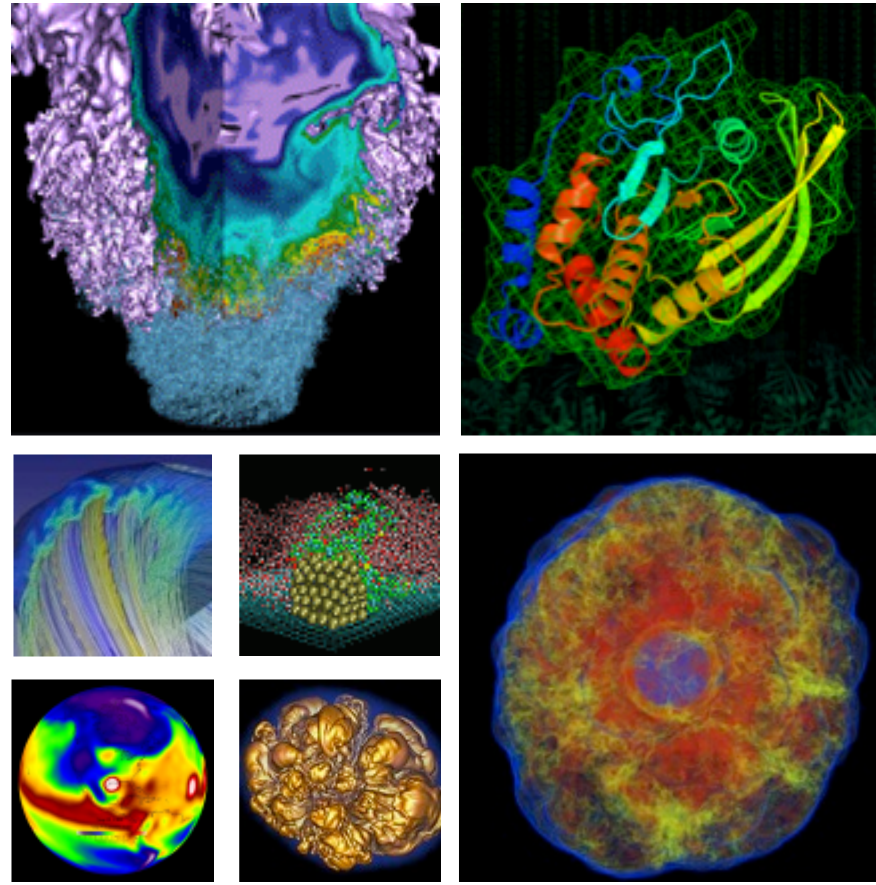
Preparing the Broad DOE Office of Science User Community for Advanced Manycore Architectures



Katie Antypas

NERSC-8 Project Leader

NERSC Deputy for Data Science

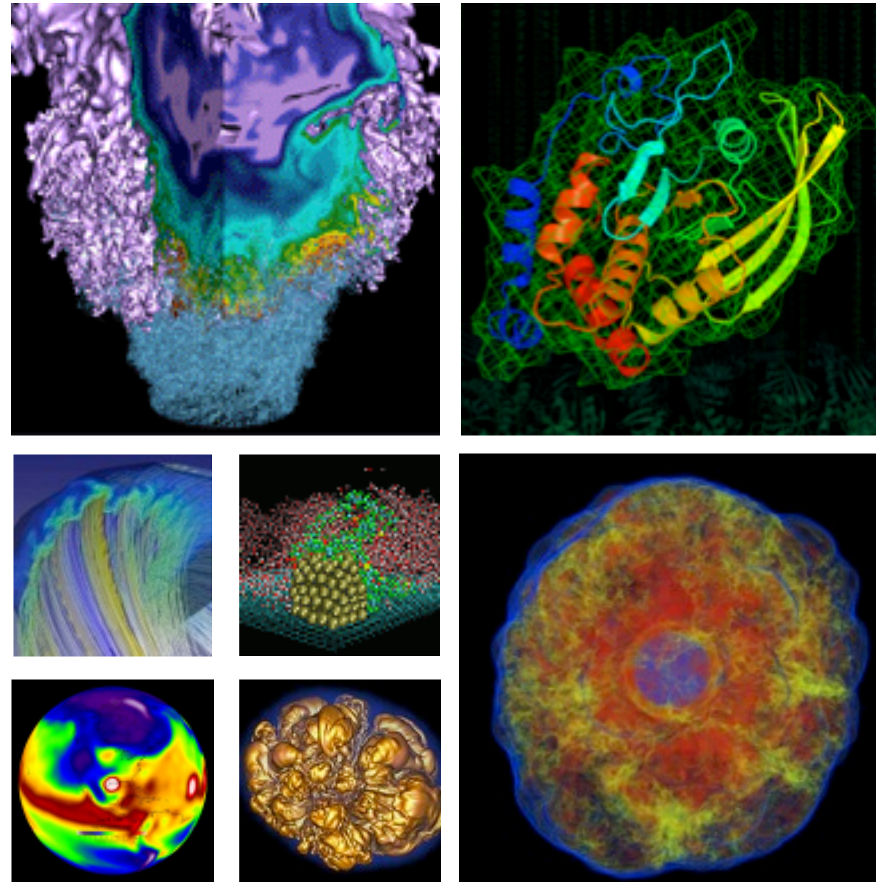
March 15, 2015

NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Preparing the Broad DOE Office of Science User Community for Advanced Manycore Architectures – and some implications for the interconnect



NERSC 40 YEARS at the FOREFRONT 1974-2014

Katie Antypas
NERSC-8 Project Leader
NERSC Deputy for Data Science

March 15, 2015

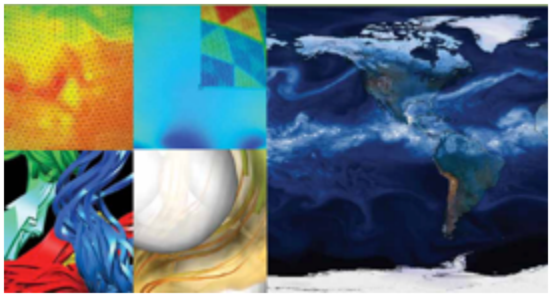
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

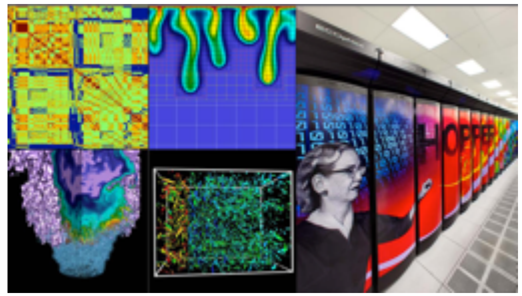
NERSC is the Production HPC & Data Facility for DOE Office of Science Research

U.S. DEPARTMENT OF ENERGY | Office of Science

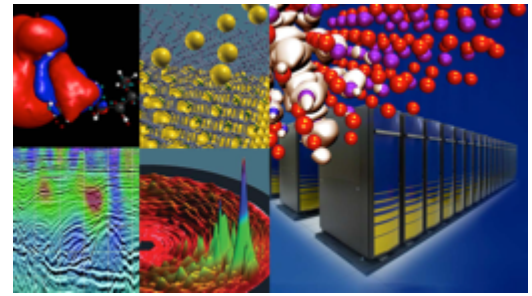
Largest funder of physical science research in U.S.



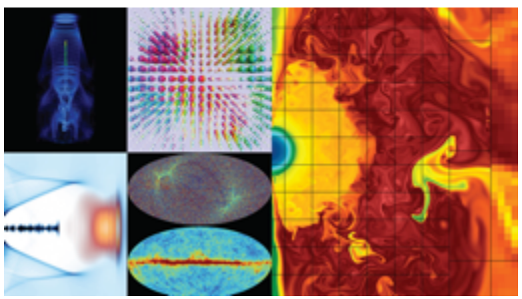
Bio Energy, Environment



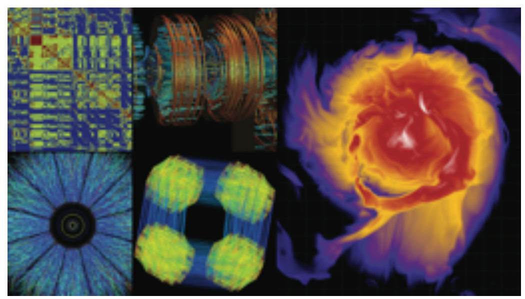
Computing



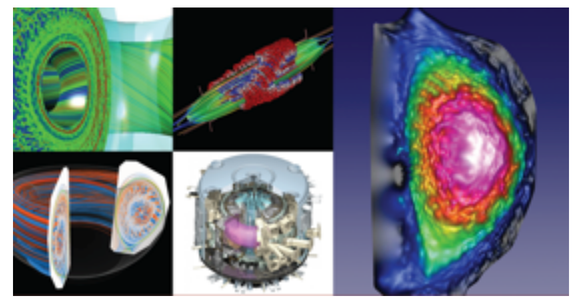
Materials, Chemistry, Geophysics



Particle Physics, Astrophysics

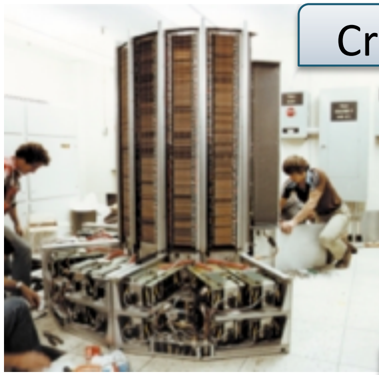


Nuclear Physics



Fusion Energy, Plasma Physics

NERSC's 40th Anniversary!



Cray 1 - 1978

Cray 2 – 1985

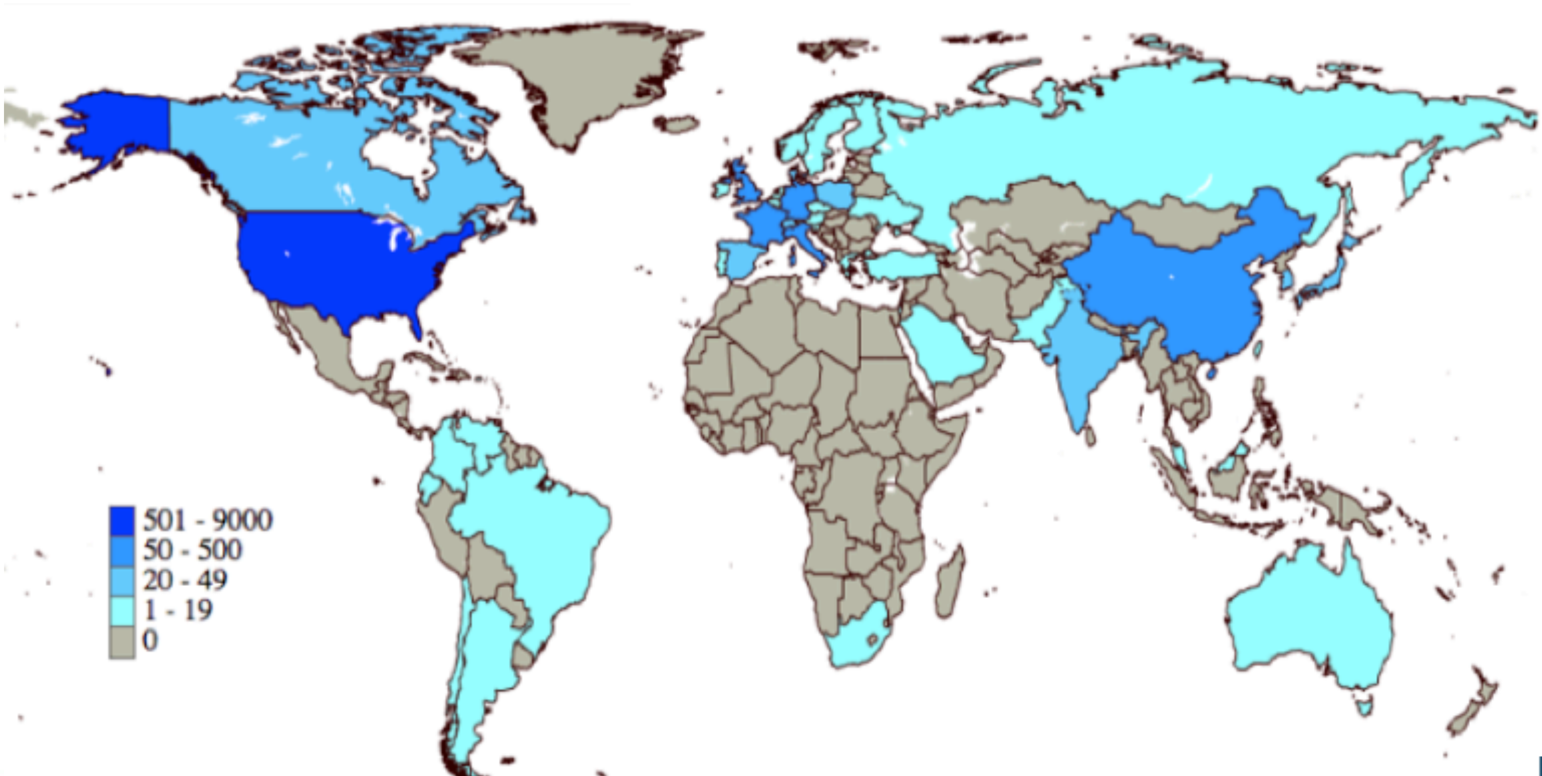
Cray T3E Mcurie - 1996

IBM Power3 Seaborg - 2001

1974	Founded at Livermore to support fusion research with a CDC system
1978	Cray 1 installed
1983	Expanded to support today's DOE Office of Science
1986	ESnet established at NERSC
1994 - 2000	Transitioned users from vector processing to MPP
1996	Moved to Berkeley Lab
1996	PDSF data intensive computing system for nuclear and high energy physics
1999	HPSS becomes mass storage platform
2006	Facility wide filesystem
2010	Collaboration with JGI
2013	Petascale Cray HPCS system

We support a broad user base

- ~6000 users, and we typically add 300-500 per year
- Geographically distributed: 48 states as well as multinational projects



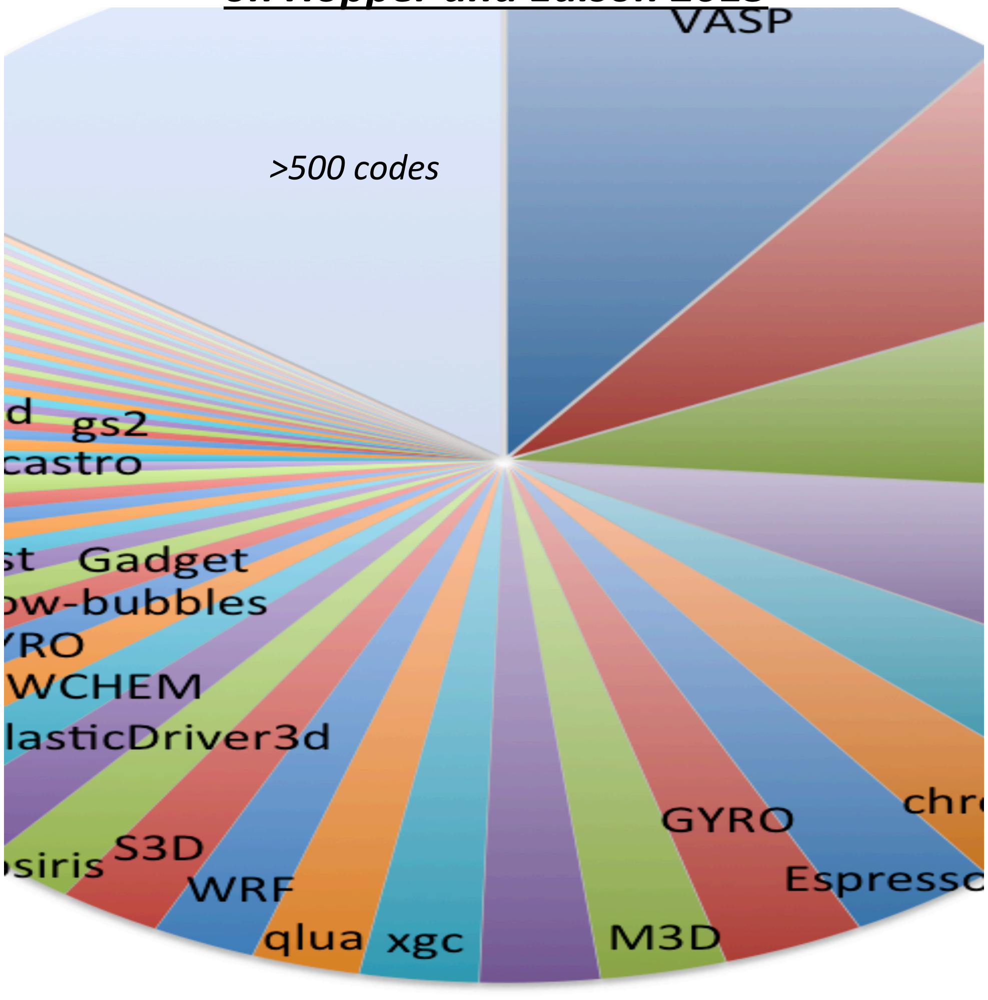
NERSC has two major systems on the floor currently

- **Hopper (NERSC-6)**
  - Along with Cielo (ACES) was the first Cray petascale systems with a Gemini interconnect

- **Edison (NERSC-7)**
  - First Cray petascale system with Intel processors, Aries interconnect and Dragonfly topology (serial #1)

NERSC's workload is highly concentrated and unequally distributed

Breakdown of Application Hours on Hopper and Edison 2013



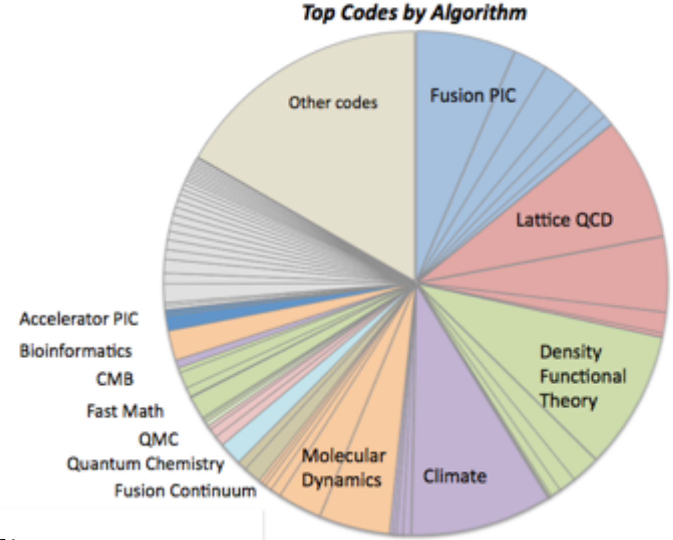
- **10 codes make up 50% of the workload**
- **25 codes make up 66% of the workload**

U.S. DEPARTMENT OF ENERGY | Office of Science

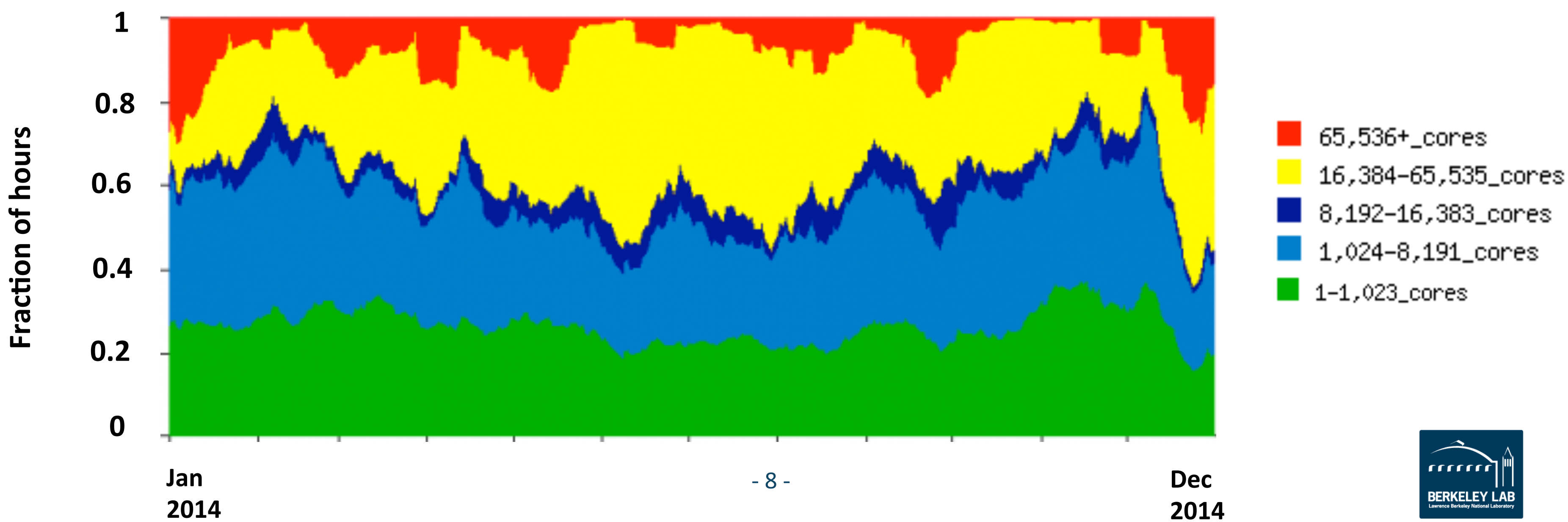
BERKELEY LAB

We support a diverse workload

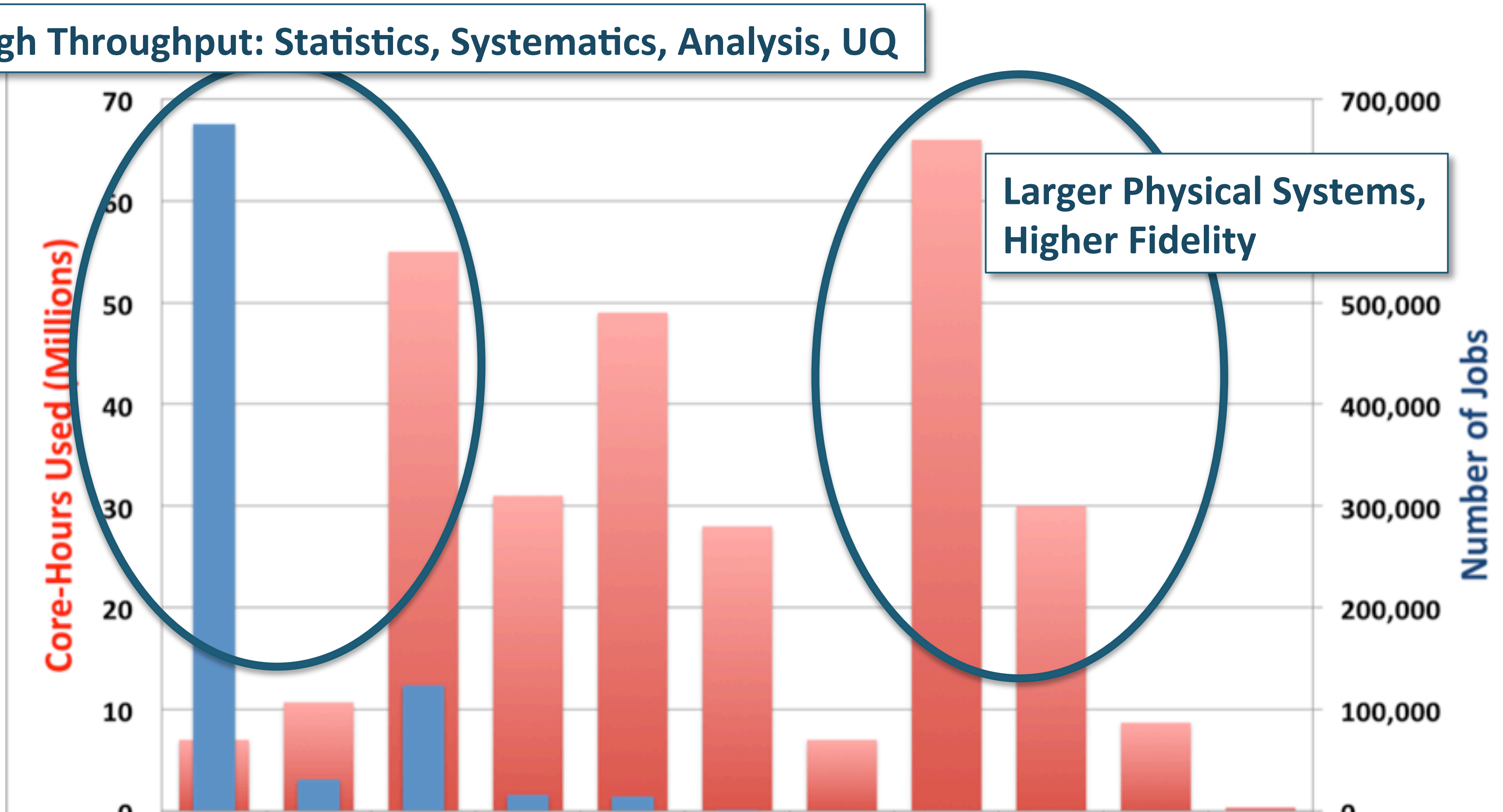
- **Many codes (700+) and algorithms**
- **Computing at scale and at high volume**



2014 Job Size Breakdown on Edison

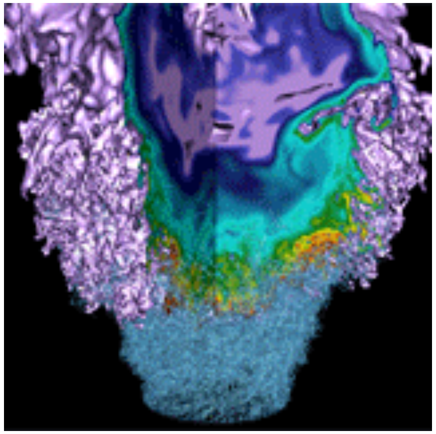


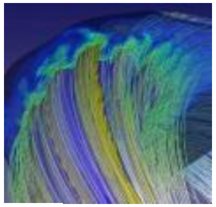
NERSC Supports Science Needs at Many Difference Scales and Sizes

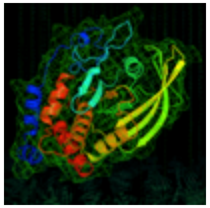


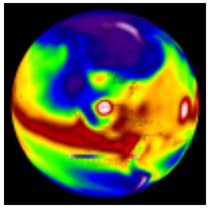
	1-31	32-63	64-511	512-1K	1K-4K	4K-8K	8K-16K	16K-32K	32K-64K	65-128K	128K+
Hours	7	10.7	55	31	49	28	7	66	30	8.7	0.382
Number	675,233	30,991	123,776	15,672	14,618	1,604	826	667	202	56	15

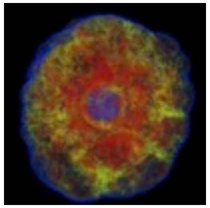
Interconnect experiences in a production computing environment

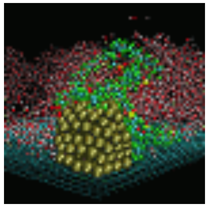












NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

What were my expectations of the interconnect as an application developer?

- If MPI communication time was < 20%, I didn't really worry about it
- Interconnect should be fast
- Performance should be consistent (consistently fast)

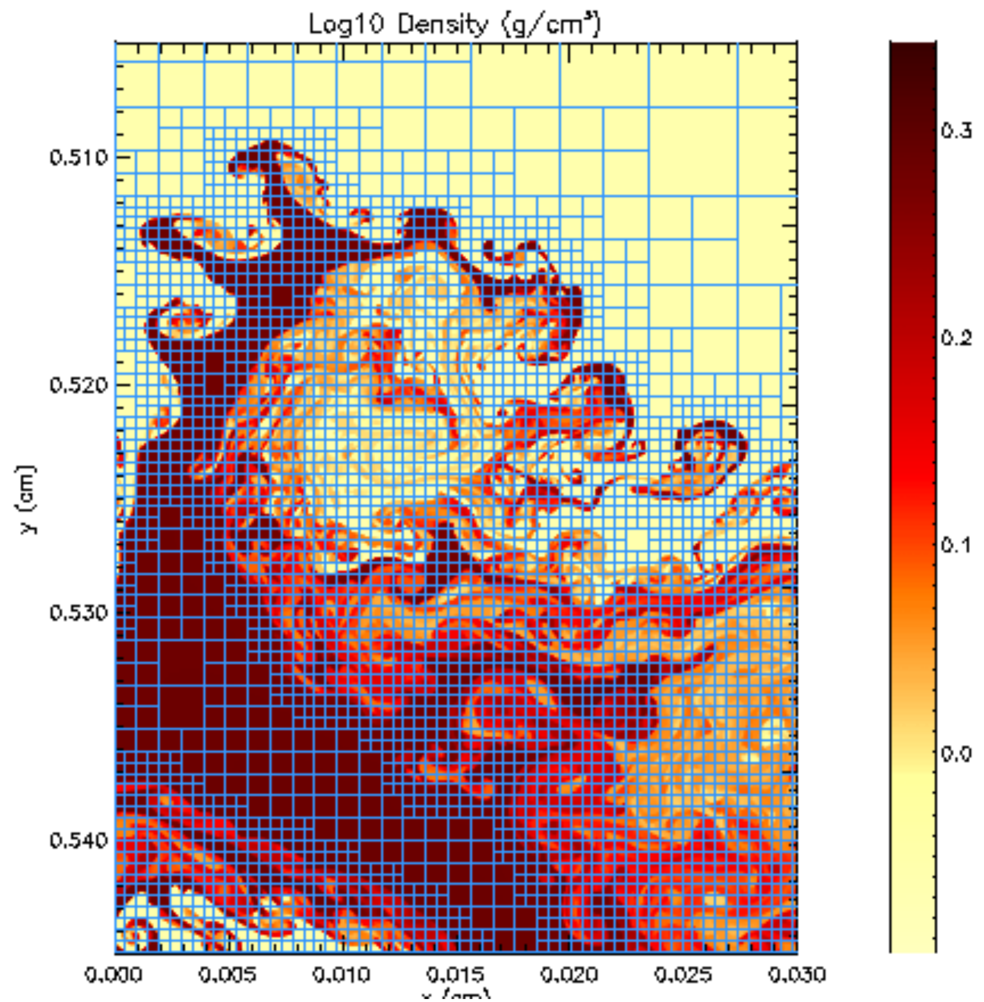


Image from Flash Center

Hopper Supercomputer

Performance

1.2 PF Peak

1.05 PF HPL (#5)

Processor

AMD MagnyCours, 2.1 GHz, 12 core

8.4 GFLOPs/core

32-64 GB DDR3-1333 per node

> 6300 total nodes

Interconnect

Gemini Interconnect (3D torus)

Adaptive routing

Link bandwidth 9.3 GB/sec

MPI bandwidth 2.9-5.8 GB/sec

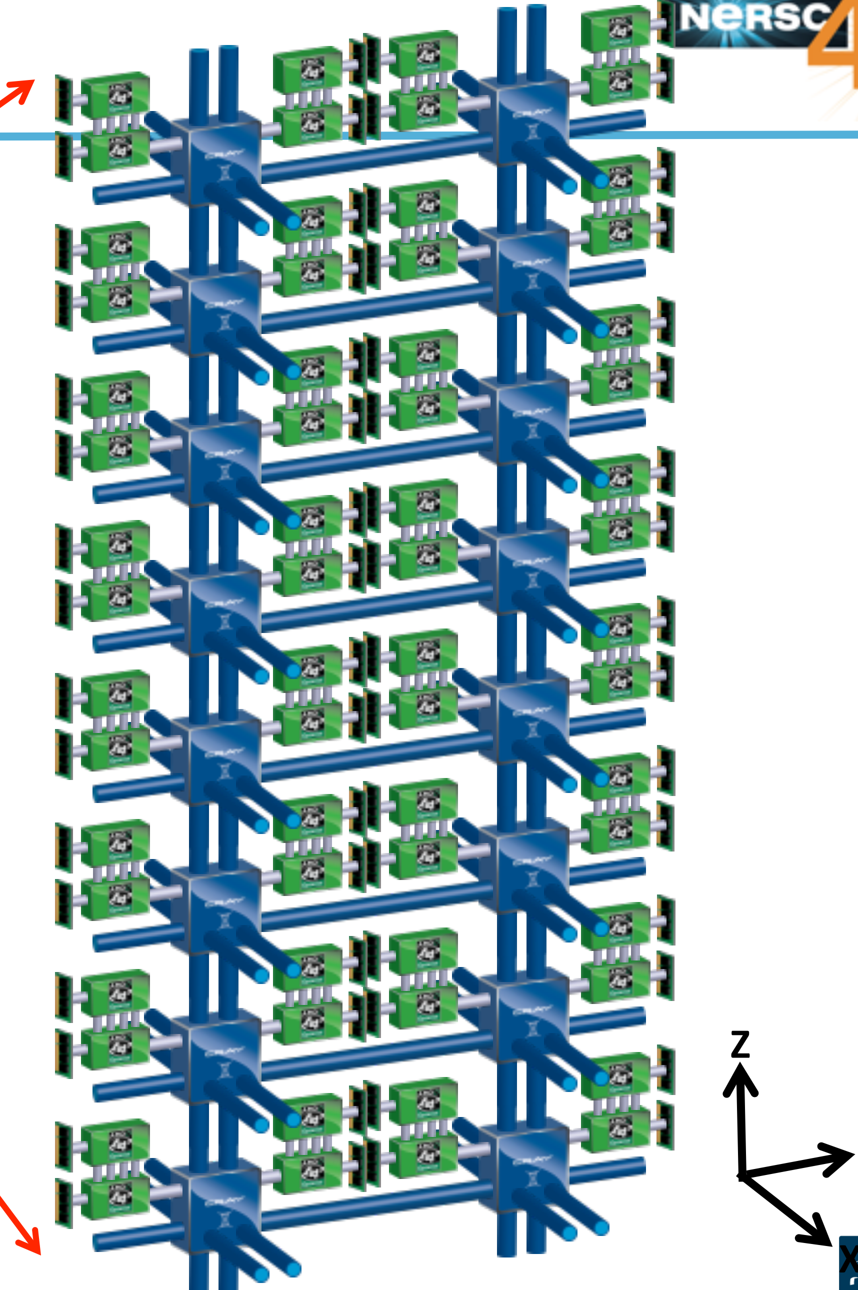
Injection bandwidth 20GB/s/node

I/O

2PB disk space

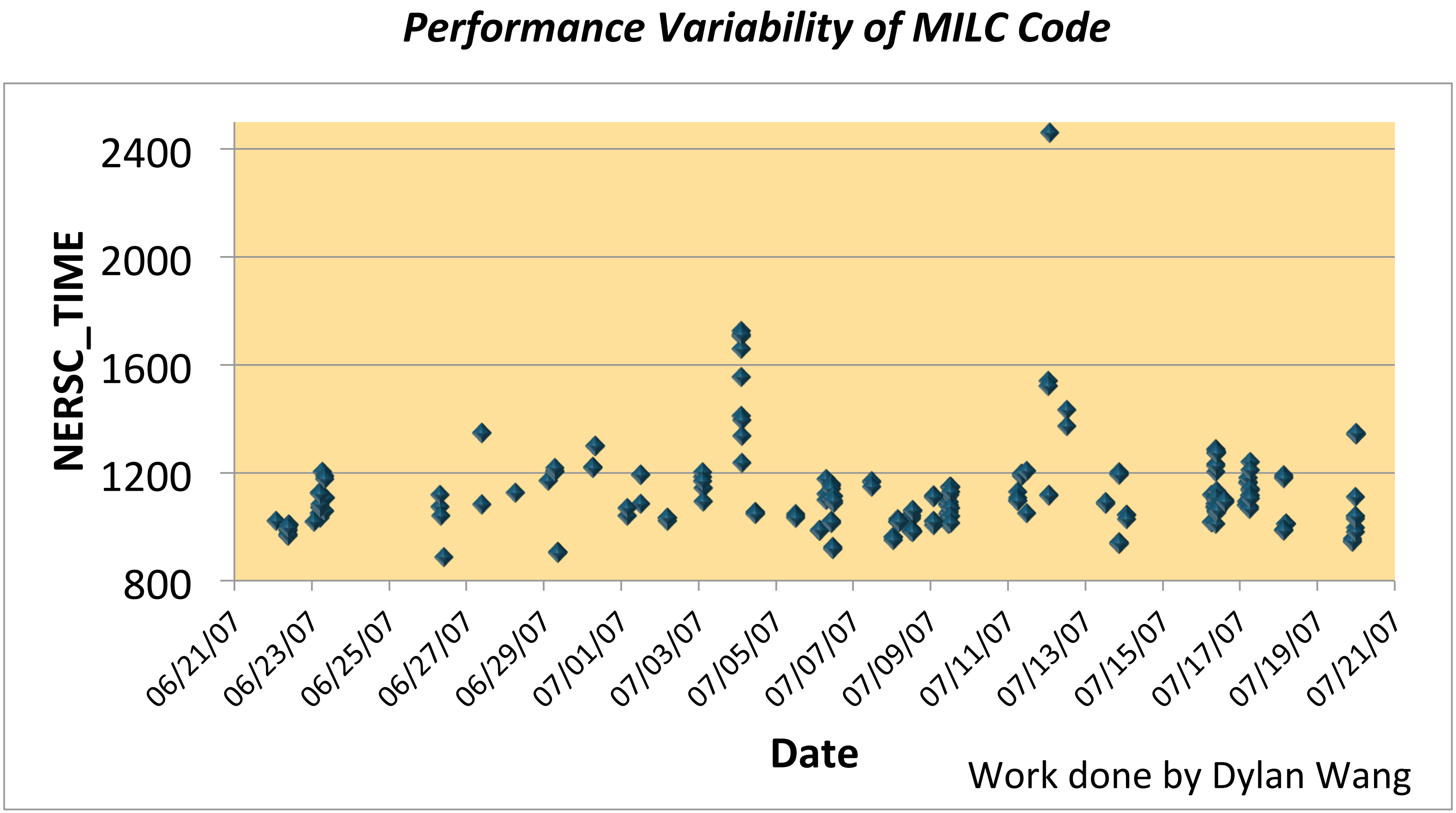
70GB/s peak I/O Bandwidth

Interconnect topology



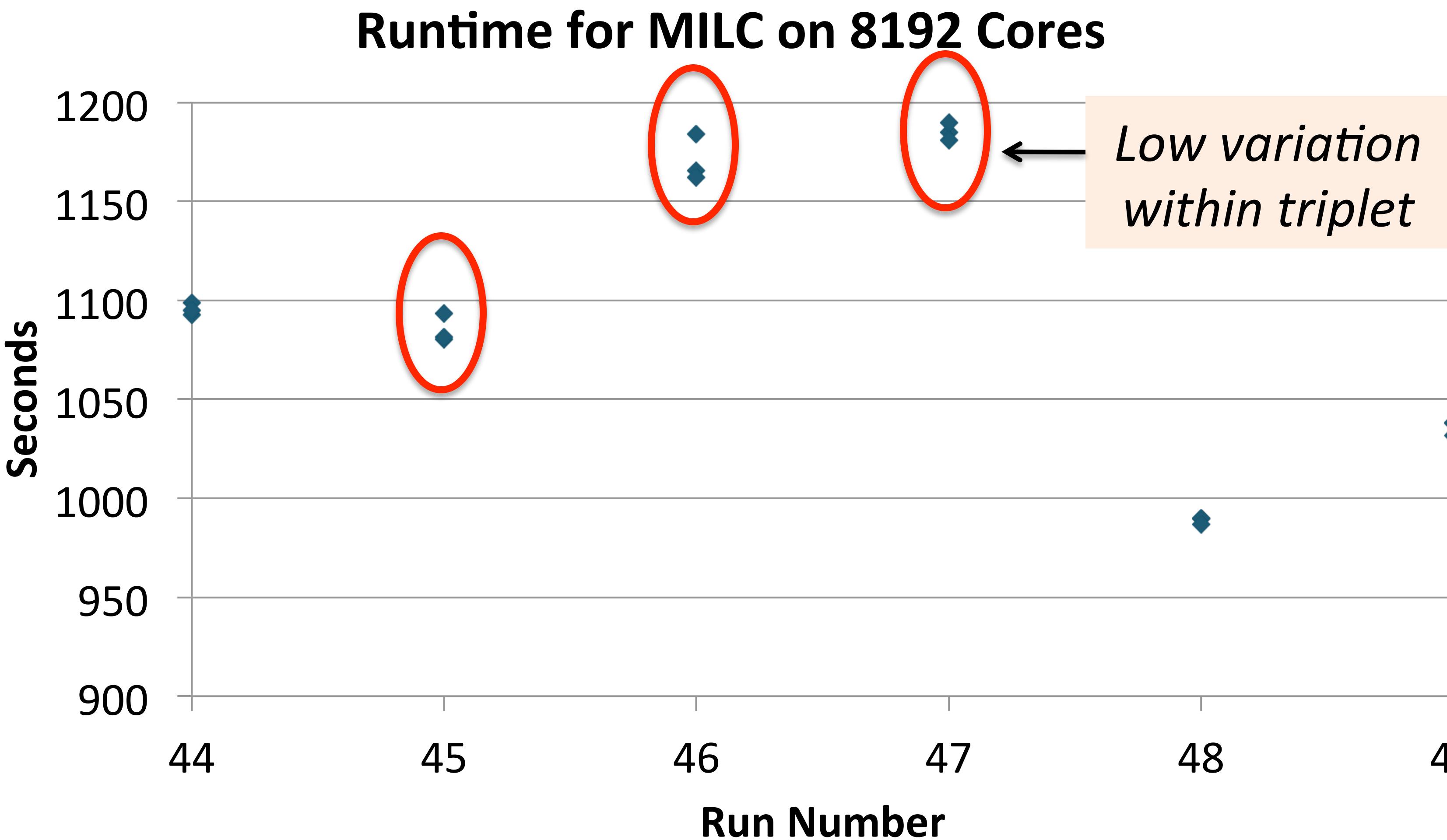
Images Courtesy of Cray Inc.

In production variability averaged 10-15% between runs, often with significant outliers



Runtime varies from ~900 seconds to 1700 seconds with one far outlier

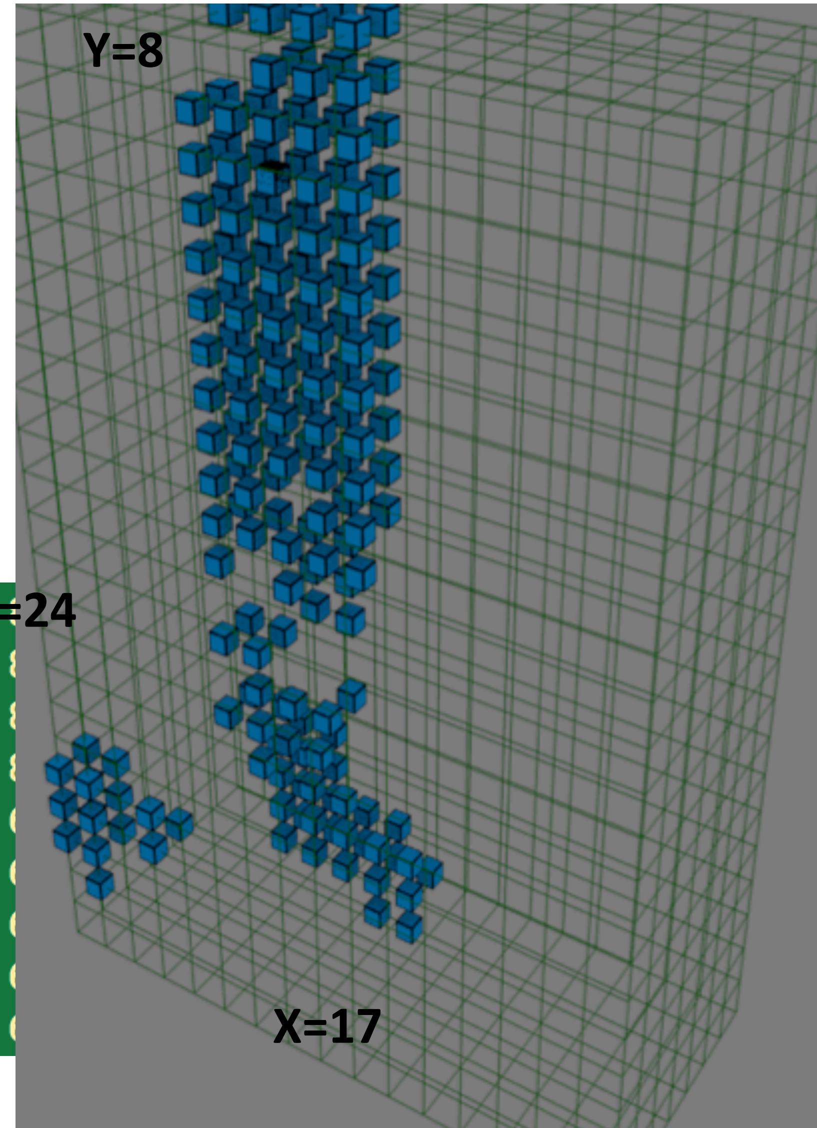
A Closer Look



Variability with a triplet (a fixed set of nodes) is low, only 3%
– so how does an allocation of nodes affect variability?

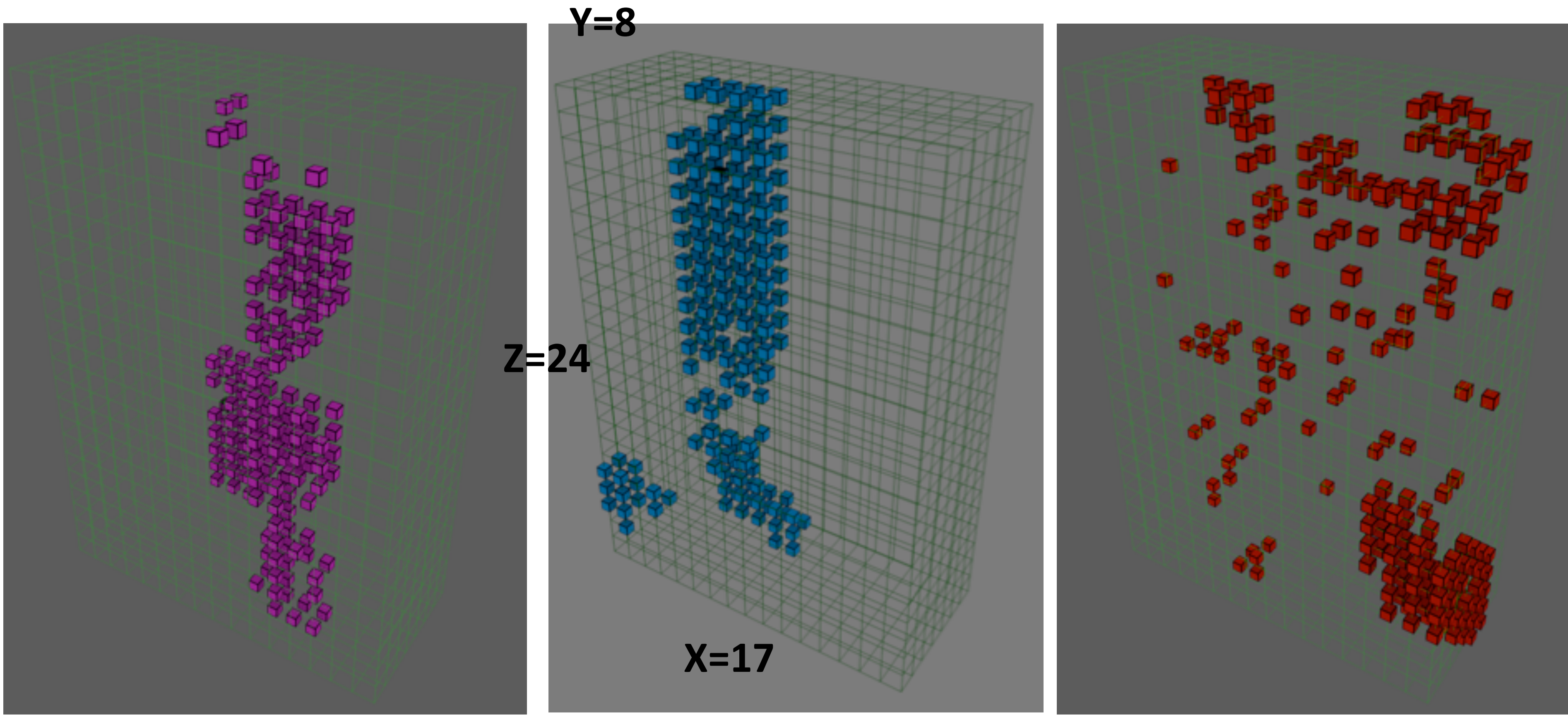
Determine coordinates on the Torus

- Run program to determine node location on torus before MILC application
- Hopper network is 17x8x24




```
mpirank 20 with host nid00089 node_id                3
mpirank 21 with host nid00089 node_id                3
mpirank 22 with host nid00089 node_id                3
mpirank 23 with host nid00089 node_id                3
mpirank 24 with host nid00006 node_id
mpirank 25 with host nid00006 node_id
mpirank 26 with host nid00006 node_id
mpirank 27 with host nid00006 node_id
mpirank 28 with host nid00006 node_id
```

Node placement of a fast, average and slow run



Fast run: 940 seconds

Average run: 1100 seconds

Slow run: 2462 seconds

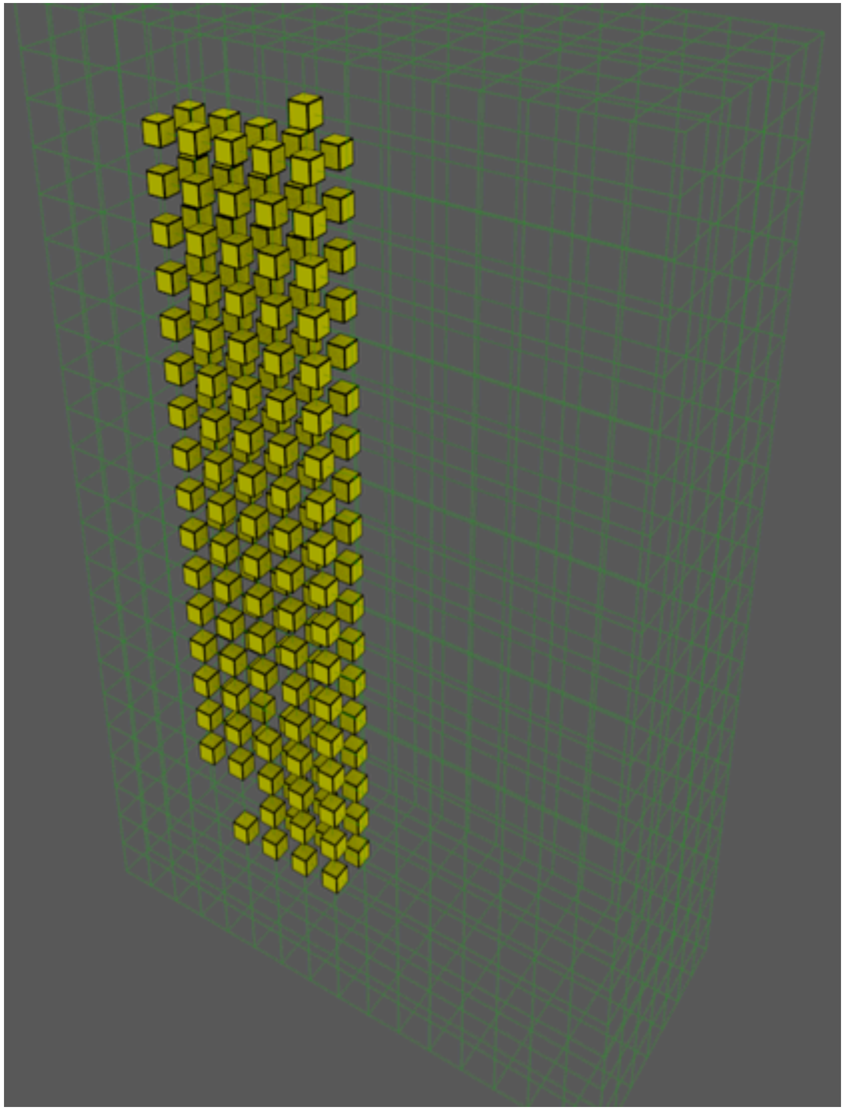
Fastest Run

Fastest run: 877 seconds

Fastest run shows a tightly packed set of nodes

Our eyes can clearly see differences in node allocation, question is what heuristics we can create for general application workload

Work by Dylan Wang



Summary

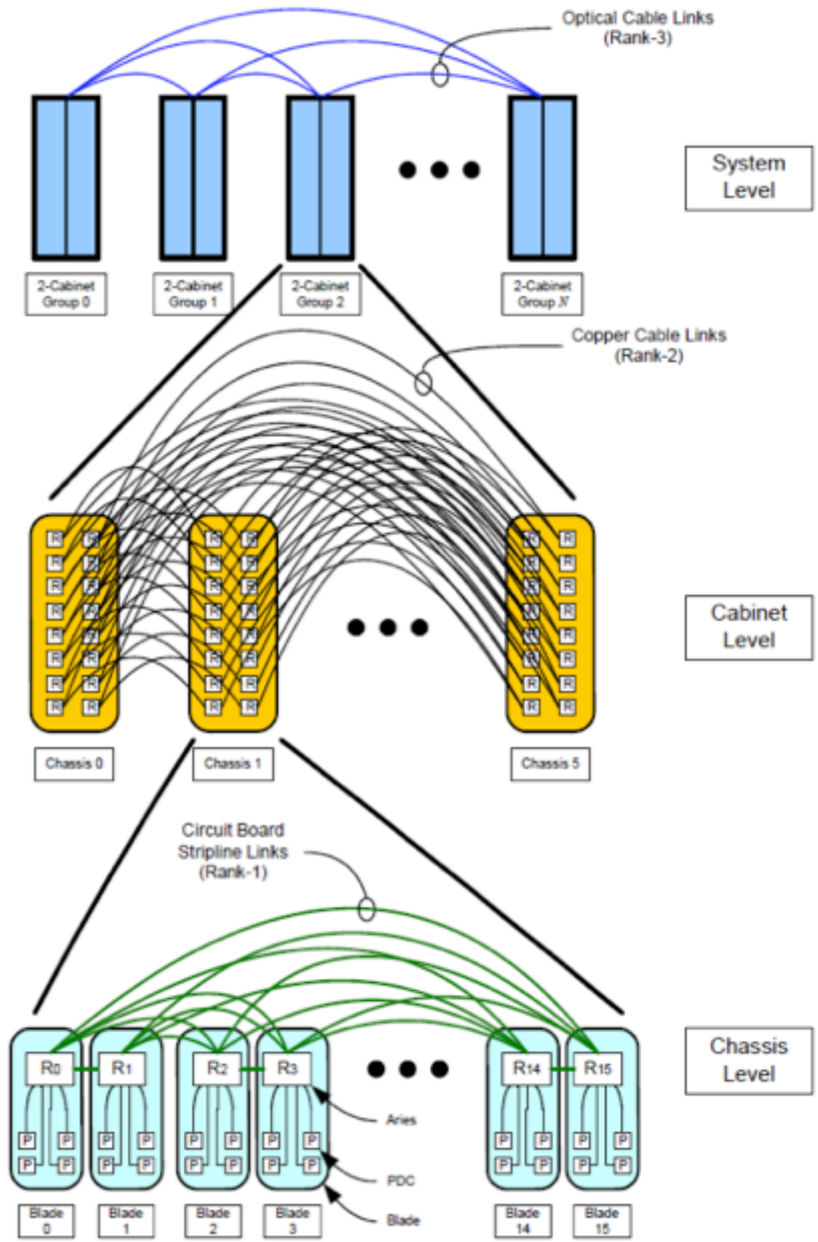
- **At least some of the variability appears to be from variation in application placement on nodes**
- **Next steps**
  – Share information with Cray and look into placement aware scheduling. Could some applications choose to spend a longer time in the queue in order to get tighter node allocation?
  – Procure a system where application runtime performance is not dependent on application placement

NERSC's latest system is Edison

- First Cray Petascale system with Intel processors, Aries interconnect and Dragonfly topology
- Very high memory bandwidth (100 GB/s per node), interconnect bandwidth and bisection bandwidth
- 64 GB/node
- Exceptional application performance

Aries Interconnect – 3 tiers

- Global: Rank3: 23.7 TB/s
- 4032 GB/s within a group (rank-1 and rank-2)
- 672 GB/s within a chassis (rank-1)

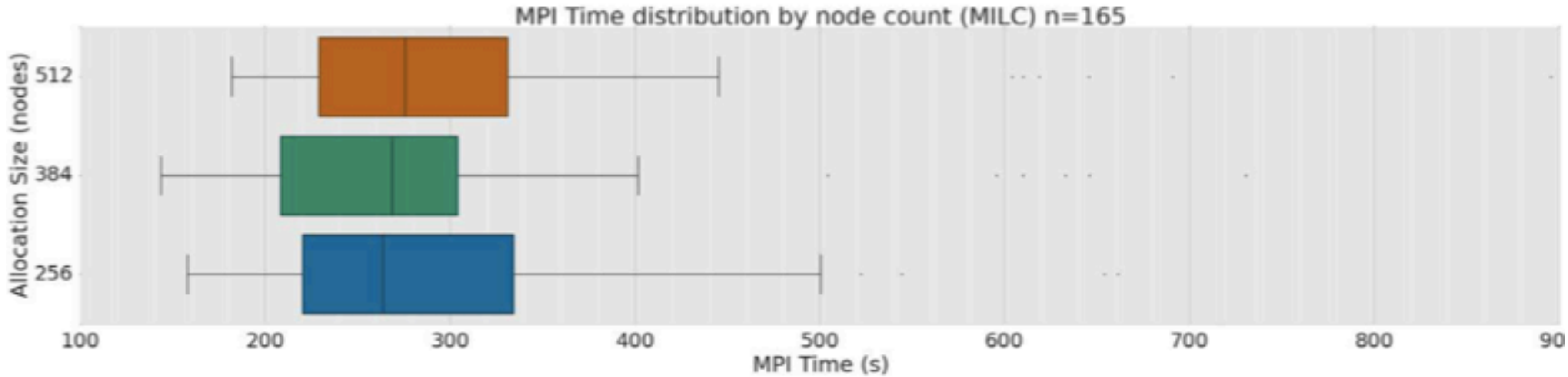


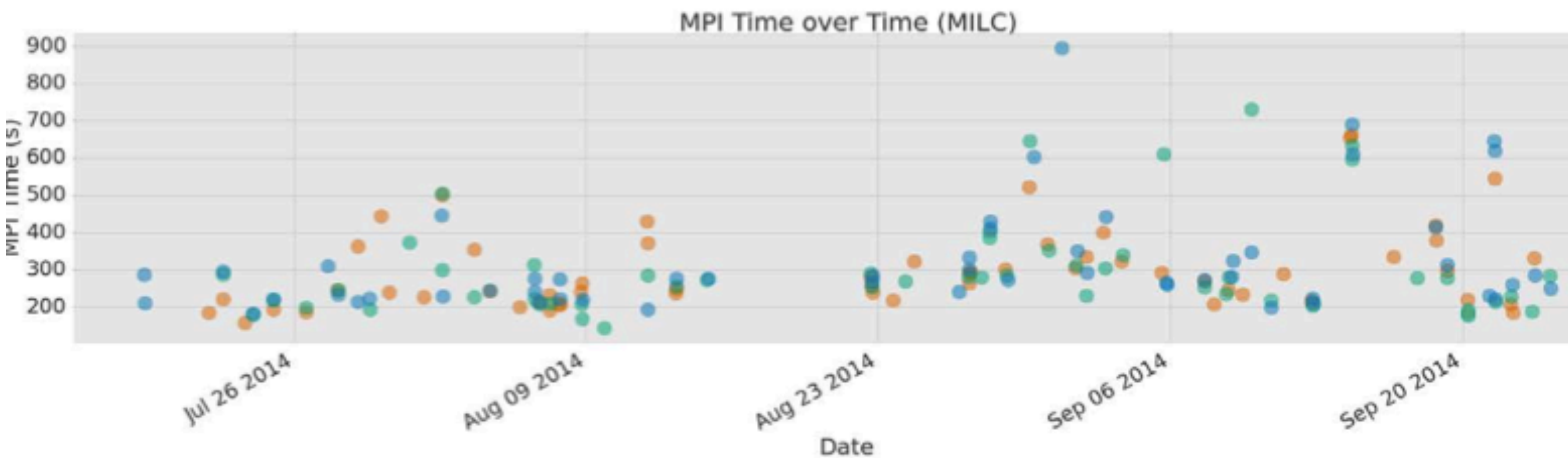
Edison system has 15 groups

A group has 6 chassis, (2 cabinets)

A chassis has 16 routers

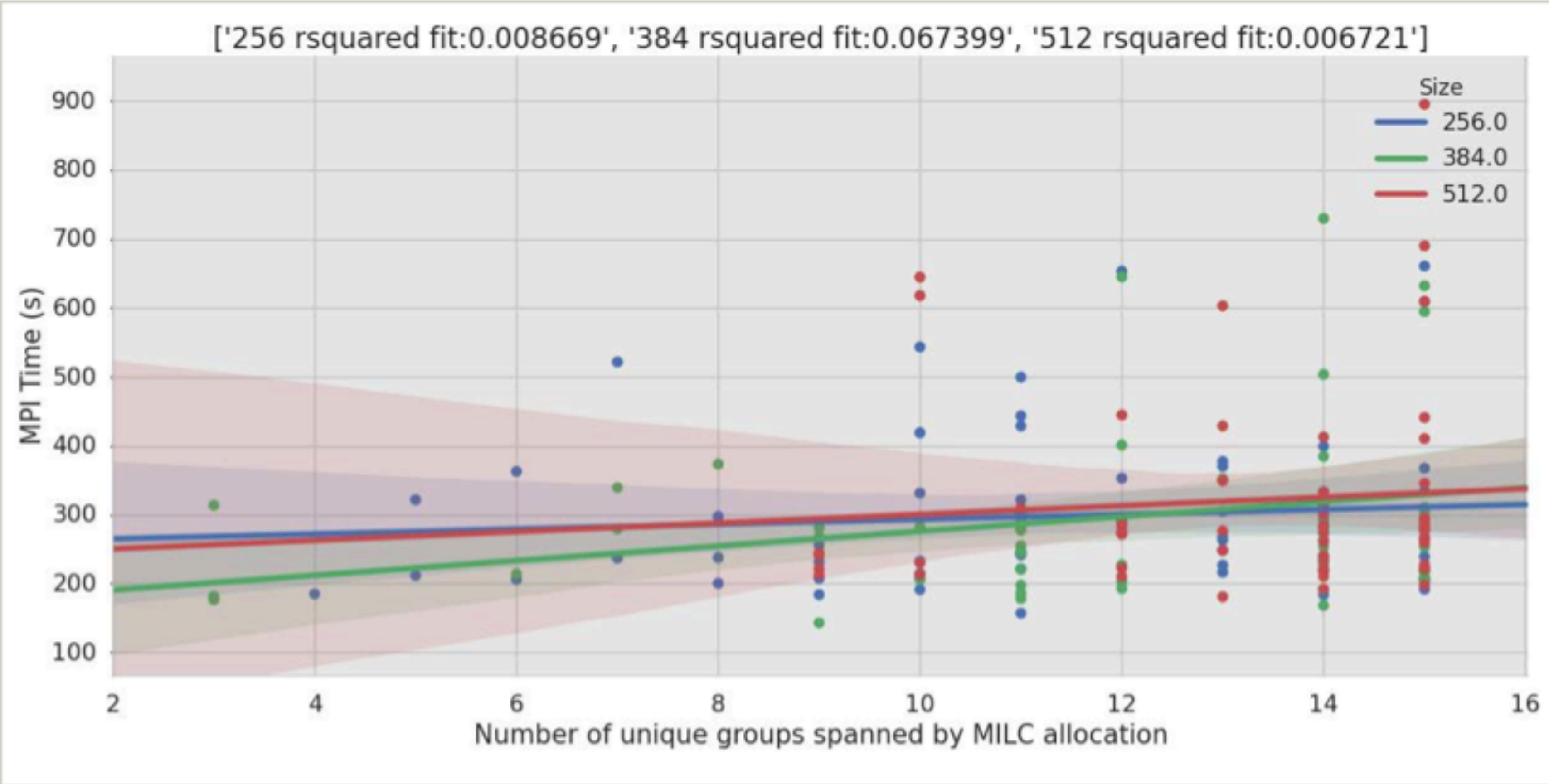
Variability of MILC Code on Edison – average of 25%, up to 300%



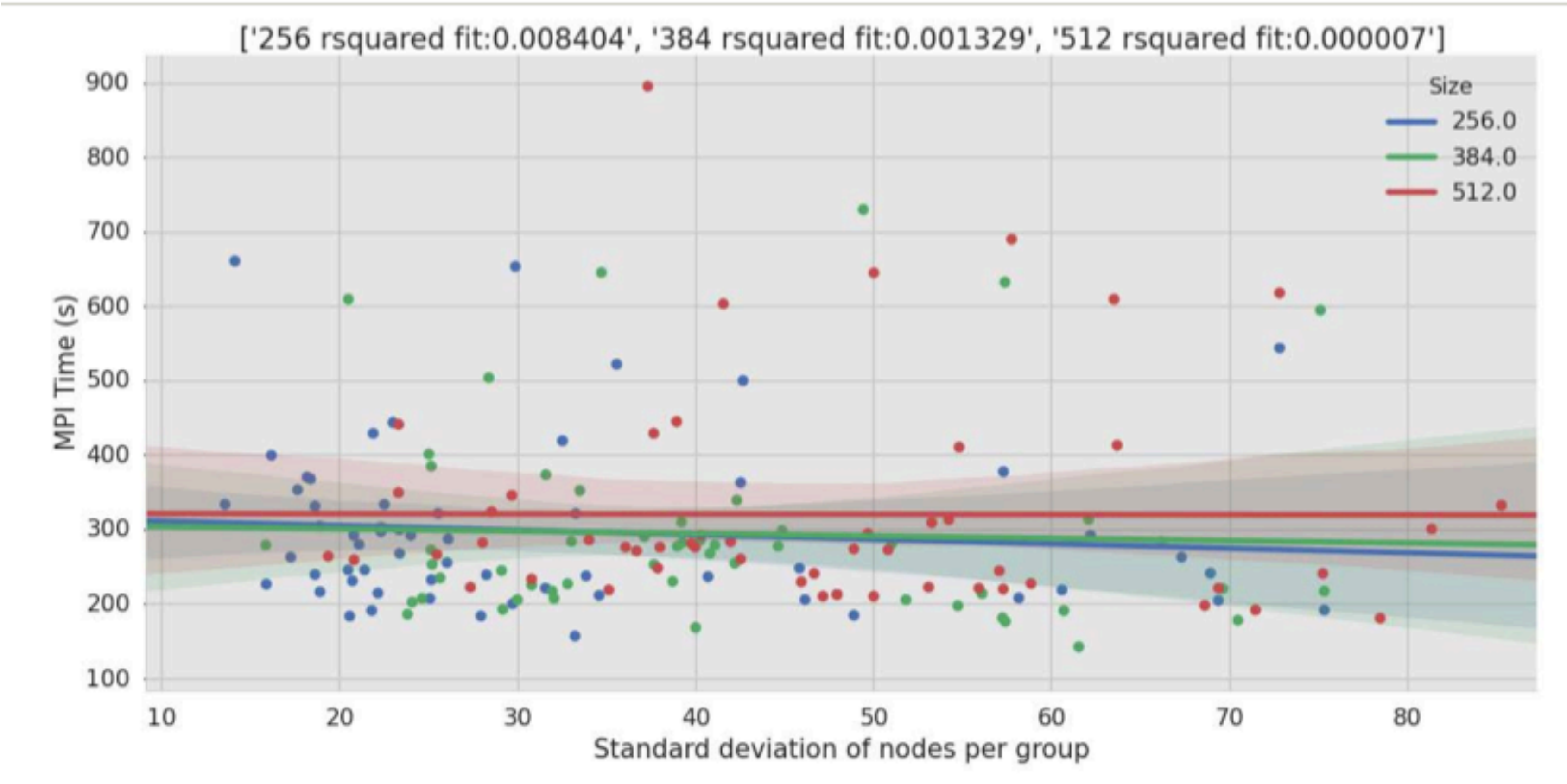


Work by Dylan Wang, Abhinav Bhatele, Dipak Ghosal at LLNL

Unique number of groups spanned



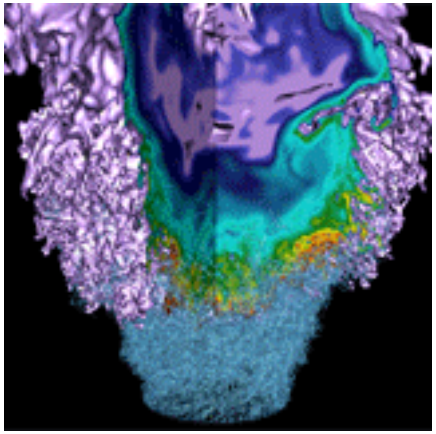
Standard deviation of number of nodes allocated to MILC per group

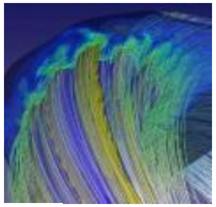


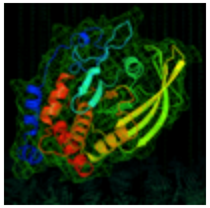
Results

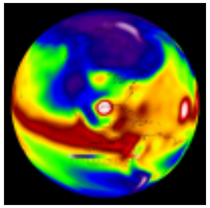
- **Node placement not correlated with variation – (What we expect from a dragon fly network)**
- **However, variability is actually higher than on Hopper**
- **Still exploring other possibilities**
  - Interference from other jobs, I/O travels on same network
  - System maintenance – warm swapping a node, throttles network
  - Bad node/memory dimm
  - Under provisioned network?

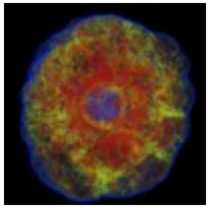
Cori – a pre-exascale system arriving in 2016

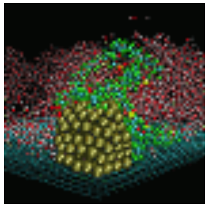












Cori: A pre-exascale supercomputer for the Office of Science workload

- **System will begin to transition the workload to more energy efficient architectures**
- **Will showcase technologies expected in exascale systems**
  - Processors with many ‘slow’ cores and longer vector units
  - Deepening memory and storage hierarchies

Image source: Wikipedia

System named after Gerty Cori, Biochemist and first American woman to receive the Nobel prize in science.

Cori Configuration – 64 cabinets of Cray XC system

- **Over 9,300 ‘Knights Landing’ compute nodes**
  - Self-hosted (not an accelerator)
  - Greater than 60 cores per node with four hardware threads each
  - High bandwidth on-package memory
- **~1,900 ‘Haswell’ compute nodes as a data partition**
- **Aries Interconnect (same as on Edison)**
- **>5x application performance of Edison system**
- **Lustre File system**
  - 28 PB capacity, >700 GB/sec I/O bandwidth
- **NVRAM “Burst Buffer” for I/O acceleration**
  - ~1.5PB capacity, > 1.5 TB/sec I/O bandwidth
- **Significant Intel and Cray application transition support**
- **Delivery in two phases, summer 2015 and summer 2016**
- **Installation in new LBNL CRT Facility**

Intel "Knights Landing" Processor

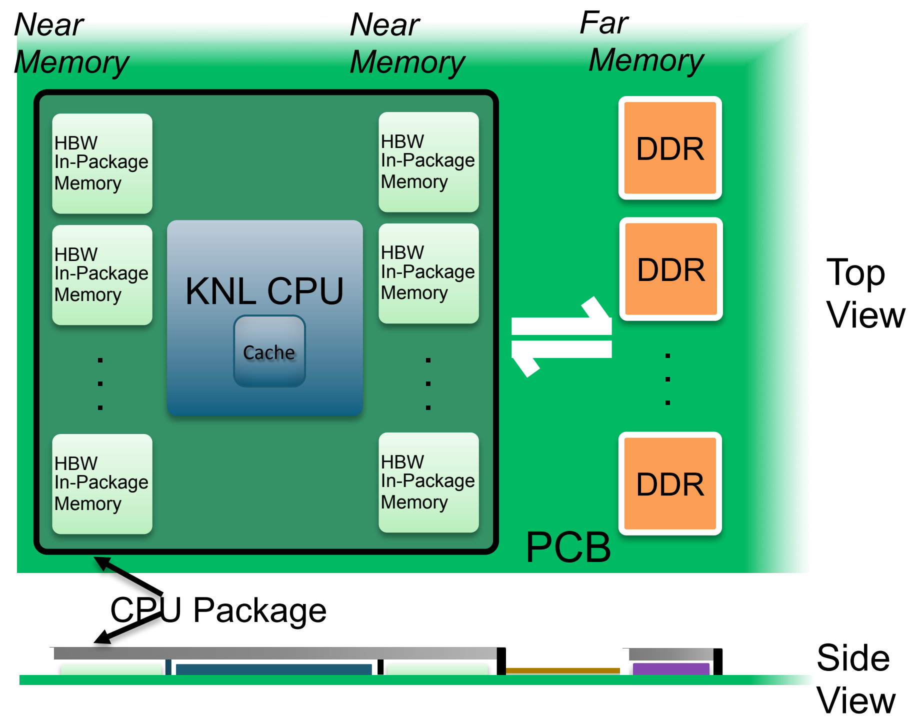
- Next generation Xeon-Phi, >3TF peak
- Single socket processor - Self-hosted, not co-processor/accelerator
- Greater than 60 cores per processor with four hardware threads each – MPI+OpenMP suggested programming model
- Intel® "Silvermont" architecture enhanced for HPC
- Cores connected via a 2D mesh network
- Multiple NUMA domains per socket
- 512b vector units (32 flops/clock – AVX 512)
- 3X single-thread performance over current generation Xeon-Phi
- High bandwidth on-package memory, up to 16GB capacity with bandwidth projected to be 5X that of DDR4 DRAM memory

Knights Landing Integrated On-Package Memory

Cache Model Let the hardware automatically manage the integrated on-package memory as an "L3" cache between KNL CPU and external DDR

Flat Model Manually manage how your application uses the integrated on-package memory and external DDR for peak performance

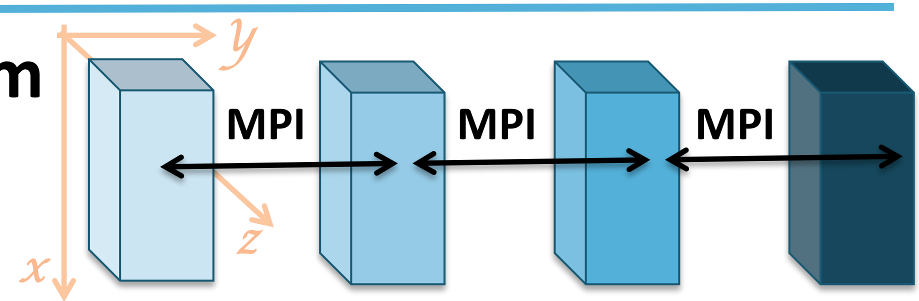
Hybrid Model Harness the benefits of both cache and flat models by segmenting the integrated on-package memory



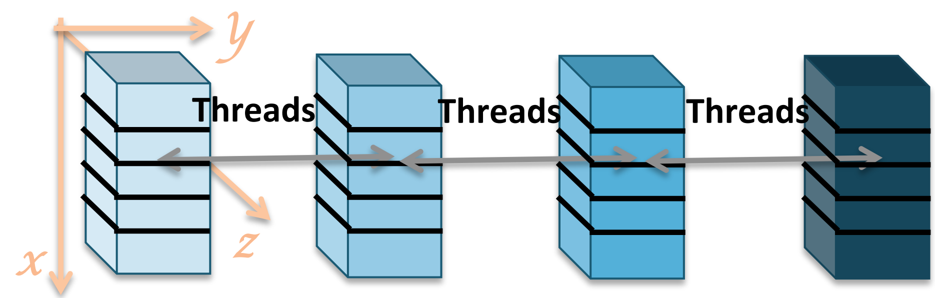
Maximum performance through higher memory bandwidth and flexibility

To run effectively on Cori users will have to:

- **Manage Domain Parallelism**
  - independent program units; explicit



- **Increase Thread Parallelism**
  - independent execution units within the program; generally explicit



- **Exploit Data Parallelism**
  - Same operation on multiple elements

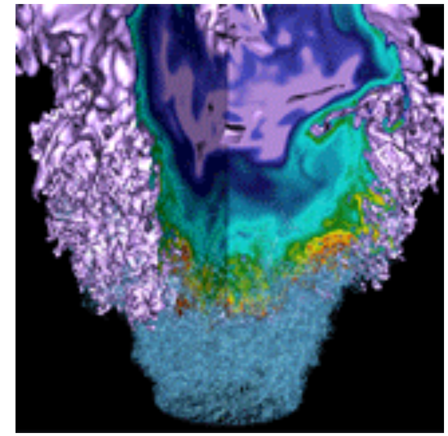
```
|--> DO I = 1, N
|        R(I) = B(I) + A(I)
|--> ENDDO
```

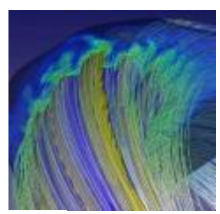
- **Improve data locality**
  - Cache blocking; Use on-package memory

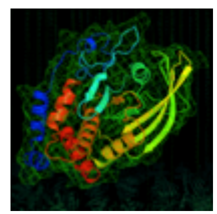
How will the KNL architecture affect network performance?

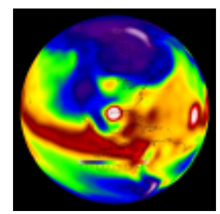
- **Significantly more on-node parallelism**
  - If MPI only programming models dominate – results in smaller domains and smaller message sizes
  - OpenMP can counter-act this
- **Slower cores -- will they be able to drive the injection rates required?**
- **How will RDMA work for a node with multiple levels of memory and how can programmers express this?**
- **Should we be pushing communication avoiding algorithms more strongly?**

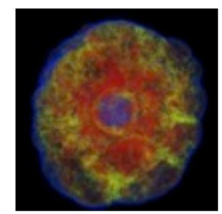
NERSC Exascale Science Application Program (NESAP)

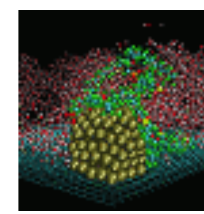












NERSC 40 YEARS at the FOREFRONT

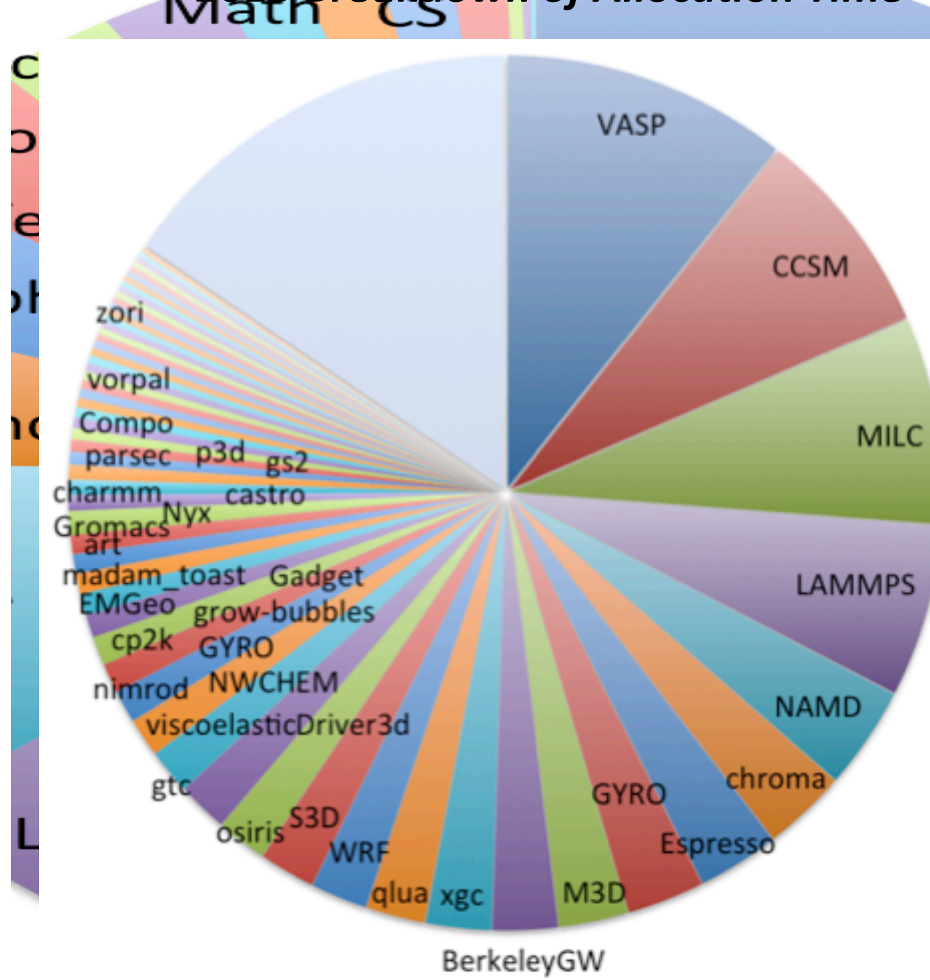
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC's Challenge

- Thousands of users
- More than 700 projects
- Hundreds of codes >600
- We don't select our users!
- Users have an insatiable demand for computing but we have a limited power budget – driving the need to move to more energy efficient computing

2013 Breakdown of Allocation Time



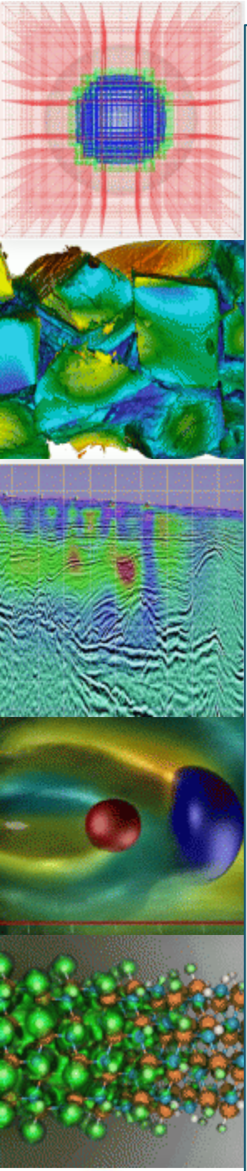
NERSC Exascale Science Application Program

- **Goal: Prepare DOE SC user community for Cori manycore architecture**
- **Partner closely with ~20 application teams and apply lessons learned to broad SC user community**
- **NESAP activities include:**
  - Strong support from vendors
  - Developer Workshops for 3rd-Party SW
  - Early engagement with code teams
  - Leverage existing community efforts
  - Postdoc Program
  - NERSC training and online modules
  - Early access to KNL technology

We solicited user proposals to be part of NESAP

- **Tier 1: 8 Application teams**
  - Each team will have an embedded post-doc
  - Access to an Intel dungeon session
  - Support from NERSC Application Readiness and Cray COE staff
  - Early access to KNL testbeds and Cori system
  - User training sessions from Intel, Cray and NERSC staff
- **Tier 2: 12 Application teams**
  - All the resources of the Tier 1 teams except for an embedded post-doc
- **Tier 3: Another 20 Application teams + library and tools teams**
  - Access to KNL testbeds, Cori system and user trainings and NDA briefings
  - Many advanced and motivated teams we were not able to accept into NESAP

NESAP Codes



Advanced Scientific Computing Research

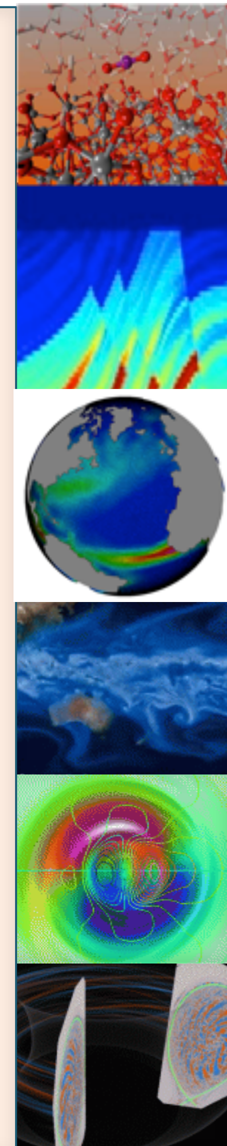
Almgren (LBNL)	**BoxLib AMR Framework**
Trebotich (LBNL)	**Chombo-crunch**

High Energy Physics

Vay (LBNL)	**WARP & IMPACT**
Toussaint(Arizona)	**MILC**
Habib (ANL)	**HACC**

Nuclear Physics

Maris (Iowa St.)	**MFDn**
Joo (JLAB)	**Chroma**
Christ/Karsch (Columbia/BNL)	**DWF/HISQ**



Basic Energy Sciences

Kent (ORNL)	**Quantum Espresso**
Deslippe (NERSC)	**BerkeleyGW**
Chelikowsky (UT)	**PARSEC**
Bylaska (PNNL)	**NWChem**
Newman (LBNL)	**EMGeo**

Biological and Environmental Research

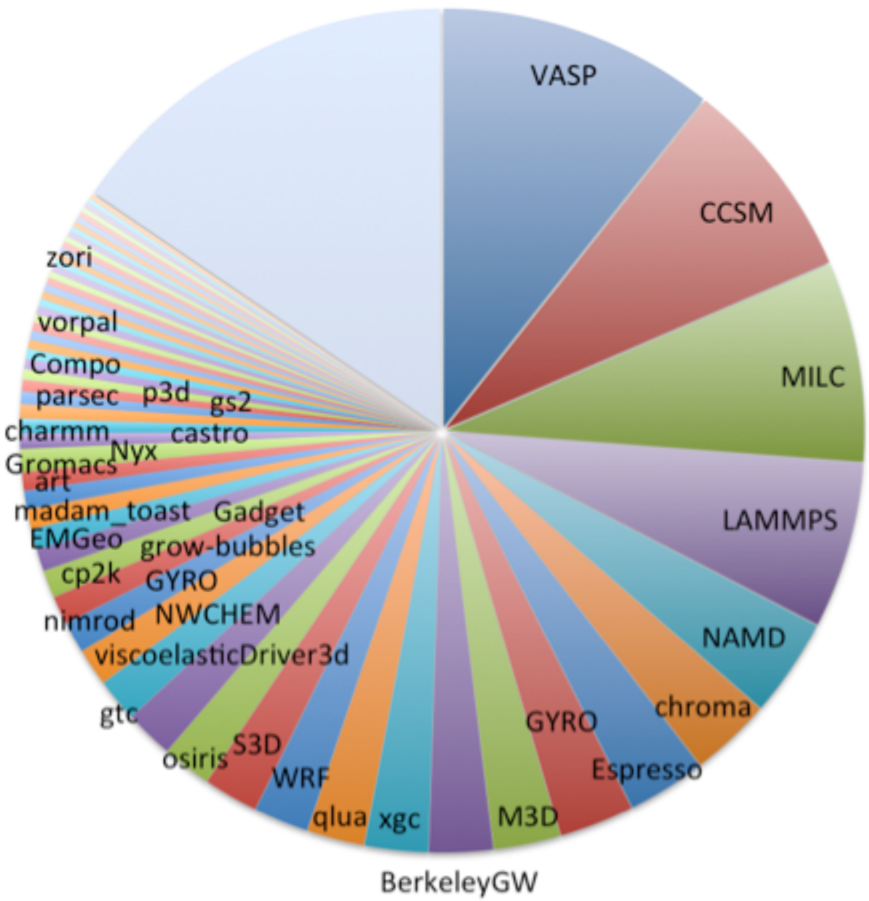
Smith (ORNL)	**Gromacs**
Yelick (LBNL)	**Meraculous**
Ringler (LANL)	**MPAS-O**
Johansen (LBNL)	**ACME**
Dennis (NCAR)	**CESM**

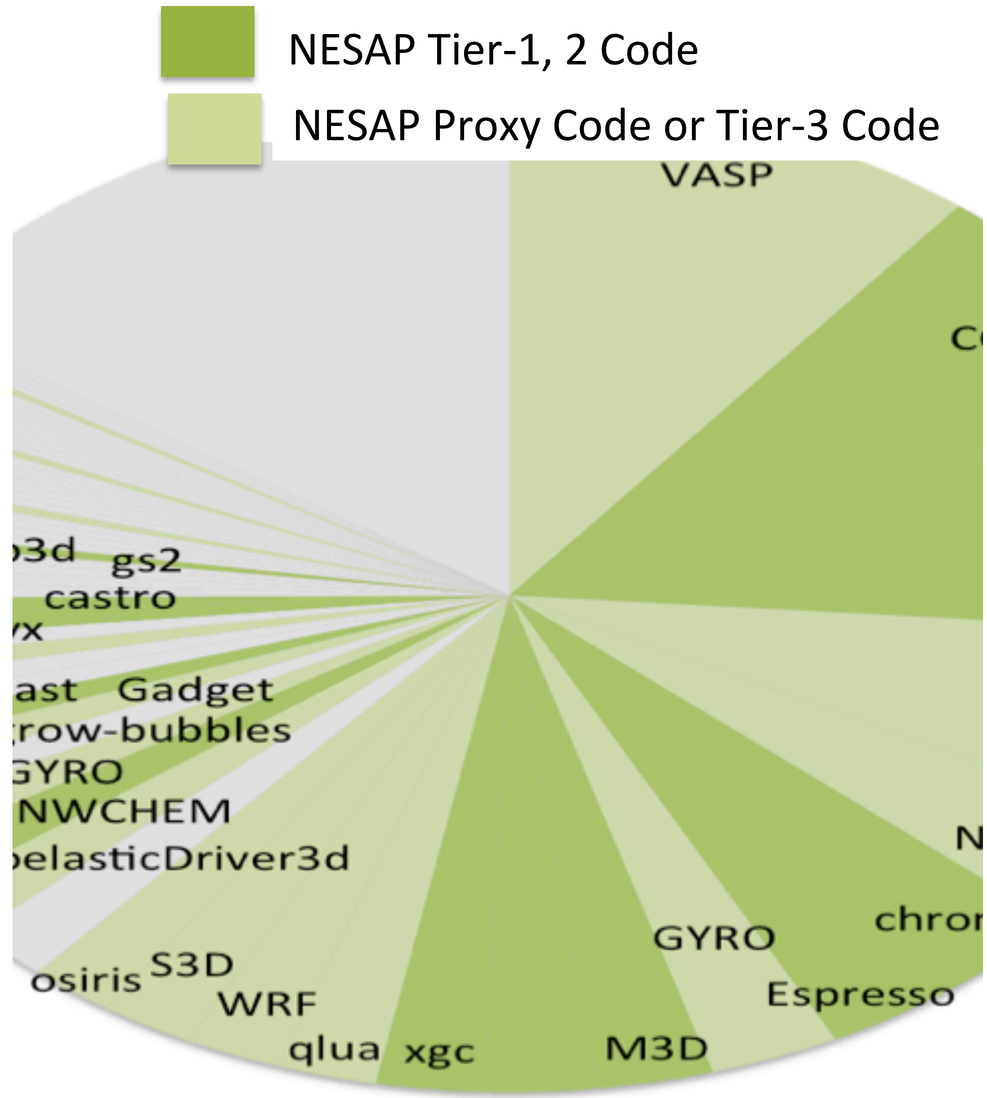
Fusion Energy Sciences

Jardin (PPPL)	**M3D**
Chang (PPPL)	**XGC1**

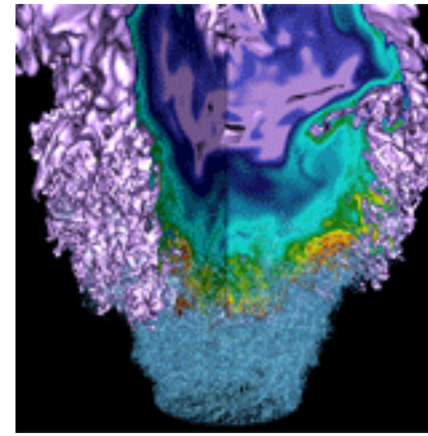
NESAP applications represent a large fraction of NERSC workload

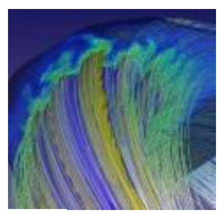
Breakdown of Application Hours on Hopper and Edison 2013

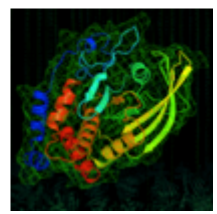


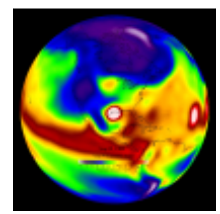


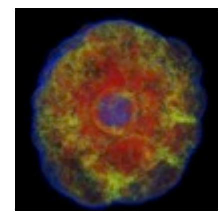
Case Study: FLASH code

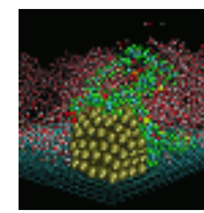












NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Case Study on the Xeon-Phi Coprocessor Architecture: NERSC's Babbage Testbed

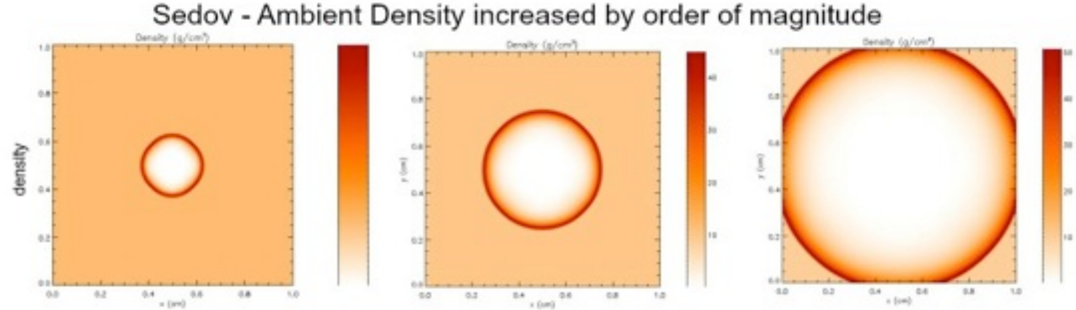
- **45 Sandy-bridge nodes with Xeon-Phi Co-processor**
- **Each Xeon-Phi Co-processor has**
  - 60 cores
  - 4 HW threads per core
  - 8 GB of memory
- **Multiple ways to program with co-processor**
  - As an accelerator
  - Reverse accelerator
  - As a self-hosted processor (ignore Sandy-bridge)

  - We chose to test as if the Xeon-Phi was a stand alone processor to mimic Knight's Landing architecture

FLASH application readiness

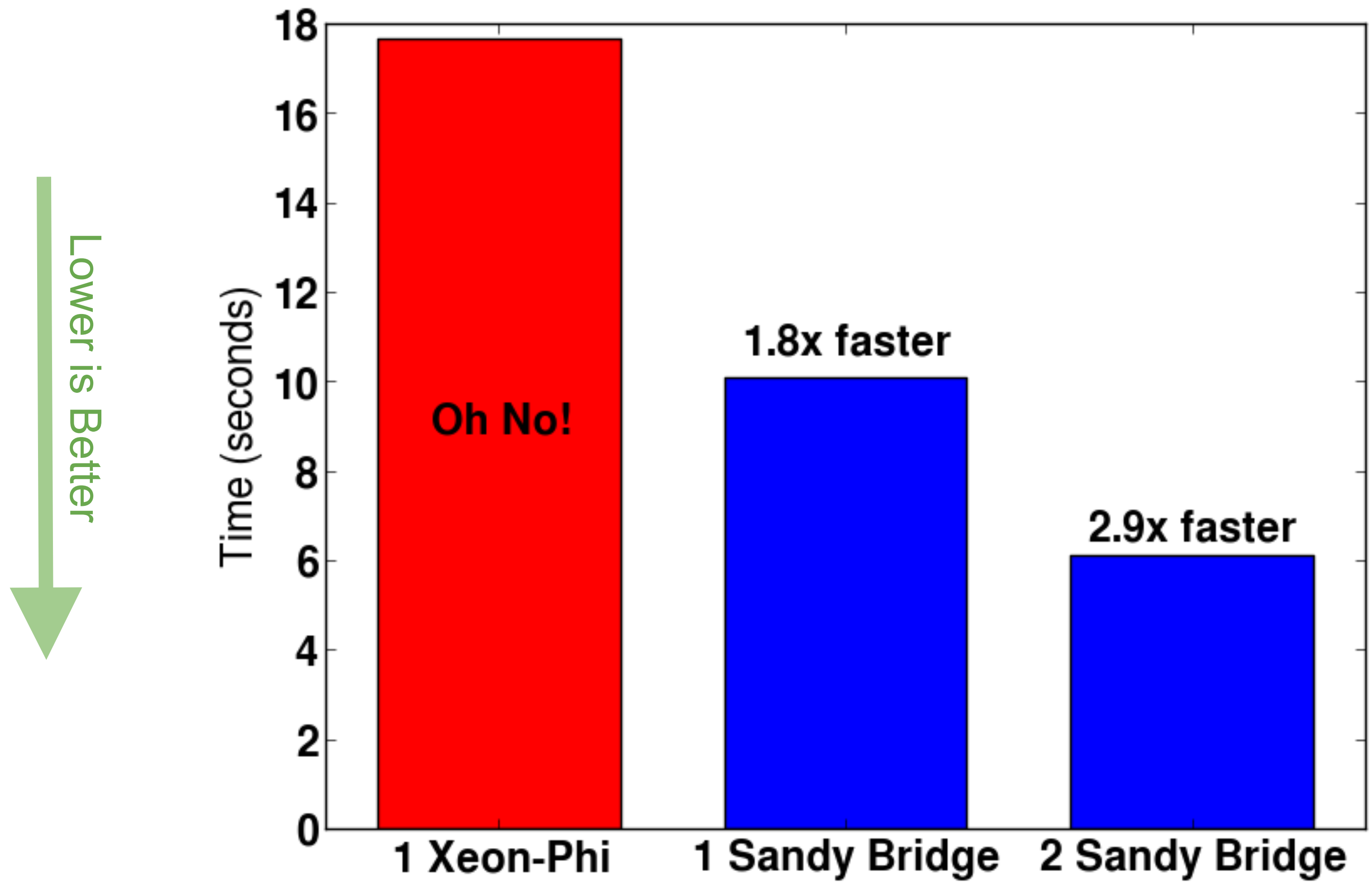
- **FLASH is astrophysics code with explicit solvers for hydrodynamics and magneto-hydrodynamics**
- **Parallelized using**
  - MPI domain decomposition AND
  - OpenMP multithreading over local domains or over cells in each local domain
- **Target application is a 3D Sedov explosion problem**
  - A spherical blast wave is evolved over multiple time steps
  - Use configuration with a uniform resolution grid and use 100^3 global cells
- **The hydrodynamics solvers perform large stencil computations.**

Sedov - Ambient Density increased by order of magnitude



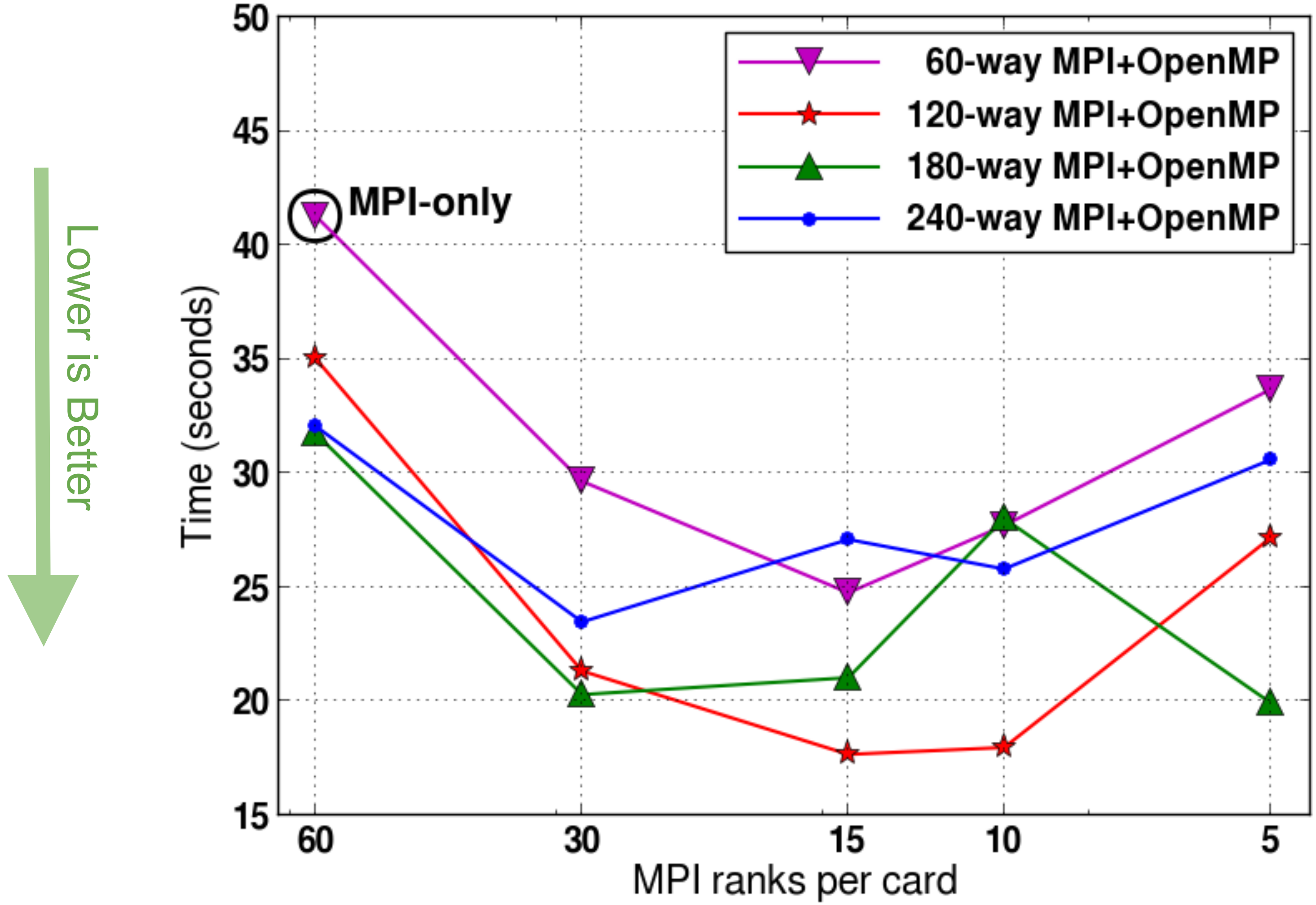
Case study by Chris Daley

Initial best KNC performance vs host



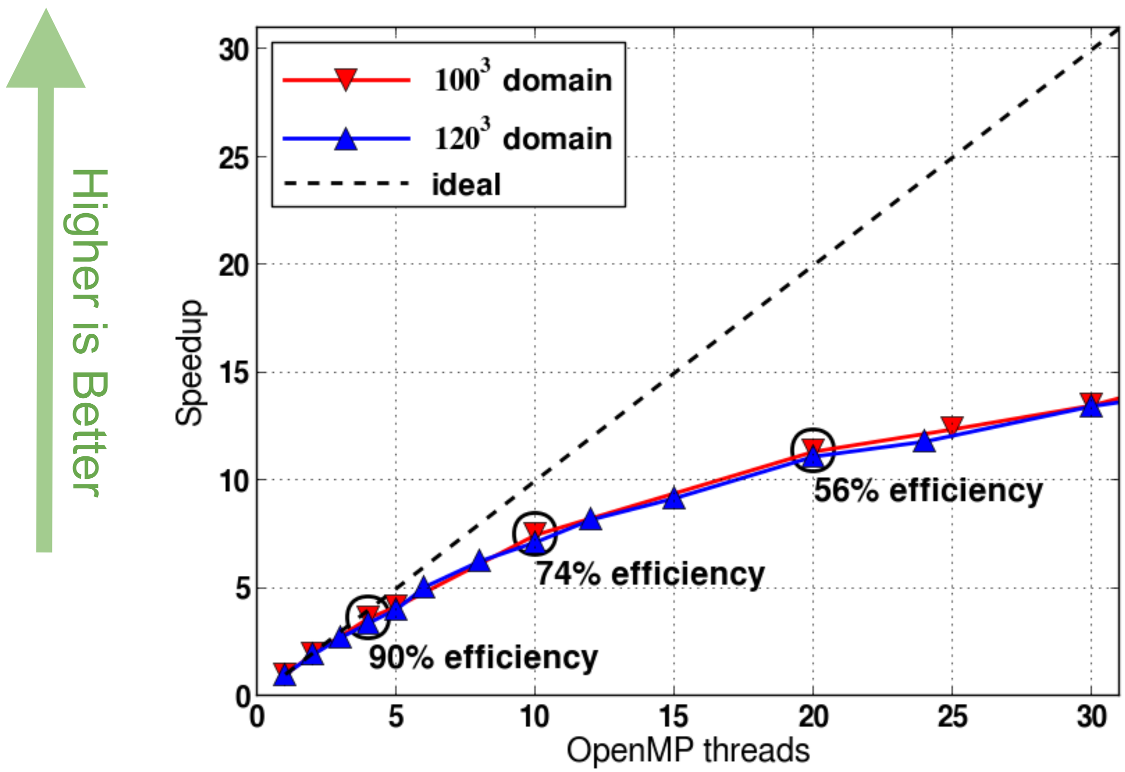
Case study by Chris Daley

Best configuration on 1 KNC card



Case study by Chris Daley

MIC performance study 1: thread speedup

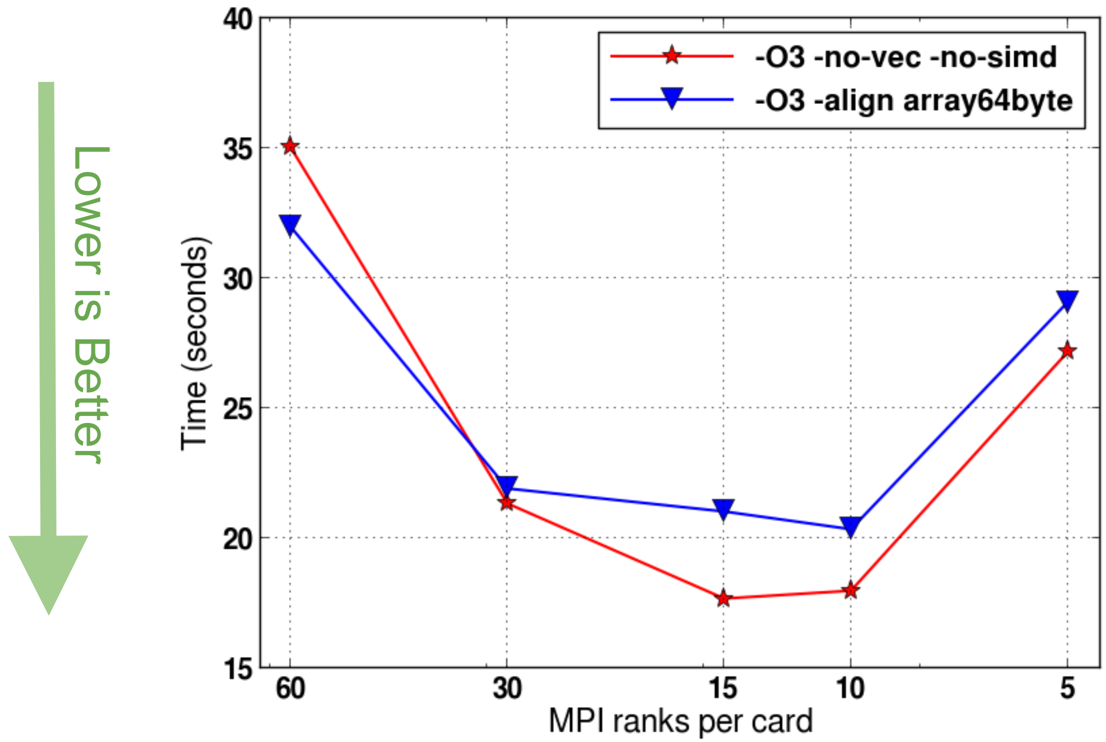


- 1 MPI rank per MIC card and various numbers of OpenMP threads
- Each OpenMP thread is placed on a separate core
- 10x thread count ideally gives a 10x speedup

- Speedup is not ideal
  - But it is not the main cause of the poor MIC performance
  - ~70% efficiency @ 12 threads (as would be used with 10 MPI ranks per card)

Case study by Chris Daley

FLASH KNC vectorization study



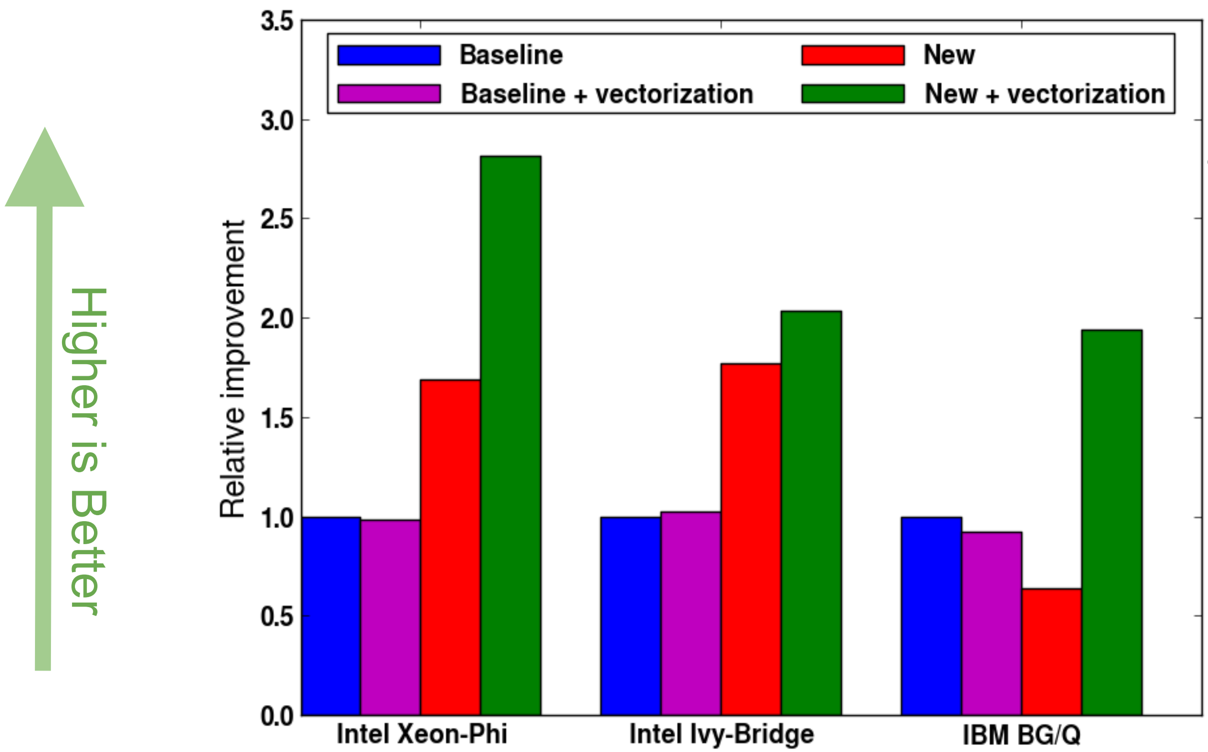
No vectorization gain!

- We find that most time is spent in subroutines which update fluid state 1 grid point at a time
  - The data for 1 grid point is laid out as a structure of fluid fields, e.g. density, pressure, …, temperature next to each other: A(HY_DENS:HY_TEMP)
  - Vectorization can only happen when the same operation is performed on multiple fluid fields of 1 grid point!

Case study by Chris Daley

Enabling vectorization

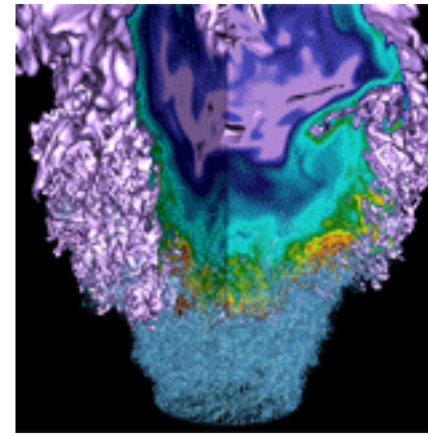
- **Must restructure the code**
  - The fluid fields should no longer be next to each other in memory
  - A(HY_DENS:HY_TEMP) should become A_dens(1:N), …, A_temp(1:N)
    - The 1:N indicates the kernels now operate on N grid points at a time
- **We tested these changes on part of a data reconstruction kernel**

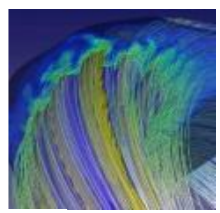


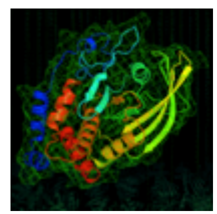
- **The new code compiled with vectorization options gives the best performance on 3 different platforms**

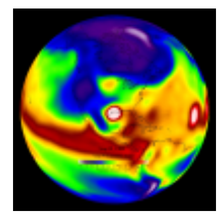
Case study by Chris Daley

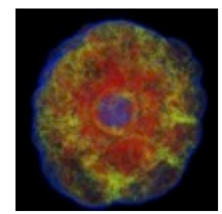
BerkeleyGW Case Study

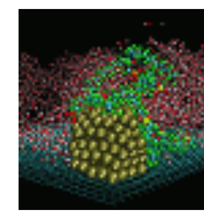












NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

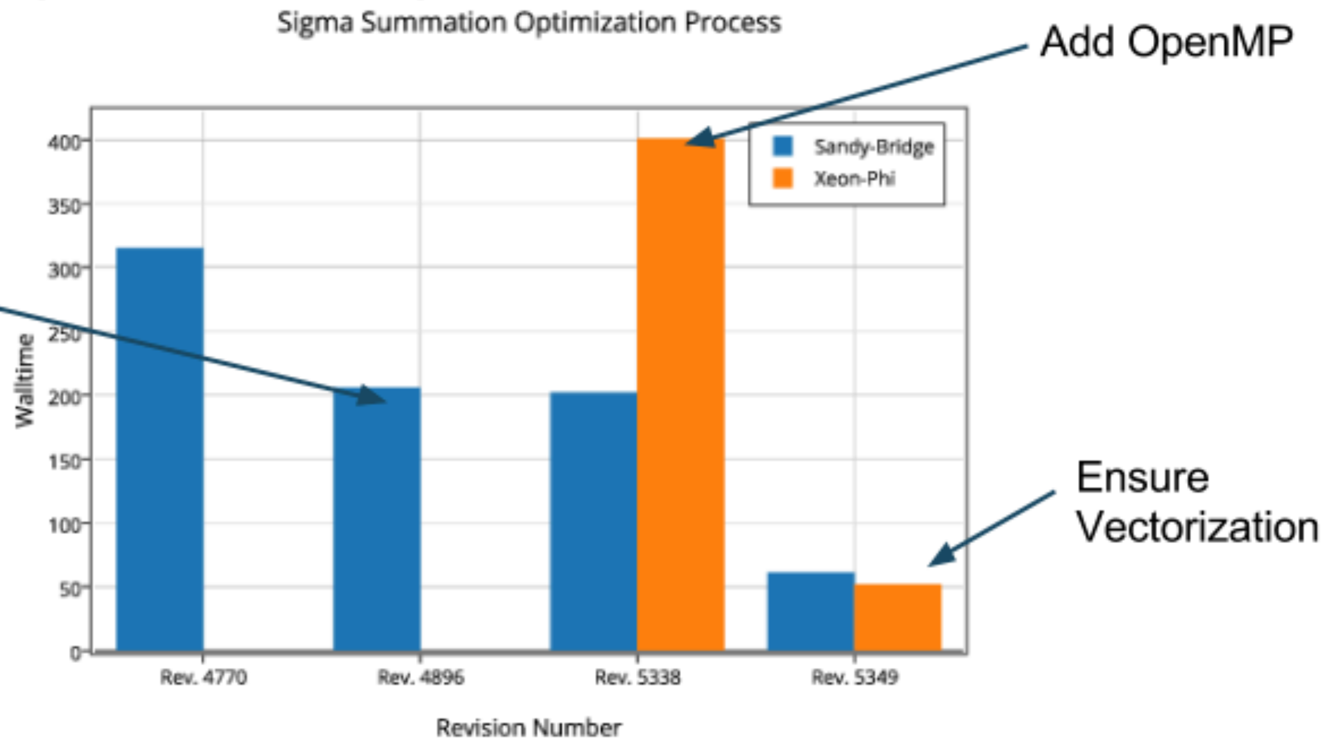
Case Study: BerkeleyGW

1. Target more on-node parallelism. (MPI model already failing users)
2. Ensure key loops/kernels can be vectorized.

Example: Optimization steps for Xeon Phi Coprocessor

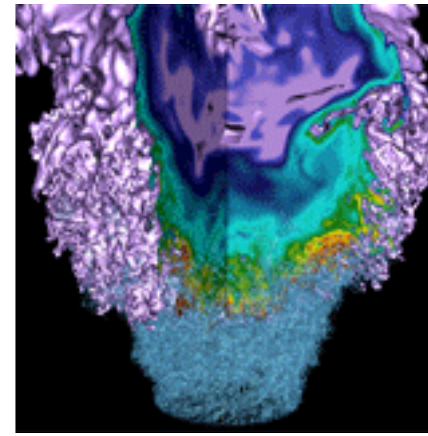
Refactor to Have 3 Loop Structure:

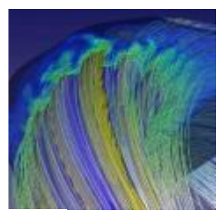
Outer: MPI
Middle: OpenMP
Inner: Vectorization

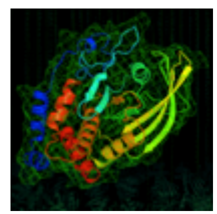


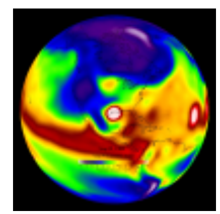
Case study by Jack Deslippe

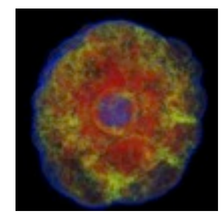
Case Study: DWF-HIST

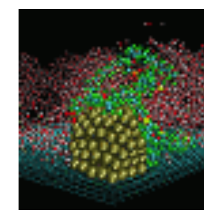






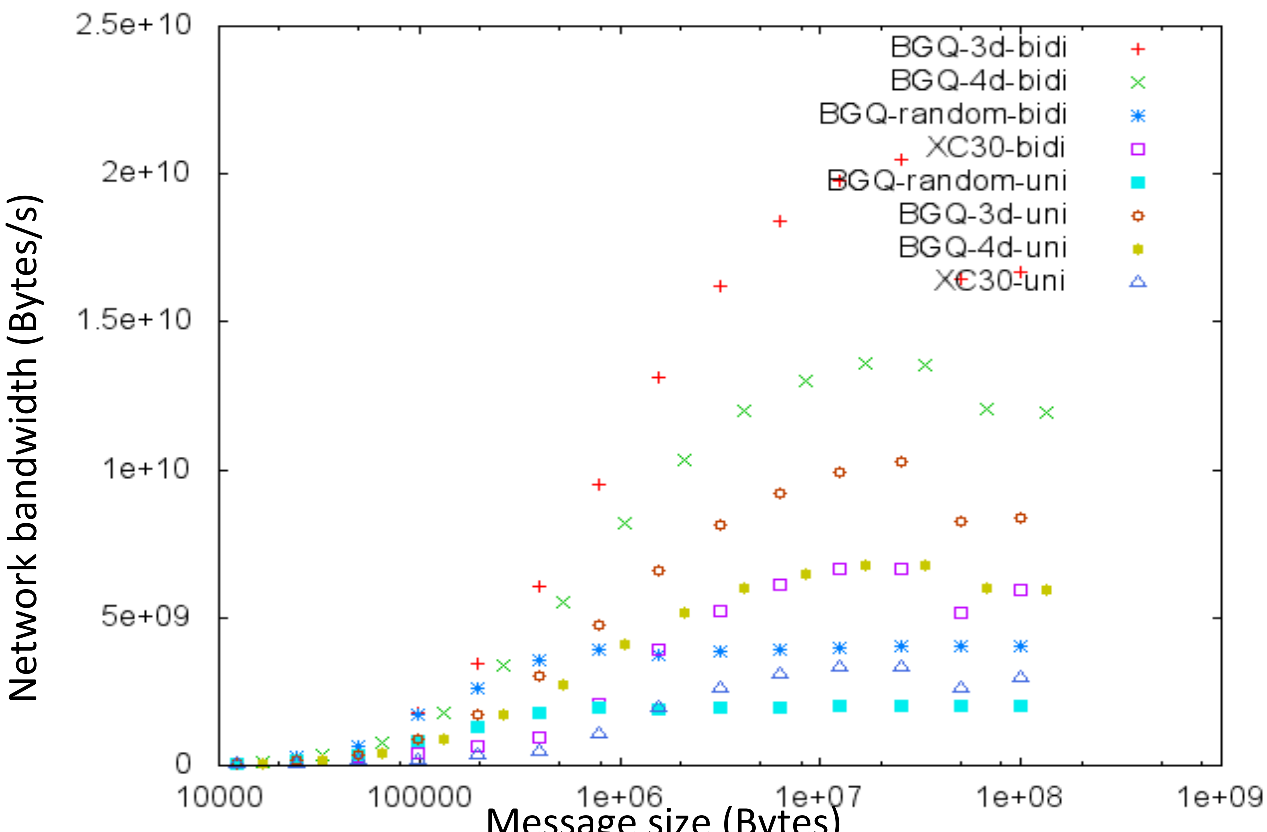






DWF-HISQ NESAP Status

- **Lattice QCD code used to study standard model**
- **Conjugate gradient test reached 206 Gflops/ on KNC (before NESAP)**
- **Strong scaling performance strongly dependent on network bandwidth**
- **Team currently conducting network topology bandwidth tests on Edison and BG/Q systems**



3d: BG/Q's 5D torus as nearest in a 3D logical torus
4d: BG/Q's 5D torus as nearest in a 4d logical torus
random: topology unaware

uni: uni-directional
bidi: bidirectional

In Summary

- Cori system will showcase a number of capabilities and architectural elements expected in future exascale systems
- The transition to more energy efficient architectures will be labor intensive for many apps
- How the deepening memory hierarchy on KNL will be used in practice is not yet known
- Many NERSC users are NOT focused on interconnect issues, but rather on-node performance at current time
- Burst Buffer has the potential to accelerate performance for many workloads – for next time.

Thank you! (And we are hiring!)