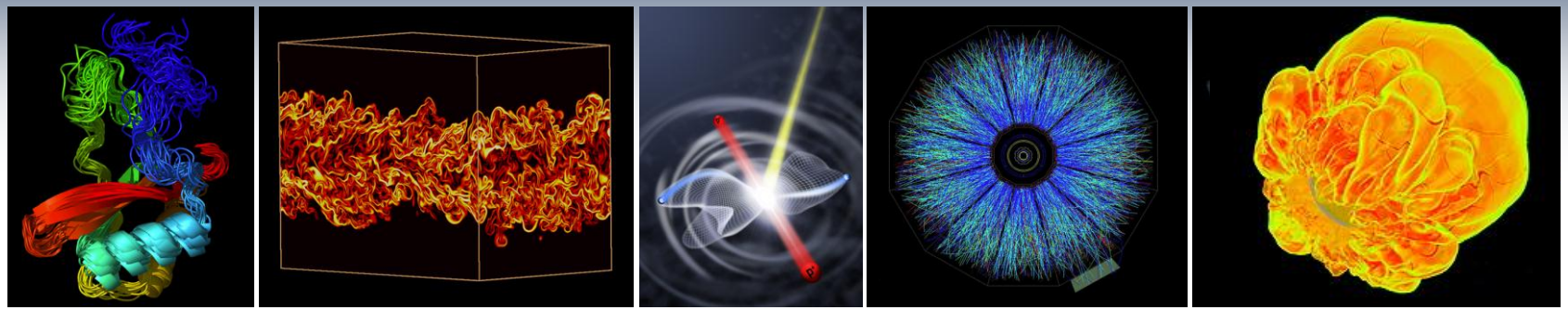


NERSC Systems and Services Available to CSGF

CSGF HPC Workshop, Arlington, VA

Katie Antypas
Group Leader, NERSC User Services

July 19, 2011

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

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

Systems designed for science

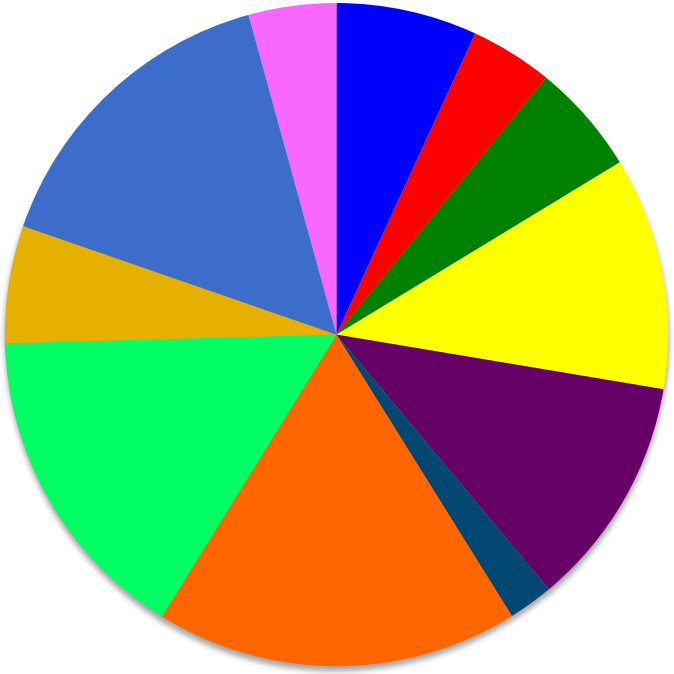
- 1.3PF Petaflop Cray system, Hopper
  - 3rd Fastest computer in US
  - Additional .5 PF in Franklin system and smaller clusters

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC is the Primary Computing Center for DOE Office of Science

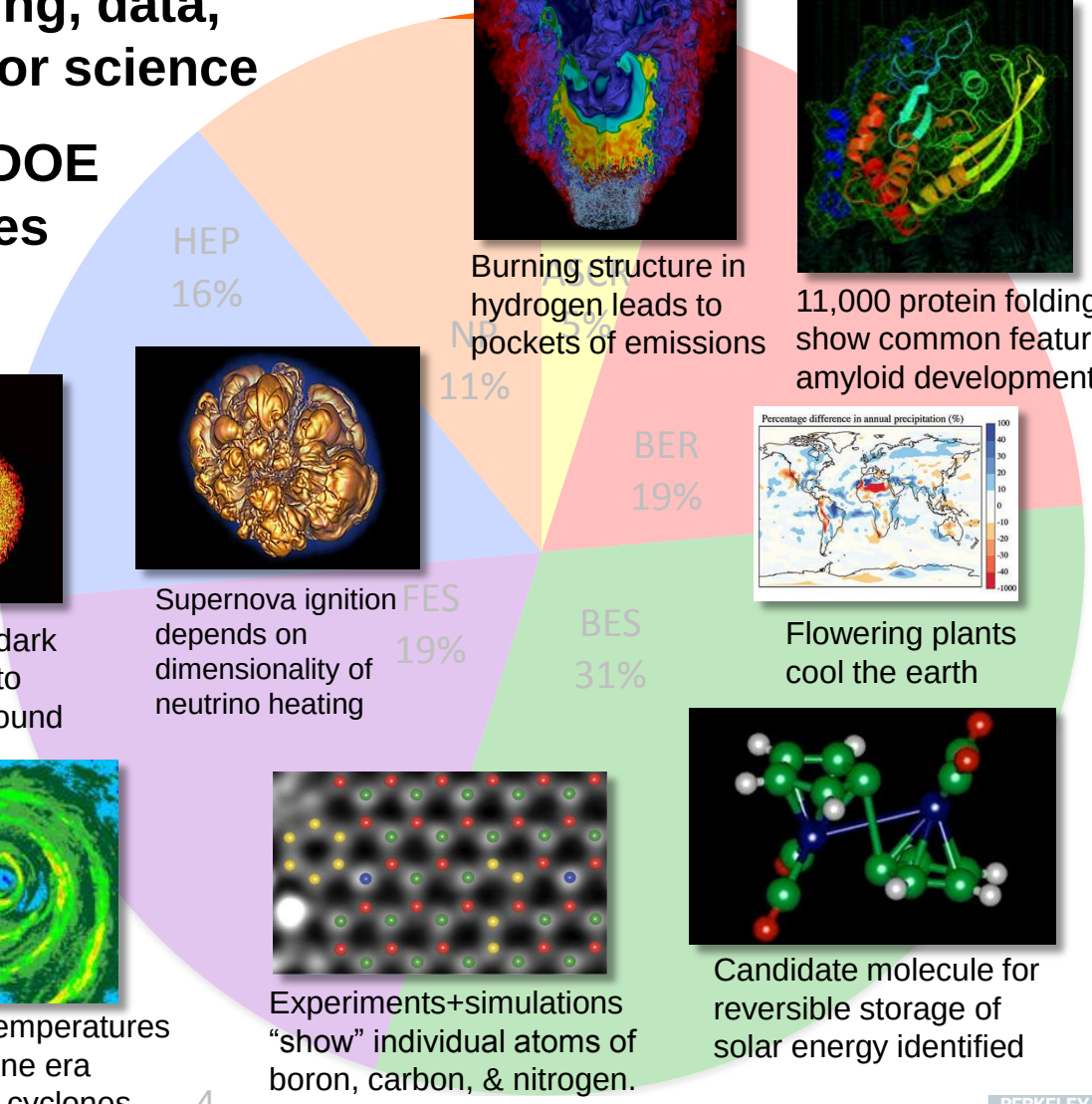
- **NERSC serves a large population**
- **Focus on "unique" resources**
  - **Expert consulting and other services**
  - **High end computing systems**
  - **High end storage systems**
- **NERSC is known for:**
  - **Outstanding services**
  - **Large and diverse user workload**

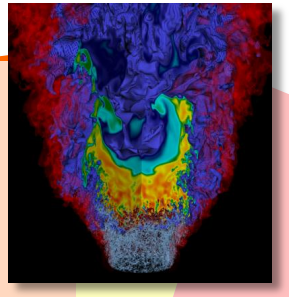


"NERSC continues to be a gold standard of a scientific High Performance Computational Facility."
– HPCOA,Review August 2008

NERSC Serves the Computing and Data Needs of Science

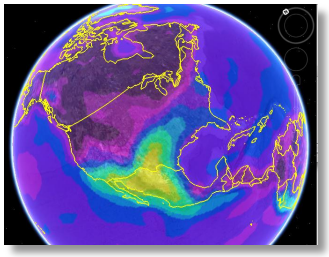
- **NERSC provides computing, data, and consulting services for science**
- **Allocations managed by DOE based on mission priorities**



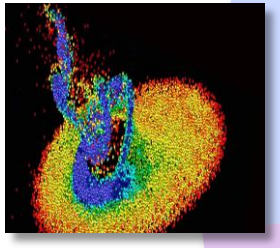


Burning structure in hydrogen leads to pockets of emissions

11,000 protein foldings, show common feature in amyloid development,



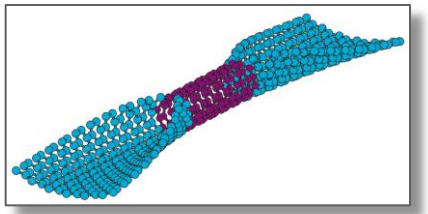
20th Century 3D climate maps reconstructed and in public database



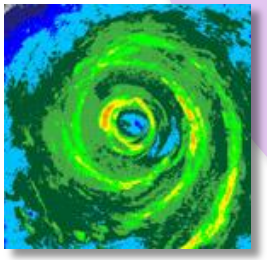
Location of dark companion to Milky Way found

Supernova ignition depends on dimensionality of neutrino heating

Flowering plants cool the earth



Carbon-based transistor junction created



Higher temperatures in Pliocene era linked to cyclones

Experiments+simulations "show" individual atoms of boron, carbon, & nitrogen.

Candidate molecule for reversible storage of solar energy identified

NERSC Systems

Large-Scale Computing Systems

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 XE6

- 6,384 compute nodes, 153,216 cores
- 120 Tflop/s on applications; 1.3 Pflop/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 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

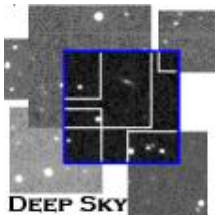
Euclid

(512 GB shared memory)

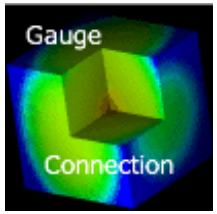
Dirac GPU testbed (48 nodes)

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
  - Write a Gateway: HTML + Javascript
- **30+ projects use the NGF -> web**

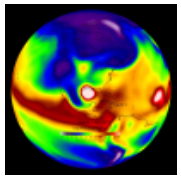


Deep Sky: 450+ Supernovae



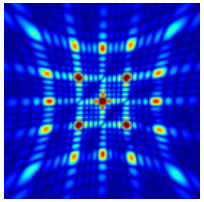
Gauge Connection: QCD

Daya Bay: Real-time processing and monitoring



20th Century Reanalysis

Earth Systems Grid



Coherent X-Ray Imaging Data Bank

HPC Architecture

7

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**
- **The technology is cutting edge**

Definitions & Terminology

- **HPC**
  - High Performance Computing
  - Scientific computing at scale
- **CPU**
  - Central Processing Unit
  - Now ambiguous terminology
  - Generic for “some unit that computes”
  - Context-sensitive meaning

- **Memory**
  - Volatile storage of data or computer instructions
- **Bandwidth**
  - The rate at which data is transferred between destinations (typically GB/s)
- **Latency**
  - The time needed to initialize a data transfer (ranges from 10^{-9} to 10^{-6} secs or more)
- **FLOP: Floating Point Operation**
  - e.g., a+b, a*b+c
  - FLOPs/sec is a common performance metric
- **Interconnect**
  - A high-performance data network that connects nodes to each other and possibly other

What are the “5 major parts”?

Google

five major parts of a computer | Search

About 302,000,000 results (0.21 seconds) Advanced search

Everything
Images
Videos
News
Shopping
More

Oakland, CA
Change location

Show search tools

The **Five** Main **Parts of a Computer** | eHow.com
May 5, 2010 ... The **Five** Main **Parts of a Computer**. Computers may look very different, but the components installed are standard. The **major** difference among ...
www.ehow.com › ... › Install a Hard Drive - Cached

Answers.com - What are **five parts** of the **computer** system
Computers question: What are **five parts** of the **computer** system? The **five parts** of the **computer** are CPU, Monitor, Printer, Mouse and Keyboard.
wiki.answers.com › ... › Categories › Technology › Computers - Cached - Similar

Answers.com - What are the main **parts** of computers
What are **five** main **parts of a computer**? ram cpu hard disk drive optical ...
wiki.answers.com › ... › Technology › Computers › Computer Hardware - Cached

Show more results from answers.com

What are the main **parts of a computer**?
What are the main component **parts of a computer**? ... a processor, and inputs and outputs.

Five Major Parts

eHow.com	Answers.com	Fluther.com	Yahoo!	Wikipedia
CPU	CPU	CPU	CPU	Motherboard
RAM	Monitor	RAM	RAM	Power Supply
Hard Drive	Printer	Storage	Power Supply	Removable Media
Video Card	Mouse	Keyboard/Mouse	Video Card	Secondary Storage
		Monitor		
Motherboard	Keyboard	Motherboard	Motherboard	Sound Card
		Case / Power Supply		IO Peripherals

It Depends on Your Perspective

- What is a computer?
  - It depends what you are interested in.
    - CPU, memory, video card, motherboard, ...
    - Monitor, mouse, keyboard, speakers, camera, …
- We'll take the perspective of an application programmer or a scientist running a code on an HPC system
- What features of an HPC system are important for you to know about?

5 Major Parts of an HPC System

1. CPUs
2. Memory (volatile)
3. Nodes
4. Inter-node network
5. Non-volatile storage (disks, tape)

Hopper

NERSC-6 Grace "Hopper"

Cray XE6

Performance

- 1.3 PF Peak
- 1.05 PF HPL (#8)

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

Evolution from Franklin (XT4) to Hopper (XE6)

Cray XT4: Franklin

Performance: 0.352 PF Peak
0.266 TF HPL (#27, debut@ #8)
Processor: AMD Budapest
4-core 2.3 GHz (9.2 GF/core)
4 cores/node
Memory: DDR2 667MHz
8 GB/node @ 21GB/s
2 GB/core
System
9,572 nodes (38,288 total cores)
Interconnect: SeaStar2 3D torus,
1.6GB/s measured @ 6-8usec
I/O
12GB/s peak I/O Bandwidth
0.436 PB disk space

Cray XE6: Hopper

Performance: 1.288 PF Peak
1.05 PF HPL (#8, debut@ #5)
Processor: AMD MagnyCours
12-core 2.1 GHz (8.4 GF/core)
24 cores/node
Memory: DDR3 1333MHz
32-64 GB/node @ 84GB/s
1.3 - 2.6 GB/core
System
6,384 nodes (153,216 total cores)
Interconnect: Gemini 3D torus,
8.3GB/s measured @ 2usec
I/O
70GB/s peak I/O Bandwidth
2PB disk space

Evolution from Franklin (XT4) to Hopper (XE6)

Cray XT4: Franklin

Performance: 0.352 PF Peak
0.266 TF HPL (#27, debut@ #8)

Processor: AMD Budapest
4-core 2.3 GHz (9.2 GF/core)
4 cores/node

Memory: DDR2 667MHz
8 GB/node @ 21GB/s
2 GB/core

System
9,572 nodes (38,288 total cores)

Interconnect: SeaStar2 3D torus,
1.6GB/s measured @ 6-8usec

I/O
12GB/s peak I/O Bandwidth
0.436 PB disk space

Cray XE6: Hopper

Performance: 1.288 PF Peak
1.05 PF HPL (#8, debut@ #5)

Processor: AMD MagnyCours
12-core 2.1 GHz (8.4 GF/core)
24 cores/node

Memory: DDR3 1333MHz
32-64 GB/node @ 84GB/s
1.3 - 2.6 GB/core

System
6,384 nodes (153,216 total cores)

Interconnect: Gemini 3D torus,
8.3GB/s measured @ 2usec

I/O
70GB/s peak I/O Bandwidth
2PB disk space

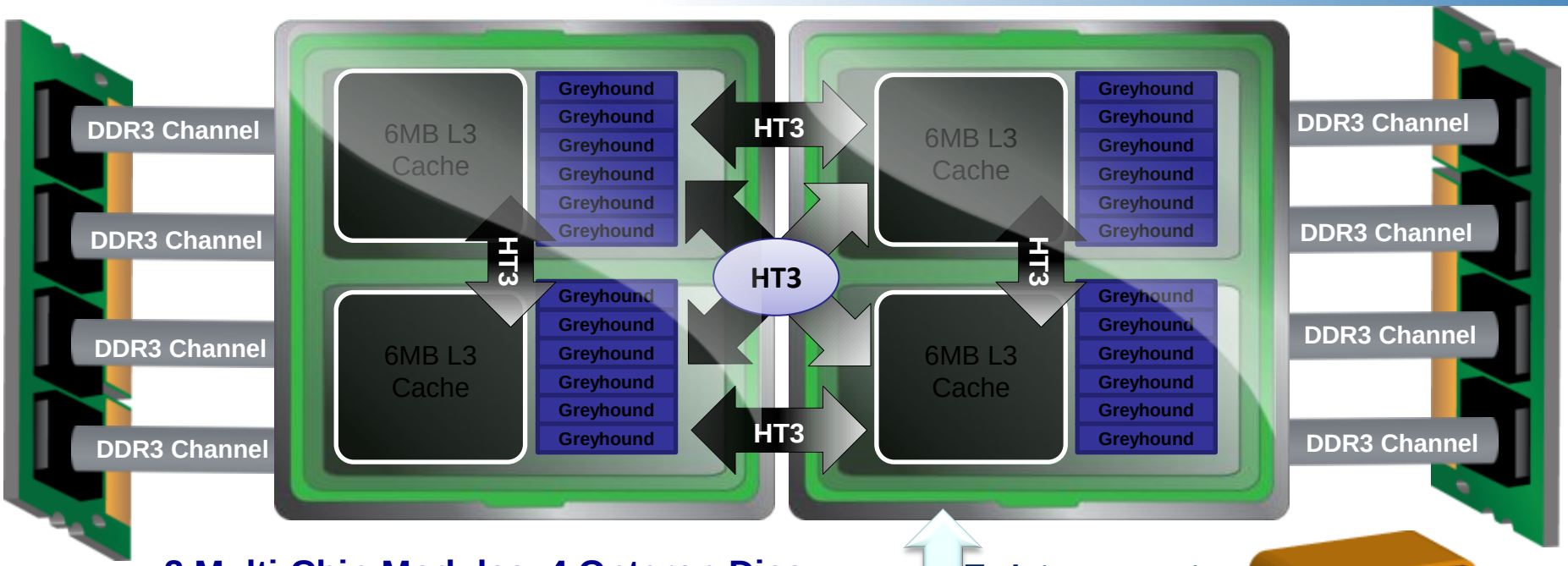
Preparing yourself for future hardware trends

- **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



To Interconnect

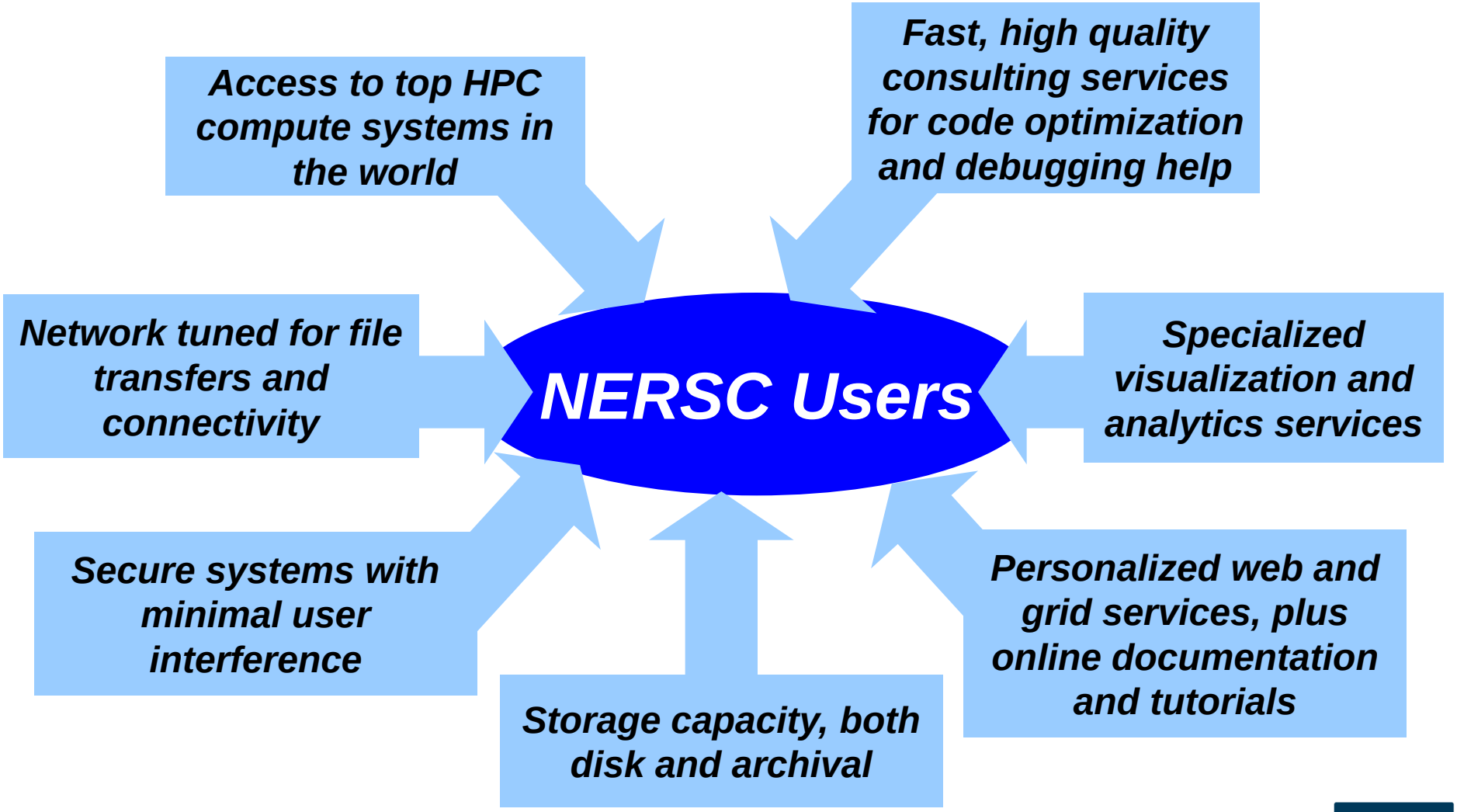
HT1 / HT3

- **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.

What services are available to CSGF fellows?

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

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
- Builds relationship with DOE program managers

ASCR's Computing Facilities

NERSC at LBNL

- Thousands of users, hundreds 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 and ANL

- Hundreds of users, tens of projects
- Allocations:
  - 60% ANL/ORNL managed INCITE process
  - 30% ACSR Leadership Computing Challenge*
  - 10% LCF reserve
- Science limited to largest scale; not just DOE/SC
- Machines procured through partnerships

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

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?

2,019 tickets

Running Jobs

- 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!

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

Programming

430 tickets

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

642 tickets

Data and Storage

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

Software on NERSC systems

- NERSC supports and maintains a large array of software
  - Chemistry/Material Science
  - Math libraries
  - I/O libraries
  - Visualization
  - Performance and Debugging tools

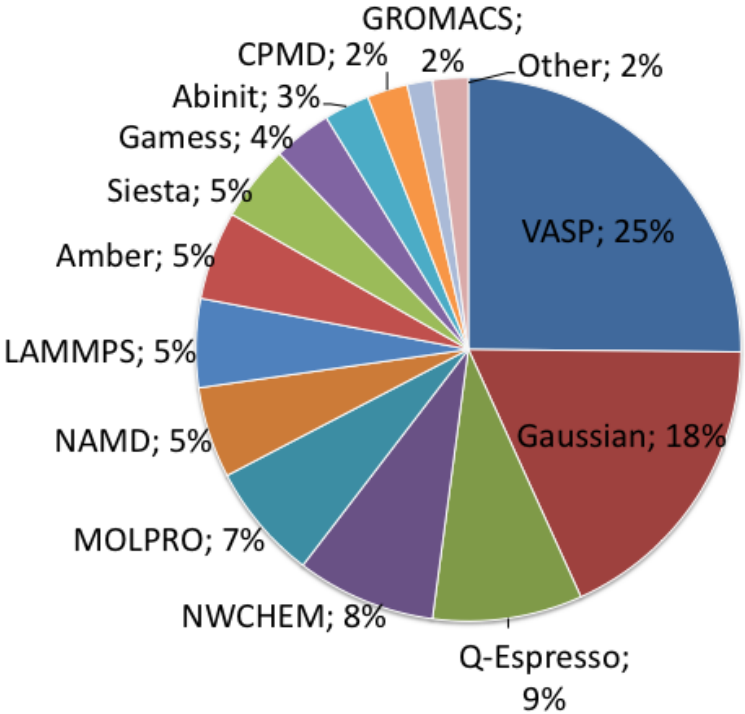
Software Support: Chemistry & Materials Applications

- **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**

Third-Party Application Usage Growing Rapidly

Over 400 researchers at NERSC use 3rd party applications

2009 Third-Party Application Breakdown by Number of Users



"Precompiled codes are a lifesaver. One machine in the system always has what you need. I get very good performance from everything I use and it all scales wonderfully over the number of processors I use..."
– NERSC User, 2010 Survey

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

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.

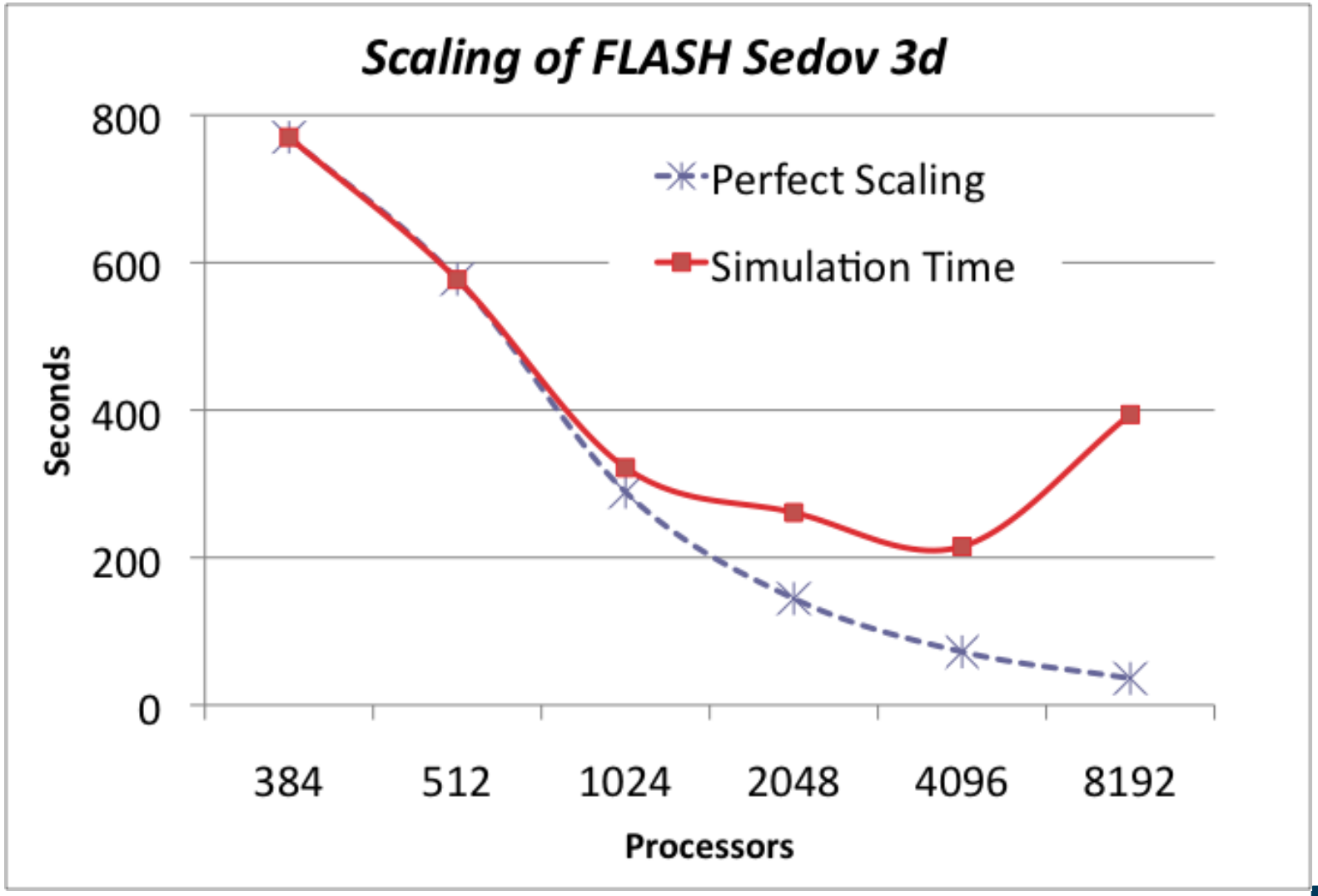
```
kantypas@login2:~> module show python
-------------------------------------------------------------------
/soft/modulefiles/compilers/python/2.7.1:

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

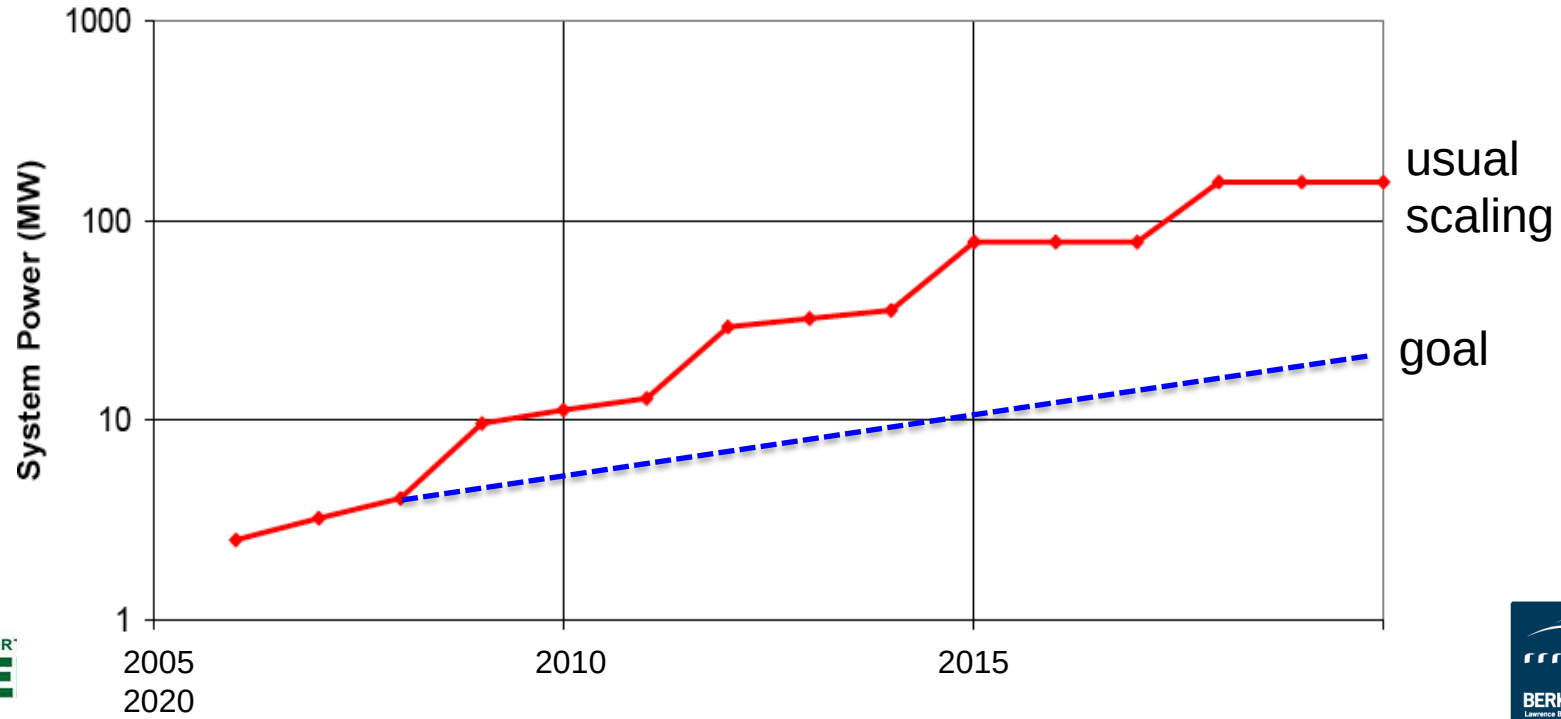
- Challenge yourself to learn a little bit about HPC architecture
  - To use systems well you need to understand conceptual design, otherwise too many things are mysterious
  - It is hard to program for HPC systems, to get your code working properly AND running efficiently
- Attend workshops and online tutorials
- Ask consultants questions – many of us have worked with these systems for a long time and we are here to help.

FLASH Sedov 3d problem with Particles

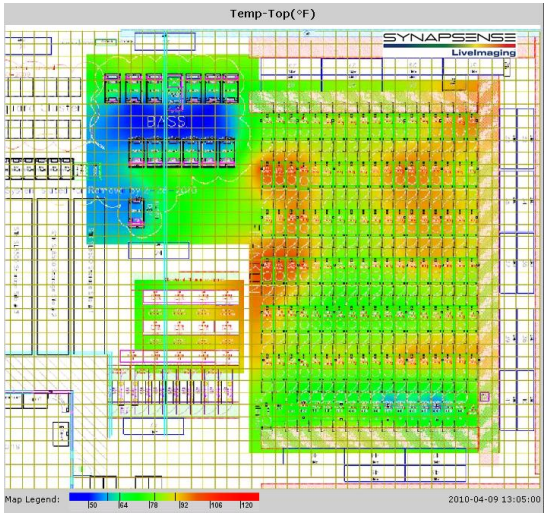


Energy Efficiency is Necessary for Computing

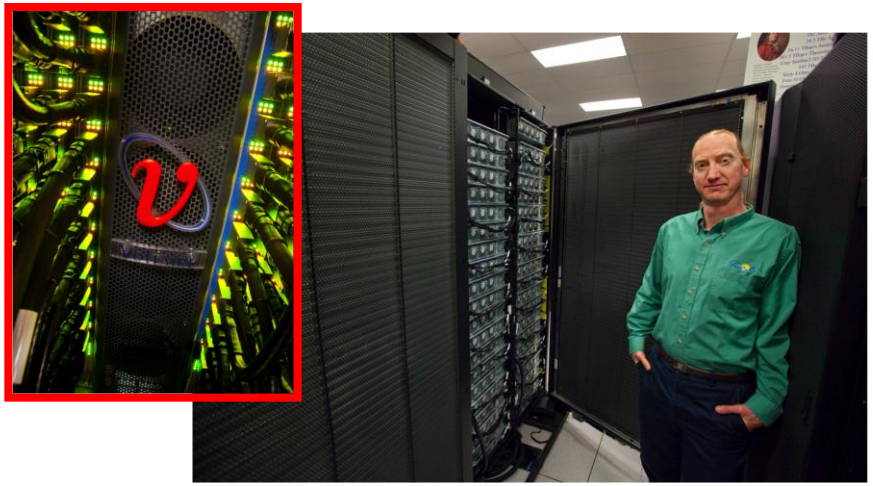
- Systems have gotten about 1000x faster over each 10 year period
- 1 petaflop (10^{15} ops) in 2010 will require 3MW
  → 3 GW for 1 Exaflop (10^{18} ops/sec)
- DARPA committee suggested 200 MW with "usual" scaling
- Target for DOE is 20 MW in 2018



Energy Efficiency Partnerships with Synapsense and IBM



600 Sensors for temperature, etc.

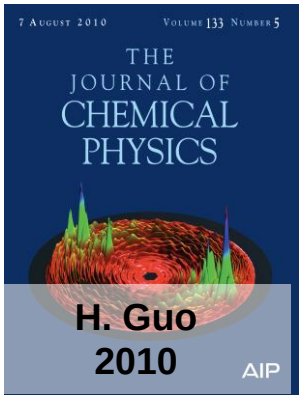


Rear door heat exchangers

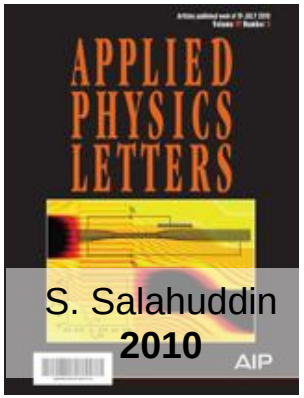
- **Monitoring for energy efficiency (and reliability!)**
- **Liquid cooling on IBM system uses return water from another system, with modified CDU design**
  - **Reduces cooling costs to as much as ½**
  - **Reduces floor space requirements by 30%**

Air is colder coming out than going in!

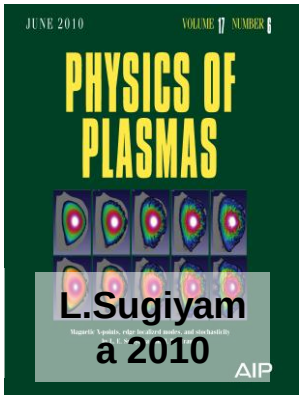
Recent Cover Stories from NERSC Research



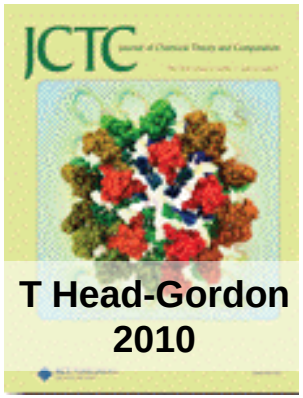
H. Guo 2010



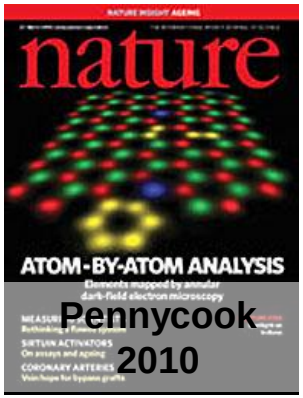
S. Salahuddin 2010



L.Sugiyama 2010

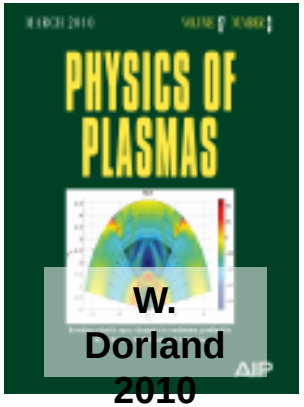


T Head-Gordon 2010

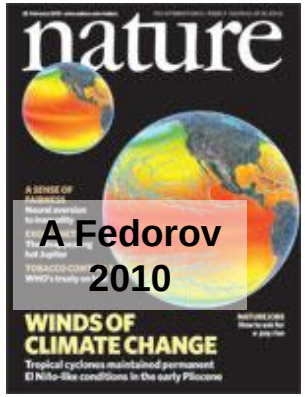


Pennycook 2010

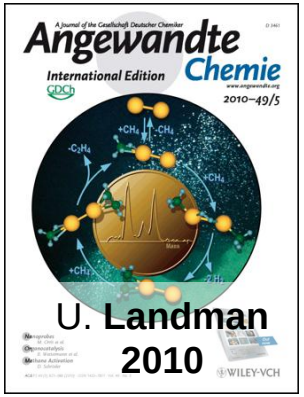
V. Daggett 2010



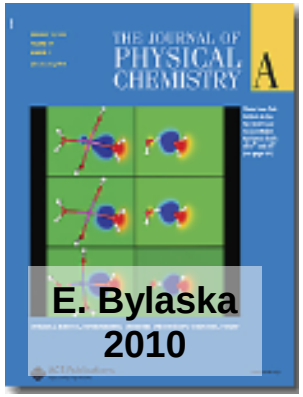
W. Dorland 2010



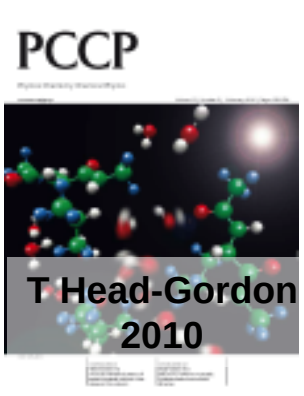
A Fedorov 2010



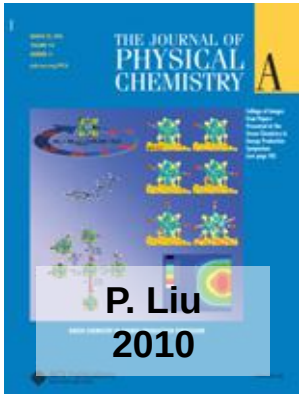
U. Landman 2010



E. Bylaska 2010



T Head-Gordon 2010



P. Liu 2010

NERSC is enabling new high quality science across disciplines, with over *1,600* refereed publications last year

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

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/XE6-feb-2011/compile

- **Copy necessary files to your $HOME directory as you don't have write permissions in the directory XE6-feb-2011 copy mpi_test.f90 and submit_static.scr to your home directory**
- **If you haven't run on a supercomputer before, take some time to go over a few simple examples**
- **Use Hopper website as a reference**
- **https://www.nersc.gov/systems/hopper-cray-**

Compile Hands On

In directory
/project/projectdirs/training/XE6-feb-2011/compile

- **Follow README for first example, or:**

% cp ***/project/projectdirs/training/XE6-feb-2011/compile/mpi_test.f90 ~***

```
% 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
- **Smallest allocatable unit**
  - Not shared

MOM Nodes

- 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

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

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

For this workshop

- Submit jobs to the “debug” queue – 30 min limit, 512 nodes, 12,288 cores.
- Debug queue has fast turn around
- Each participant has ~10k hours
- You are welcome to run larger jobs in “regular” queue if you have enough time/

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]*

Submit the job

% qsub submit_static.pbs

```
140979.sdb
```

Keep this jobid. It is often useful for debugging

Examine job output:

```
% cat my_job.o63731
```

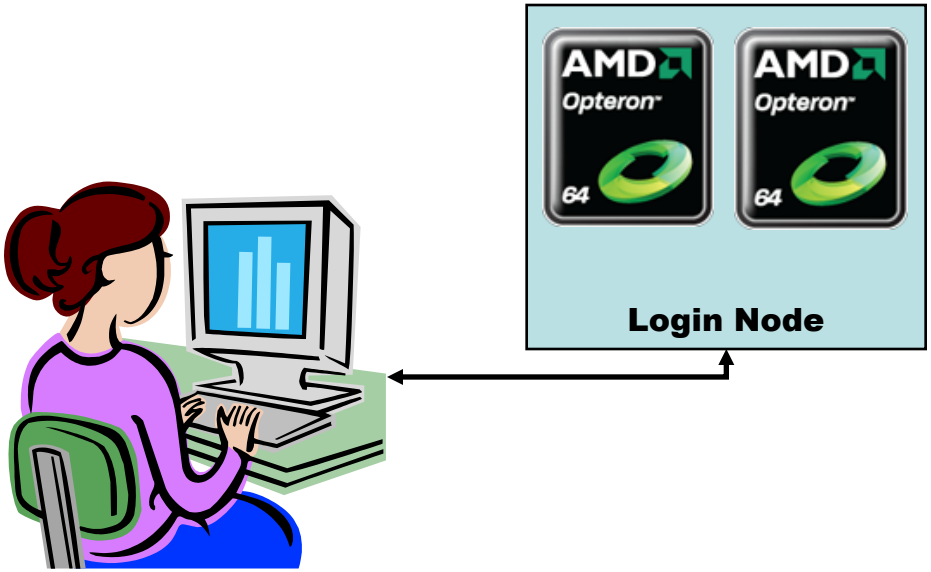
Monitoring Batch Jobs

- **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***

Summary -- Running a Job on the XE6



Login nodes run a full version of Linux

1. Log in from your desktop using SSH
2. Compile your code or load a software module
3. Write a job script
4. Submit your script to the batch system
5. Monitor your job's progress
6. Archive your output
7. Analyze your results

Running Interactively

```
% qsub -I -V
-l walltime=00:10:00
-l mppwidth=24 -q batch
qsub: waiting for job 140979.sdb
to start
qsub: job 140979.sdb ready
% cd $PBS_O_WORKDIR
% aprun -n 24 ./mpi_test
```

Basic aprun Options

Option	Description
-n	Number of MPI tasks.
-N	(Optional) Number of tasks per Beagle Node. Default is 24.

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

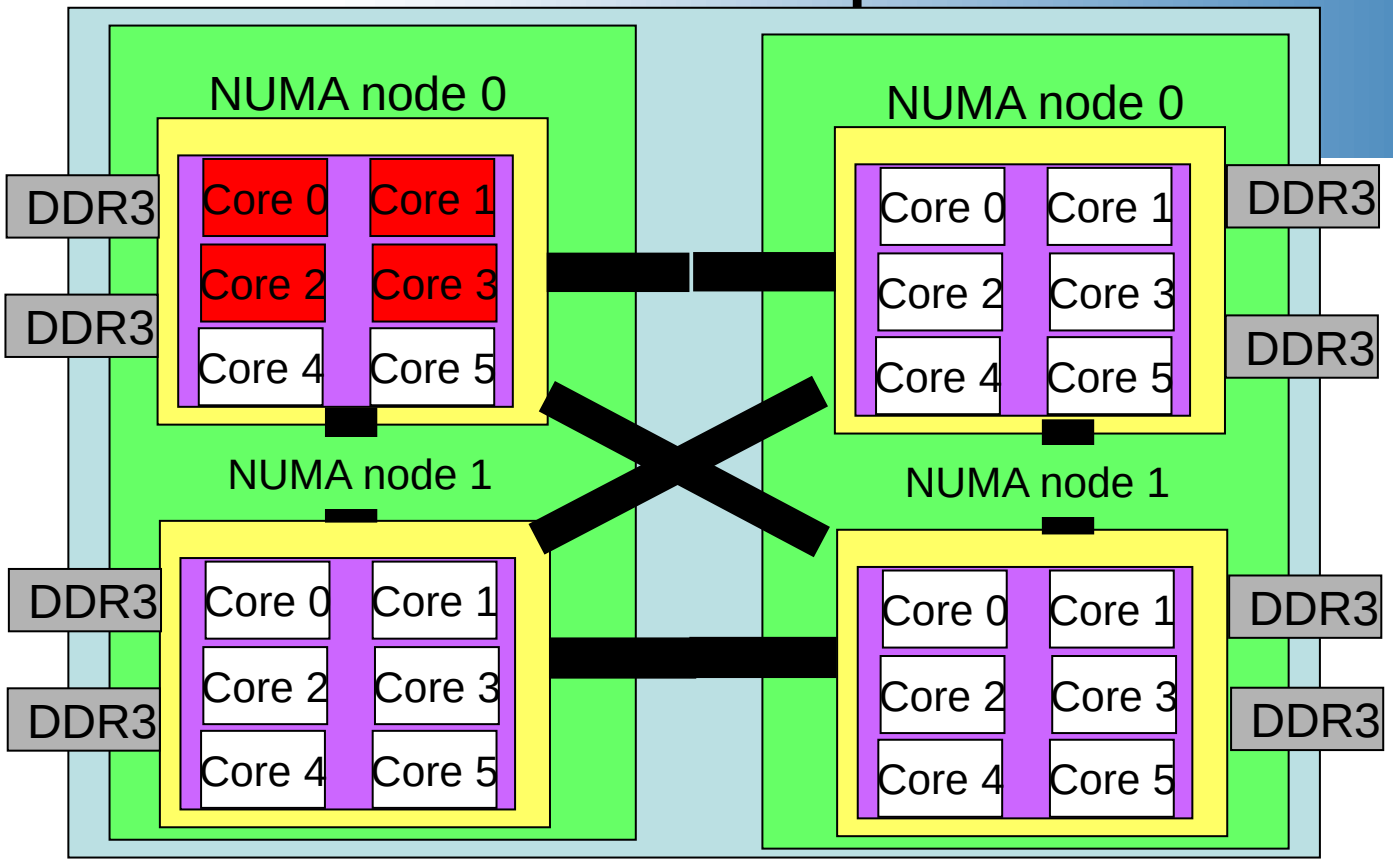
Unpacked

```
#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

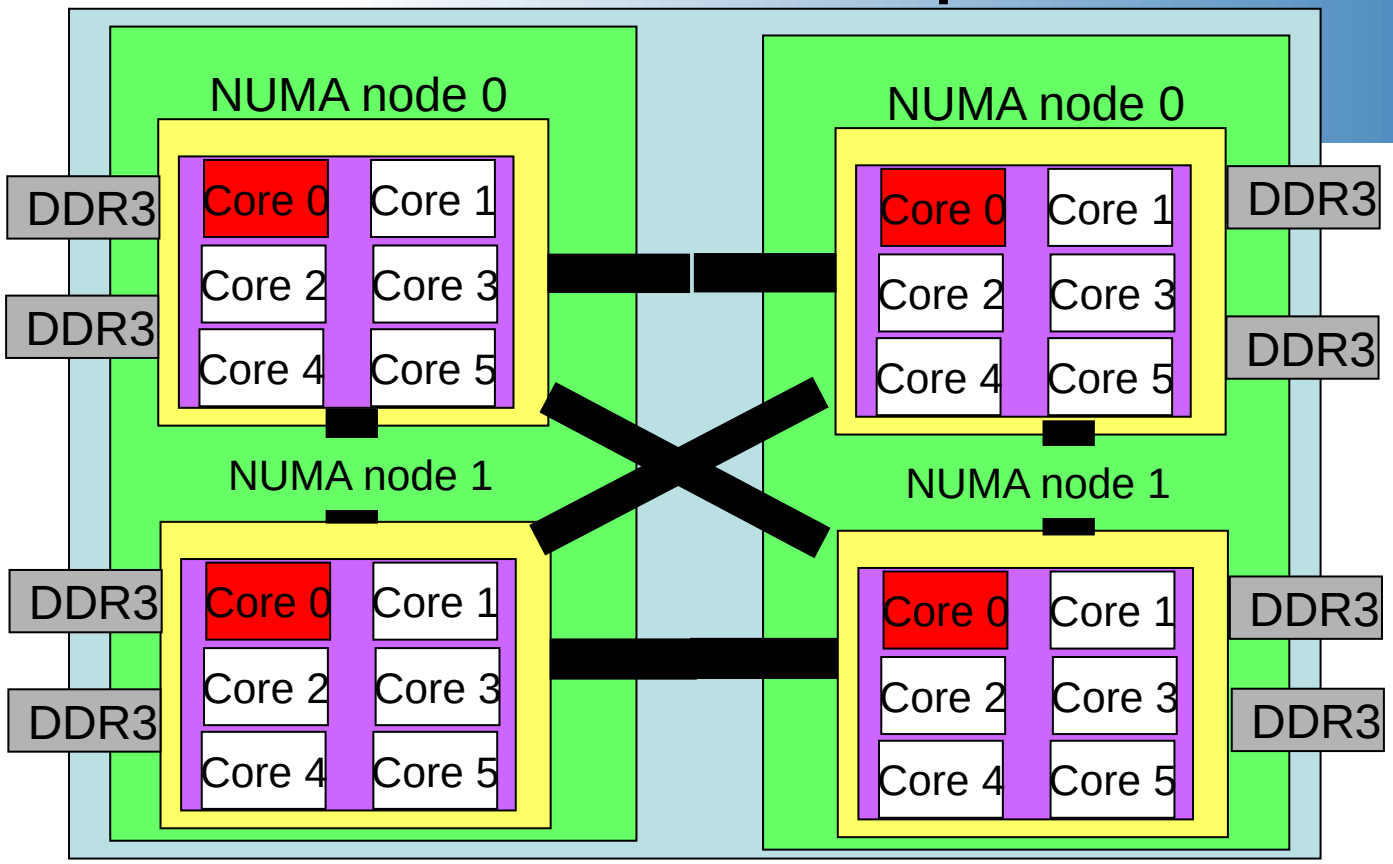


- *Example: 4 MPI tasks per node*
- *Default placement is not ideal when fewer than 24 cores per node are used.*

```
#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
```

Hands-On

/project/projectdirs/training/XE6-feb-2011/RunningParallel

```
jacobi_mpi.f90
jacobi.pbs
indata

mmsyst.f
mmsyst.pbs
```

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
```

XE6 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
```

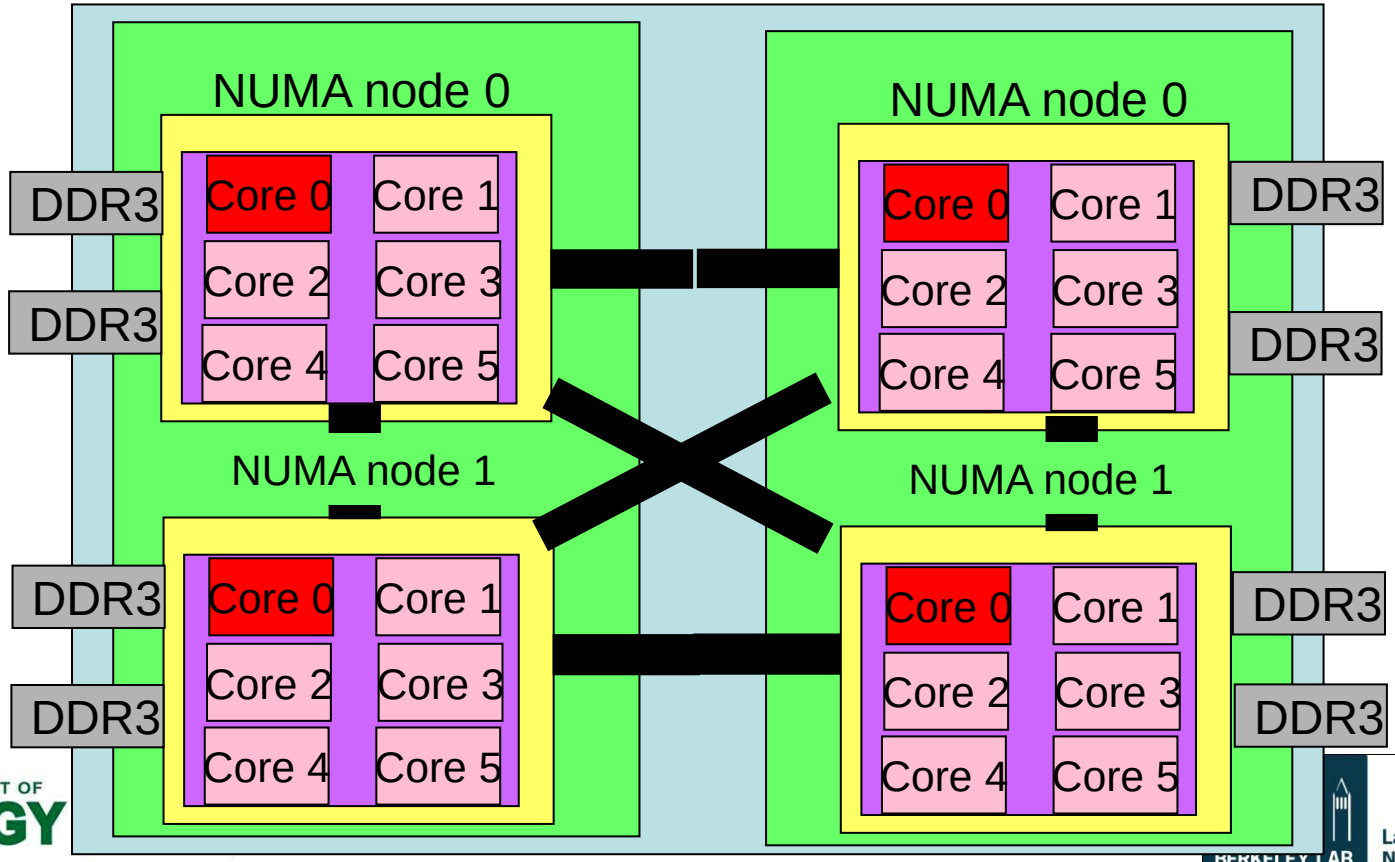
Useful aprun Options

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
```



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
```

Hands-On

/project/projectdirs/training/XE6-feb-2011/Mixed

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

Dynamic Shared Objects and Libraries

- **Using system provided dynamic shared libraries**
  - Swap modules “module swap xt-mpt xt-mpich2” (this should be the default soon)
  - Link codes with –dynamic
  - Set runtime variable CRAY_ROOTFS=DSL

See example in:

/project/projectdirs/training/XE6-feb-2011/compile

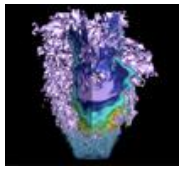
Dynamic Shared Objects and Libraries

- **User defined dynamic shared libraries**
  - Compile with –shared -fPIC
  - Set runtime variable CRAY_ROOTFS=DSL
  - Set runtime variable LD_LIBRARY_PATH

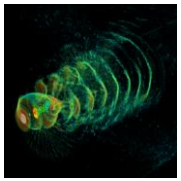
See example in:

/project/projectdirs/training/XE6-feb-2011/compile

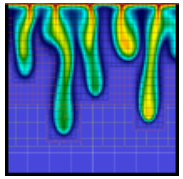
About the Cover



Low swirl burner combustion simulation. Image shows flame radical, OH (purple surface and cutaway) and volume rendering (gray) of vortical structures. Red indicates vigorous burning of lean hydrogen fuel; shows cellular burning characteristic of thermodiffusively unstable fuel. Simulated using an adaptive projection code. Image courtesy of John Bell, LBNL.

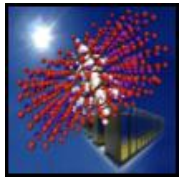


Hydrogen plasma density wake produced by an intense, right-to-left laser pulse. Volume rendering of current density and particles (colored by momentum orange - high, cyan - low) trapped in the plasma wake driven by laser pulse (marked by the white disk) radiation pressure. 3-D, 3,500 Franklin-core, 36-hour LOASIS experiment simulation using VORPAL by Cameron Geddes, LBNL. Visualization: Gunther Weber, NERSC Analytics.

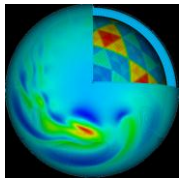


Numerical study of density driven flow for CO_2 storage in saline aquifers. Snapshot of CO_2 concentration after convection starts. Density-driven velocity field dynamics induces convective fingers that enhance the rate by which CO_2 is converted into negatively buoyant aqueous phase, thereby improving the security of CO_2 storage. Image courtesy of George Pau, LBNL

False-color image of the Andromeda Galaxy created by layering 400 individual images captured by the Palomar Transient Factory (PFT) camera in February 2009. NERSC systems analyzing the PTF data are capable of discovering cosmic transients in real time. Image courtesy of Peter Nugent, LBNL.



The exciton wave function (the white isosurface) at the interface of a ZnS/ZnO nanorod. Simulations performed on a Cray XT4 at NERSC, also shown. Image courtesy of Lin-Wang Wang, LBNL.



Simulation of a global cloud resolving model (GCRM). This image is a composite plot showing several variables: wind velocity (surface pseudocolor plot), pressure (b/w contour lines), and a cut-away view of the geodesic grid. Image courtesy of Professor David Randall, Colorado State University.

Extra Slides

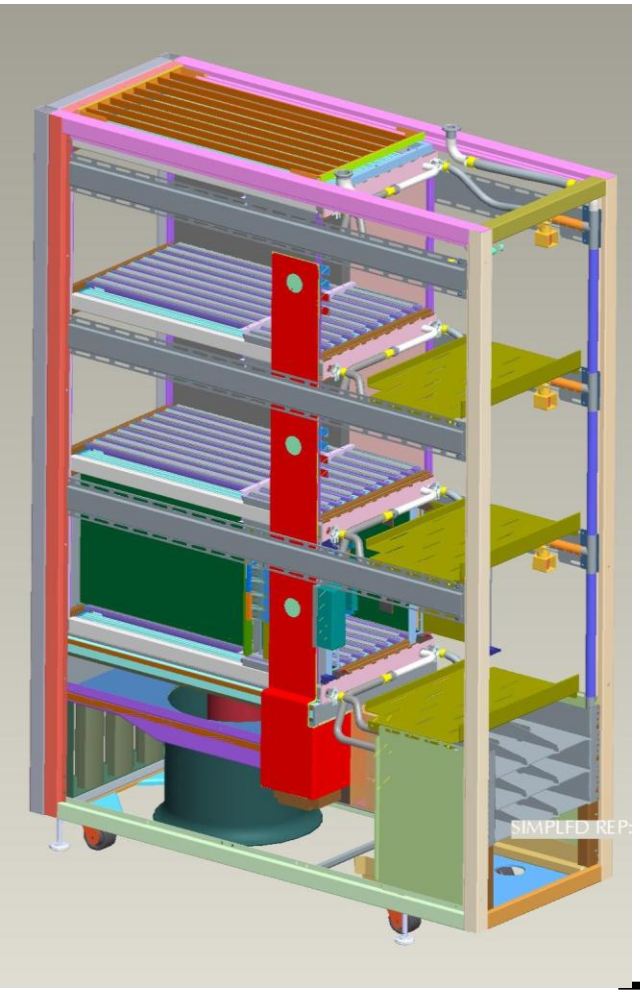
71

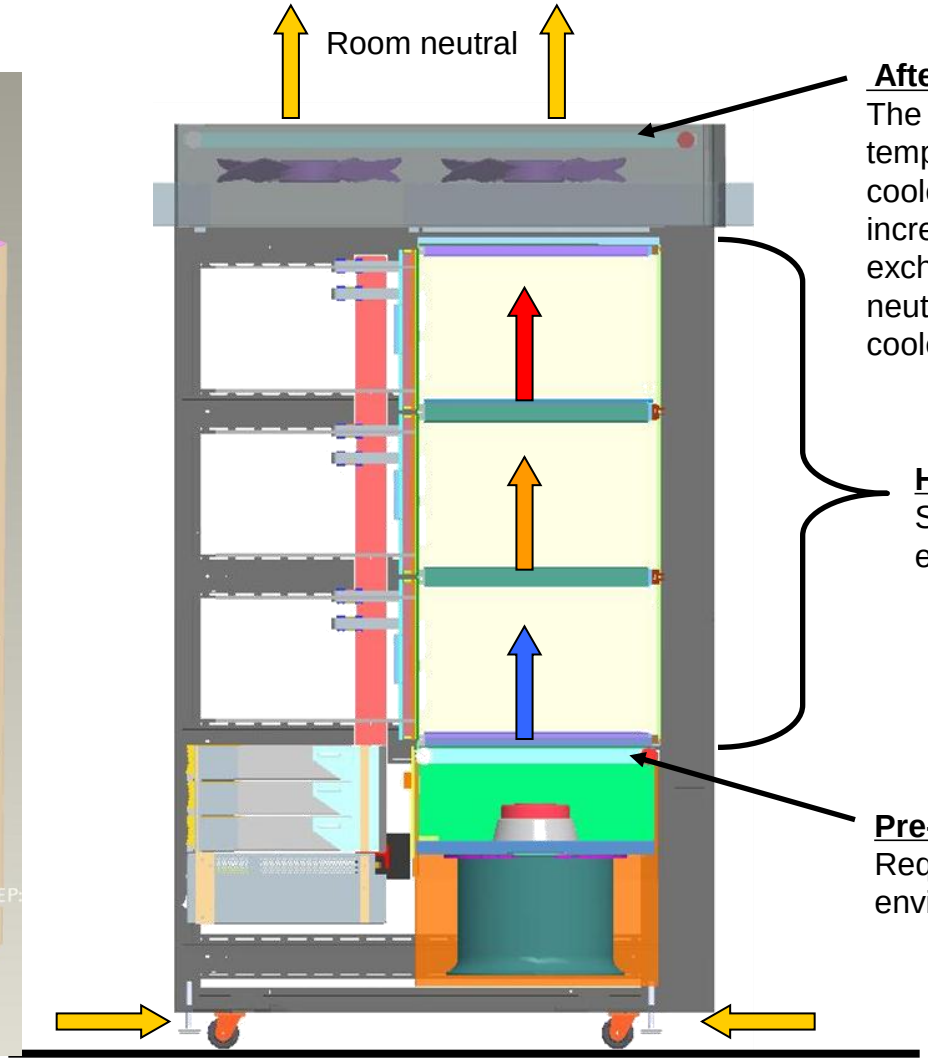
XE6 Cabinet Design

What's up with the hat??

Cray Cabinet Design

Energy Efficient Liquid Cooling





After-cooler assembly: The extremely hot exhaust temperature of the HD air cooled chassis dramatically increases the capability of heat exchanger. This makes room neutral possible with single cooler assembly at exit.

HD Air Cooled Chassis: Sandwich with R134a evaporators.

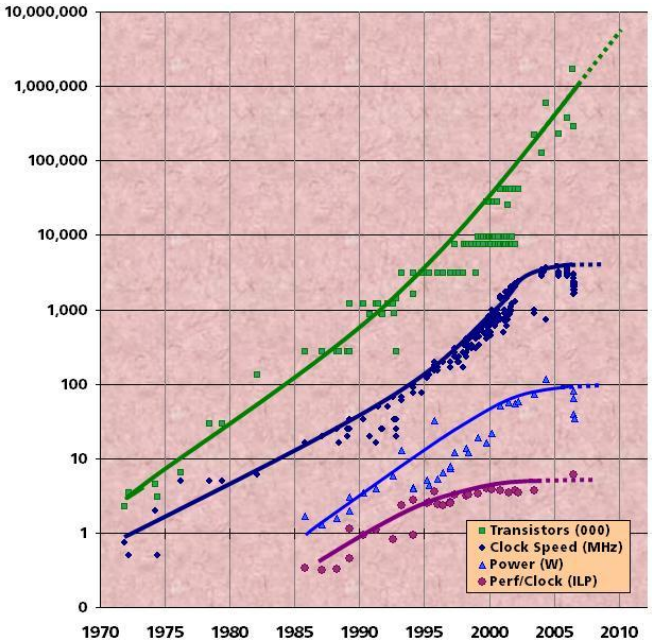
Pre-cooler assembly: Required to operate in room environments over 20C.

Images Courtesy of Cray Inc.

Hopper Cooling Apparatus

What About the Future?

- **The technology trends point to**
  - **Little or no gain in clock speed or performance per core;**
  - **Rapidly increasing numbers of cores per node;**
  - **Decreased memory *capacity* per core (possible slight increase per node)**
  - **Decreased memory *bandwidth* per core**
  - **Decreased interconnect bandwidth per core**
  - **Deeper memory hierarchy**

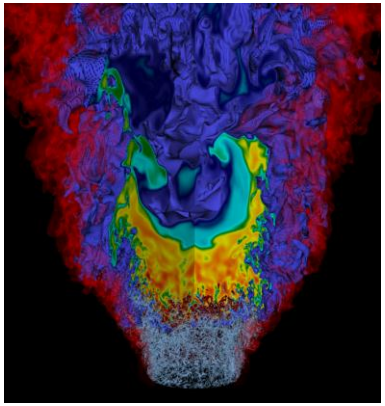


- **Hopper is the first example at NERSC but surely not the last**

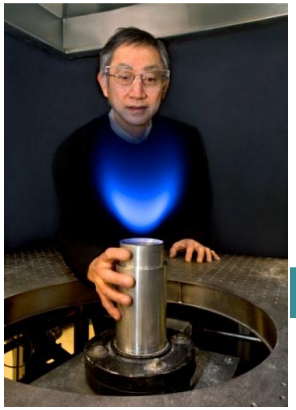
Computation and Experiments at Berkeley Lab Improve Efficiency of Burners

- **Low Swirl Burners used by Solar Turbines (Caterpillar) and Maxon Corp. (Honeywell) to improve commercial burners**
  - **Efficient, low-emissions, Fuel-flexible (oil, gas, hydrogen-rich fuels)**
- **Simulations explain combustion process to improve designs**
  - **Modeled kinetics and chemical transport (15 species, 58 reactions)**
  - **Uses advanced math algorithms (AMR) equivalent to $4K^3$ mesh**
  - **Scales and runs in production at 20K cores**

Simulations show cellular burning in lean hydrogen leads to pockets of enhanced emissions, & increasing the turbulence enhances the effect.



Simulations reveal features not visible in lab (John Bell, PI, LBNL)



Experiments show feasibility: 50KW-50MW (Robert Cheng, PI, LBNL)

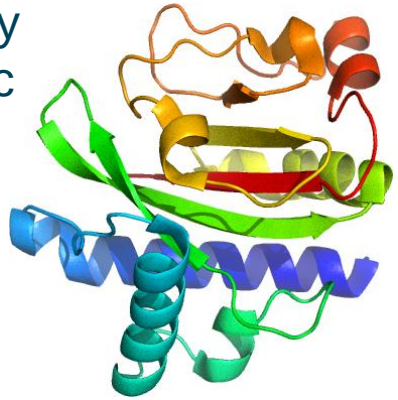
Low NOx technology licensed by industry

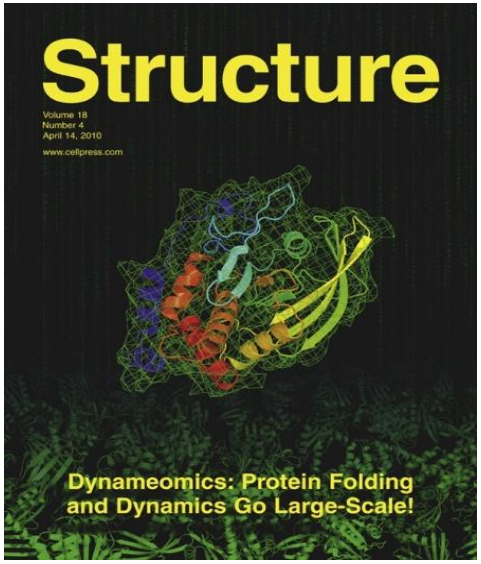
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Simulations Populate a Database of Molecular Dynamics and Protein Folds

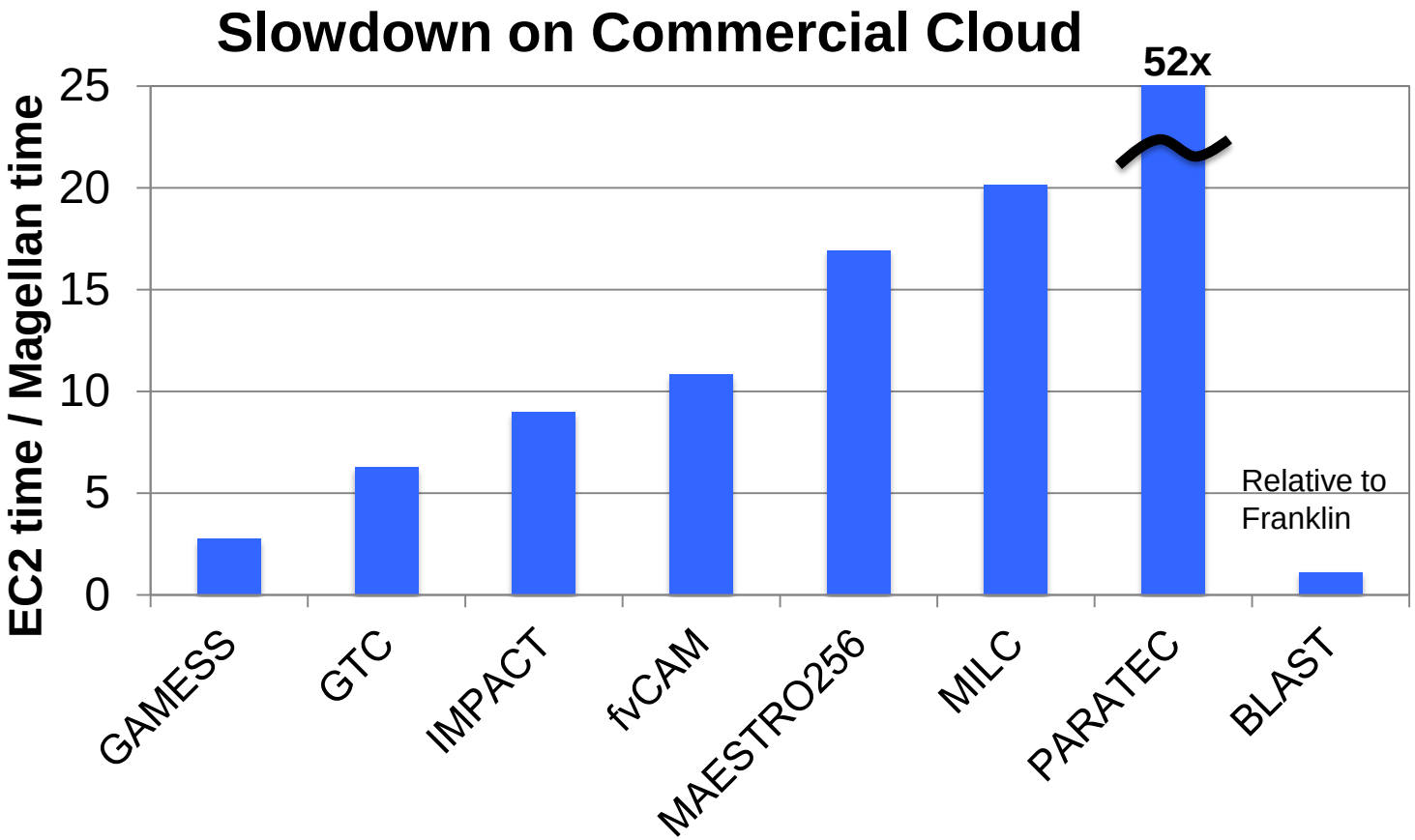
- **Produced public catalog of the unfolding dynamics of 11,000 proteins, covering all 807 self-contained autonomous folds**
- **Simulations used 12M hours of NERSC on custom code and help from NERSC on load balancing, optimizations, and workflow**
- **Mined amyloid producing proteins and found common structural feature between normal and toxic forms.**
  - Custom-designed complementary compounds, which bind with toxic forms of proteins that cause multiple diseases, including Alzheimer's and mad cow.
  - Results suggest drug designs, screening for blood/food supply, and diagnostic tools for up to 25 amyloid diseases.





Valerie Daggett, PI, U. Washington

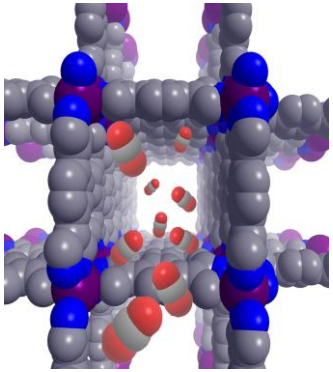
Provide Cloud Computing Testbed and Evaluation

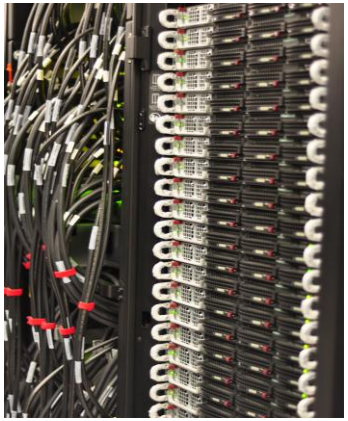


On traditional science workloads, standard cloud configurations see significant slowdown (up to 50x), but independent BLAST jobs run well

Provide GPU Testbed and Evaluation

- **Installed "Dirac" GPU testbed**
  - About100 users so far
  - Popular with SciDAC-E postdocs
- **Example: Q-Chem Routine**
  - Impressive single node speedups relative to 1 core on CPU
  - Highly variable with input structure



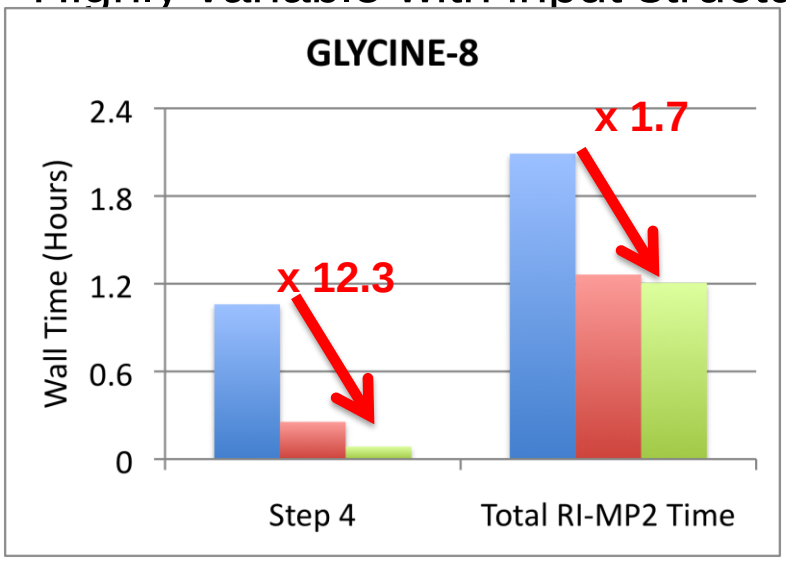


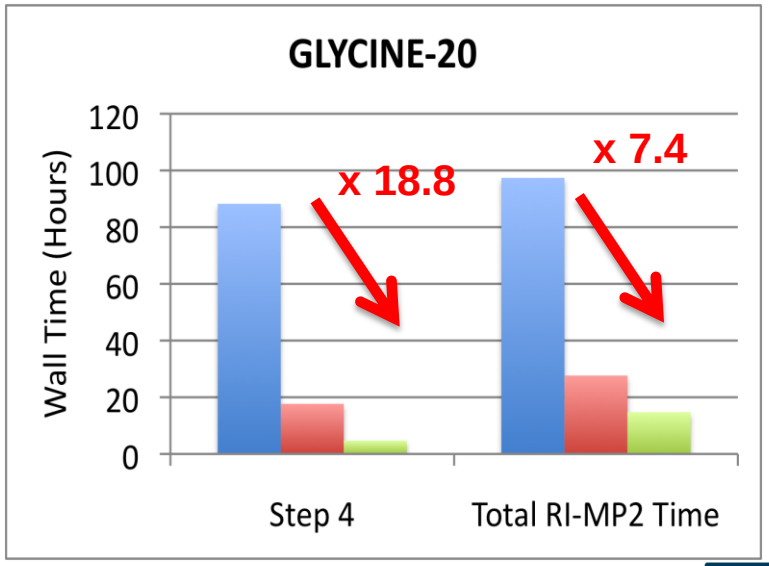
Fermi GPU Racks - NERSC

Blue: CPU 1 thread

Red: CPU 8 threads

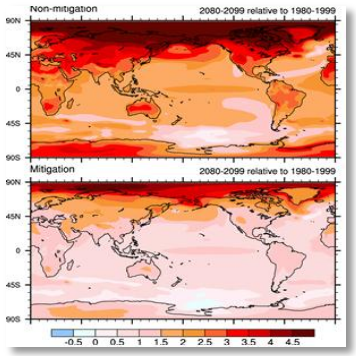
Green: GPU





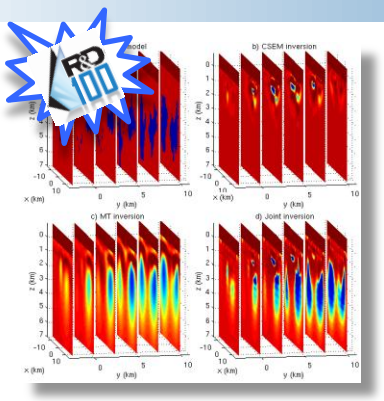
Jihan Kim (SciDAC-E NERSC Postdoc)

Sample Scientific Accomplishments at NERSC



Climate

Studies show that global warming can still be diminished if society cuts emissions of greenhouse gases.
(Warren Washington, NCAR)

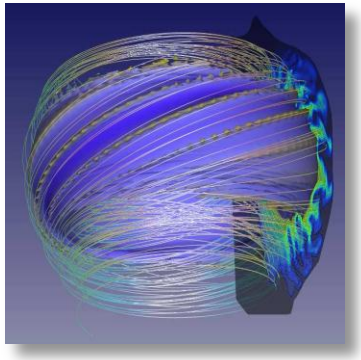


Energy Resources

Award-winning software uses massively-parallel supercomputing to map hydrocarbon reservoirs at unprecedented levels of detail.
(Greg Newman, LBNL)

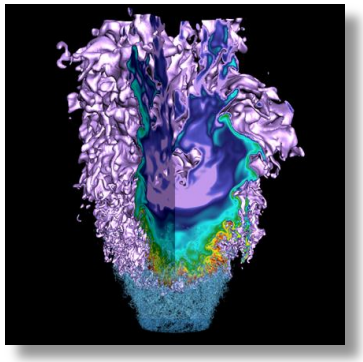
Fusion Energy

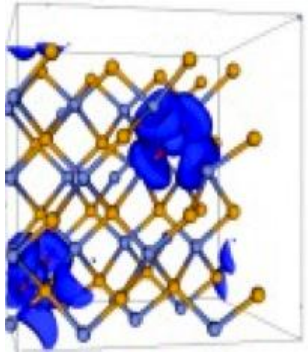
A new class of non-linear plasma instability has been discovered that may constrain design of the ITER device.
(Linda Sugiyama, MIT)



Combustion

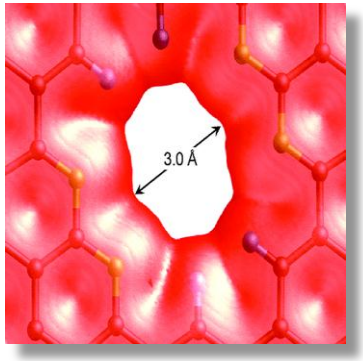
Adaptive Mesh Refinement allows simulation of a fuel-flexible low-swirl burner that is orders of magnitude larger & more detailed than traditional reacting flow simulations allow.
(John Bell, LBNL)





Materials

Electronic structure calculations suggest a range of inexpensive, abundant, non-toxic materials that can produce electricity from heat.
(Jeffrey Grossman, MIT)

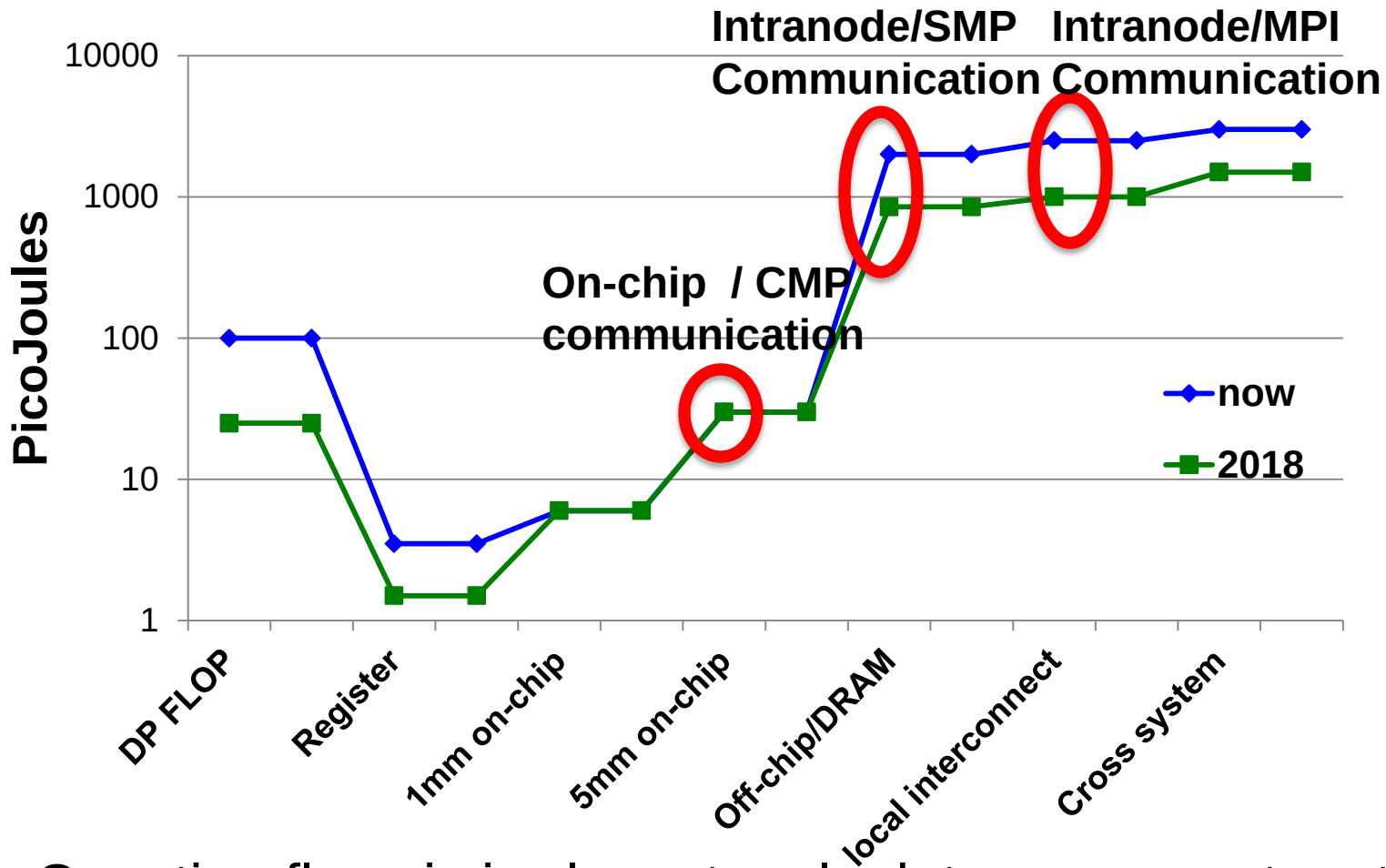


Nano Science

Using a NERSC NISE grant researchers discovered that Graphene may be the ultimate gas membrane, allowing inexpensive industrial gas production.
(De-en Jiang, ORNL)

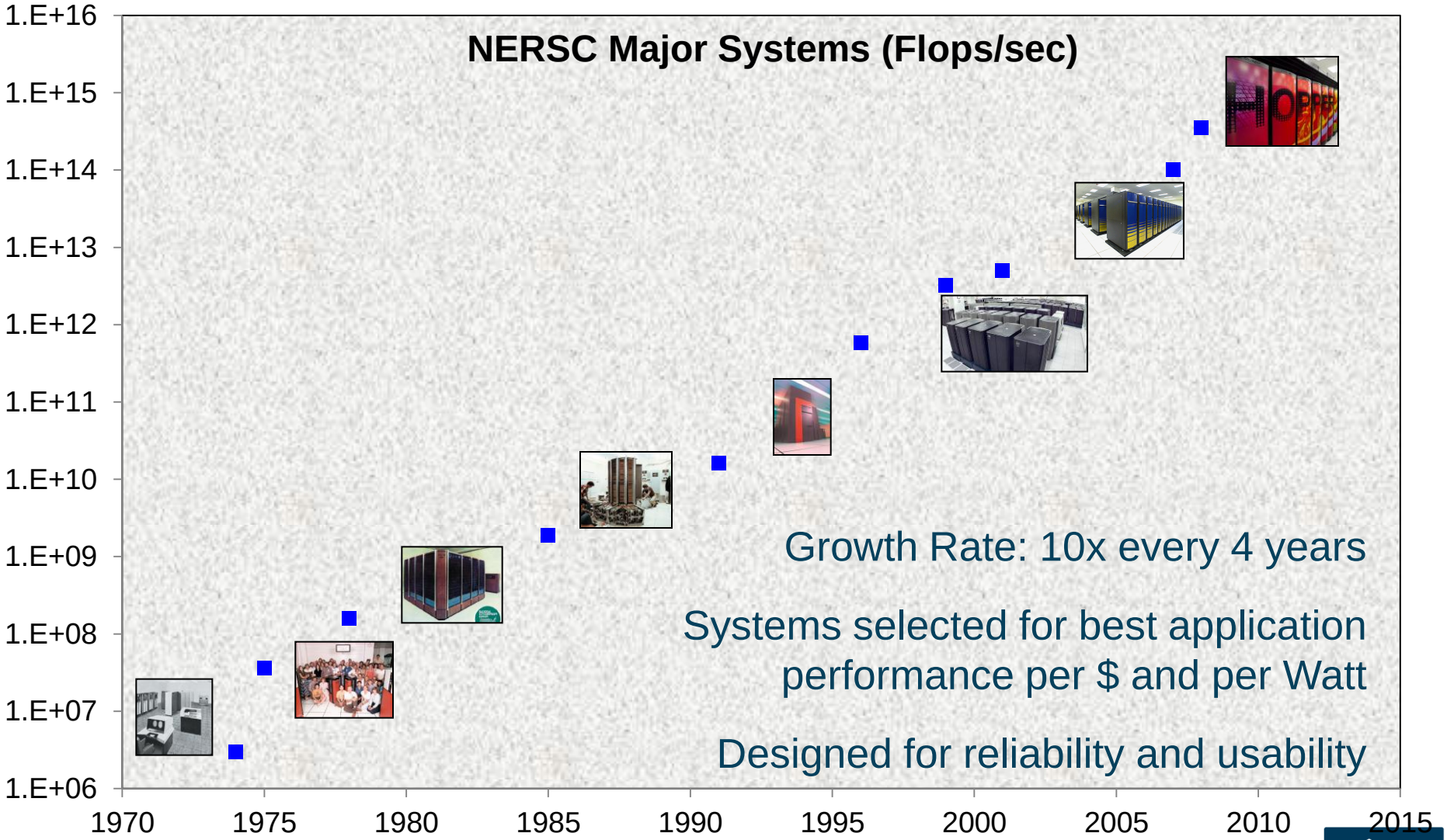
U.S. DEPARTMENT OF ENERGY | Office of Science

Where does the Energy (and Time) Go?



Counting flops is irrelevant, only data movement matters

NERSC Responds to Scientific Demands for Computing and Services



Growth Rate: 10x every 4 years

Systems selected for best application performance per $ and per Watt

Designed for reliability and usability

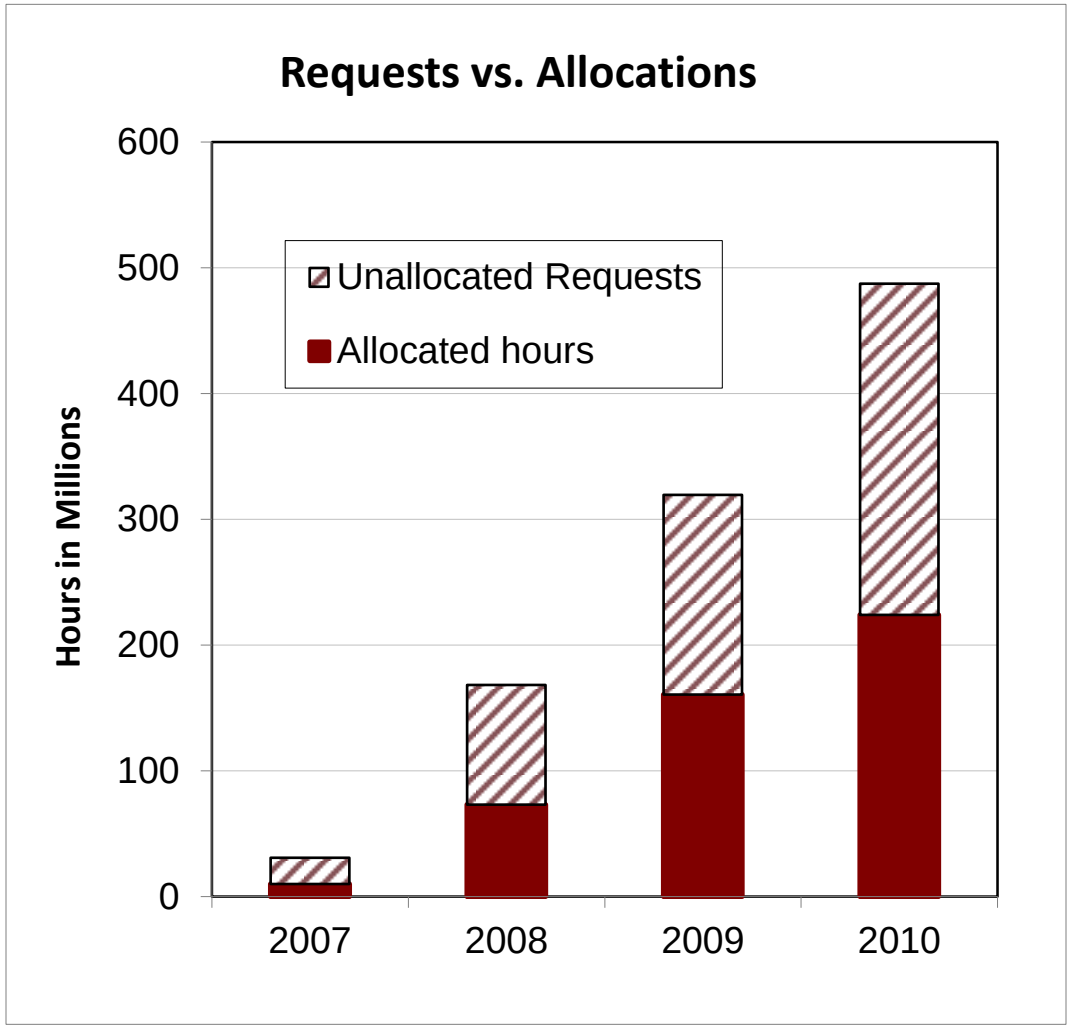
Challenges to ~~Exascale~~

Performance Growth

1) **System power** is the primary constraint
2) **Concurrency** (1000x today)
3) **Memory** bandwidth and capacity are not keeping pace
4) **Processor** architecture is an open question
5) **Programming model** heroic compilers will not hide this
6) **Algorithms** need to minimize data movement, not flops
7) **I/O bandwidth** unlikely to keep pace with machine speed
8) **Reliability and resiliency** will be critical at this scale
9) **Bisection bandwidth** limited by cost and energy

Unlike the last 20 years most of these (1-7) are equally important across scales, e.g., 100 10-PF machines

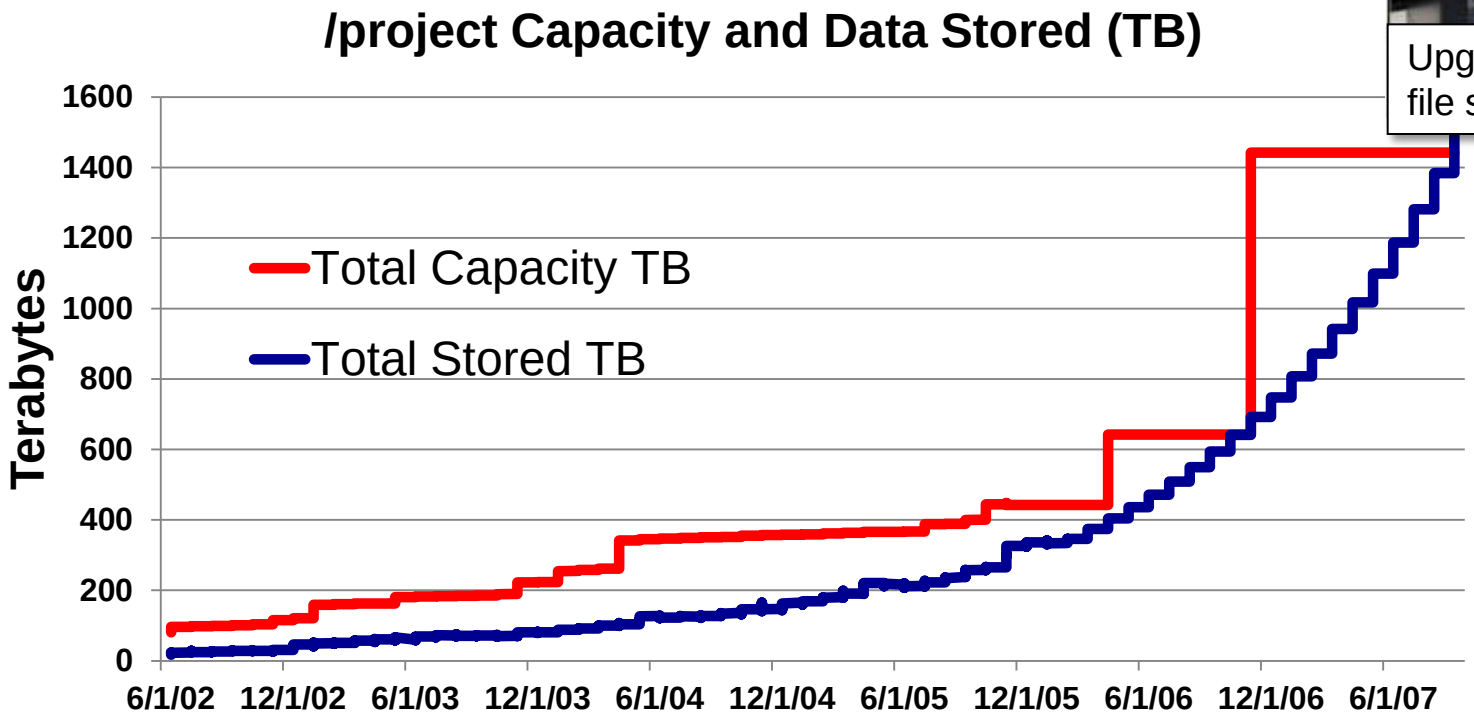
Demand for More Computing



- *Each year DOE users requests ~2x as many hours as can be allocated*
- *This 2x is artificially constrained by perceived availability*
- *Unfulfilled allocation requests amount to hundreds of millions of compute hours in 2010*

NERSC Global Filesystem Upgrades & Enhancements

Upgraded global file system

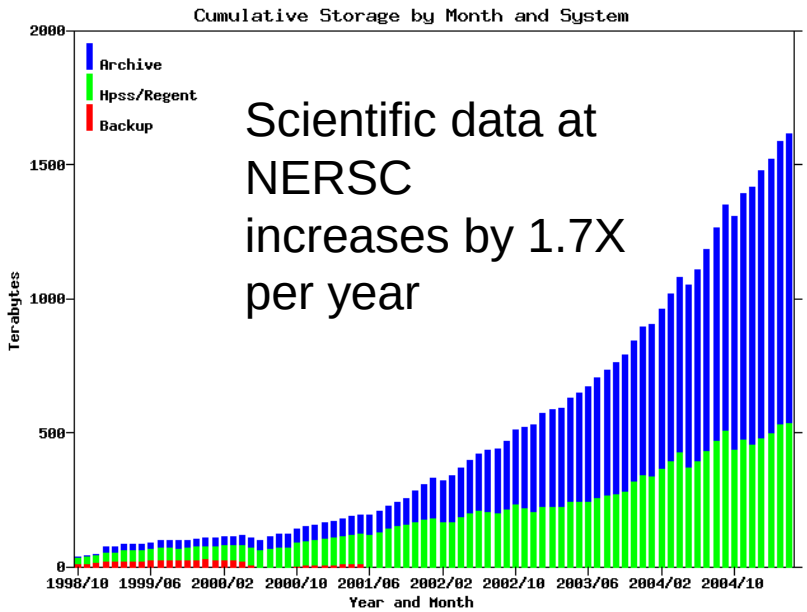


- Extended global filesystem from “project” to scratch and home directories for convenience
- Different service models for capacity (project), random access performance (home), temporary data (scratch)

NERSC Strategy: *Science First*

- **Response to scientific needs**
  - **Requirements setting activities**
- **Support computational science:**
  - **Provide effective machines that support fast algorithms**
  - **Deploy with flexible software**
  - **Help users with expert services**
- **NERSC future priorities are driven by science:**
  - **Increase application capability: "usable Exascale"**
  - **For simulation and data analysis**

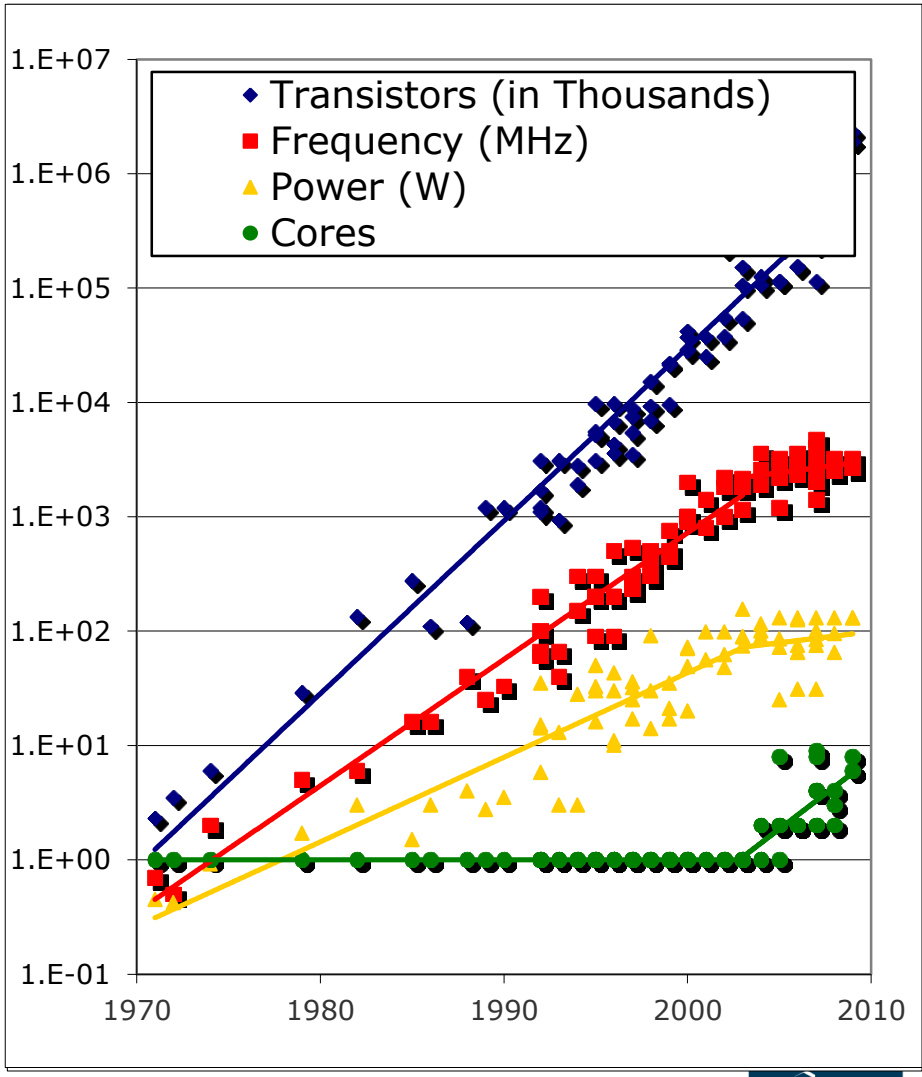
Tape Archives: Green Storage



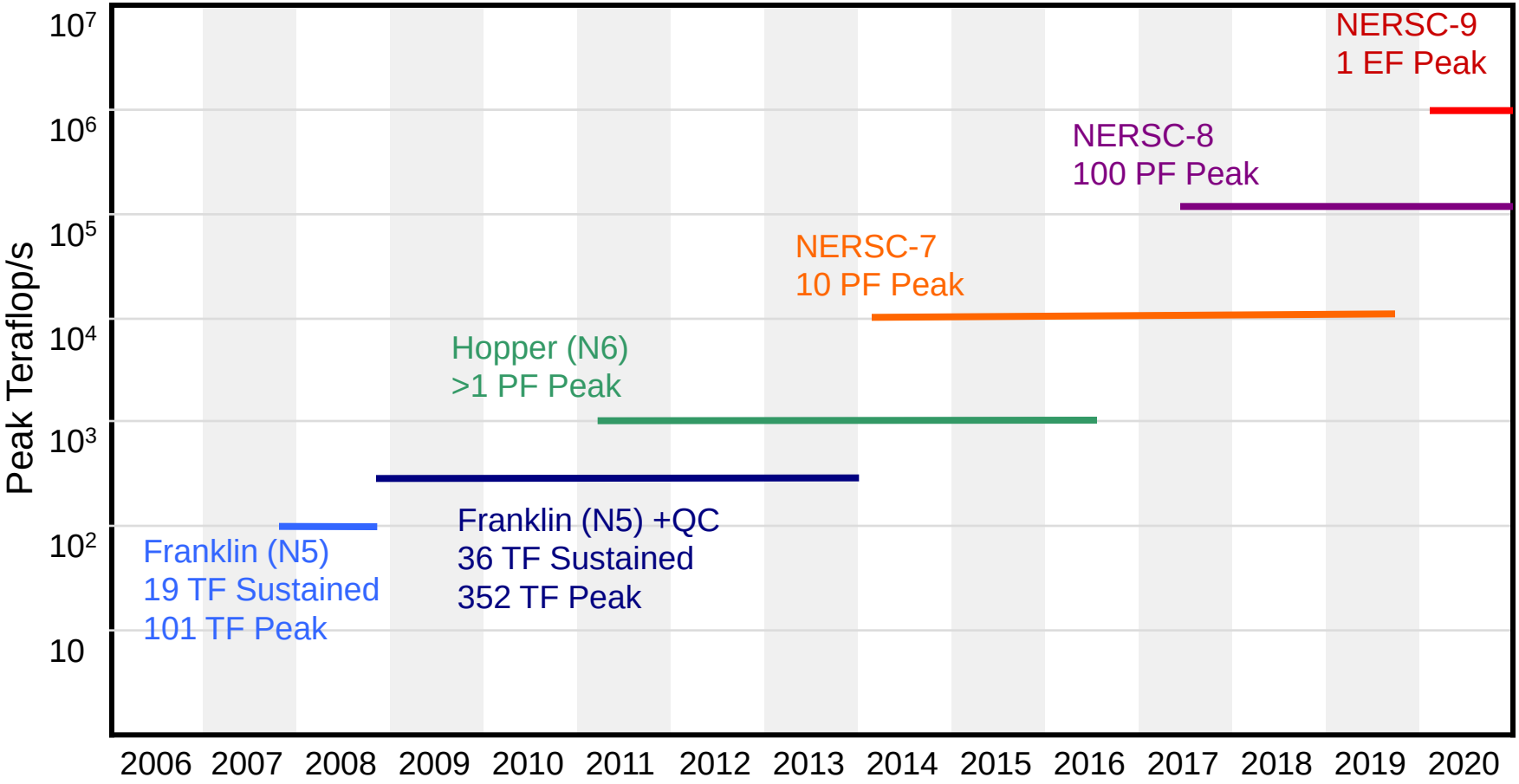
- **Tape archives are important to efficient science**
  - **2-3 orders of magnitude less power than disk**
  - **Requires specialized staff and major capital investment**
  - **NERSC participates in development (HPSS consortium)**
- **Questions: What are your data sets sizes and growth rates?**

Moore's Law Continues, but Only with Added Concurrency

- Power density limit single processor clock speeds
- Cores per chip is growing
- Simple doubling of cores is not enough to reach exascale
  - Also a problem in data centers, laptops, etc.
- Two paths to exascale:
  - Accelerators (GPUs)
  - Low power embedded cores
  - (Not x86 clusters)



NERSC Aggressive Roadmap



- NERSC goal is application performance (~10x every 3 years)
- Peak numbers assume (generous) 10% of peak for applications

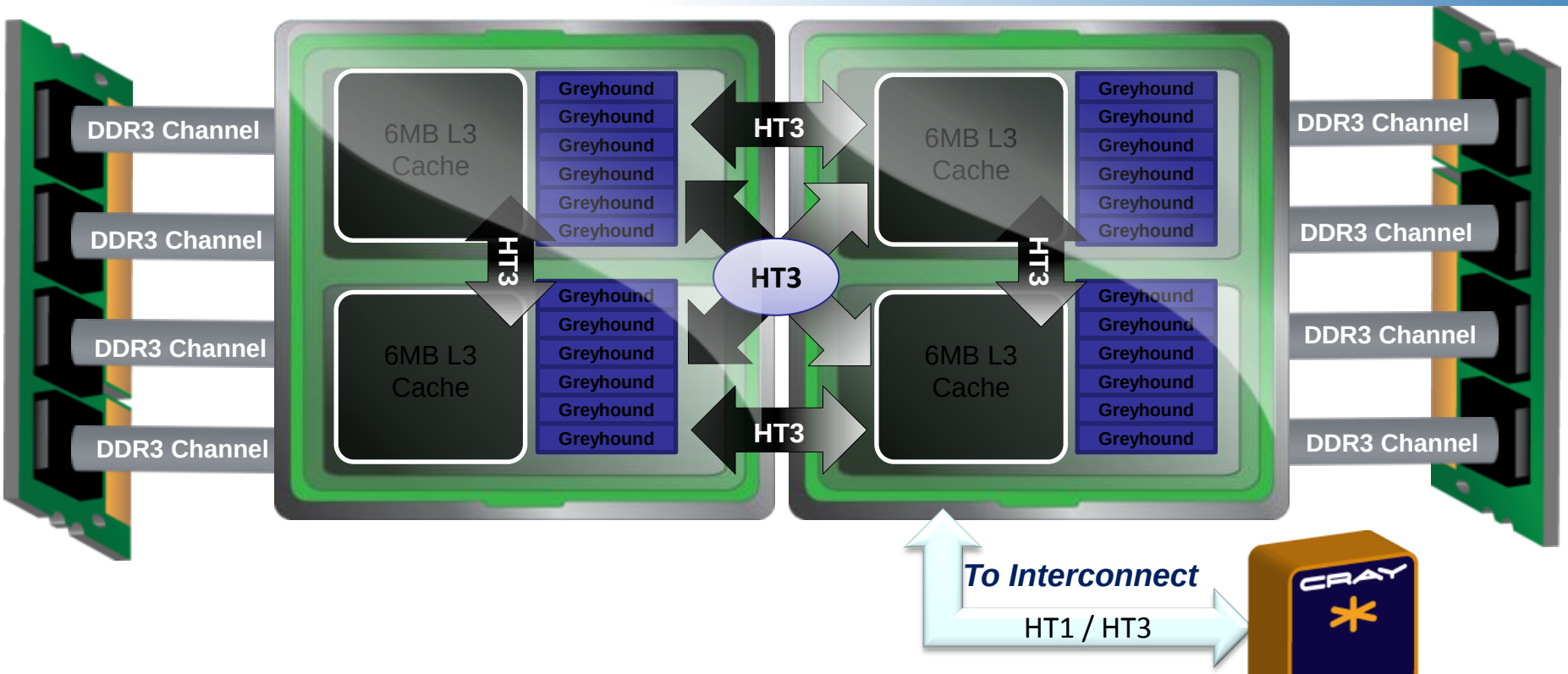
NERSC Mission

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.

Developing HPC Applications for Optimal Performance

91

What is Different About Hopper?



- **Hopper system has 24 cores per node.**
- **The way that you use the new Hopper system may have to change as a result.**

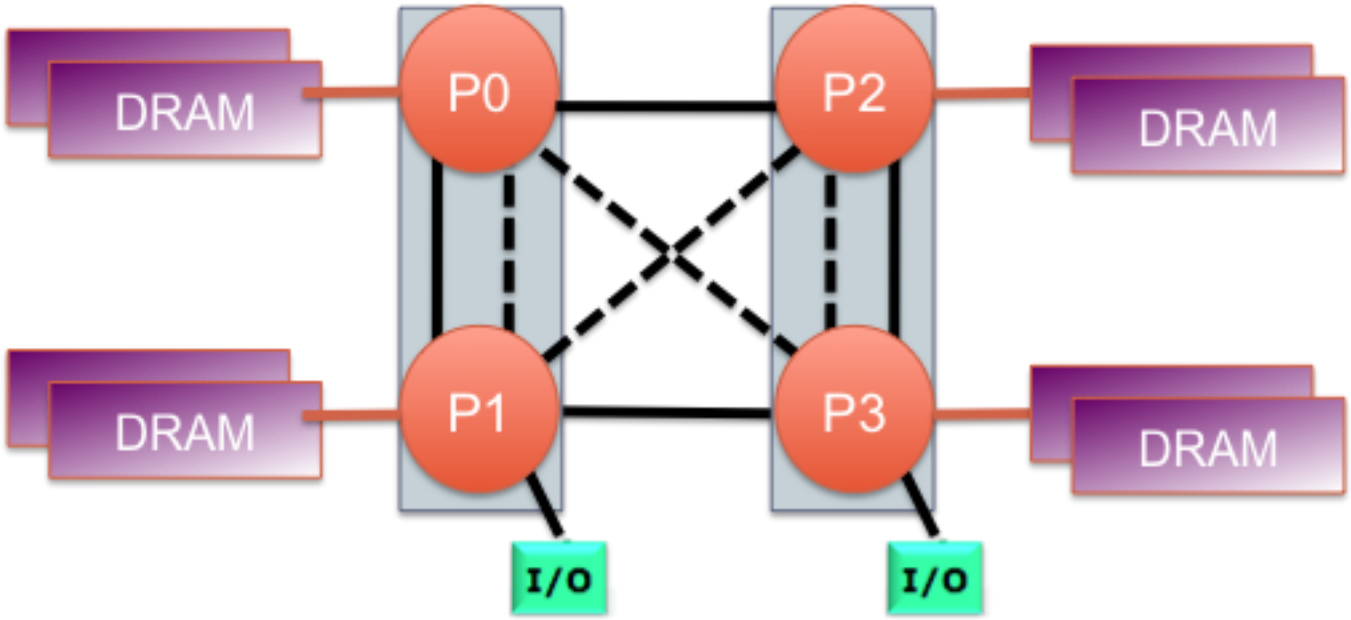
Hopper Node Topology
Understanding NUMA Effects

- **Heterogeneous Memory access between dies**
- **"First touch" assignment of pages to memory.**

2xDDR1333 channel
21.328 GB/s

3.2GHz x8 lane HT
6.4 GB/s bidirectional

3.2GHz x16 lane HT
12.8 GB/s bidirectional



- **Locality is key** *(just as per Exascale Report)*
- **Only *indirect* locality control with OpenMP**

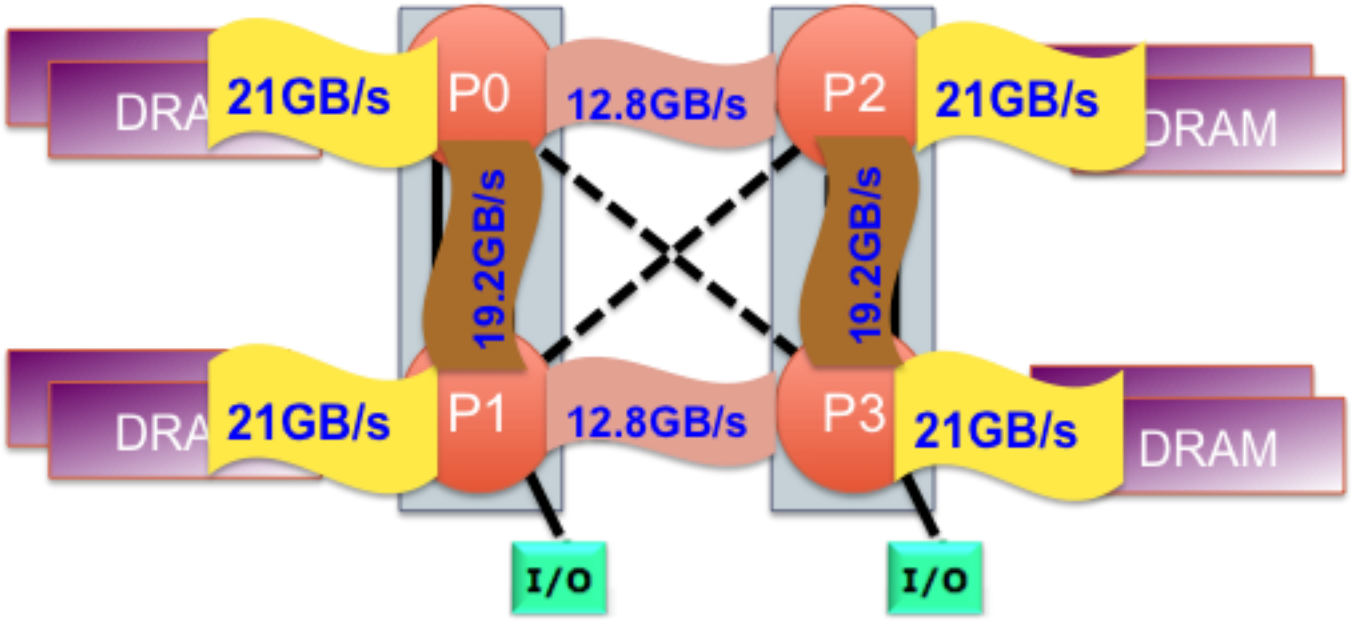
Hopper Node Topology
Understanding NUMA Effects

- **Heterogeneous Memory access between dies**
- **"First touch" assignment of pages to memory.**

2xDDR1333 channel
21.328 GB/s

3.2GHz x8 lane HT
6.4 GB/s bidirectional

3.2GHz x16 lane HT
12.8 GB/s bidirectional



- **Locality is key** *(just as per Exascale Report)*
- **Only *indirect* locality control with OpenMP**

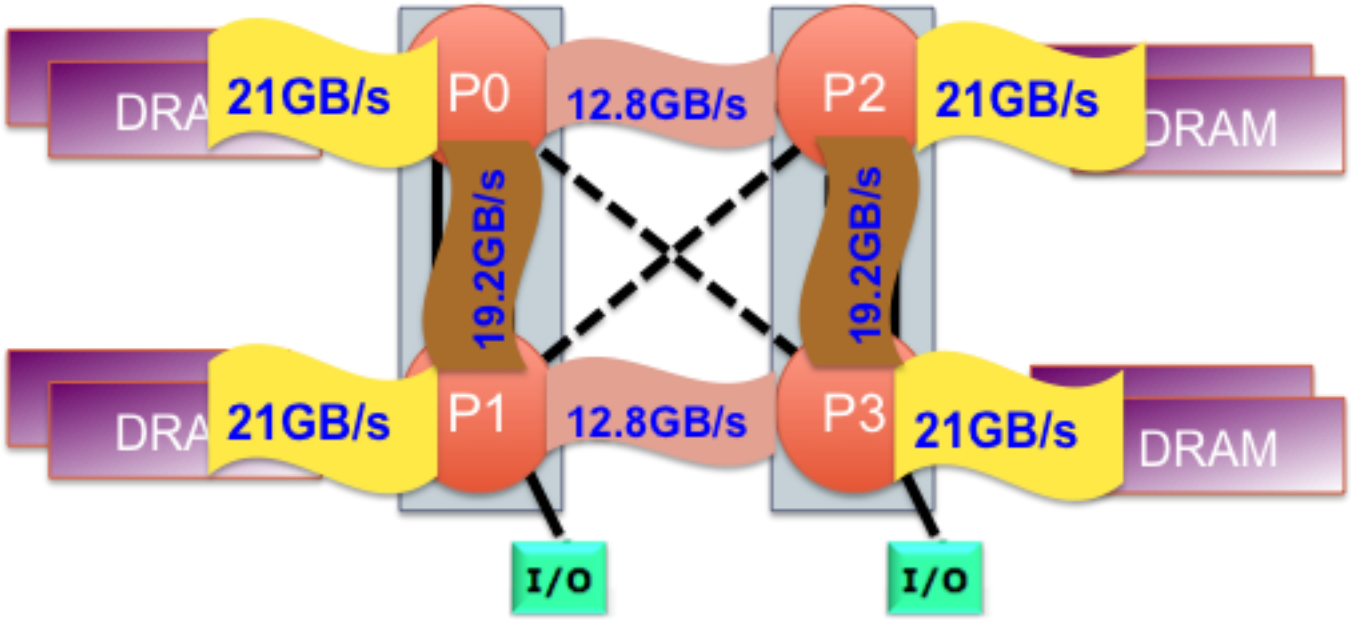
Hopper Node Topology
Understanding NUMA Effects

- **Heterogeneous Memory access between dies**
- **"First touch" assignment of pages to memory.**

2xDDR1333 channel
21.328 GB/s

3.2GHz x8 lane HT
6.4 GB/s bidirectional

3.2GHz x16 lane HT
12.8 GB/s bidirectional



- **Locality is key** *(just as per Exascale Report)*
- **Only *indirect* locality control with OpenMP**

What Else is Different ?

- **Less memory per core: 1.33 GB *vs.* 2.0 GB**
  - **8 GB per node (Franklin);**
  - **32 GB per node (Hopper, 6,008 nodes)**

- **"OOM killer terminated this process" error**

  OOM = Out of Memory

- **(Hopper has 384 larger-memory nodes 64 GB.)**

Will My Existing Pure MPI Code Run?

- **Probably, yes, your MPI code will run.**

- **But the decrease in memory available per core may cause problems ...**
  - **May not be able to run the same problems.**
  - **May be difficult to continue "weak" scaling (problem size grows in proportion to machine size).**

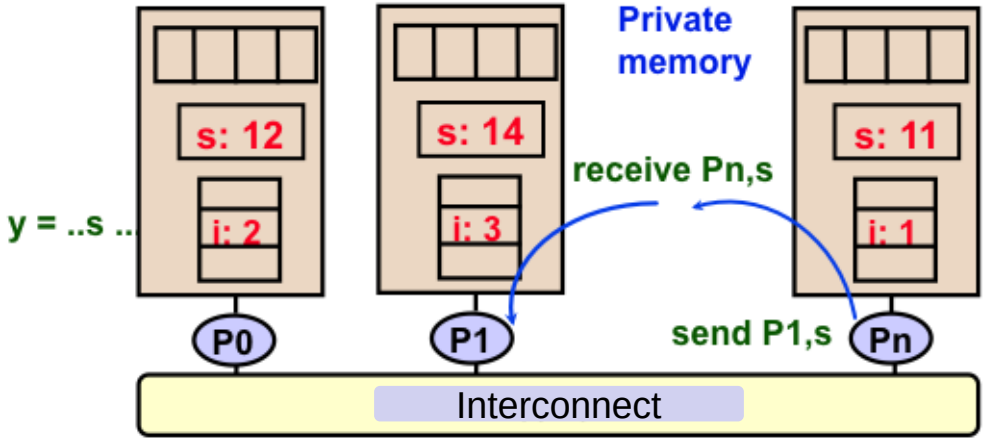
- **(and your MPI code might not use the machine most effectively.)**

- **Time to consider alternative programming models?**

What Does NERSC Recommend?

- NERSC recognizes the huge investment in MPI.
- But given the technology trends…
- We suggest a move towards programming models other than pure MPI
- A good place to start: MPI + OpenMP ("Hybrid")
  - MPI for domain decomposition and OpenMP threads within a domain
  - Suggested primarily to help with memory capacity

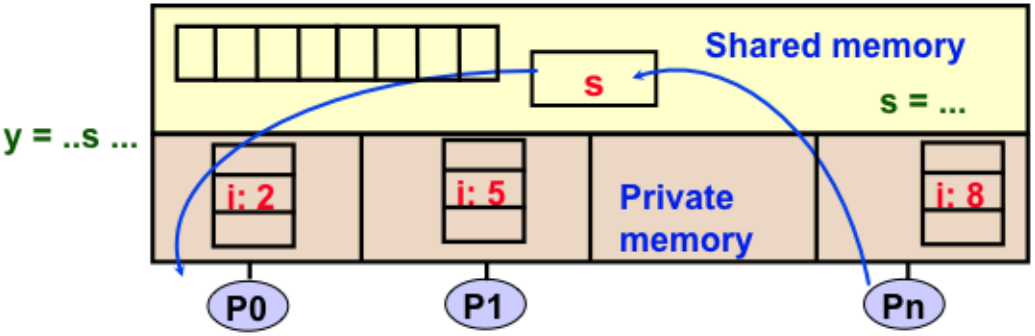
What are the Basic Differences Between MPI and OpenMP?



Message Passing Model

- Program is a collection of processes.
  - **Usually fixed at startup time**
- Single thread of control plus private address space -- NO shared data.
- Processes communicate by explicit send/receive pairs
  - **Coordination is implicit in every communication event.**
- MPI is most important example.

Shared Address Space Model

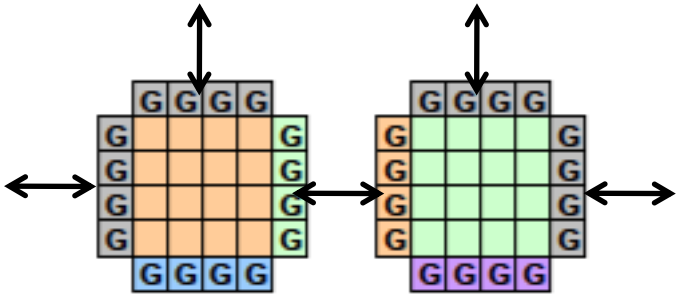


- Program is a collection of threads.
  - **Can be created dynamically.**
- Threads have private variables and shared variables
- Threads communicate implicitly by writing and reading shared variables.
  - **Threads coordinate by synchronizing on shared variables**
- OpenMP is an example

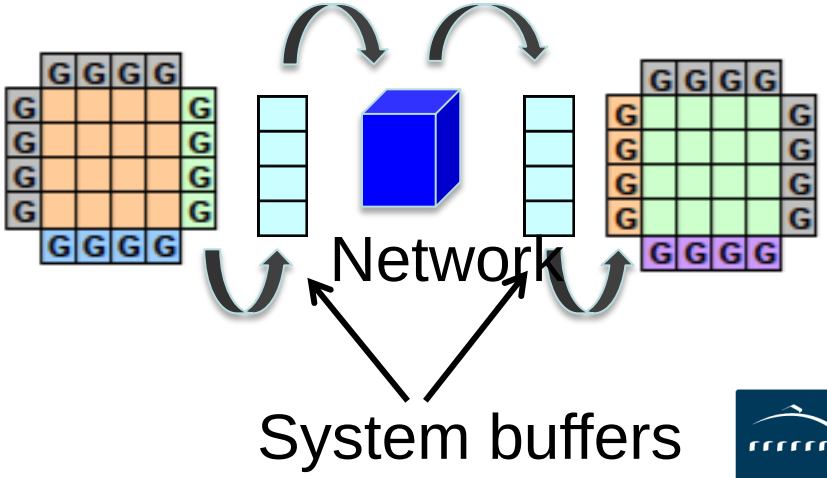
K.Yelick, CS267 UCB

Why are MPI-only Applications Memory Inefficient?

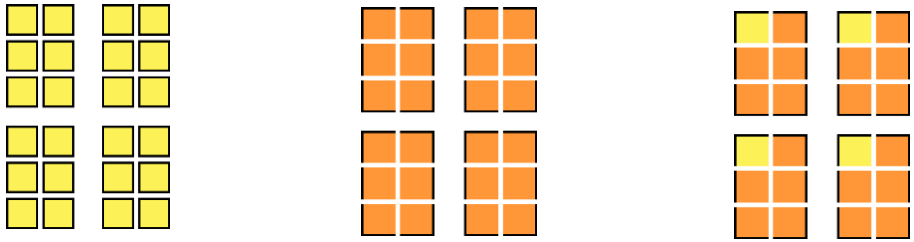
- **MPI codes consist of *n* copies of the program**
- **MPI codes require *application-level* memory for messages**
  - **Often called "ghost" cells**



- **MPI codes require *system-level* memory for messages**
  - **Assuming the very common synchronous/blocking style**



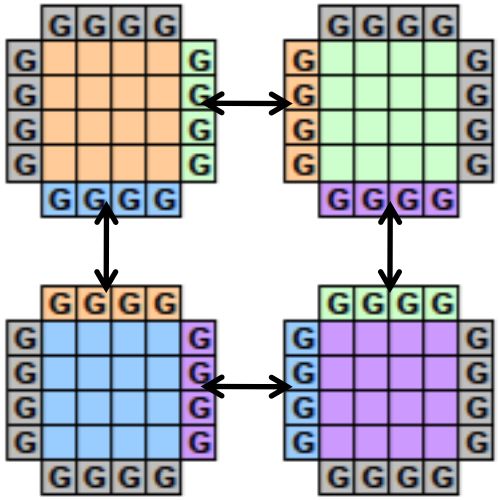
Why Does Hybrid/OpenMP Help?



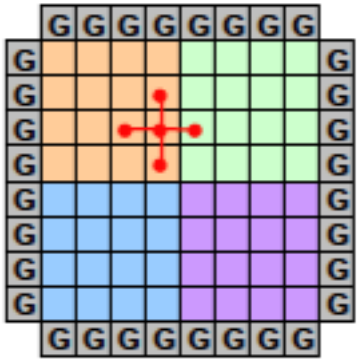
"Pure" MPI

"Pure" OpenMP

Hybrid: 4 MPI tasks, 6 threads per MPI



Distributed memory subgrid distribution

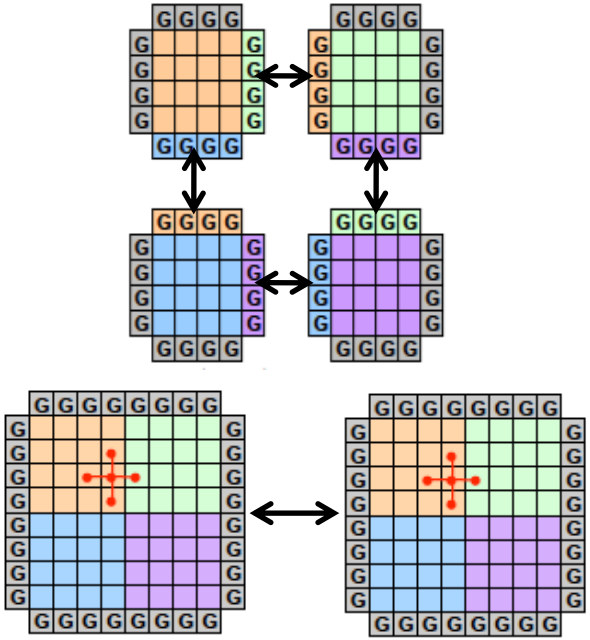


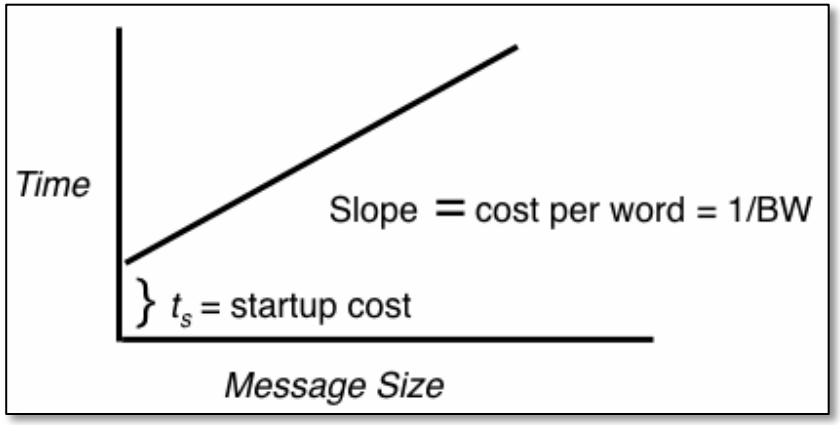
Shared memory subgrid distribution

- **Reduced Memory Usage:**
  - **Fewer instances of your program on the node**
  - **Eliminate some ghost cell memory**

Figures from Kaushik Datta, Ph.D. Dissertation, UC Berkeley, 2009

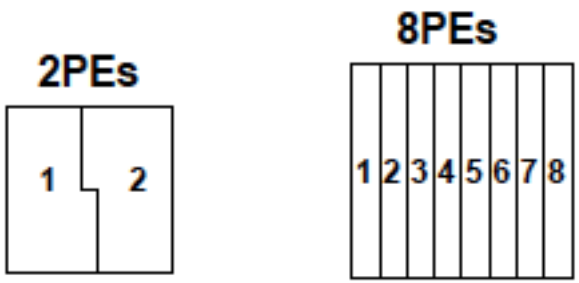
Why Does Hybrid/OpenMP Help?





- **Send larger MPI messages**
  - **small messages are expensive**
- **No intra-node messages**

Why Does Hybrid/OpenMP Help?



- There may be scalability limits to domain decomposition
- OpenMP adds fine granularity (larger message sizes) and allows flexibility of dynamic load balancing.
- Some problems have two levels of parallelism

What are the Benefits of OpenMP?

- **Uses less memory per node**
- **Typically, at least equal performance**
- **Additional parallelization may fit algorithm well**
  - **especially for applications with limited domain parallelism**
- **Possible improved MPI performance and load balancing**
  - **Avoid MPI within node**
- **OpenMP is a standard so code is portable**
- **Some OpenMP code can be added incrementally**
  - **Can focus on performance-critical portions of code**
- **Better mapping to multicore architecture**

What are the Disadvantages of OpenMP?

- **Additional programming complexity**
- **Can be difficult to debug race conditions**
- **Requires explicit synchronization**
- **Additional scalability bottlenecks:**
  - **thread creation overhead, critical sections, serial sections for MPI**
- **Cache coherence problems (false sharing) and data placement issues**
  - **Memory locality is key...**
  - **but OpenMP offers no direct control**

Are There Additional Solutions?

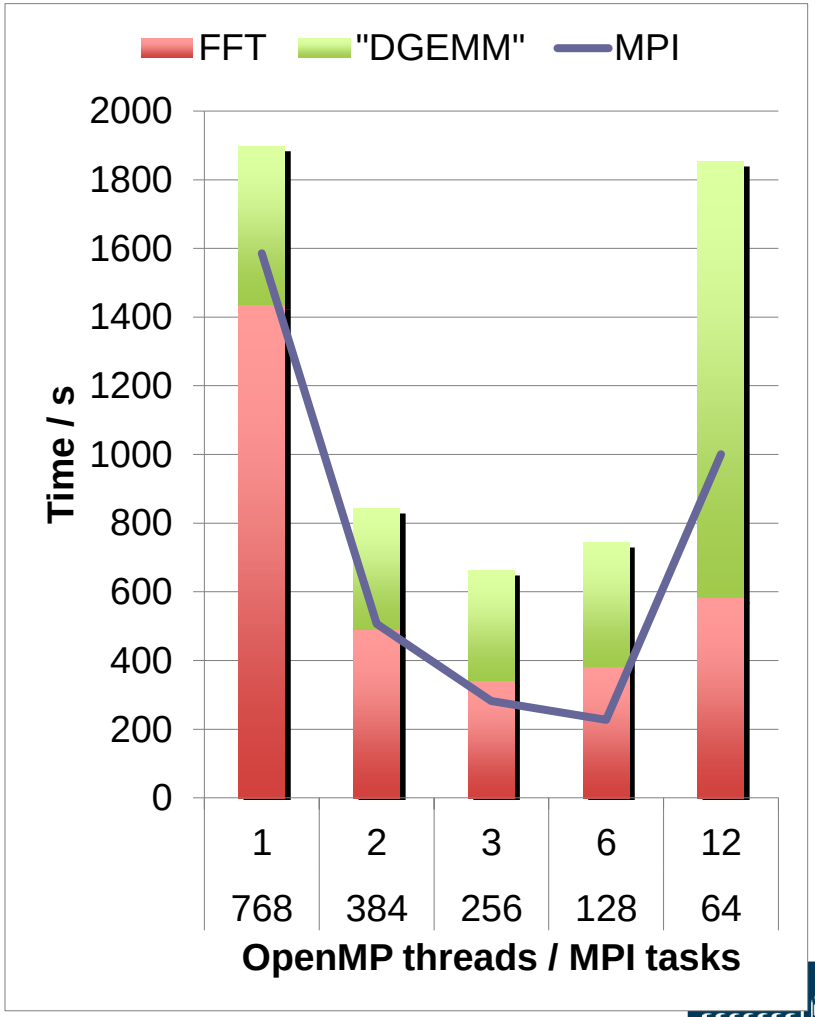
- **Sometimes it may be better to leave cores idle**
  - **Improves memory capacity and bandwidth**
  - **Improves network bandwidth**

- **However, you are charged for all cores**

Advice to NERSC Users

- OpenMP + MPI can be faster than pure MPI and is often comparable in performance
- Mixed OpenMP/MPI saves significant memory
- Beware of NUMA ! – don't use more than 6 OpenMP threads unless you know how to first-touch memory perfectly.

Paratec MPI+OpenMP Performance



DOE Explores Cloud Computing

- **In spite of NERSC and other DOE centers**
  - **Many scientists still by their own clusters**
  - **No coordinated plan for clusters in SC**
- **NERSC received funding for Magellan**
  - **$16M project at NERSC from Recovery Act**
- **Cloud questions to explore on Magellan:**
  - **Can a cloud serve DOE's mid-range computing needs?**
  - **What features (hardware and software) are needed of a "Science Cloud"?**
  - **What requirements do the jobs have?**
  - **How does this differ, if at all, from commercial clouds which serve primarily independent serial jobs?**
- **Magellan testbed installed in early 2010**

What HPC Can Learn from Clouds

- **Need to support surge computing**
  - **Predictable: monthly processing of genome data; nightly processing of telescope data**
  - **Unpredictable: computing for disaster recovery; response to facility outage**
- **Support for tailored software stack**
- **Different levels of service**
  - **Virtual private cluster: guaranteed service**
  - **Regular: low average wait time**
  - **Scavenger mode, including preemption**

NERSC-6 Hopper

Hopper provides over 3 million computing hours per day to scientists

- 1.28 PFlop/s peak performance
- Over 1 billion annual core-hours facility wide
- Gemini high performance resilient interconnect
- Two 12-core AMD Magny-Cours chips per node
- Collaboration with NNSA ACES on testing

Hopper installation, August 2010

NERSC/Cray Center of Excellence

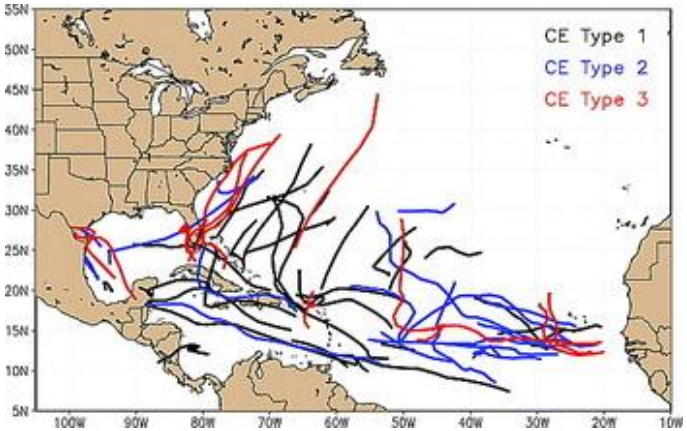
- Programming Models for Multicore systems
- Ensures effective use of new 24-core nodes

20th Center Climate Data Reconstructed

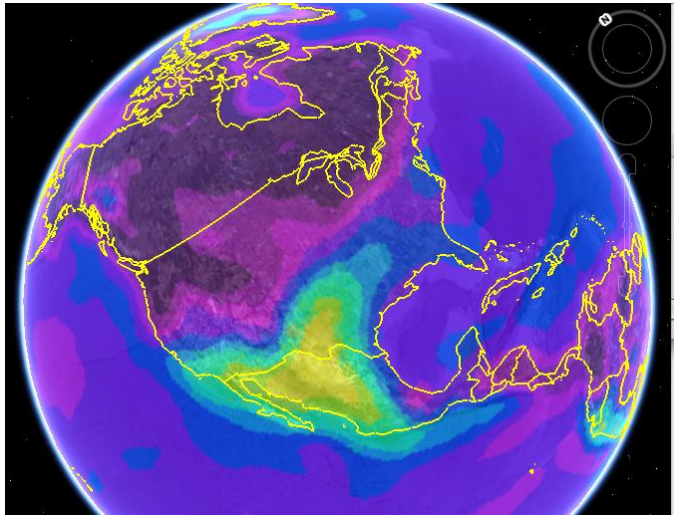
Reconstructed global weather conditions in 6-hour intervals from 1871-2010

- **Based on data from meteorologists, military, volunteers and ships' crews**
- **Over 10M hours at NERSC using reverse Kalman filter algorithms**
- **Data used in 16 papers to date: reproduced 1922 Knickerbocker storm, understand causes of the 1930 Dust Bowls, and determine whether recent extremes are sign of climate change**

NERSC has 2PB of online storage and up to 44 PB of archive for scientific data sets. New "Science Gateways" make it easy to make data accessible on the web



Previously undetected warm-core cyclones, *Geophys. Res. Letters*, 2011



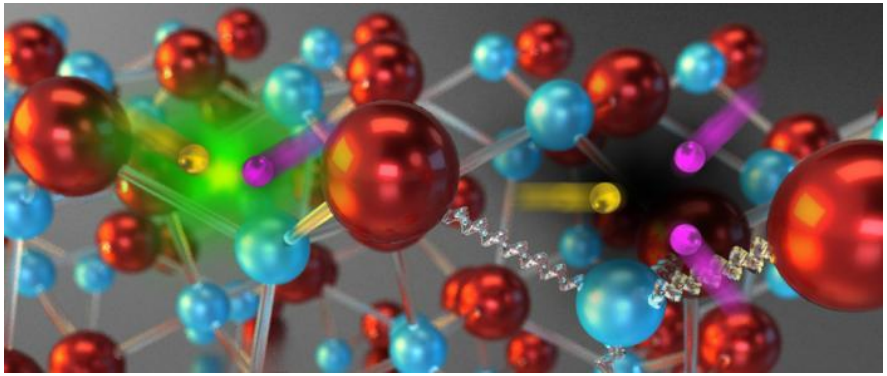
Relative Humidity for 1920-1929
Gil Compo, PI (U. Colorado)

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Material Science for Energy Efficient Lighting

- ***LEDs are up to 3x more energy efficient than fluorescent lights and last 10x longer***
  - ***"LED droop" makes them unusable for lighting rooms, since efficiency drops when current is scaled***
  - ***Cause? Auger recombination combined with carrier scattering.***
- **Science discovery explains cause of droop, allowing university and industry researchers to work on solutions.**



The illustration shows nitride-based LEDs. At left, an electron and electron hole recombine and release light. In Auger recombination (right) the electron and hole combine with a third carrier, releasing no photon. The energy loss is also assisted by indirect processes, vibrations in the crystal lattice shown as squiggles.