

# NERSC Systems and Services Available to CSGF Fellows

Jack Deslippe  
HPC Consultant, NERSC

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



## NERSC computing for science

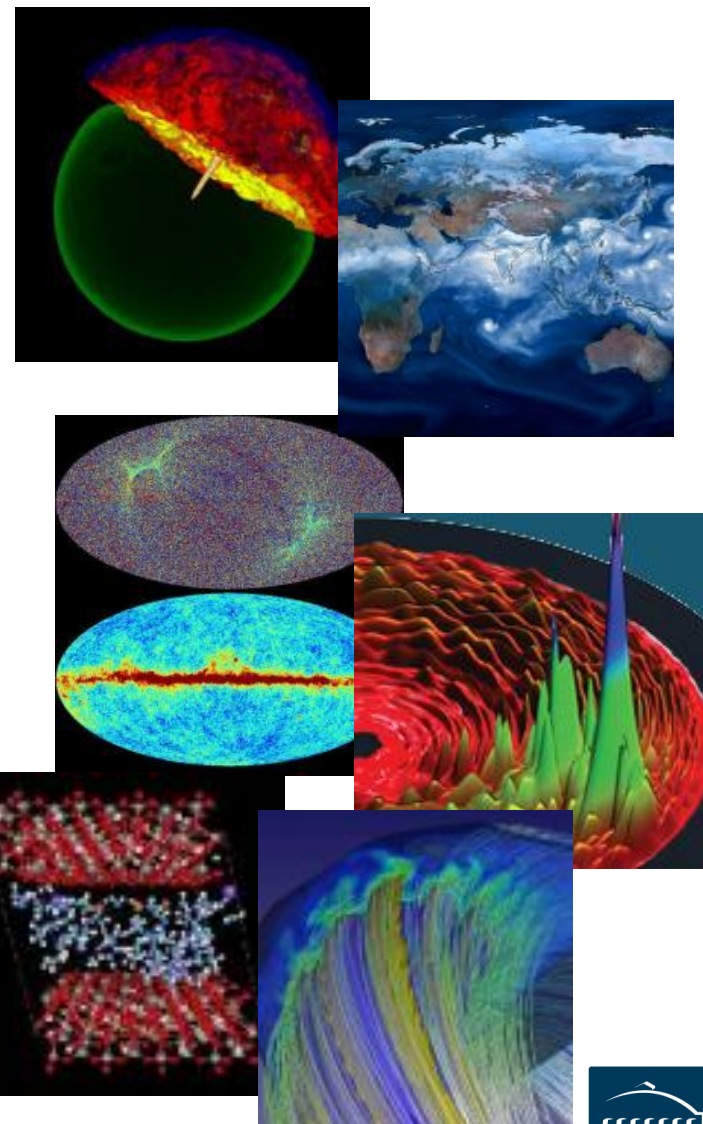
- 5000 users, 650 projects
- From 48 states; 65% from universities
- Hundreds of users each day
- **1500 publications per year**

## Systems designed for science

- 1.3PF Petaflop Cray system, Hopper
- N7 Coming in Next Year
  - Additional smaller clusters



- **Support computational science:**
  - Provide effective machines that **support fast algorithms**
  - Deploy with **flexible systems software** to run a **broad range** of applications
  - 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





## Large-Scale Computing Systems

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

- 6,384 compute nodes, 153,216 cores
- 120 Tflop/s on applications; 1.3 Pflop/s peak
- N7 Coming in 2013



### Clusters

140 Tflops total

#### Carver

- IBM iDataplex cluster

#### PDSF (HEP/NP)

- ~1K core cluster

#### GenePool (JGI)

- ~5K core cluster



### NERSC Global Filesystem (NGF)

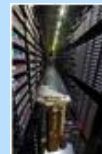
Uses IBM's GPFS

- 1.5 PB capacity
- 10 GB/s of bandwidth



### HPSS Archival Storage

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



### Analytics



#### Euclid

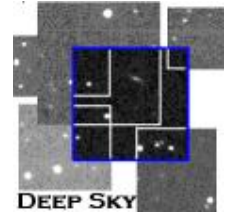
(512 GB shared memory)

#### Dirac GPU

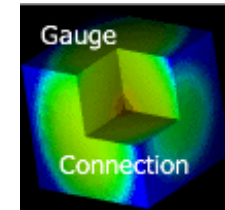
testbed (48 nodes)

# 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](http://newt.nersc.gov)
- **30+ projects use the NGF -> web**



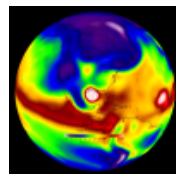
Deep Sky: 450+ Supernovae



Gauge Connection: QCD



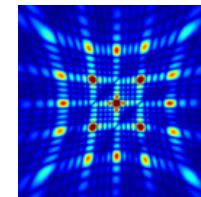
Daya Bay: Real-time processing and monitoring





20<sup>th</sup> Century Reanalysis



Earth Systems Grid



Coherent X-Ray Imaging Data Bank

[Job Control](#)
[About](#)
[VASP Manual](#)

Logged in as [jdeslip](#) | [Logout](#)

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:  Functional:

Click elements below to select available potentials.  
Remember to *select in the order they occur in your POSCAR file.*

Selected potentials:

1 H																	2 He
3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
55 Cs	56 Ba	71 Lu	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
87 Fr	88 Ra	103 Lr	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Nh	114 Fl	115 Mc	116 Lv	117 Ts	118 Og

57

58

59

60

61

62

63

64

65


66

67

68

69

70



[login](#)

## NERSC MOBILE beta

Please login.

### System Status:

Host	Status
hopper	up
carver	up
pdsf	up
genepool	up
euclid	up
archive	up

[NERSC MOTO](#)
[NOW COMPUTING](#)

### System Status



# NERSC Machines

# Why Do You Care About Architecture?

- **To use HPC systems well, you need to understand the basics and conceptual design**
  - Otherwise, too many things are mysterious
- **Programming for HPC systems is hard**
  - To get your code to work properly
  - To make it run efficiently (performance)
- **You want to efficiently configure the way your job runs**



# 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

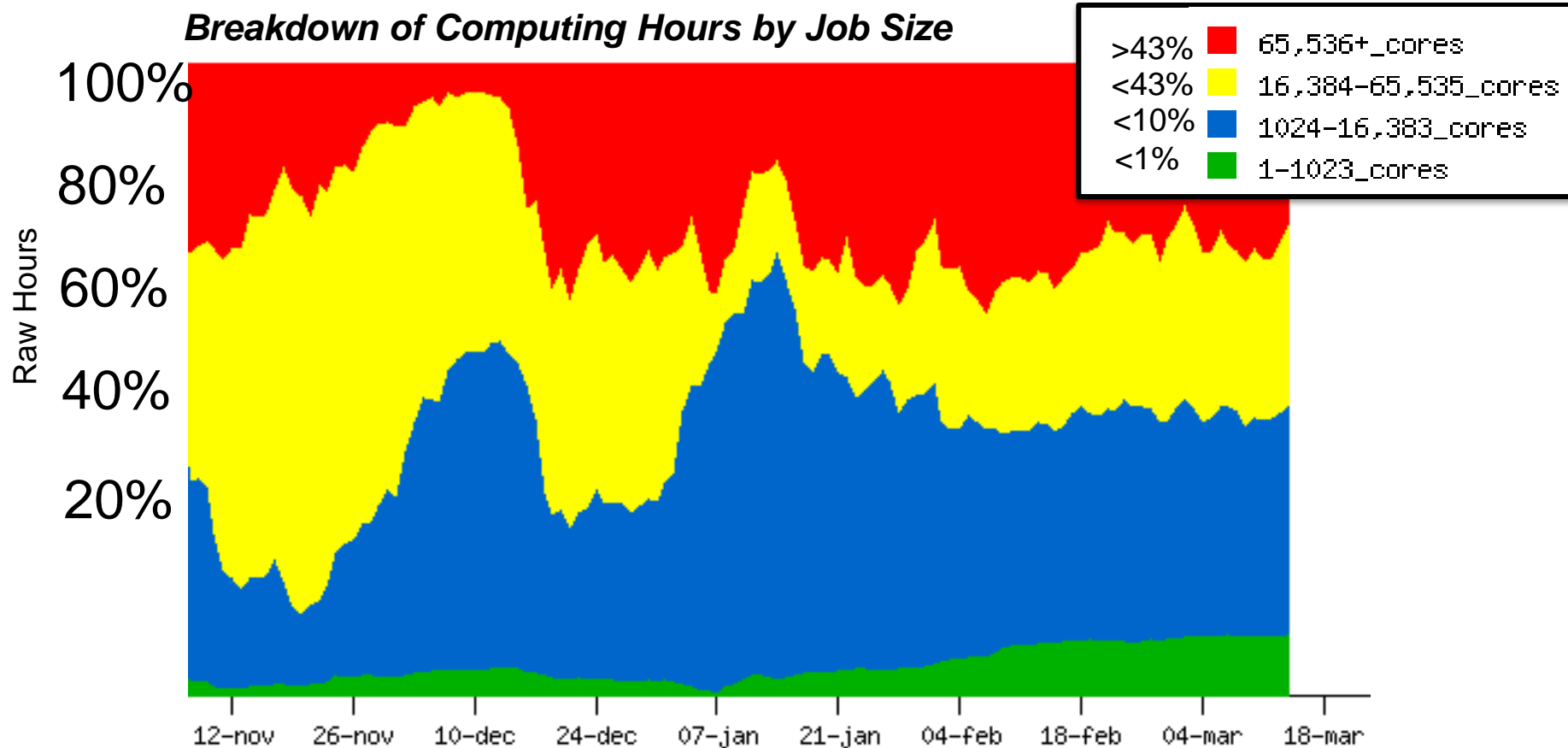
## I/O

2PB disk space

70GB/s peak I/O Bandwidth



# Hopper Job Size Mix

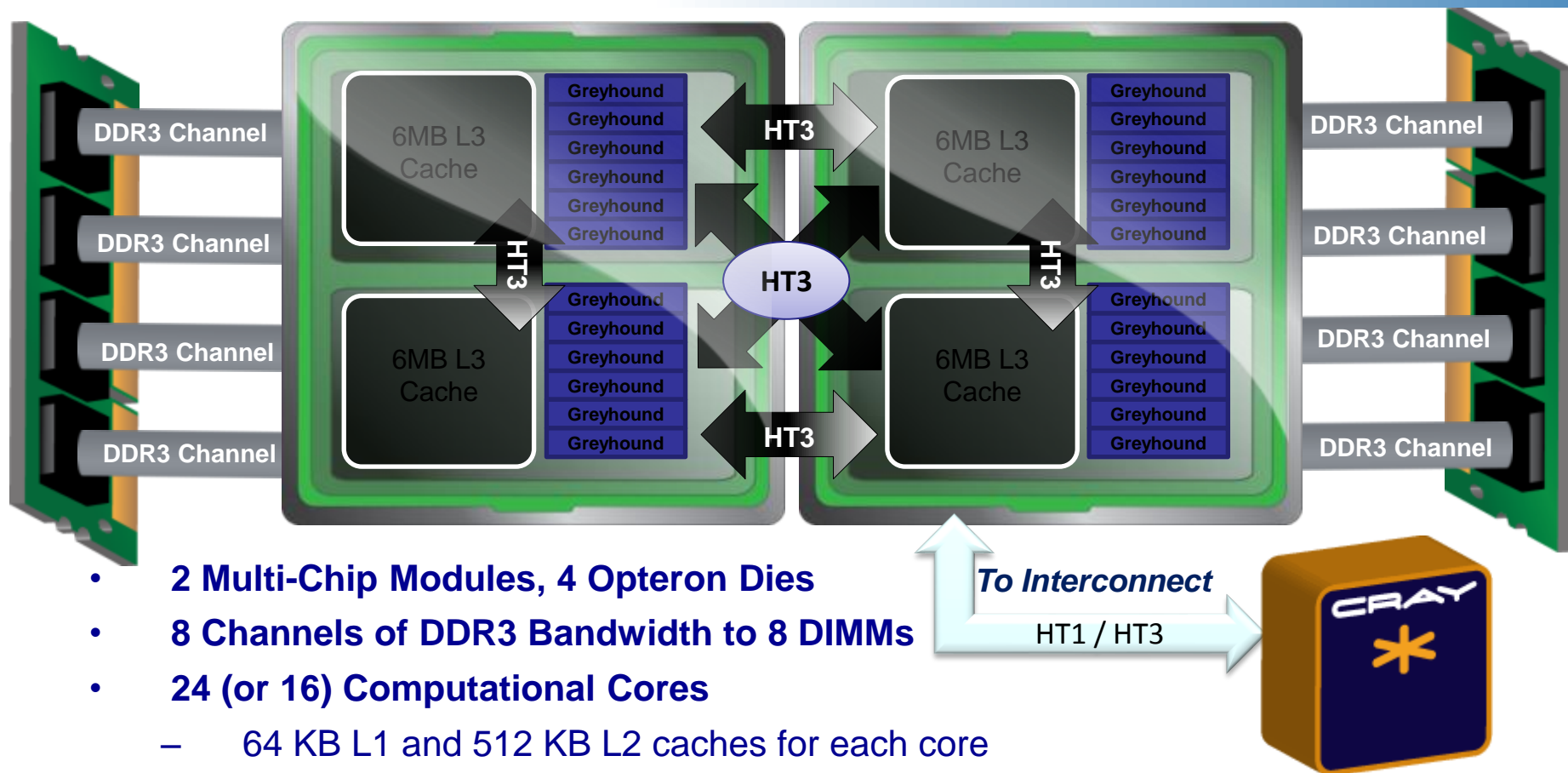


- Hopper is a 153,216 core system.*

# Preparing yourself for future hardware 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**



- **\$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



- 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

# 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.

- **NERSC will install a Cray “Cascade” system in 2013**
  - First all new Cray design since Red Storm; developed for the DARPA HPCS program
  - Intel Processors with combined > 2PF peak performance
  - New “Aries” interconnect using a “dragonfly” topology
  - 6.5PB storage using Cray Sonexion Lustre appliances
- **Good match for diverse NERSC user needs**
  - Both High-throughput and high-concurrency workloads.



# What services are available to CSGF Fellows?

# 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
  - Good news, you already have an allocation
  - All fellows have access to ~10k hours in m1266



- 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](mailto: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](#)



U.S. DEPARTMENT OF  
**ENERGY**

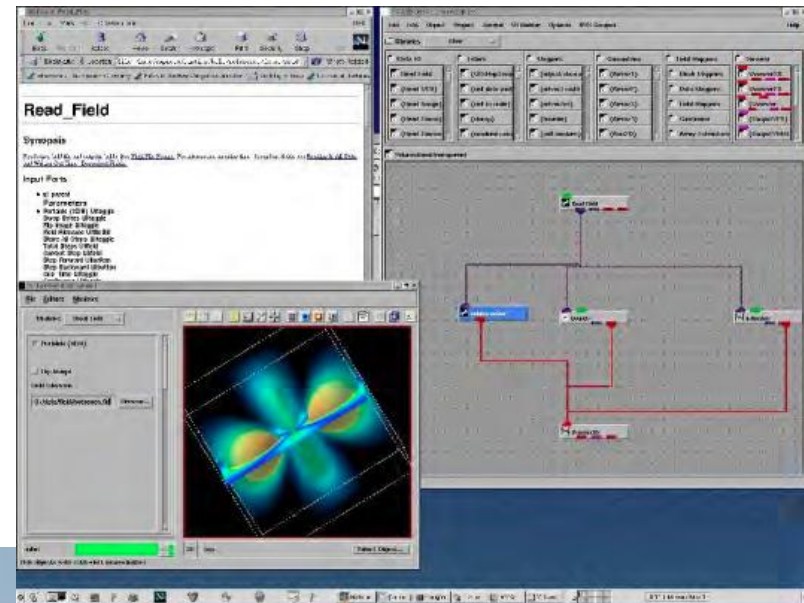
Office of  
Science



Lawrence Berkeley  
National Laboratory

# NX Provides Faster Remote Visualization

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



# Getting Your Own Production Allocation

- If you have exhausted your CSGF allocation, apply for your own allocation with DOE
- Research must be relevant to the mission of the DOE
- <https://www.nersc.gov/users/accounts/>
- ASCR Program managers are very supportive of CSGF program

# 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: 1/2 of consultants have PhDs in science domain, other 1/2 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

1,313  
tickets

## Account Support

- I forgot my password
- I'm a new user
- I'm out of time, can I have more?
- I want to add a new user to project
- How do I log in?

Network  
and  
Security

87 tickets

785 tickets

## Software

- How do I use this package?
- My job is failing with this software
- This software has a bug
- I'd like to request new software

## Running Jobs

2,019  
tickets

- My job failed
  - User failures
  - System Failures
- This worked on my local cluster, how can I run it on at NERSC?
- How do I submit my job?
- My application is running slowly.
- I'm new, help!

## Programming

- Need help porting code to new machine
- My compilation is failing
- I found a compiler bug

430  
tickets

## Data and Storage

642  
tickets

- I need help backing up data
- I need more disk space
- How can I transfer files to local system or another facility



# Software Support: Chemistry & Materials Applications

QUANTUM ESPRESSO

CPMD consortium page

CPMD

b-initio

abinit.org

- 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

www.gaussian.com  
THE OFFICIAL GAUSSIAN WEBSITE

NWCHEM

# 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”

# 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**
- **Ask consultants questions – we are here to help.**
- **Profile your code with CrayPat, IPM, HPCToolkit**
- **Use parallel debuggers like DDT.**

# Hands On Activities!

1. **Logging In**
2. **Compiling + Submitting a Parallel 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***



**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 supercomputer 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

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

UNIX commands run on a  
service node (Full Linux)

```
cd $PBS_O_WORKDIR
```

**mpi\_test** runs in  
parallel on compute nodes

```
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-pathscales
  - module swap PrgEnv-pgi PrgEnv-pathscales

# 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

- Launch and manage parallel applications on compute nodes
- Commands in batch script are executed on MOM nodes
- No user (ssh) logins

*This is a key difference between a vanilla cluster and a Cray system*

# Batch Queues

Submit Queue	Execution Queue <sup>1</sup>	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 <sup>4</sup>	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

# Batch Options

Specify the max wall clock time

**#PBS -l walltime=*hh:mm:ss***

Specify the number of cores

**#PBS -l mppwidth=*num\_cores***

Specify the queue name

**#PBS -q *queue\_name***

Import environment

**#PBS -V**

Charge job to account

**#PBS -A *account***

# More Batch Script Options

Name of job

**#PBS -N *job\_name***

Name output and error files

**#PBS -o *output\_file***

**#PBS -e *error\_file***

Join output and error files

**#PBS -j oe**

Specifies email address for notifications

**#PBS -M email address**

Email notification (abort/begin/end/never)

**#PBS -m [*a/b/e/n*]**



```
% qsub submit_static.scr  
140979.sdb
```

*Keep this jobid. It is often useful for debugging*

**Examine job output:**

```
% cat my_job.o63731
```

# 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
- **apstat, showstart, checkjob, xtnodestat**

# Manipulating Batch Jobs

- **qsub *job\_script***
- **qdel *job\_id***
- **qhold *job\_id***
- **qrls *job\_id***
- **qalter *new\_options job\_id***
- **qmove *new\_queue job\_id***

# 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

```
#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



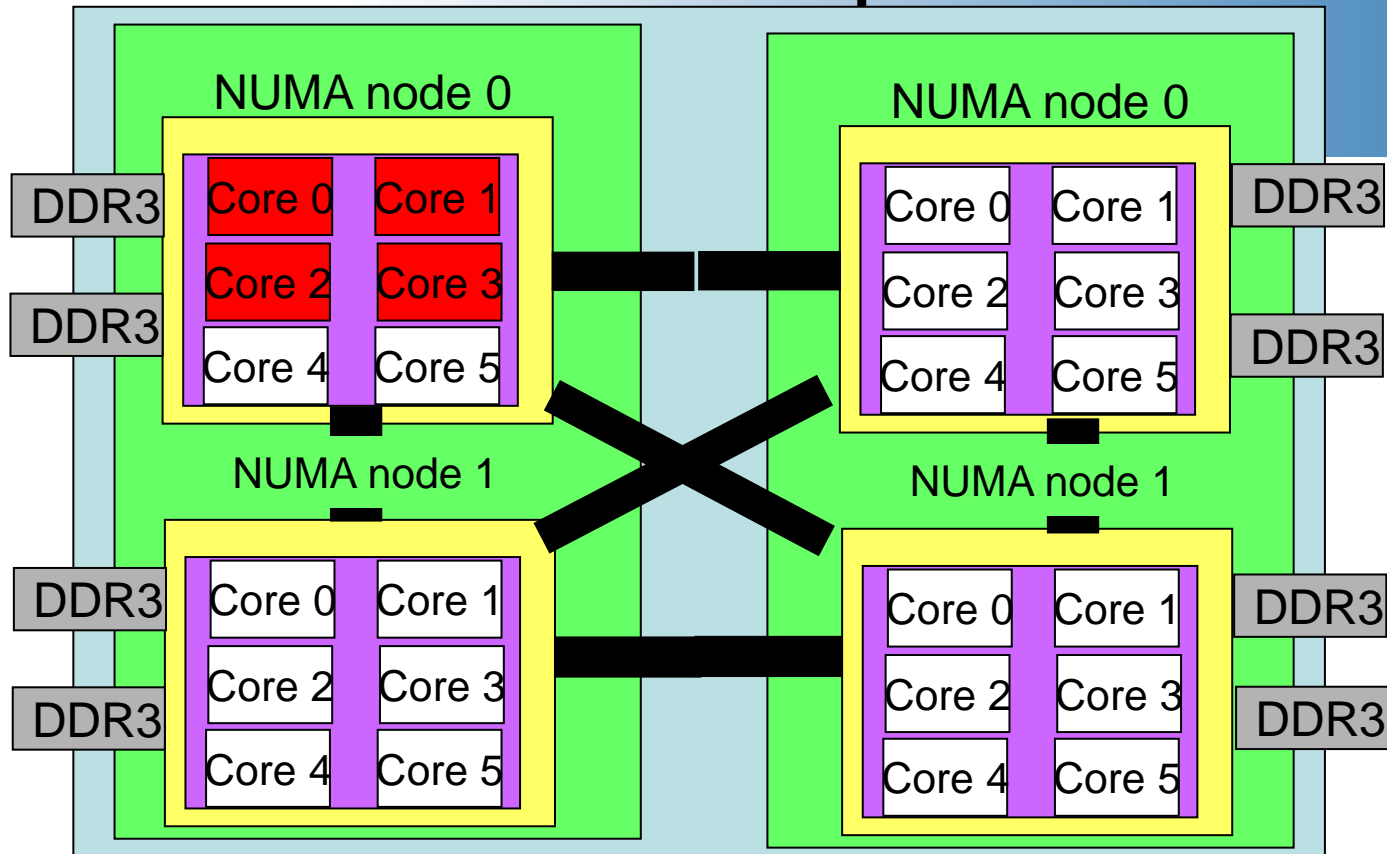
```
#PBS -l mppwidth=2048  
aprun -n 1024 -N 12 ./a.out
```

- **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.*

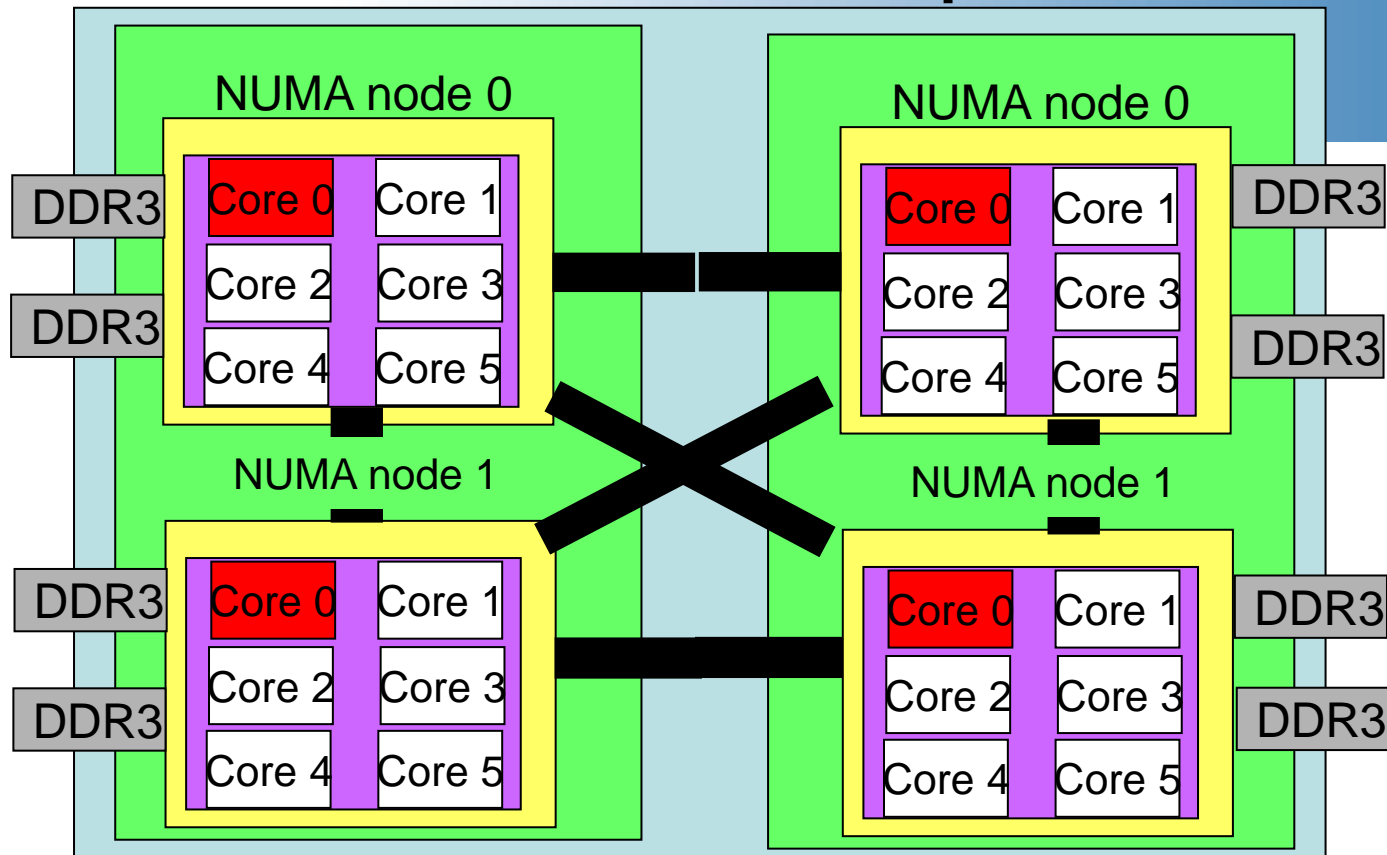


```
#PBS -l mppwidth=24
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS -q batch
#PBS -V
```

```
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
```

# Activity 3: Hands-On

`/project/projectdirs/training/jul-2012/mpi`

`jacobi_mpi.f90`

`jacobi.pbs`

`indata`

# A Hybrid Pseudo Code

```
program hybrid
call MPI_INIT (ierr)
call MPI_COMM_RANK (...)
call MPI_COMM_SIZE (...)
... some computation and MPI communication
call OMP_SET_NUM_THREADS(4)
!$OMP PARALLEL DO PRIVATE(i) SHARED(n)
do i=1,n
... computation
enddo
!$OMP END PARALLEL DO
... some computation and MPI communication
call MPI_FINALIZE (ierr)
end
```



- **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 <i>in addition to</i> <b>OMP_NUM_THREADS</b> . 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.

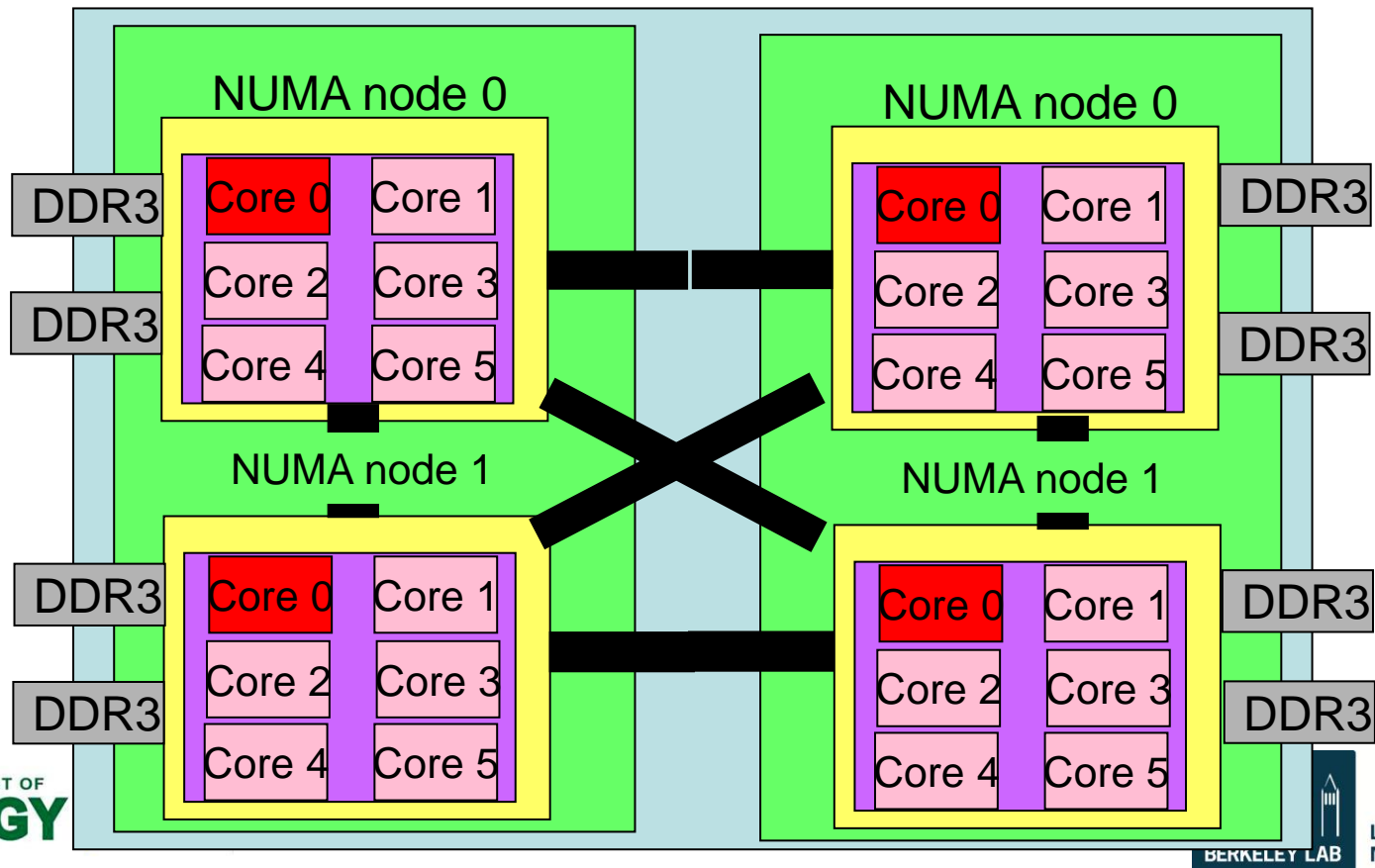
# Hybrid MPI/OpenMP example on 6 nodes

- 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
```



# Controlling NUMA Placement

```
#PBS -l mppwidth=144    (so 6 nodes!)
```

- 1 MPI task per NUMA node with 6 threads each

```
setenv OMP_NUM_THREADS 6
```

```
aprun -n 24 -N 4 -d 6 ./a.out
```

- 2 MPI tasks per NUMA node with 3 threads each

```
setenv OMP_NUM_THREADS 3
```

```
aprun -n 48 -N 8 -d 3 ./a.out
```

- 3 MPI tasks per NUMA node with 2 threads each

```
setenv OMP_NUM_THREADS 2
```

```
aprun -n 72 -N 12 -d 2 ./a.out
```

## Activity 3: Hybrid Jobs

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

`jacobi_mpiomp.f90`

`jacobi_mpiomp.pbs`

`indata`

- **[www.nersc.gov](http://www.nersc.gov)**