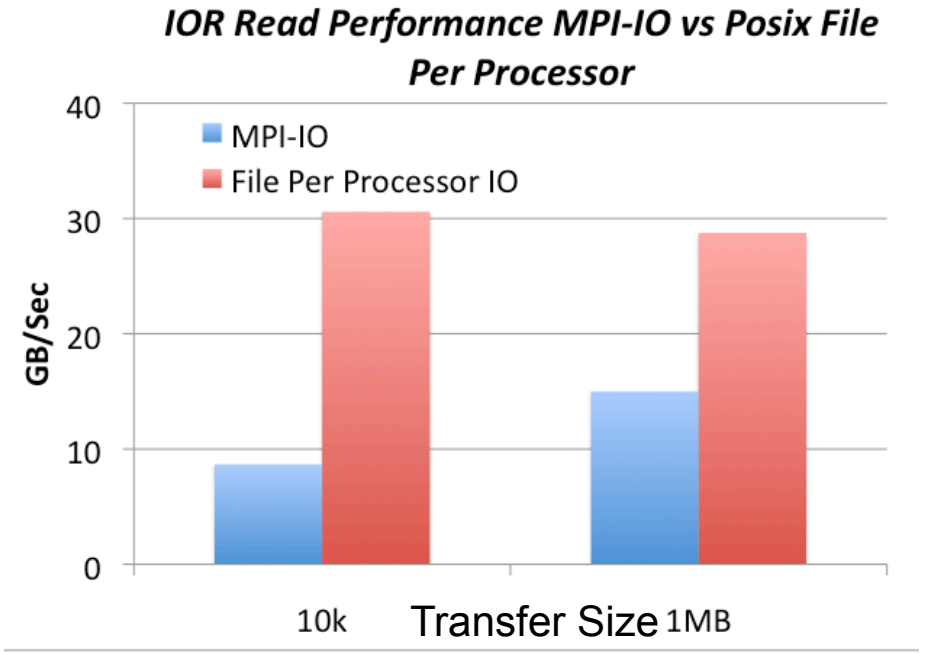
- **$HOME**
  - Where you land when you log in
  - Tuned for small files
- **$SCRATCH and $SCRATCH2**
  - Tuned for large streaming I/O
- **$GSCRATCH**
  - Mounted across all NERSC file system
- **$PROJECT**
  - Sharing between people/systems
  - By request only

# 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

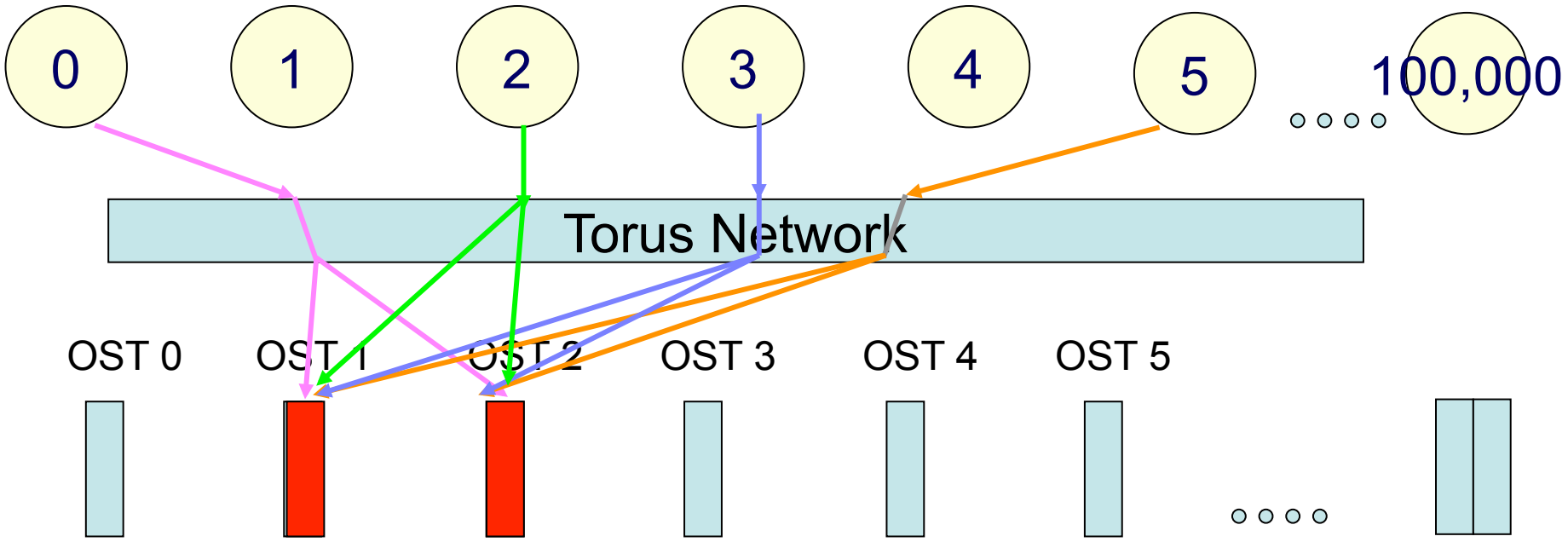
# IO Test using IOR benchmark on 576 cores on Hopper with Lustre file system



# Use striping to improve I/O Performance

Files are striped across different disks. Example stripe count = 2 – Not optimal.




For large shared files, increase the stripe count

# Striping Can Improve Performance

When striping set on a directory: all files created in that directory will inherit striping set on the directory

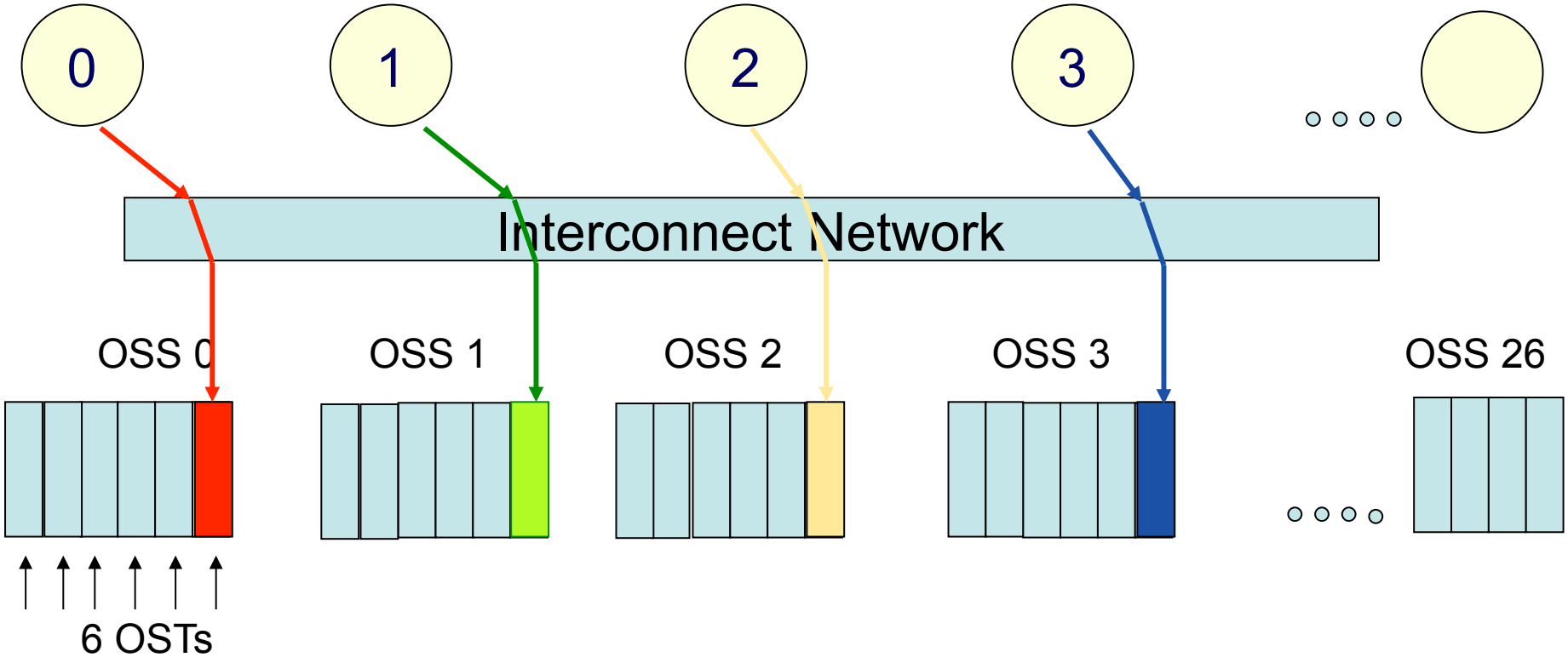
lfs setstripe <directory|file> -c stripe-count

Stripe count - # of disks file is split across

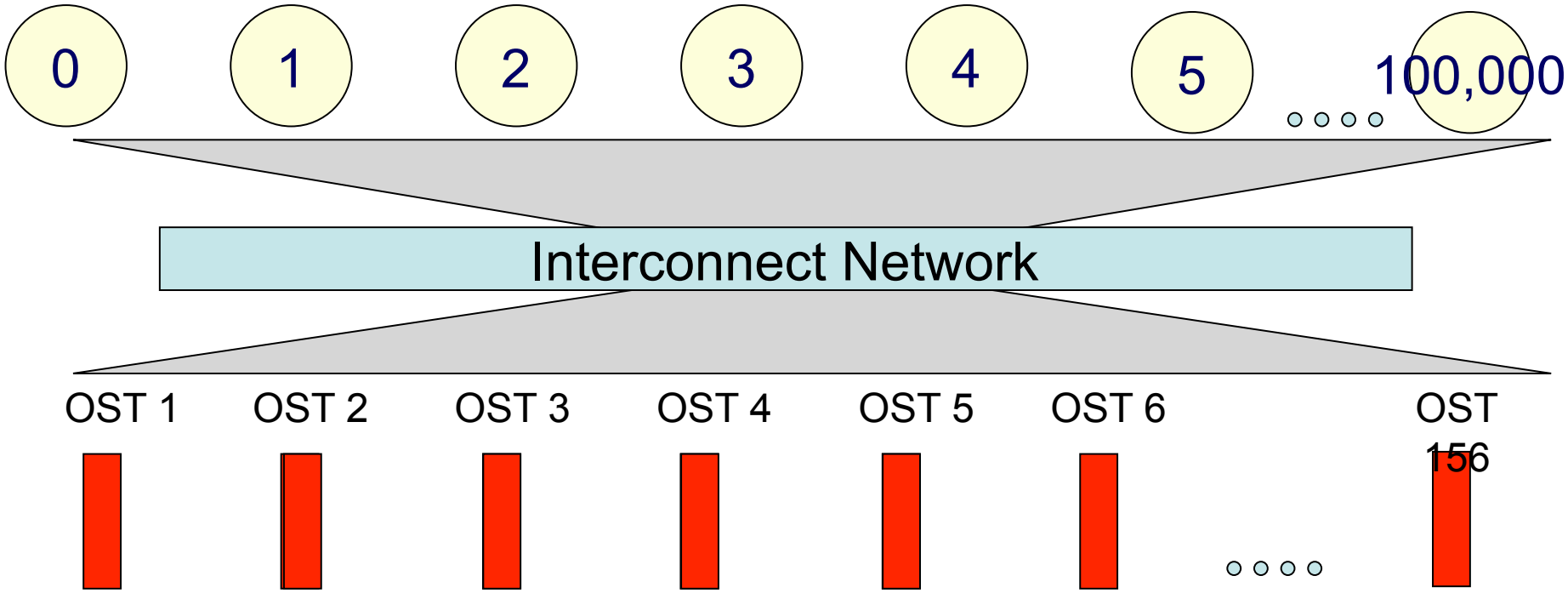
Example: change stripe count to 10

lfs setstripe mydirectory -c 10

For one-file-per-processor workloads set the stripe count to 1 for maximum bandwidth and minimal contention



# Striping over all OSTS increases bandwidth available to application




The next table gives guidelines on setting the stripe count

# 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

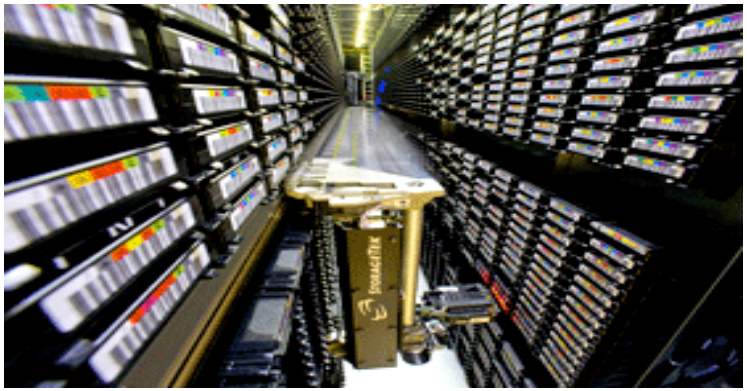
Use Carver for jobs that use up to 512 cores, need a fast CPU, need a standard Linux configuration, or need up to 48 GB of memory on a node.

# 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 more info see the NERSC web site: type “hpss getting started” in the search box



**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**


# What services are available to you?

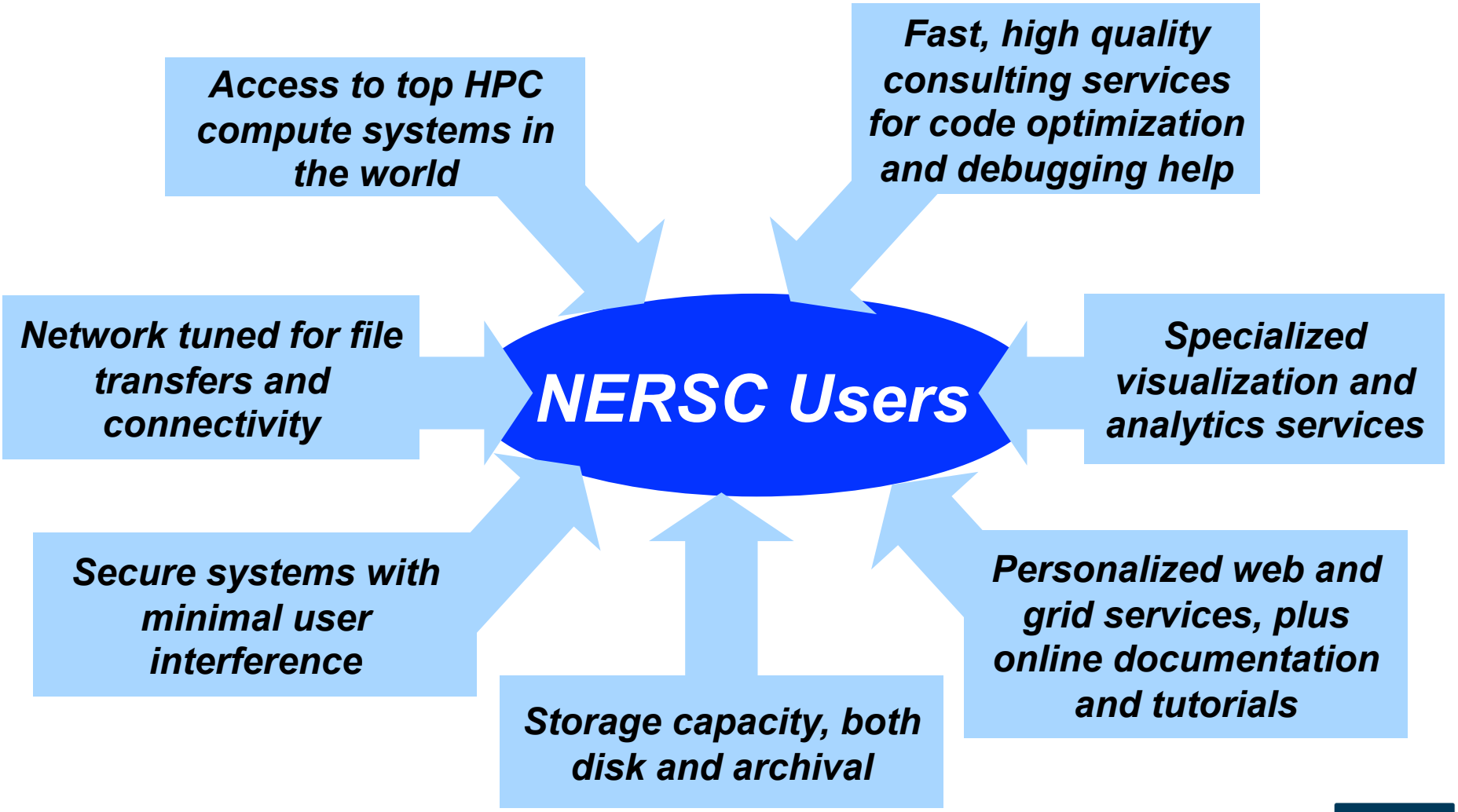


U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

# All NERSC Systems and Services are available to you

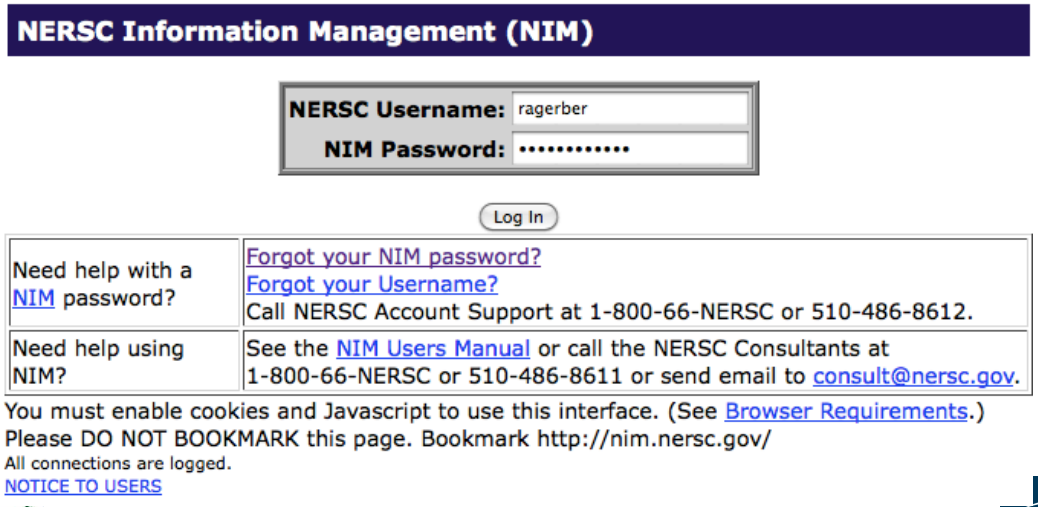


# Getting enabled to run at NERSC

- To be able to run at NERSC you need to have an *account* and an *allocation*.
- An *account* is a username and password
  - Simply fill out the Computer Use Policy Form (https://www.nersc.gov/users/accounts/user-accounts/nersc-computer-use-policies-form/)
  - Fax form to NERSC
  - Receive email with link to initial password
- An *allocation* is a repository of CPU hours

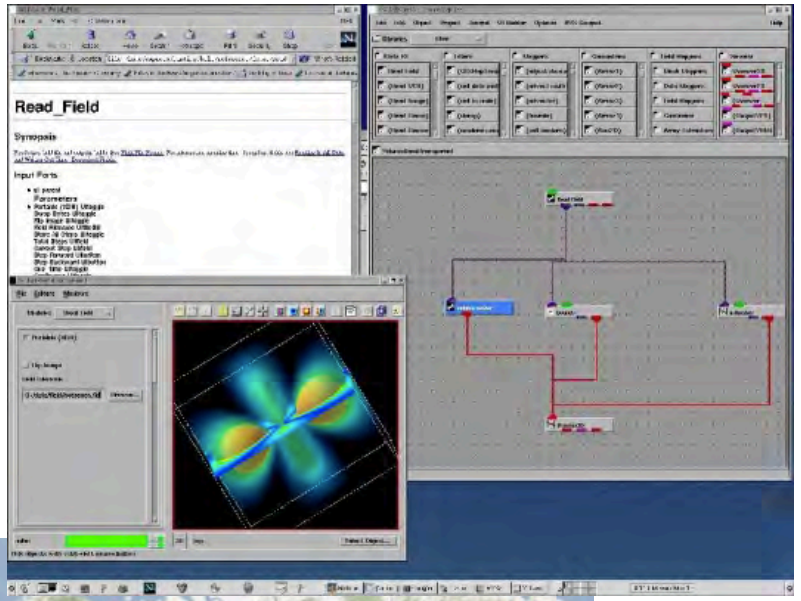
# Accounting Web Interface (NIM)

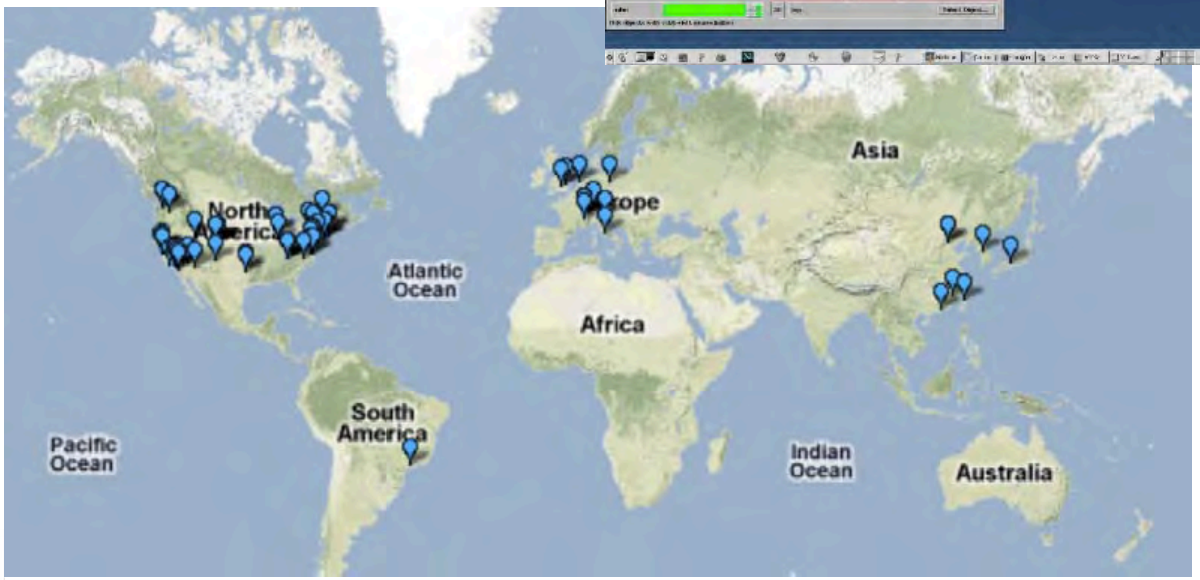
- Log into the NERSC NIM web site at https://nim.nersc.gov/ to manage your NERSC accounts.
- In NIM you can check your daily allocation balances, change your password, run reports, update your contact information, change your login shell, etc.



# NX Provides Faster Remote Visualization

- **NX Servers plus client software**
- **Used worldwide for**
  - **–Scientific data visualization**
  - **–Remote debugging with GUIs**



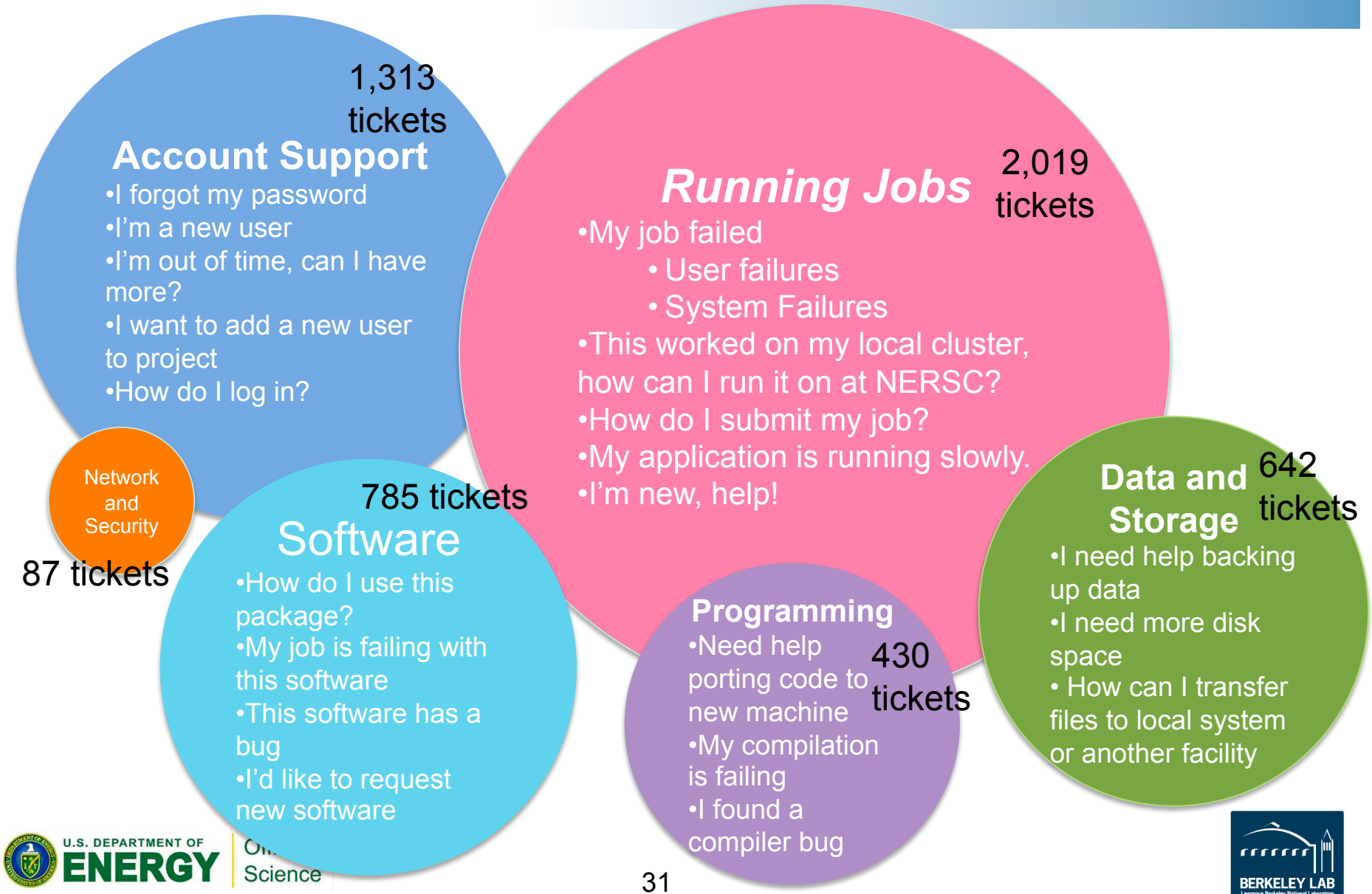



# Consulting Services are available to you

- NERSC users submit online tickets or call account support and consultants weekdays between 8am-5pm Pacific Time
- 2 Account support staff
- 8 Consultants
  - Diverse backgrounds from computer science to science domain expertise
  - Highly skilled: ½ of consultants have PhDs in science domain, other ½ have master's degrees
  - Focus on quality responses

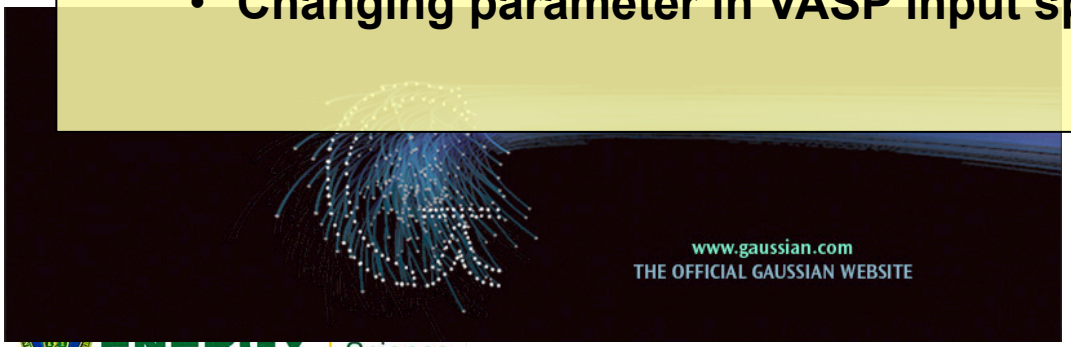
"One thing that I love about NERSC is that they think in a way that is like a researcher, not as a system administrator."
–Guoping Zhang, Indiana State University

# Common Questions to NERSC Consultants



# Software Support: Chemistry & Materials Applications

- More than 13.5 million lines of source code Compiled, Optimized, and Tested
  - *"The 3.2 version of PWSCF built by the NERSC staff is very fast. We appreciate the consulting staff's effort in providing optimized software for the users."*
- Expert advice provided on using these applications
  - Bridging gap between application science and computer science
  - Changing parameter in VASP input sped up calculations by 2X



# NERSC Uses Modules to manage Software

- Find all pgi compiler modules on the system

```
kantypas@login2:~> module avail pgi

--------------------------- /opt/modulefiles ----------------
pgi/10.9.0          pgi/11.0.0          pgi/11.1.0(default)
```

- Swap to an earlier version

```
kantypas@login2:~> module swap pgi pgi/10.9.0
```

- Other commands are “load”, “unload”, “avail”, “switch”

# Underneath Modules

- No magic in module files – simple environment variables
- The software is there, Modules files just point to it.

```
kantypas@login2:~> module show python
-------------------------------------------------------------------
/soft/modulefiles/compilers/python/2.7.1:

module-whatis     Sets up Python in your environment
Switching to GNU compiler environment
module            switch PrgEnv-pgi PrgEnv-gnu
module            switch xt-mpt xt-mpich2
prepend-path      PATH /soft/python/2.7/2.7.1/bin
prepend-path      LD_LIBRARY_PATH /soft/python/2.7/2.7.1/lib
prepend-path      MANPATH /soft/python/2.7/2.7.1/share/man
prepend-path      C_INCLUDE_PATH /soft/python/2.7/2.7.1/include
setenv            PYTHON_HOME /soft/python/2.7/2.7.1
setenv            PYTHON_VERSION 2.7
```

# Tips for new users

- Challenge yourself to learn a little bit about HPC architecture
  - To use systems well you need to understand conceptual design, otherwise too many things are mysterious
- Attend workshops and online tutorials
- Profile your code
- Ask consultants questions – we are here to help.

# Getting Started With Your First Jobs

## HomeWork Activities:

1. Logging In
2. Compiling + Submitting a Batch Job
3. Submitting a Hybrid Calculation

# Activity 1: Logging In

**% ssh *username*@hopper.nersc.gov**

***This will put you on one of the 8 Hopper login nodes***

- **These nodes have a full OS**
- **Edit files**
- **Compile programs**
- **Submit jobs to *compute nodes***
- ***DON'T use login nodes compute intensive applications***
- ***Shared between all Hopper users***

# Examples

**Basic examples are in:**

**/project/projectdirs/training/jul-2012/compile**

- **Copy necessary files to your $HOME directory as you don't have write permissions in the directory jul-2012**
- **If you haven't run on a cray before, take some time to go over a few simple examples**

# Activity 2: Compile Hands On

***In directory***

***/project/projectdirs/training/jul-2012/compile***

- **First Example:**

**% cp** ***/project/projectdirs/training/jul-2012/compile/mpi_test.f90 ~***

**% cp** ***/project/projectdirs/training/jul-2012/compile/submit_static.scr ~***

```
% ftn mpi_test.f90 -o mpi_test
% qsub submit_static.scr
```

***You just compiled and submitted a job to Hopper. Now let's take a closer look.***

# Most Basic Batch Script

A job script is a text file. Create and edit with a text editor, like vi or emacs.

Directives specify how to run your job

UNIX commands run on a service node (Full Linux)

**mpi_test** runs in parallel on compute nodes

```
#PBS -l walltime=00:10:00
#PBS -l mppwidth=24
#PBS -q debug
#PBS -N my_job

cd $PBS_O_WORKDIR

aprun -n 24 ./mpi_test
```

# Compilers on Hopper

- **Portland Group**
  - Default module PrgEnv-pgi
- **Cray**
  - PrgEnv-cray
  - module swap PrgEnv-pgi PrgEnv-cray
- **GNU**
  - PrgEnv-gnu
  - module swap PrgEnv-pgi PrgEnv-gnu
- **Pathscale**
  - PrgEnv-pathscale
  - module swap PrgEnv-pgi PrgEnv-pathscale

# Compiler Wrappers

- Use the Cray provided compiler wrappers which transparently link your application to MPI and other system libraries
- Fortran – use “ftn”
- C – use “cc”
- C++ -- use “CC”

```
% ftn parHelloWorld.F90
```

***This is one of the most common questions we answer at NERSC***


# Hopper Compute Nodes

- **6,384 nodes (153,216 cores)**
  - 6000 nodes have 32 GB; 384 have 64 GB
- **Small, fast Linux OS**
  - Limited number of system calls and Linux commands
  - No shared objects by default
    - Can support ".so" files with appropriate environment variable settings

# Batch Queues

| Submit Queue | Execution Queue[1] | Nodes | Processors | Max Wallclock |
|---|---|---|---|---|
| interactive | interactive | 1-256 | 1-6,144 | 30 mins |
| debug | debug | 1-512 | 1-12,288 | 30 mins |
| regular | reg_1hour | 1-256 | 1-6,144 | 1 hr |
| | reg_short | 1-683 | 1-16,392 | 6 hrs |
| | reg_small | 1-683 | 1-16,392 | 36 hrs |
| | reg_med | 684-2,048 | 16,393-49,152 | 36 hrs |
| | reg_big | 2,049-4,096 | 49,153-98,304 | 36 hrs |
| | reg_xbig[4] | 4,097-6,100 | 98,305-146,400 | 12 hrs |
| low | low | 1-683 | 1-16,392 | 12 hrs |
| premium | premium | 1-2,048 | 1-49,152 | 12 hrs |
| xfer | xfer | -- | -- | 12 hrs |

# Monitoring Batch Jobs

- **qstat –a [-u *username*]**
  - All jobs, in submit order
- **qstat –f *job_id***
  - Full report, many details
- **showq**
  - All jobs, in priority order
- **showstart, checkjob**

# Packed vs Unpacked

- **Packed**
  - User process on every core of each node
  - One node might have unused cores
  - Each process can safely access ~1.25 GB
- **Unpacked**
  - Increase per-process available memory
  - Allow multi-threaded processes

# Packed

```
#PBS -l mppwidth=1024
aprun -n 1024 ./a.out
```

- **Requires 43 nodes**
  - 42 nodes with 24 processes
  - 1 node with 16 processes
    - 8 cores unused
  - Could have specified mppwidth=1032

# Unpacked

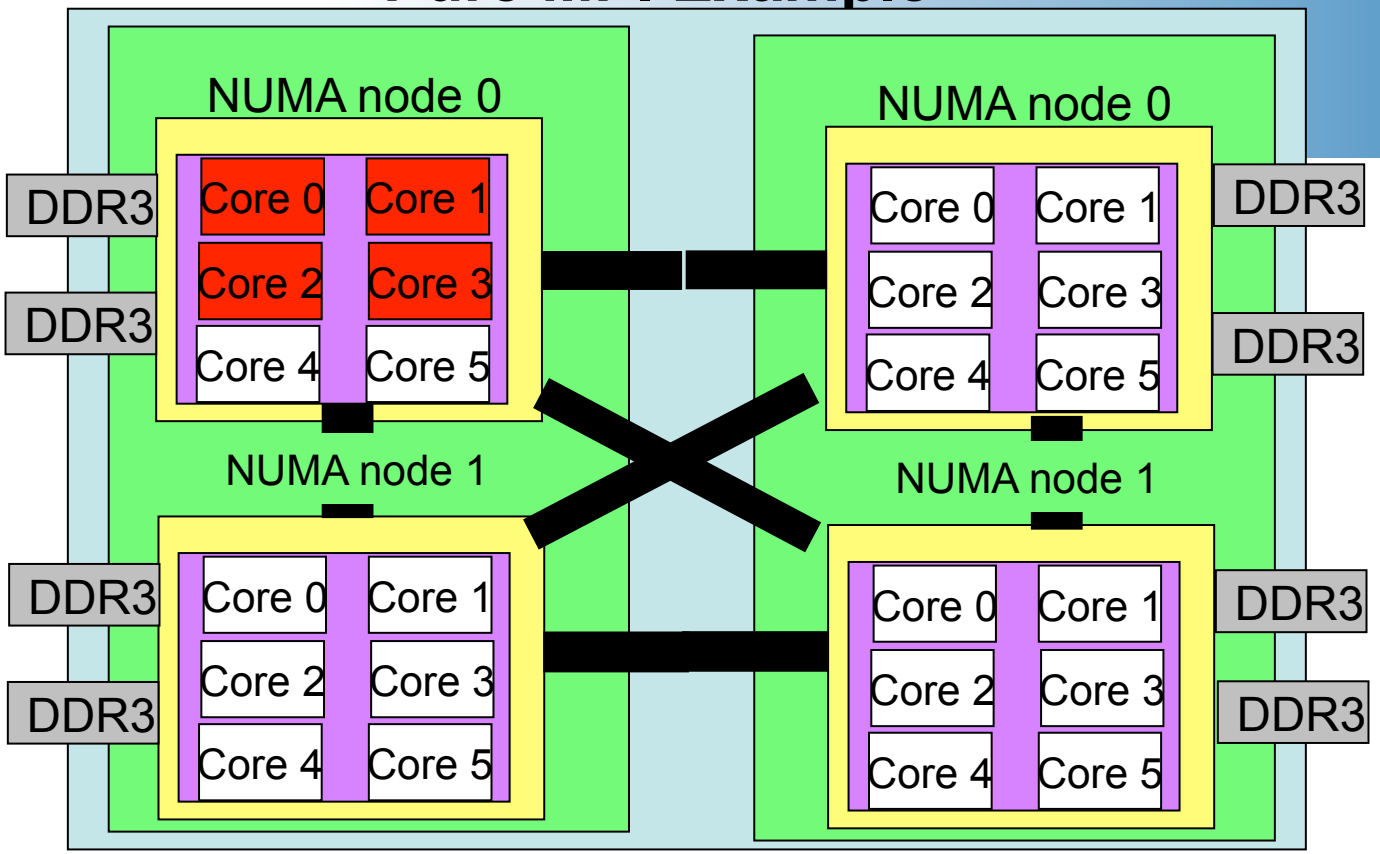
```
#PBS -l mppwidth=2048
aprun -n 1024 -N 12 ./a.out
```

- **Requires 86 nodes**
  - 85 nodes with 12 processes
  - 1 node with 4 processes
    - 20 cores unused
  - Could have specified mppwidth=2064
  - Each process can safely access ~2.5 GB

***But this isn't the most optimal way to run …***

# Pure MPI Example

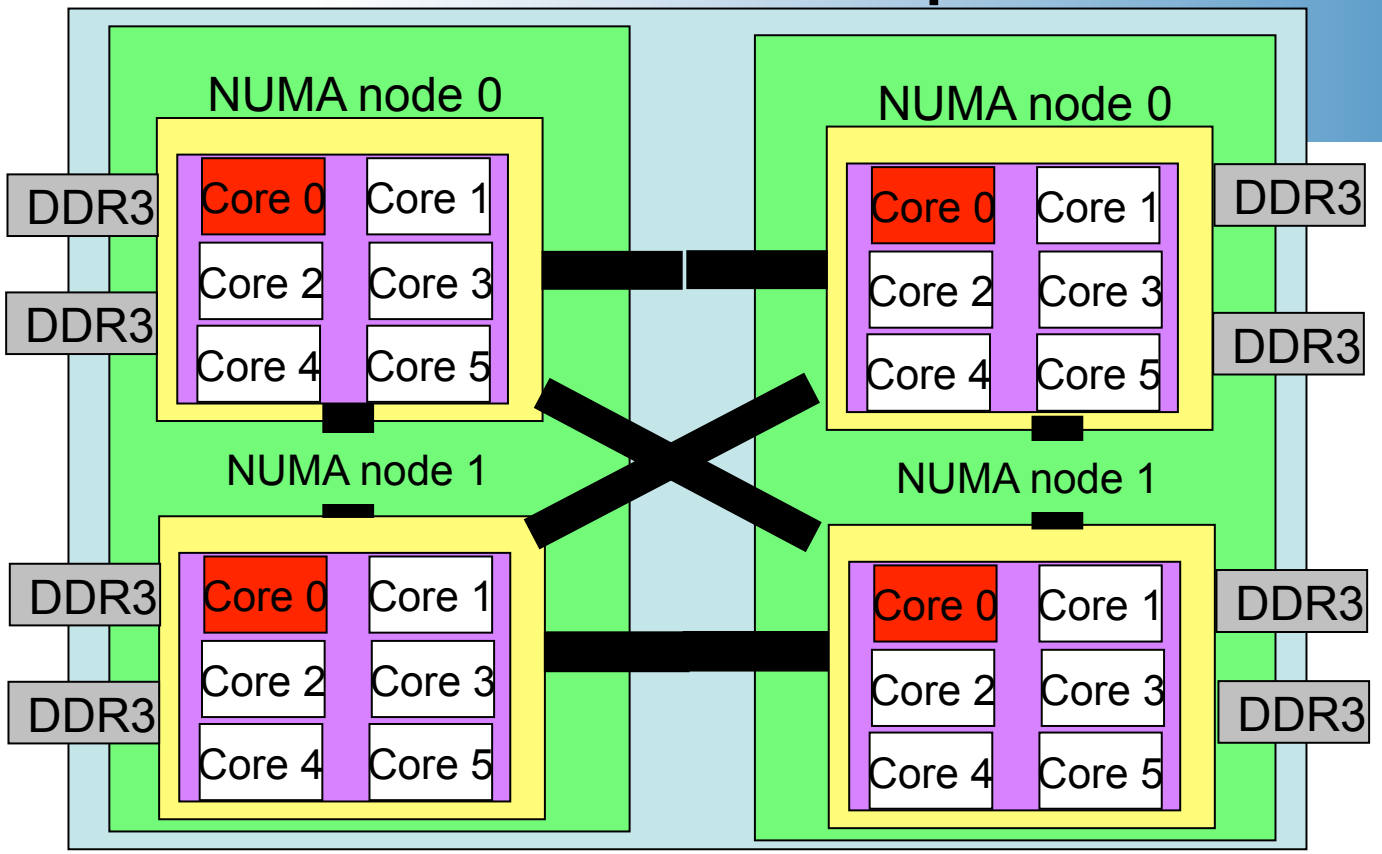
- *Example: 4 MPI tasks per node*
- *Default placement is not ideal when fewer than 24 cores per node are used.*




```
#PBS -l mppwidth=24
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS –q batch
#PBS -V

cd $PBS_O_WORKDIR
aprun -n 4 ./mpi_test
```

# Better Pure MPI Example




- *Example 4 MPI tasks per node*
- *- S 1 flag says put one core on each NUMA node*

```
#PBS -l mppwidth=24
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS –q batch
#PBS -V

cd $PBS_O_WORKDIR
aprun -n 4 –S 1 ./mpi_test
```

# XE6 Hybrid Usage Overview

- **Compile as if "pure" OpenMP**
  - -mp=nonuma for PGI
  - -mp for Pathscale
  - -fopenmp for GNU
  - no options for Cray
  - Cray wrappers add MPI environment

```
#PBS -l mppwidth=48
setenv OMP_NUM_THREADS 6
aprun –n 8 -N 4 -d 6 ./a.out
```

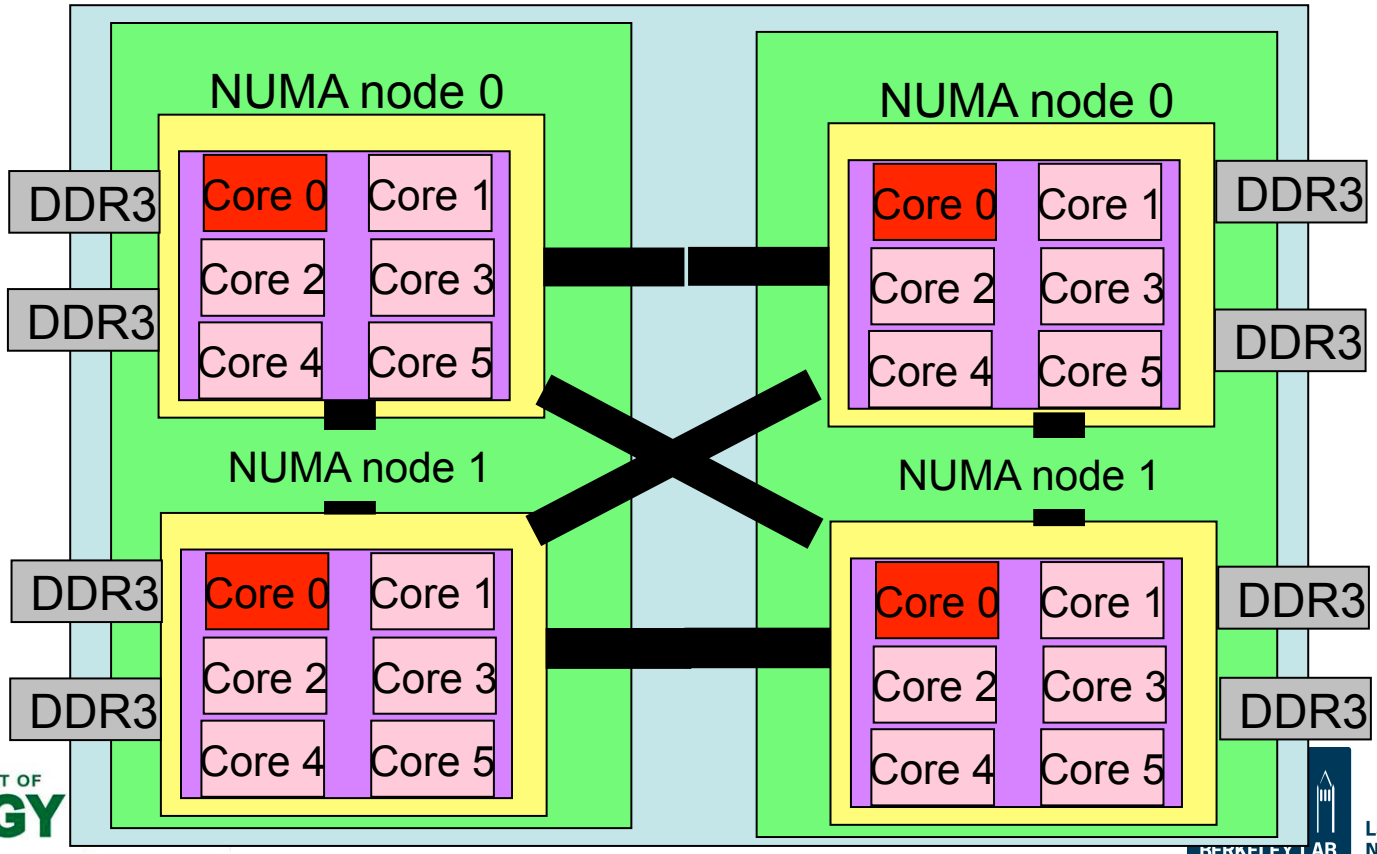
# Useful aprun Options

| Option | Description |
|---|---|
| -n | Number of MPI tasks. |
| -N | (Optional) Number of tasks per Hopper Node. Default is 24. |
| -d | (Optional) Depth, or number of threads, per MPI task. Use *in addition to* **OMP_NUM_THREADS**. Values can be 1-24; values of 2-6 are recommended. |
| -S | (Optional) Number of tasks per NUMA node. Values can be 1-6; default 6 |
| -sn | (Optional) Number of NUMA nodes to use per Hopper node. Values can be 1-4; default 4 |
| -ss | (Optional) Demands strict memory containment per NUMA node; default is to allow remote NUMA node memory access. |
| -cc | (Optional) Controls how tasks are bound to cores and NUMA nodes. Recommendation for most codes is -cc cpu which restricts each task to run on a specific core. |

# Hybrid MPI/OpenMP example on 6 nodes

- 24 MPI tasks with 6 OpenMP threads each

```
#PBS -l mppwidth=144
setenv OMP_NUM_THREADS 6
aprun -n 24 -N 4 -d 6 ./a.out
```



# Activity 3: Hybrid Hands-On

/project/projectdirs/training/jul-2012/mixed

```
jacobi_mpiomp.f90
jacobi_mpiomp.pbs
indata
```

# Find Out More

- **www.nersc.gov**