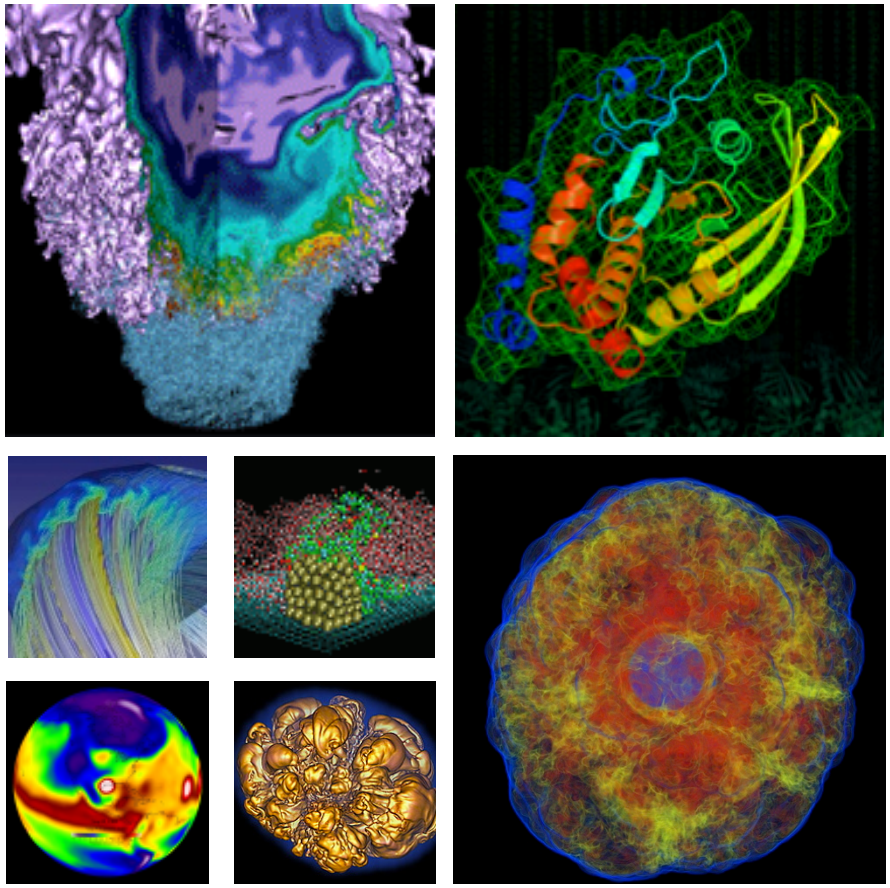
The Cori System



NERSC 40 YEARS at the FOREFRONT 1974-2014

Katie Antypas

NUG 2015
February 24, 2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

The Cori System

- **Cori will support the broad Office of Science research community and begin to transition the workload to more energy efficient architectures**

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)**
- **10x performance of Hopper system on real applications**
- **Lustre File system**
  - 28 PB capacity, >700 GB/sec peak performance
- **NVRAM "Burst Buffer" for I/O acceleration**
- **Significant Intel and Cray application transition support**
- **Delivery in two phases, summer 2015 and summer 2016**
- **Installation in new LBNL CRT**

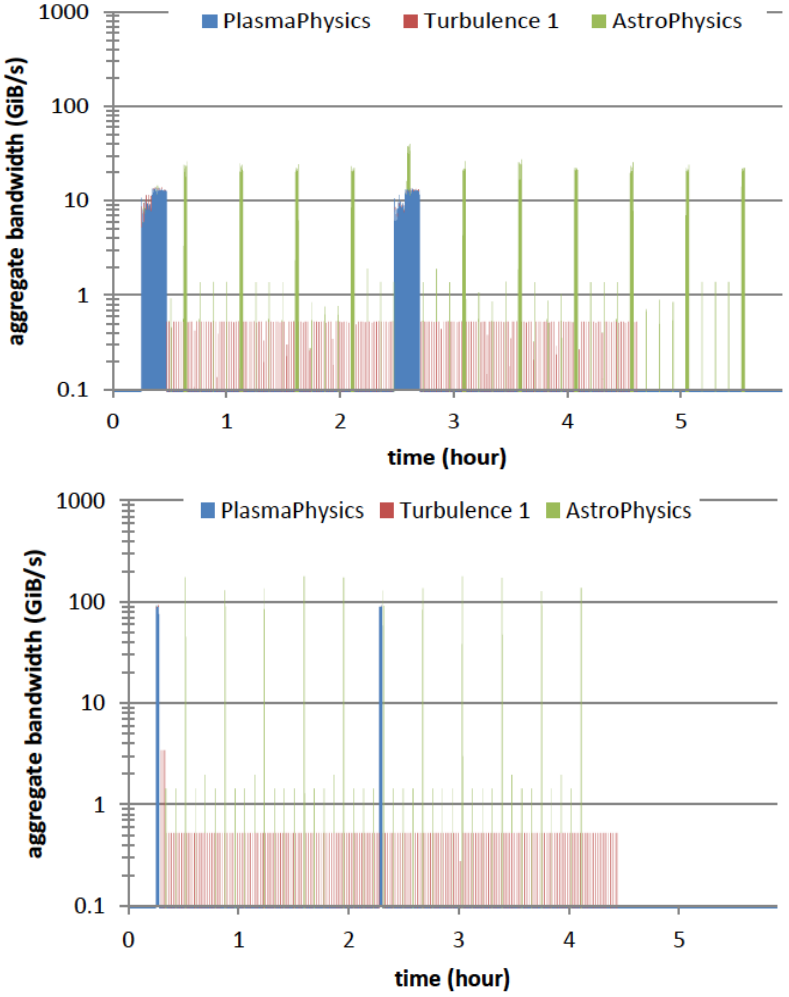
Cori Phase 1 (Data Partition) Summer 2015

- **Approximately the same computing capability as Hopper using Intel Haswell processor**
- **Goals**
  - Replace hours lost from turning off Hopper
  - Become a data partition for Cori
- **Support for data intensive science**
  - Queues for fast turn around, immediate access experiments
  - Serial and high throughput
  - External connectivity to databases from compute nodes
  - Virtualization capabilities (Collaboration with Cray, exploring Docker)
  - NVRAM Flash Burst Buffer
  - More login nodes for managing workflows

New Scheduler! SLURM

Burst Buffer Motivation

- Flash storage is significantly more cost effective at providing bandwidth than disk (up to 6x)
- Flash storage has better random access characteristics than disk, which help many SC workloads
- Users' biggest request (complaint) after wanting more cycles, is for better I/O performance



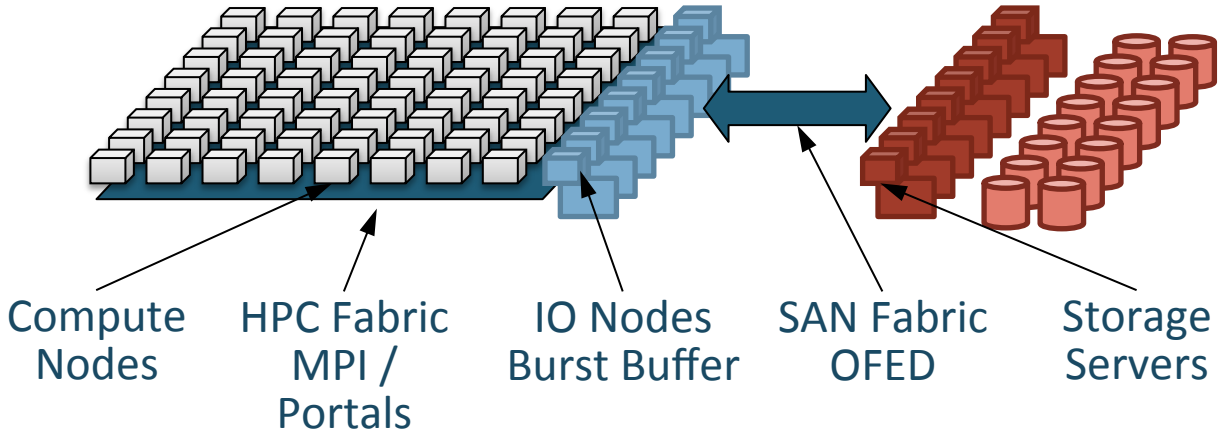
Application perceived I/O rates, with no burst buffer (top), burst buffer (bottom).

Analysis from Chris Carothers (RPI) and Rob Ross (ANL)

Burst Buffer Use Cases

- Improves application reliability (checkpoint-restart)
- Accelerates application I/O performance for small blocksize I/O and analysis files
- Provides fast temporary space for out-of-core applications (example: NWChem)
- Staging area for jobs requiring large input files or persistent fast storage between coupled simulations
- Post-processing analysis of large simulation data
- In-situ visualization and analysis
- Shared library staging area

Burst Buffer Software Development Efforts



Create Software to enhance usability and to meet the needs of all NERSC users

- Scheduler enhancements
  - Automatic migration of data to/from flash
  - Dedicated provisioning of flash resources
  - Persistent reservations of flash storage
- Caching mode – data transparently captured by the BB nodes
  - Transparent to user -> no code modifications required
- Enable In-transit analysis
  - Data processing or filtering on the BB nodes – model for exascale

Burst Buffer Software Development Timeline

Phase 3

- BB-node functionality: In Transit, filtering

Phase 2

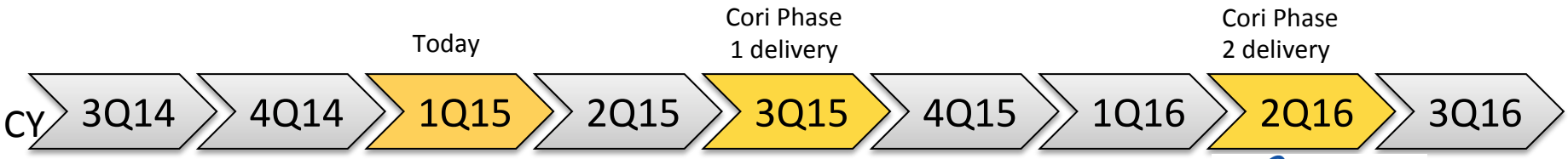
- Usability enhancements: Caching mode

Phase 1

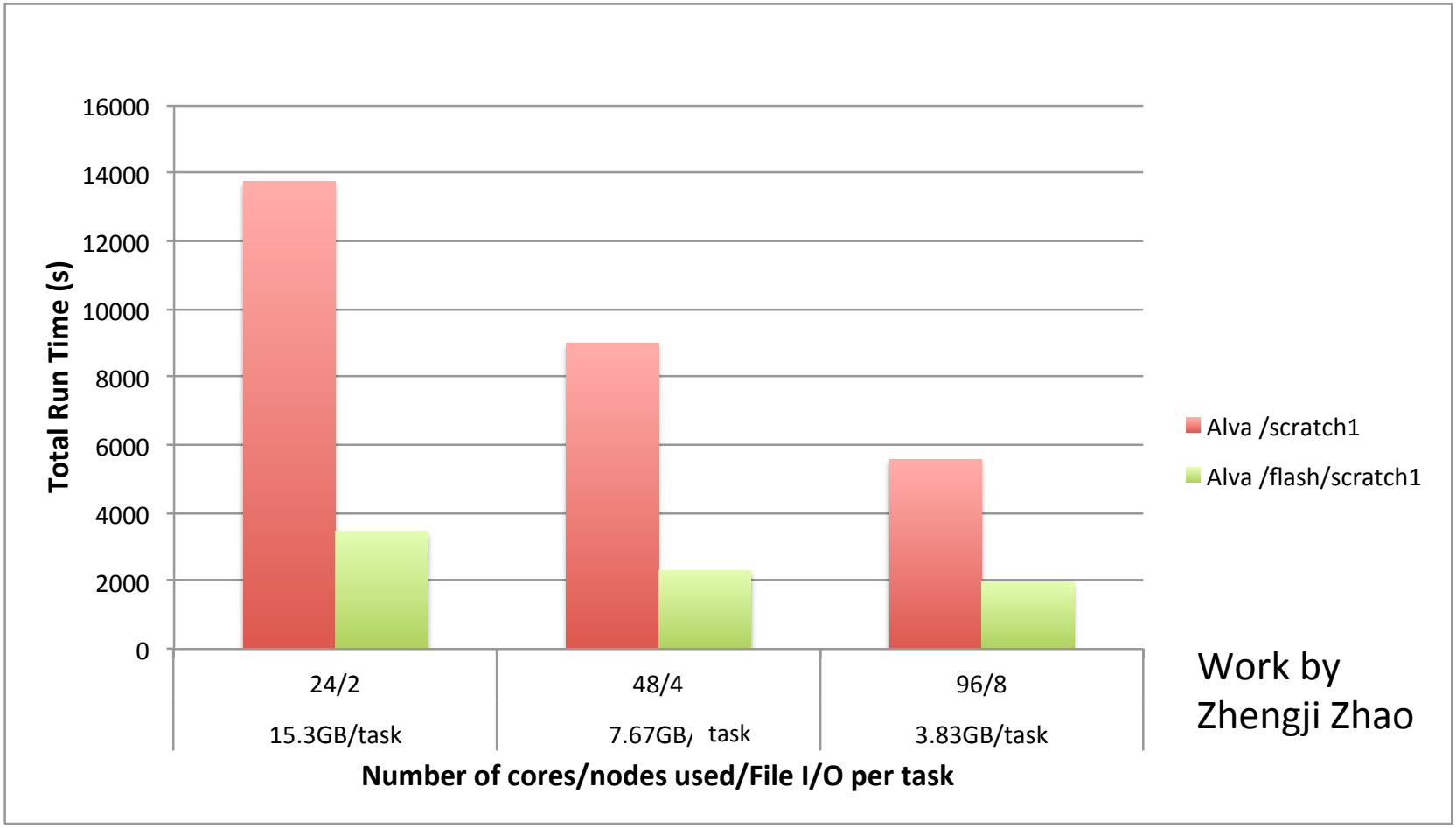
- I/O acceleration: Striping, reserved I/O bandwidth
- Job launch integration: allocation of space – per job or persistently
- Administrative functionality

Phase 0

- Static mapping of compute to BB node
- User responsible for migration of data

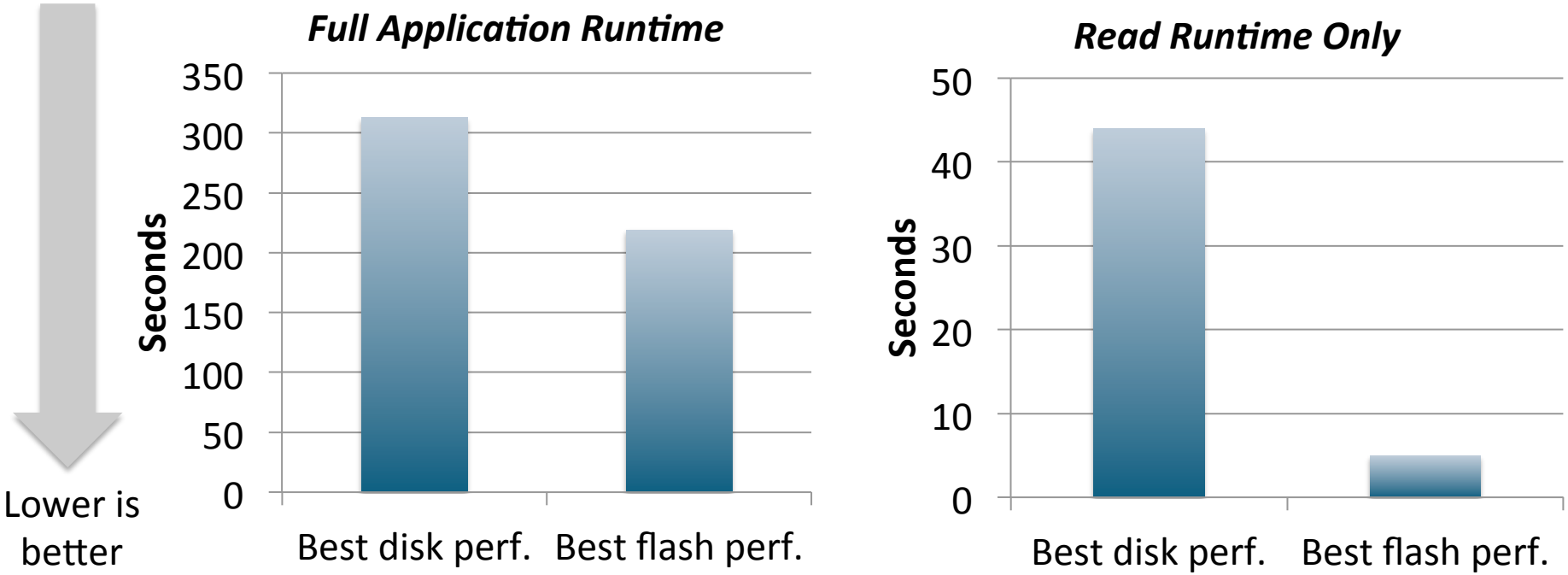


NWChem Out-of-Core Performance: Flash vs Disk on Burst Buffer testbed



- NWChem MP2 Semi-direct energy computation on 18 water cluster with aug-cc-pvdz basis set
- Geometry (18 water cluster) from A. Lagutschenkov, e.tal, *J. Chem. Phys.* **122**, 194310 (2005).

TomoPy performance comparison between flash and disk file systems

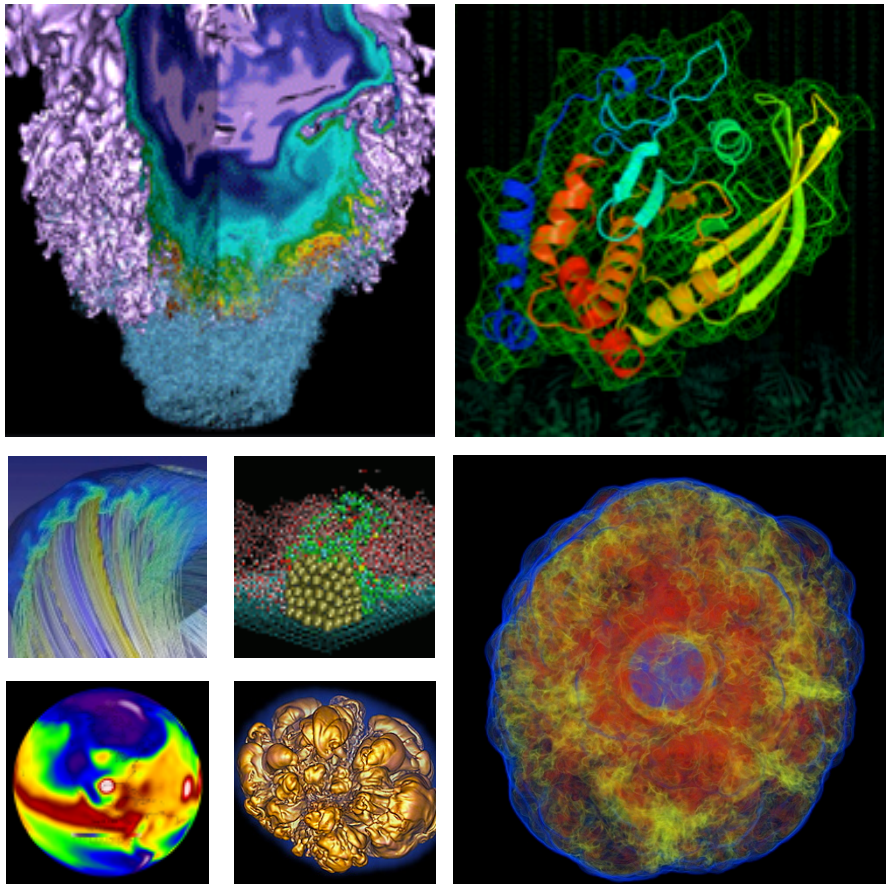


- **This I/O intensive application runtime improves by 40% with the only change switching from disk to flash**
- **Read performance is much better when using Flash: ~8-9x faster than disk**
- **Disk performance testing showed high variability (3x runtime), whereas the flash runs were very consistent (2% runtime difference)**

NERSC High Level Burst Buffer Plan

- **Deploy 3 cabinets worth of Burst Buffer nodes**
  - Each BB node provides 6GB/sec and 6.25 TB of storage
  - Each cabinet provides 575 GB/sec and 600 TB of storage
  - Total specifications for 3 cabinets 1.7TB/sec and 1.8PB storage (formatting SSDs will reduce capability a little)
- **Deployment plan**
  - Deliver ½ of burst buffer nodes with Phase 1 Cori system (Haswell)
  - Deliver remaining ½ with Phase 2 Cori system (Knights Landing)
  - Burst Buffer blades are distributed throughout the system for optimal performance

Cori Phase 2 – KNL partition



NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Intel "Knights Landing" Processor

- Next generation Xeon-Phi, >3TF peak
- Single socket processor - Self-hosted, not a co-processor, not an accelerator
- Greater than 60 cores per processor with support for four hardware threads each; more cores than current generation Intel Xeon Phi™
- Intel® "Silvermont" architecture enhanced for high performance computing
- 512b vector units (32 flops/clock – AVX 512)
- 3X single-thread performance over current generation Xeon-Phi co-processor
- High bandwidth on-package memory, up to 16GB capacity with bandwidth projected to be 5X that of DDR4 DRAM memory
- Higher performance per watt

Knights Landing Integrated On-Package Memory

Cache Model

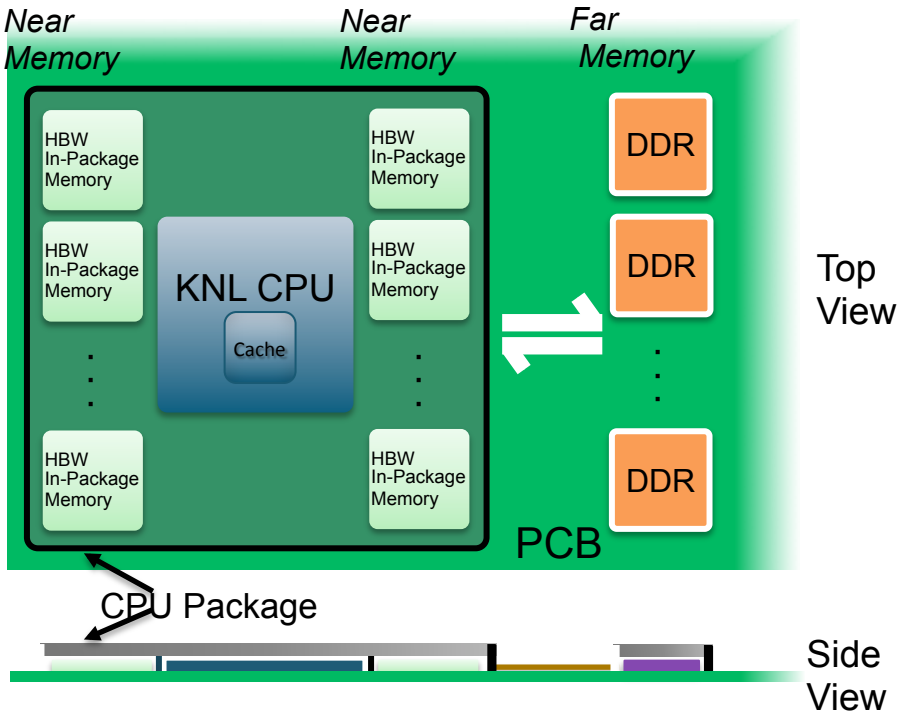
Let the hardware automatically manage the integrated on-package memory as an "L3" cache between KNL CPU and external DDR

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

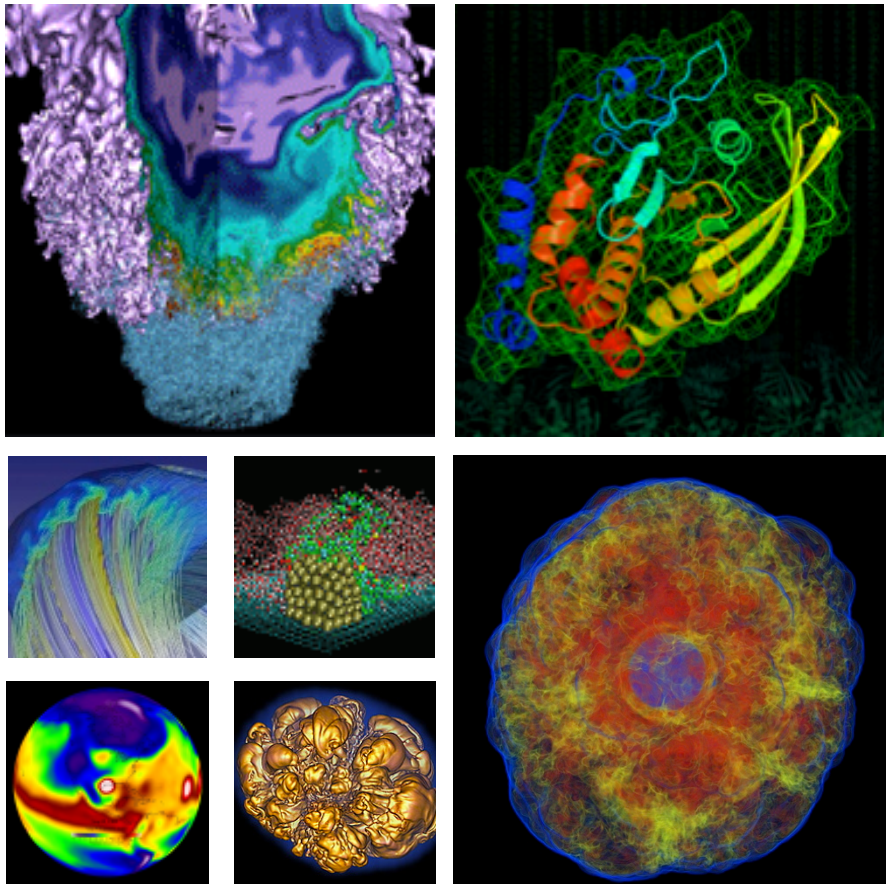
Cori's Programming Environment

- **Programming environment on Cori will look similar to Franklin, Hopper and Edison**
- **Cori is *not* a heterogeneous or accelerator system**
- **Programmed via Fortran/C/C++ plus MPI+OpenMP**
- **Intel / Cray/ GNU Programming Environments and commitment from Cray and Intel for optimized math and I/O libraries**
- **Multiple vendor profiling tools: CrayPAT and Vtune**
  - Interest from 3rd-party suppliers, too
- **DDT and Totalview support + Cray debugging tools**

Running on Cori

- Codes will run on Cori without any changes using MPI-only programming
- To take advantage of the Knights Landing architecture, applications must
  - Exploit more parallelism
  - Express thread-level parallelism
  - Exploit data-level parallelism (vectorization)
  - Manage data placement and movement
  - Accommodate less memory per process space

NERSC Exascale Science Application Program (NESAP) Update



Katie Antypas
NERSC-8 Project Manager

February 20, 2015

NERSC Exascale Science Application Program

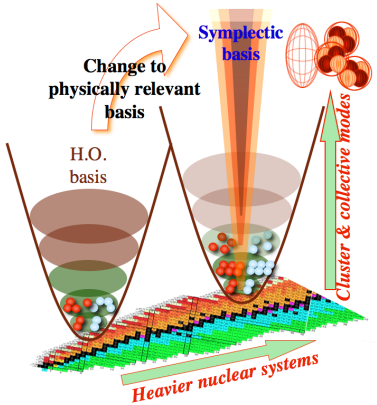
- **Goal: Prepare DOE SC user community for Cori manycore architecture**
- **Partner closely with ~20 application teams and apply lessons learned to broad SC user community**
- **NESAP activities include:**
  - User training
  - Deep technical dives with Intel and Cray staff
  - NERSC/Cray Center of Excellence
  - Early access to prototype hardware
  - Early access to Cori system
  - Post-doc program

Because of the uneven amount of resources, we are planning a 3 tiered program

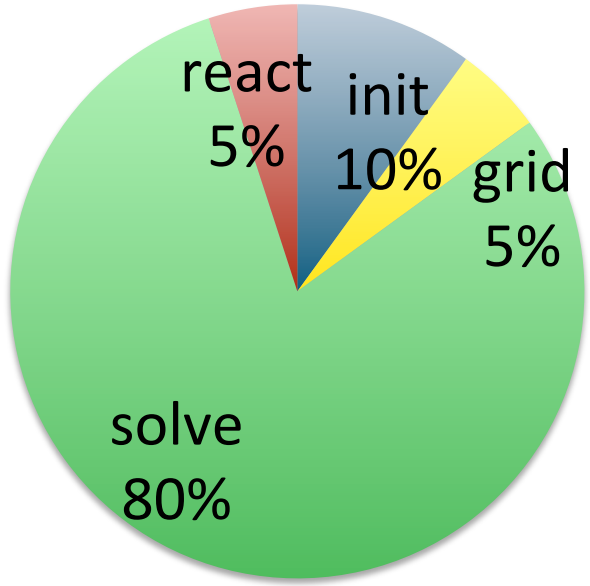
- **Tier 1: 8 Application teams**
  - Each team will have an embedded post-doc
  - Access to an Intel dungeon session
  - Support from NERSC Application Readiness and Cray COE staff
  - Early access to KNL testbeds and Cori system
  - User training sessions from Intel, Cray and NERSC staff
- **Tier 2: 12 Application teams**
  - All the resources of the Tier 1 teams except for an embedded post-doc
- **Tier 3: ~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

Key point for Tier 3 applications teams: Encourage capable teams to prepare for Cori! Early access and training sessions can be provided to more users at minimal cost.

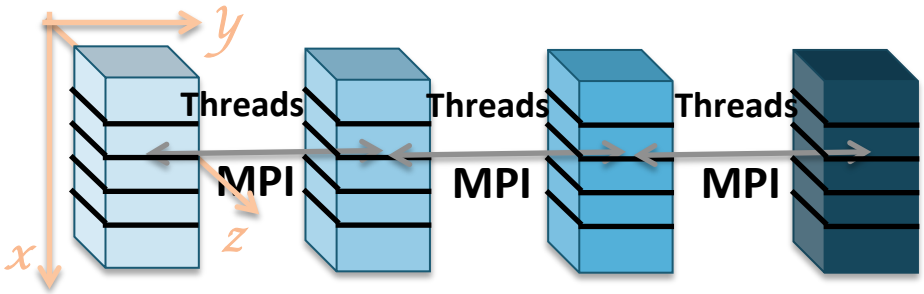
NESAP teams are getting to work



Defining and sizing science problems and extracting kernels



Profiling code

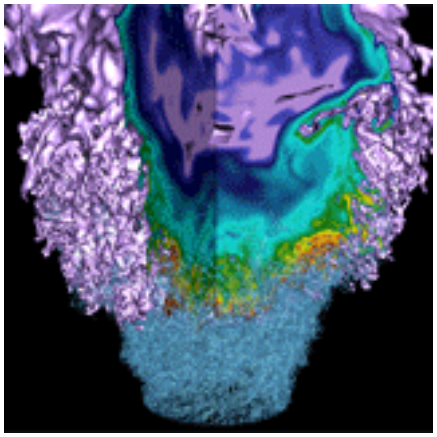


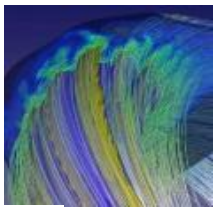
Increasing thread parallelism

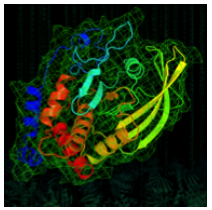
```
|--> DO I = 1, N
|       R(I) = B(I) + A(I)
|--> ENDDO
```

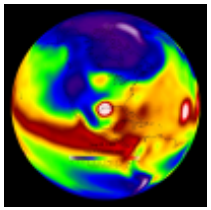
Improving data locality

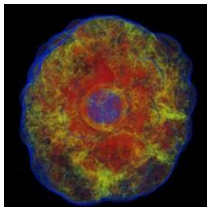
Case Study on Knights Corner system

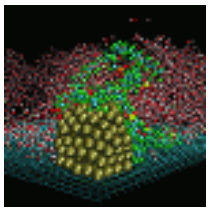












NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

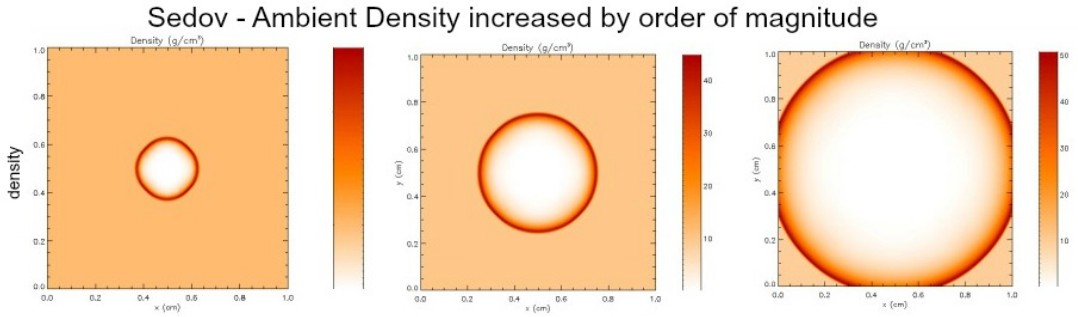
Case Study on the Xeon-Phi Coprocessor Architecture: NERSC's Babbage Testbed

- **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

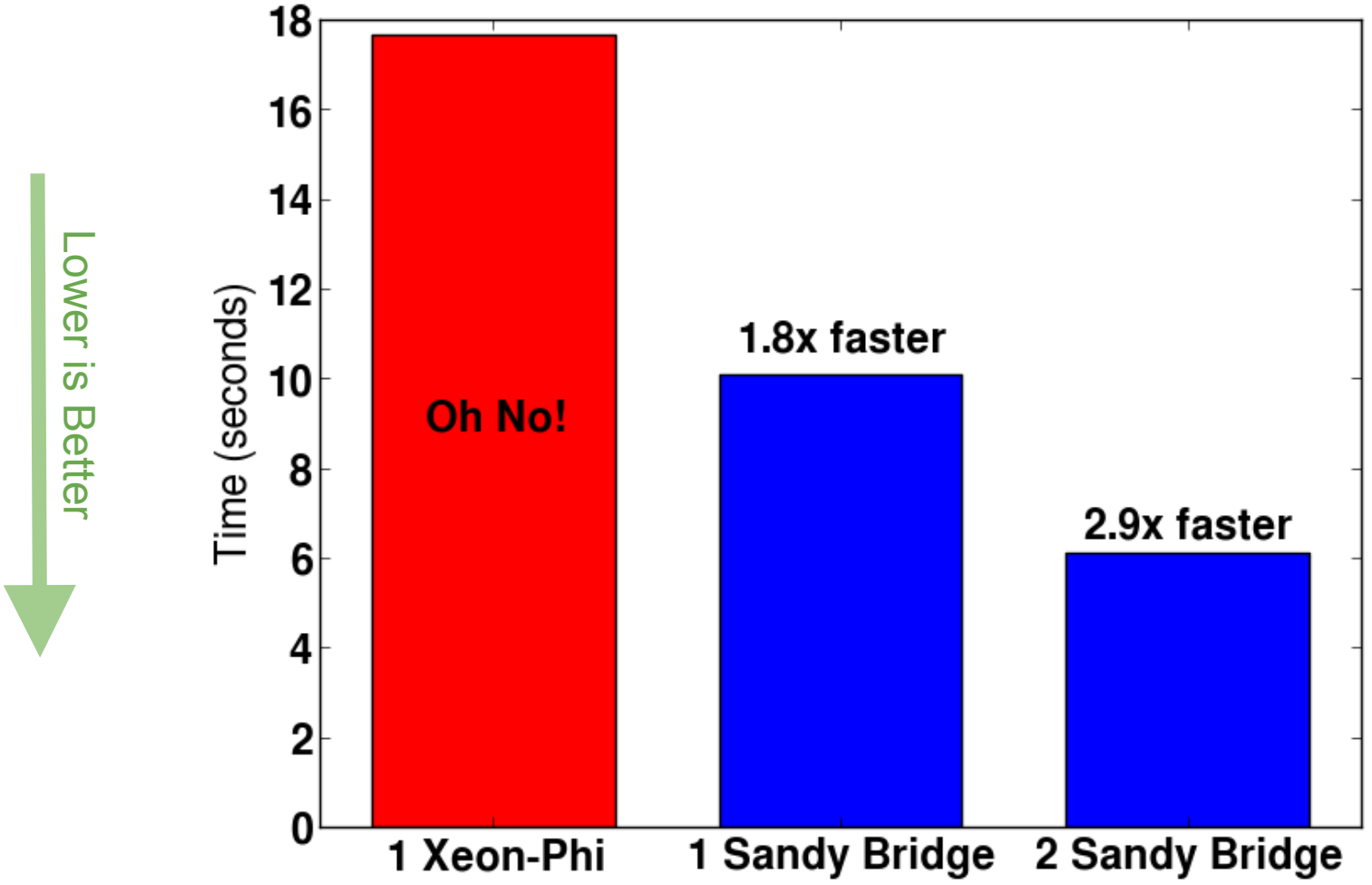
FLASH application readiness

- **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.**



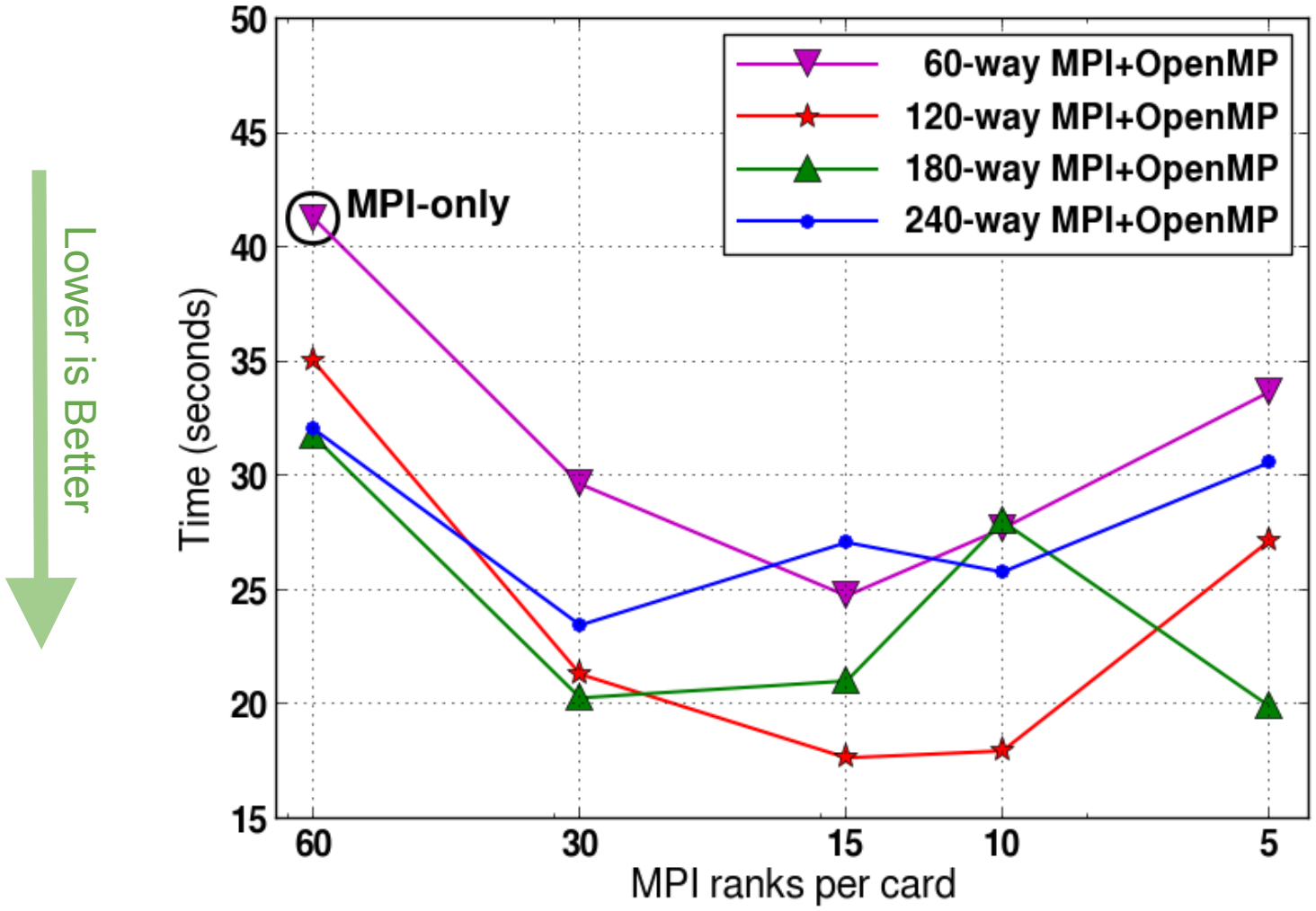
Case study by Chris Daley

Initial best KNC performance vs host



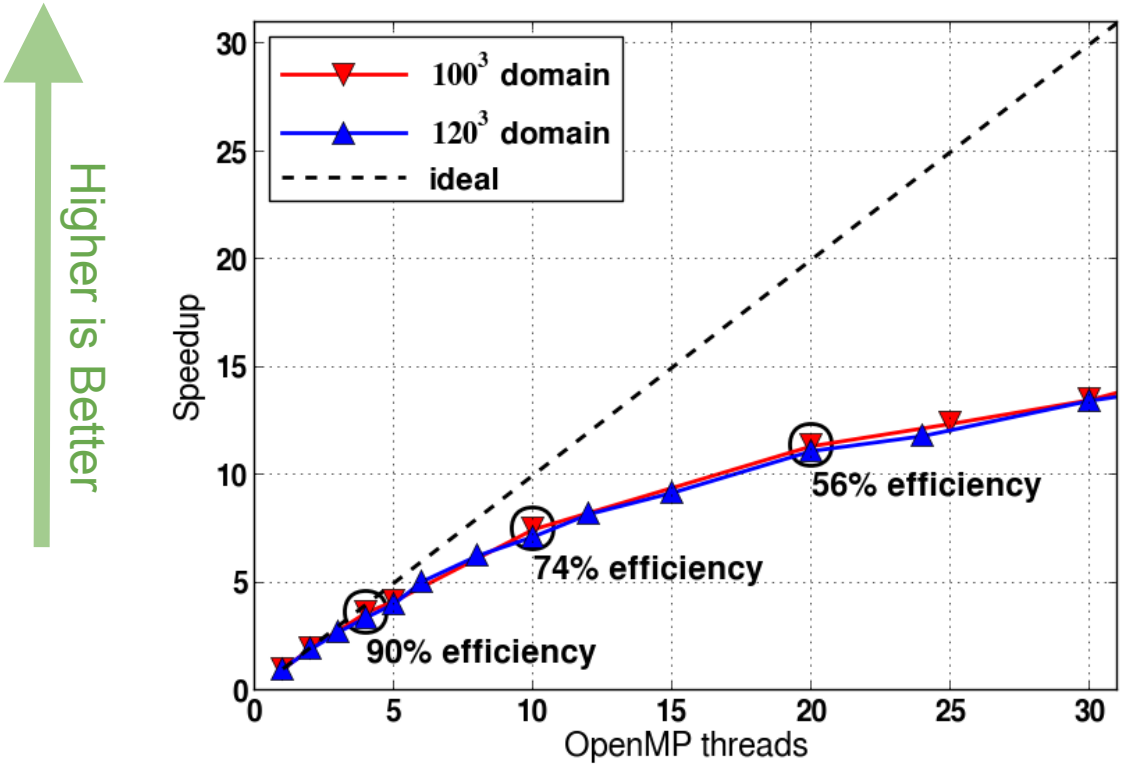
Case study by Chris Daley

Best configuration on 1 KNC card



Case study by Chris Daley

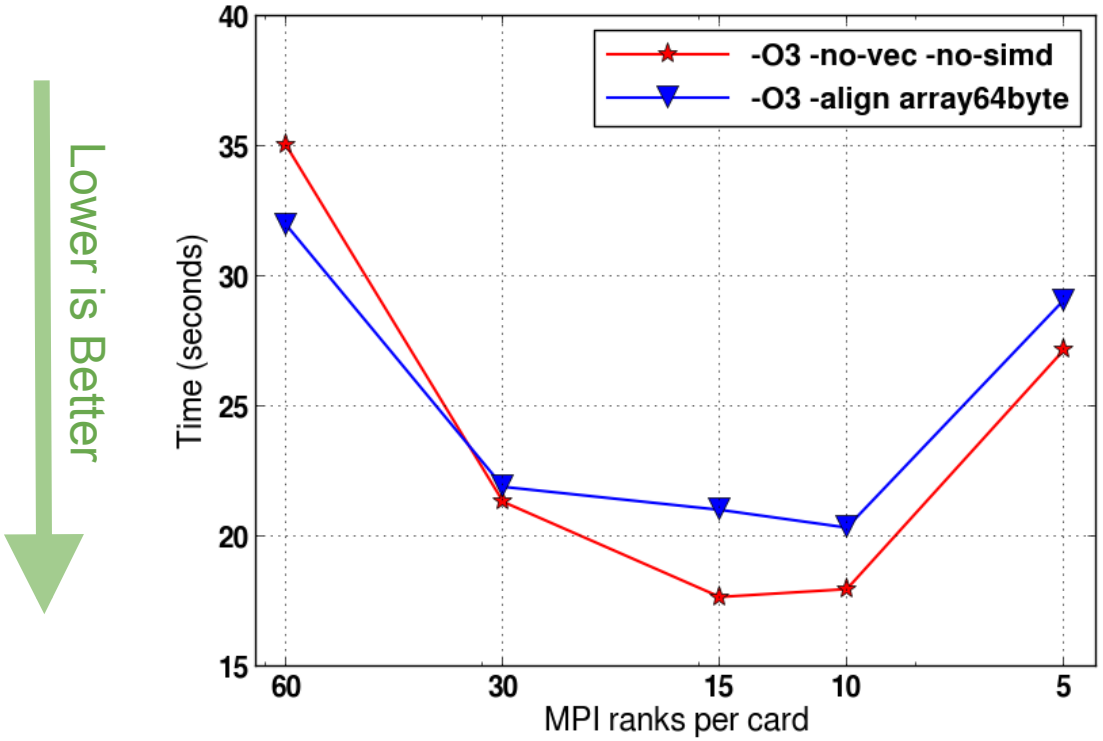
MIC performance study 1: thread speedup



- 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)

FLASH KNC vectorization study



No vectorization gain!

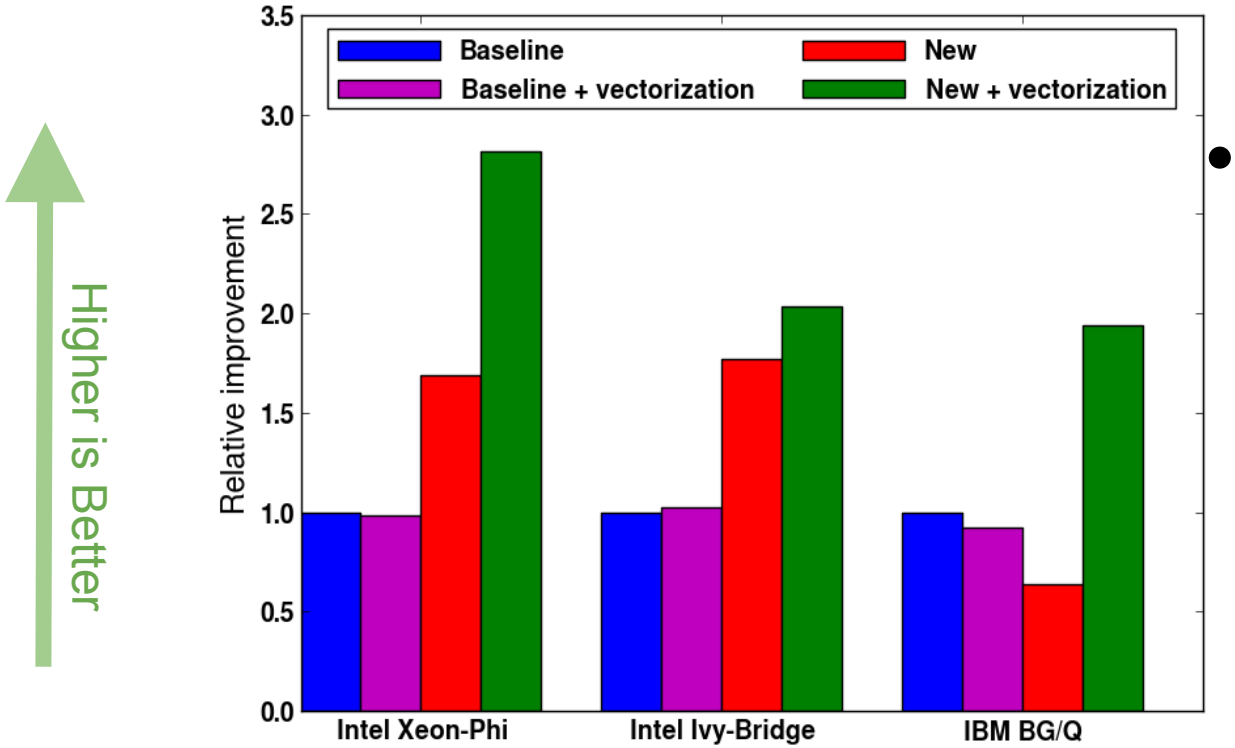
- We find that most time is spent in subroutines which update fluid state 1 grid point at a time

– 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!

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**

What about application portability?

- **NERSC, OLCF and ALCF are working together to prepare our users for our next generation architectures**
  - Application Readiness Coordination meeting in Oakland March 2014
  - Application Portability meeting in Oakland, Sept 2014
  - Staff training meeting at Oak Ridge, Jan 2015
- **Collaboration on user training and application portability**
- **NERSC, OLCF and ALCF will each help review the other's early science proposals**
- **On going partnerships and training planned with Trinity team as well**

Argonne Leadership Computing Facility

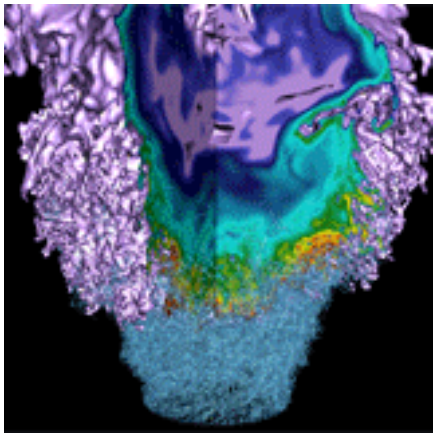
U.S. DEPARTMENT OF ENERGY | Office of Science

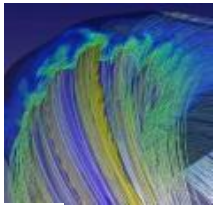
OLCF Oak Ridge Leadership Computing Facility

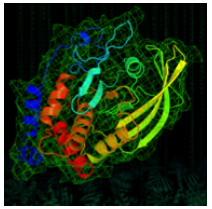
BERKELEY LAB

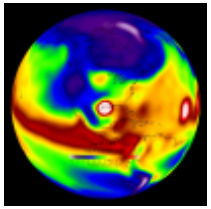
We are hiring postdocs!

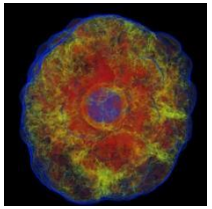
NESAP Code Updates

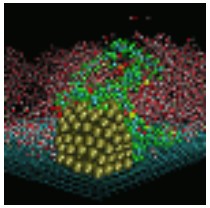










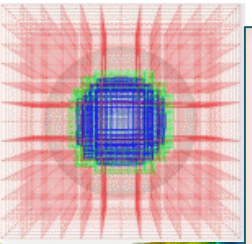
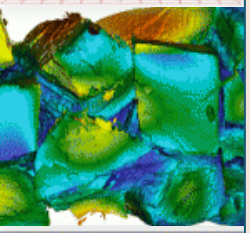
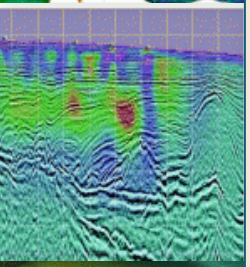
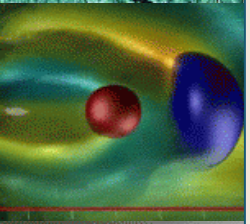
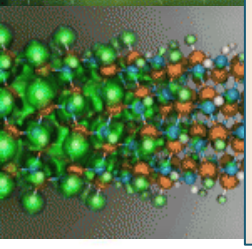


NERSC 40 YEARS at the FOREFRONT

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

NESAP Codes

Advanced Scientific Computing Research

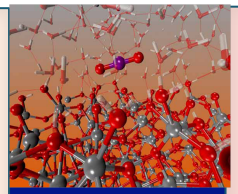
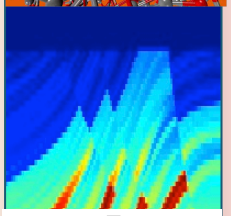
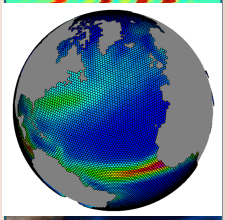
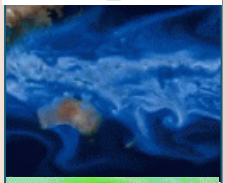
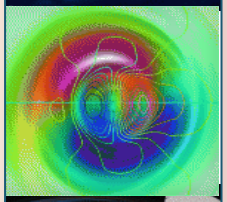
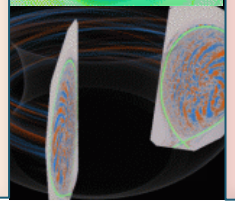
Almgren (LBNL)	**BoxLib AMR Framework**
Trebotich (LBNL)	**Chombo-crunch**

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**

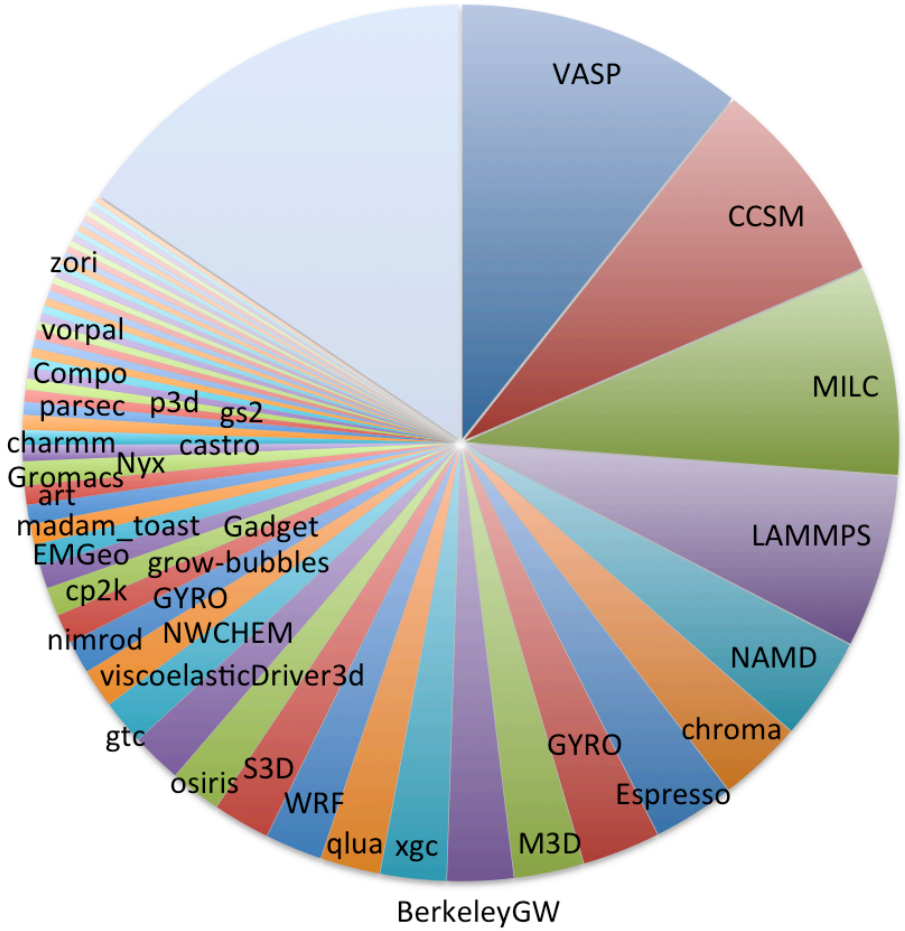
Fusion Energy Sciences

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

NERSC's Challenge

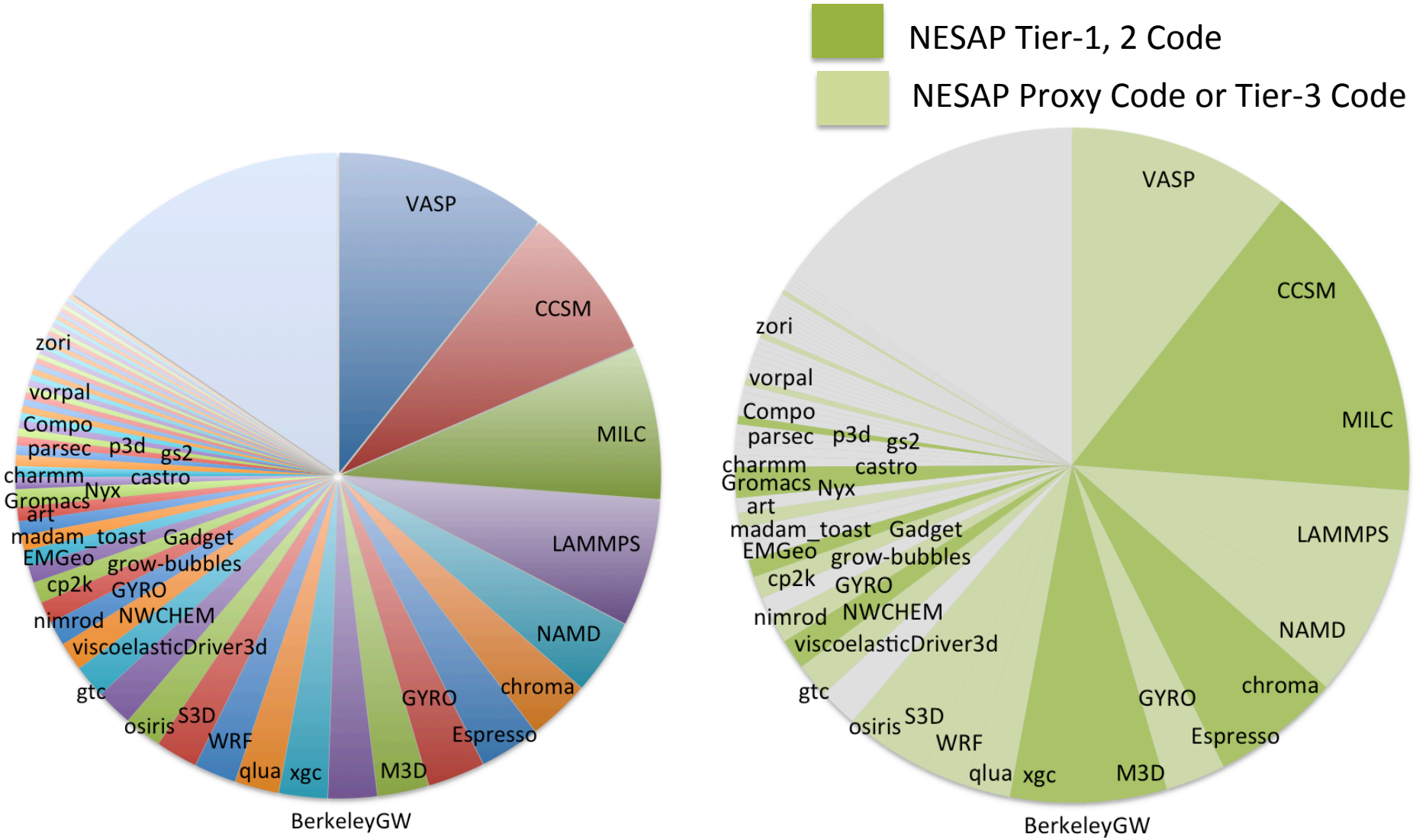
- Thousands of users – as of yesterday 5950
- More than 700 projects
- Hundreds of codes >600
- We don't select our users!
- Science output is top priority
- Our users have an insatiable demand for computing but we have a limited power budget – driving the need to move to more energy efficient computing

2013 Breakdown of Allocation Time

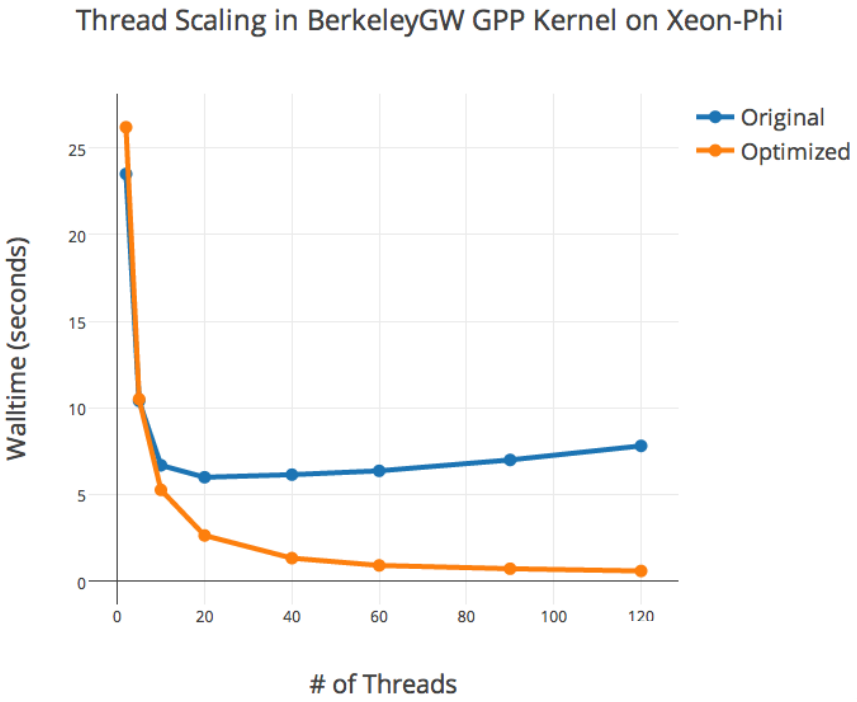


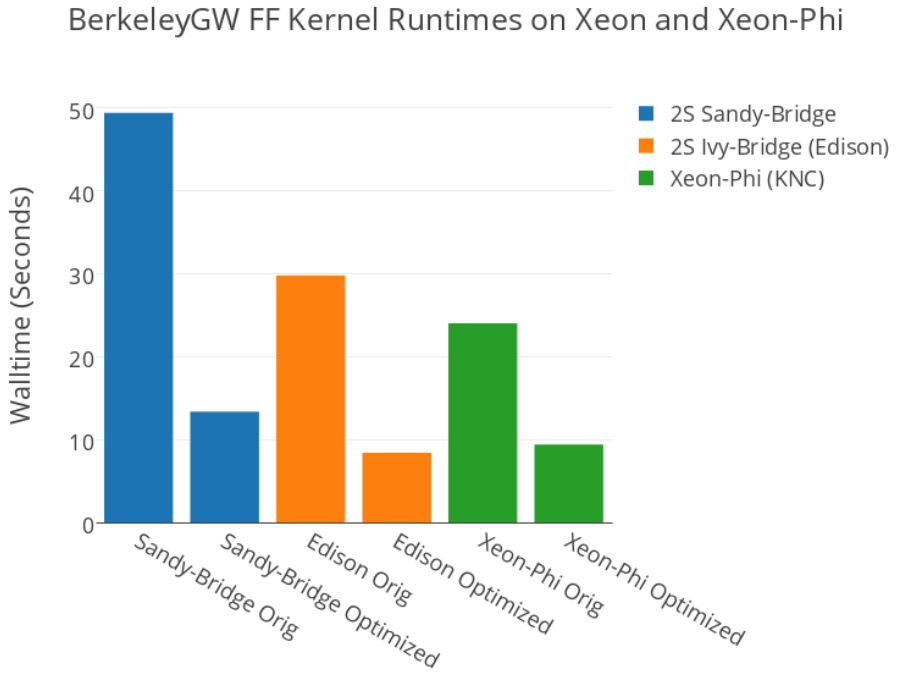
NESAP applications represent a large fraction of NERSC workload

Breakdown of Application Hours on Hopper and Edison 2013



NESAP (COE + Dungeon) Advances





	Overall Improvement	Notes
BGW GPP Kernel	0-10%	Optimized to begin with. Thread scalability improved
BGW FF Kernel	2x-4x	Unoptimized to begin with. Cache reuse improvements
BGW Chi Kernel	10-30%	Moved threaded region outward in code
BGW BSE Kernel	10-50%	Created custom vector matmuls

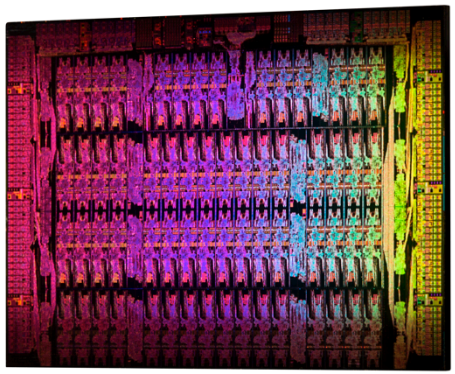
Summary of progress

- Many teams at the stage of profiling codes and extracting kernels
- 9 teams have held brainstorming sessions with with Cray and Intel
- 5-6 very advanced teams, operating mostly independently, already are ported to Knights Corner or have deep collaborations with Intel
- 1 Dungeon session completed
- Kernels in the hands of Cray and Intel staff for ~4 codes

Intel Xeon Phi Users' Group (IXPUG)

IXPUG provides a forum for the *free exchange* of information that enhances the *usability* and *efficiency* of scientific and technical applications using the Xeon Phi processor.

Charter Members
TACC
NERSC
Sandia/LANL
Zuse Inst. Berlin
Juelich
Intel



- Deep technical exchanges between Intel experts and leading HPC application developers.
- Started at TACC with two highly successful meetings in 2013 & 2014.
- Now expanding world-wide with leadership from DOE labs.
- 2015 meetings being planned for Berkeley & Berlin.

Great interest from the community

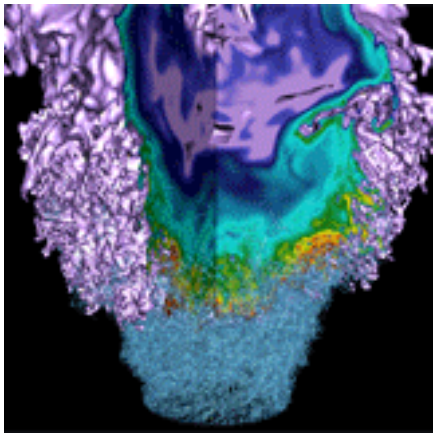
- BOF at SC14 drew 150+ attendees.
- Six "lightning talk" papers on performance tuning chosen from 30 proposals.

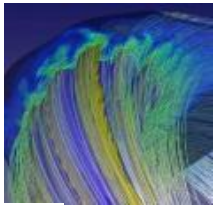
IXPUG is an independent organization of users and institutions interested in using the Phi for their scientific and technical applications.

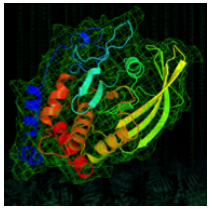
Some Initial Lessons Learned from Phase 1

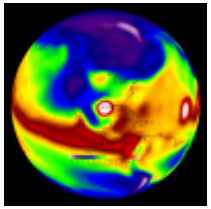
- Improving a code for advanced architectures can improve performance on traditional architectures.
- Inclusion of OpenMP may be needed just to get the code to run (or to fit within memory)
- Some codes may need significant rewrite or refactoring; others gain significantly just adding OpenMP and vectorization
- Profiling/debugging tools and optimized libraries will be essential
- Vectorization important for performance

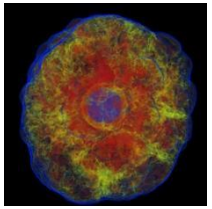
Thank you!

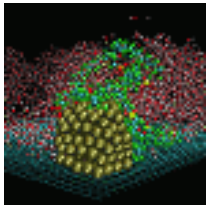












NERSC 40 YEARS at the FOREFRONT

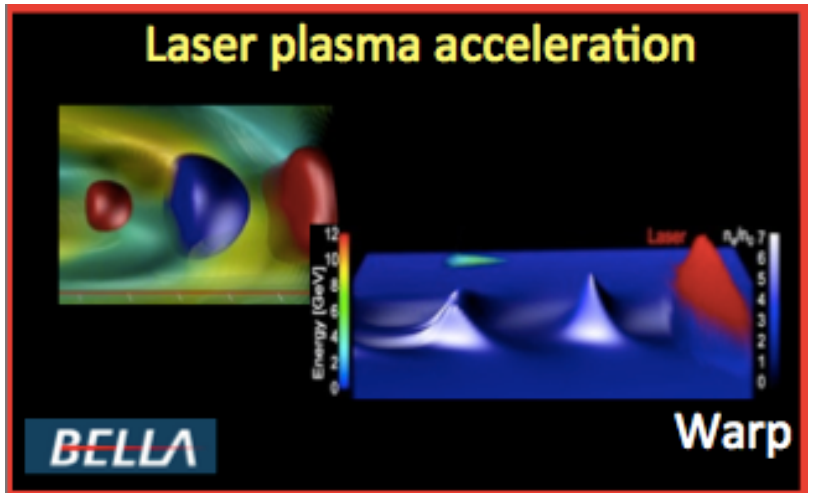
U.S. DEPARTMENT OF ENERGY | Office of Science

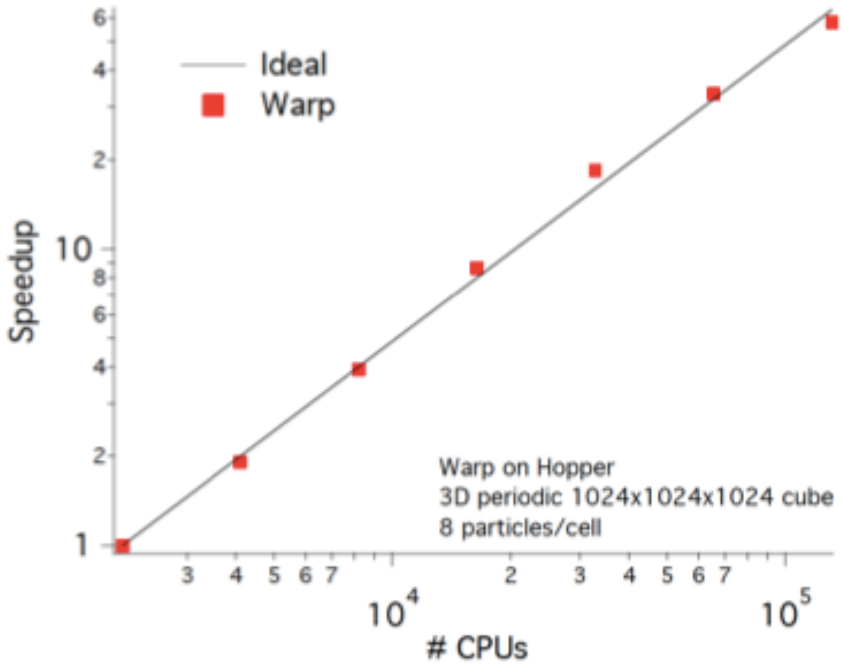
BERKELEY LAB
Lawrence Berkeley National Laboratory

WARP Code Status

- **WARP is a PIC code used for particle accelerator modeling for the generation, transport and neutralization of charged-beam particles**

- **It is a highly scalable code across many nodes**

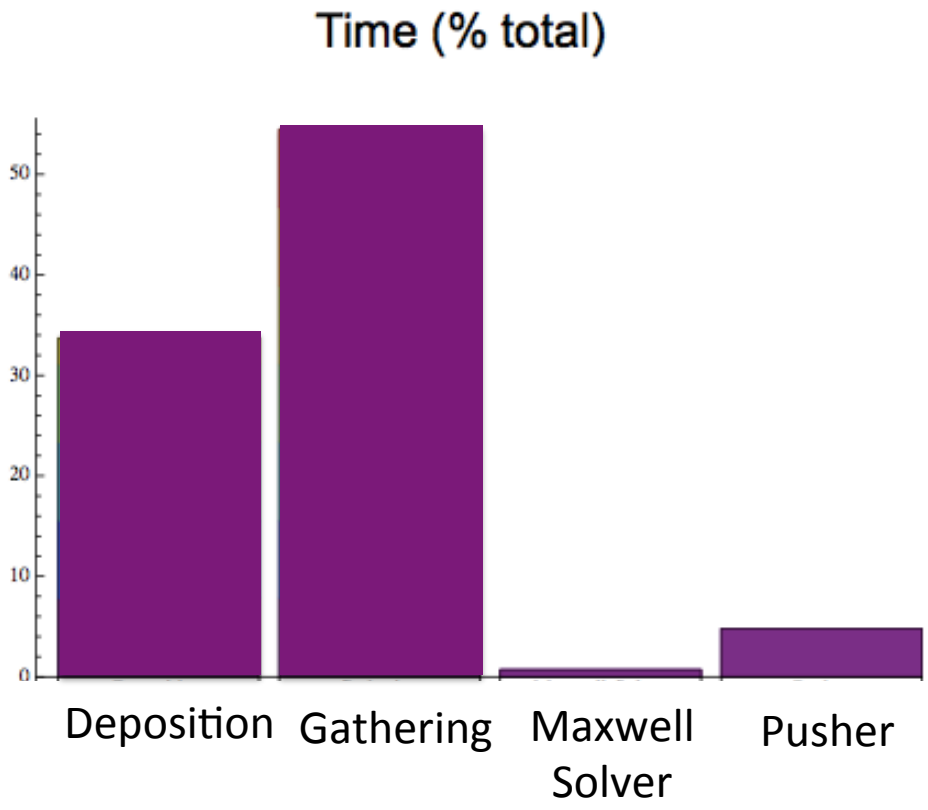


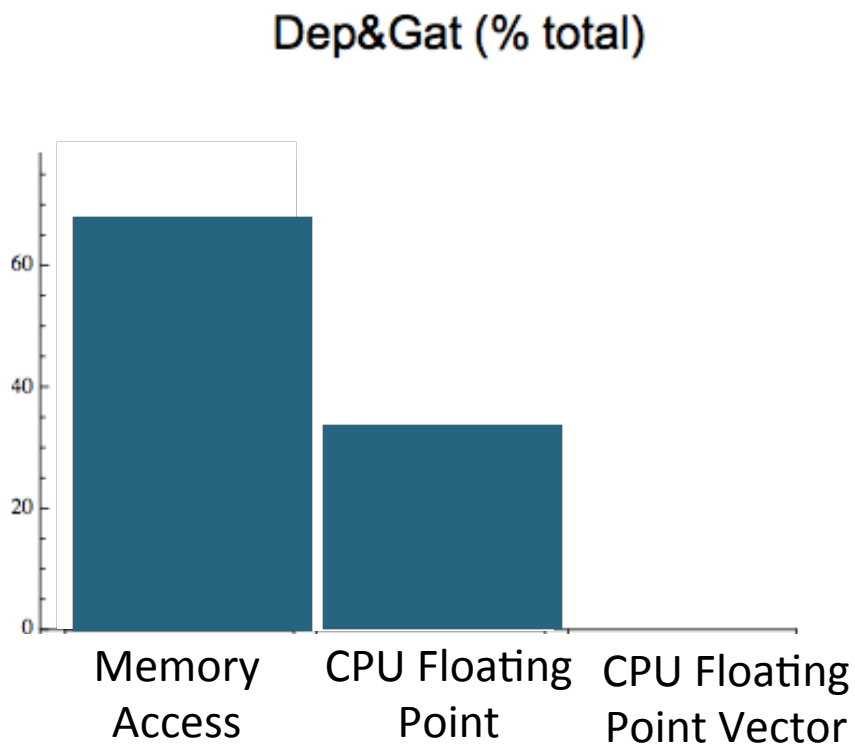


Images from JL Vay

WARP-Core Test Problem Profile

- **Majority of time in test problem spent in Deposition and Gathering routines – which spend a large % of time accessing memory with few loops vectorized**



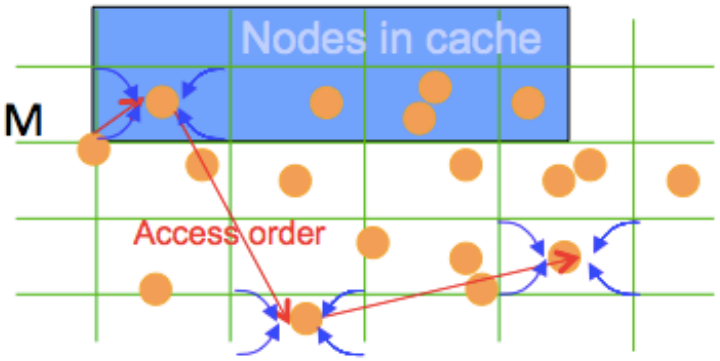


Slide data courtesy of Jean-Luc Vay, Henri Vincenti and team

WARP Hot Spots

- Similar to other PIC codes
- Memory access non-sequential, causing cache misses and increasing time spent accessing memory
- Many loops not vectorized due to flow dependencies and function calls

Ex : Field gathering process



Code structure

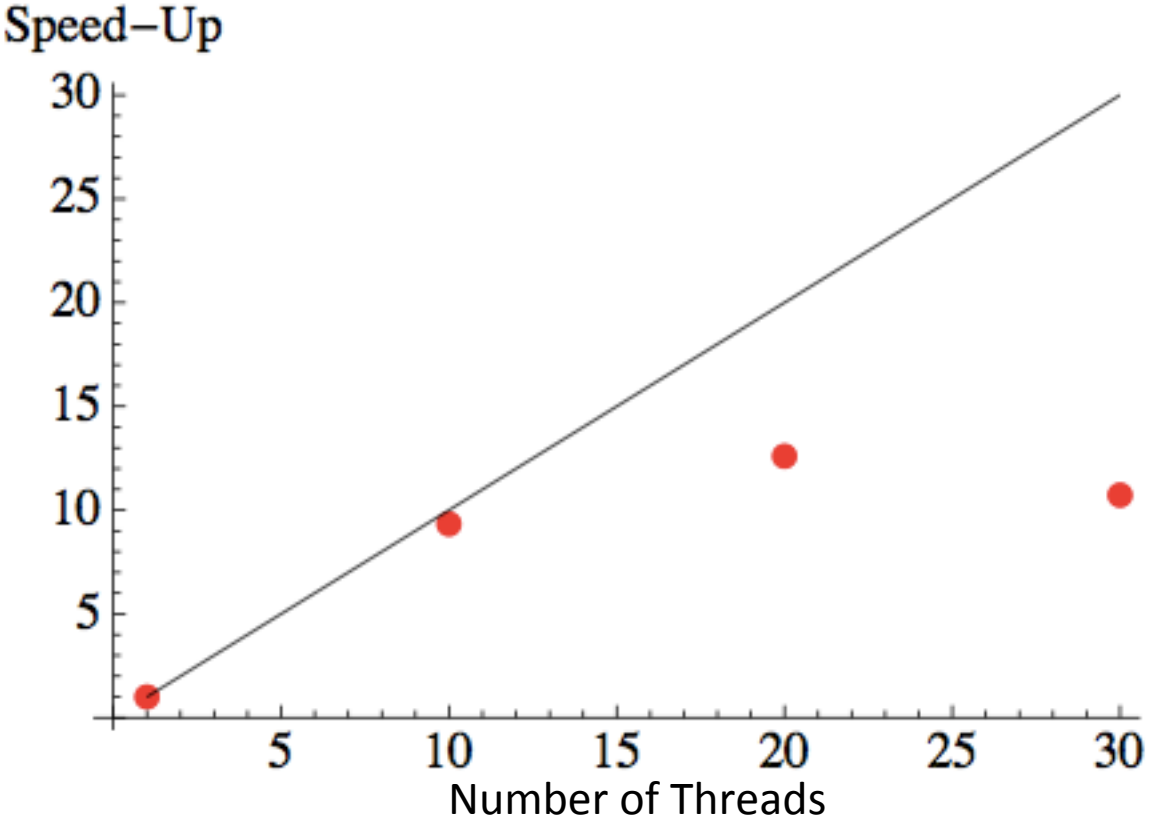
```
DO i=1,nparticles
  DO inode=1,N_adjacentnodes
          Field(particle)= Field(particle)+Weight*Field(inode)
  END DO
END DO
```


Slide data courtesy of Jean-Luc Vay, Henri Vincenti and team

WARP OpenMP Scalability

- **The baseline deposition routine shows good OpenMP Scalability up to about 10 threads**

Deposition Routine



Slide data courtesy of Jean-Luc Vay, Henri Vincenti and team

WARP Code directions and next steps

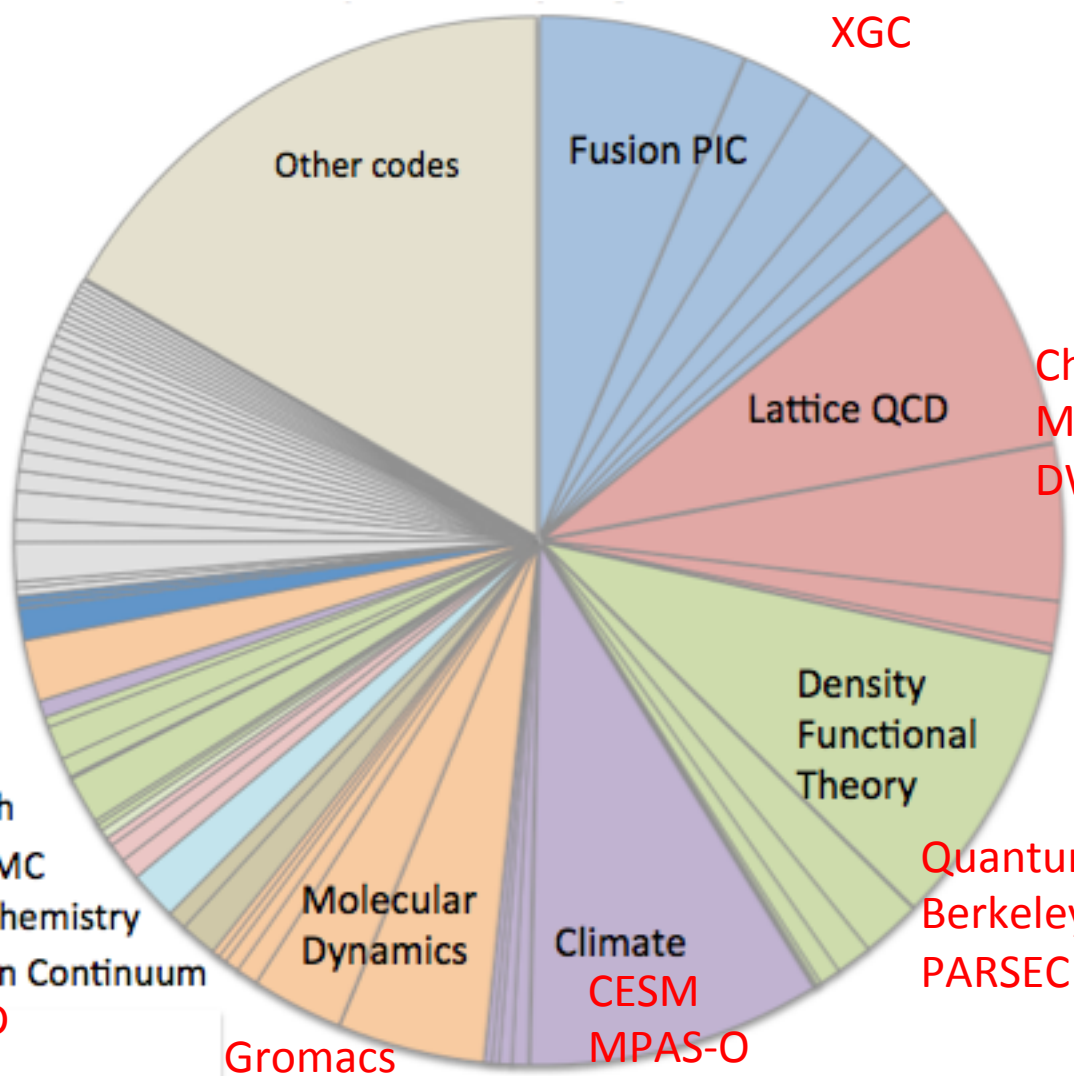
- **Possible directions**
  - Consider how particles could be sorted (or tiled) to increase data locality to fit in cache
  - Reorganize parts of code to improve vectorization
  - Because of almost perfect MPI scaling, consider if WARP code fit entirely into on-package memory
- **Next Steps**
  - Create 3D kernel, (current version is 2D)
  - Run more multi-threaded profiling tests

Slide data courtesy of Jean-Luc Vay, Henri Vincenti and team

NERSC's Key Challenges

- **Application Readiness**
  - We must prepare the broad user community for manycore architectures, not just a few codes
  - Will require deep collaboration with select code teams
  - Finding additional application parallelism is the main challenge
  - Unclear how to use on-package memory, as explicit memory or cache
- **Burst Buffer**
  - How to integrate and monitor in a production environment?
  - Which applications are best suited to use the Burst Buffer?
  - How to make the Burst Buffer user friendly
- **Integration into NERSC environment in CRT**
  - Mounting Cori file system across other systems, (Edison)
  - Integration into a new facility

NESAP Applications represent wide range of algorithms



XGC

Chroma
MILC
DWF

WARP
Meraculous

CASTRO
MAESTRO
Chombo-crunch

NWCHEM

M3D

Gromacs

Quantum Espresso
BerkeleyGW
PARSEC

CESM
MPAS-O
ACME