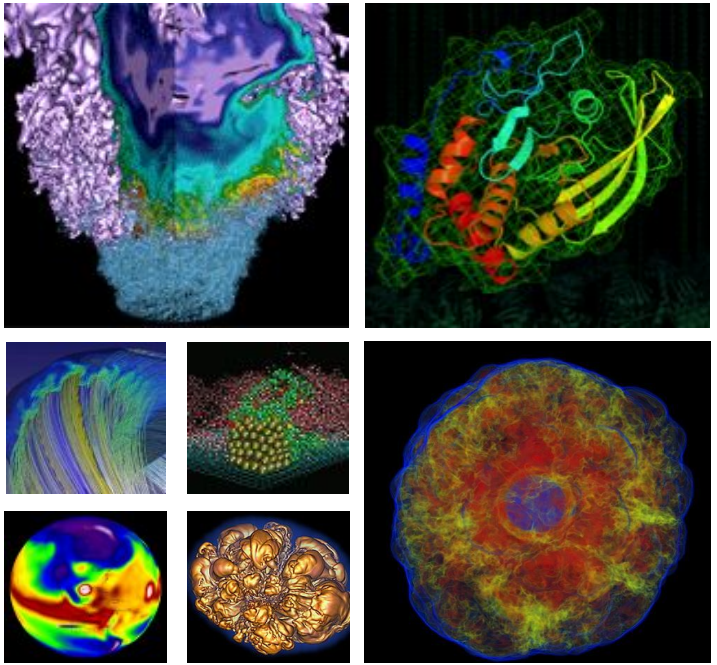
Enabling Applications for Cori KNL: NESAP



Helen He and Jack Deslippe

NUG 2017, 09/21/2017

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC's Challenge

- How to enable NERSC's diverse community of 7,000 users, 750 projects, and 700 codes to run on advanced architectures like Cori?

Cori KNL Node *vs.* Edison Node

Edison (Ivy-Bridge):

- 5500+ nodes
- 12 cores per socket
- 24 HW threads per socket
- 2.4 GHz
- 8 double precision operations per cycle
- 30 MB L3 cache (shared per socket)
- 64 GB DDR @100 GB/s

Cori (KNL):

- 9600+ nodes
- 68 physical cores per socket
- 272 HW threads per socket
- 1.4 GHz
- 32 double precision operations per cycle
- No L3 cache
- 16 GB of MCDRAM @450 GB/s
  96 GB of DDR memory @120 GB/s

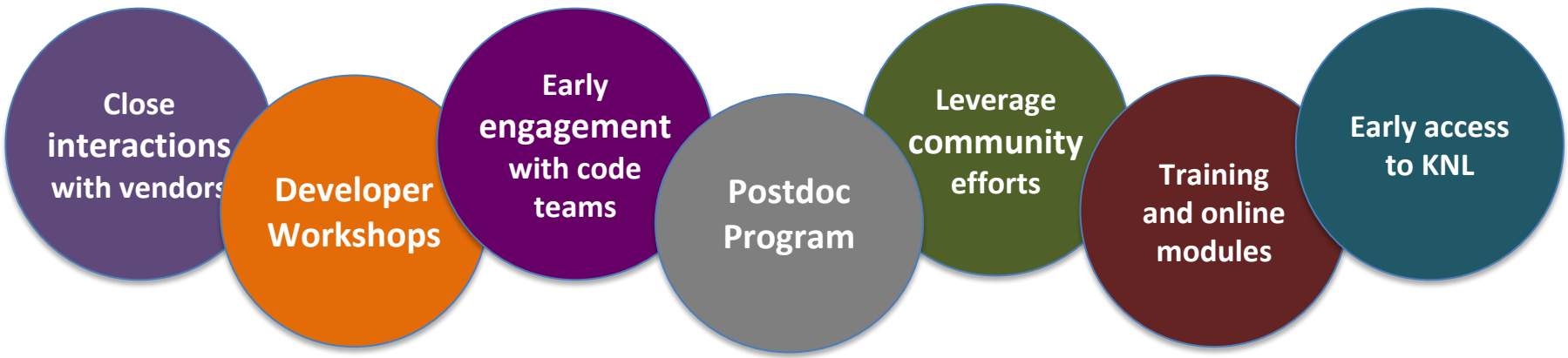
Preparing Users for KNL

- Out of the box performance on KNL is usually slower than on Haswell
- KNL has a number of features that offer opportunities to enhance performance
- For high performance, applications need to exploit thread scaling, vectorization, and on-board MCDRAM (high-bandwidth memory)
- NERSC recommends using MPI and OpenMP together to achieve thread and task scaling and maintain code portability
- **Our users told us they needed porting help**

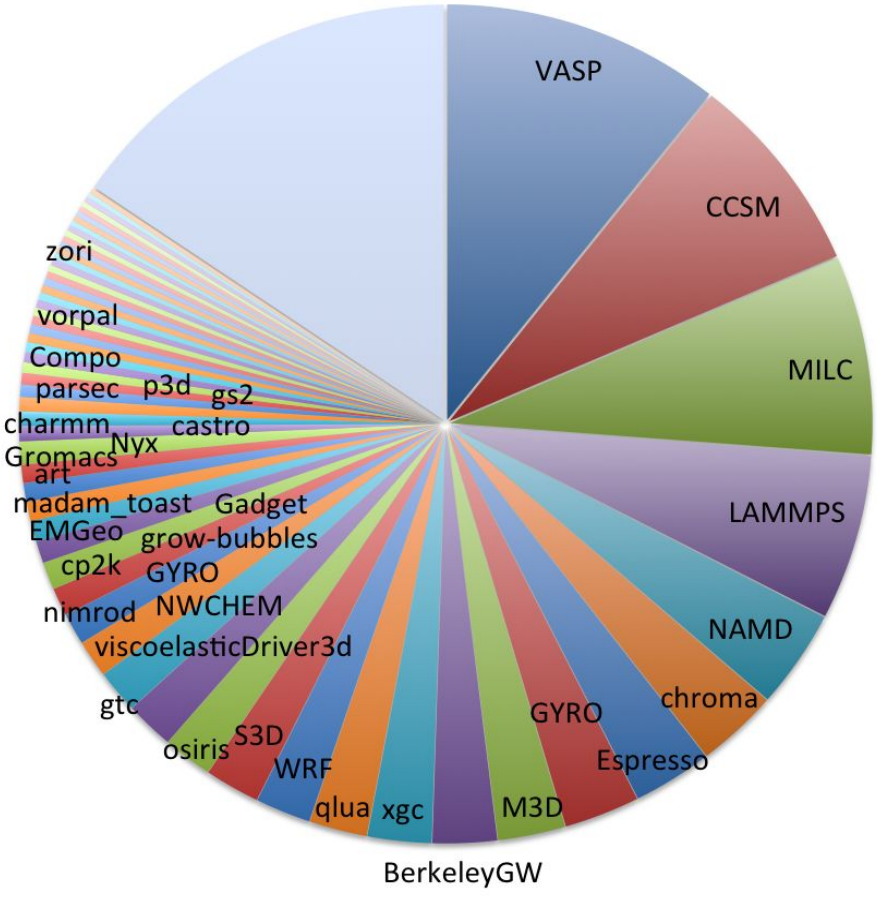
NERSC Exascale Scientific Application Program (NESAP)

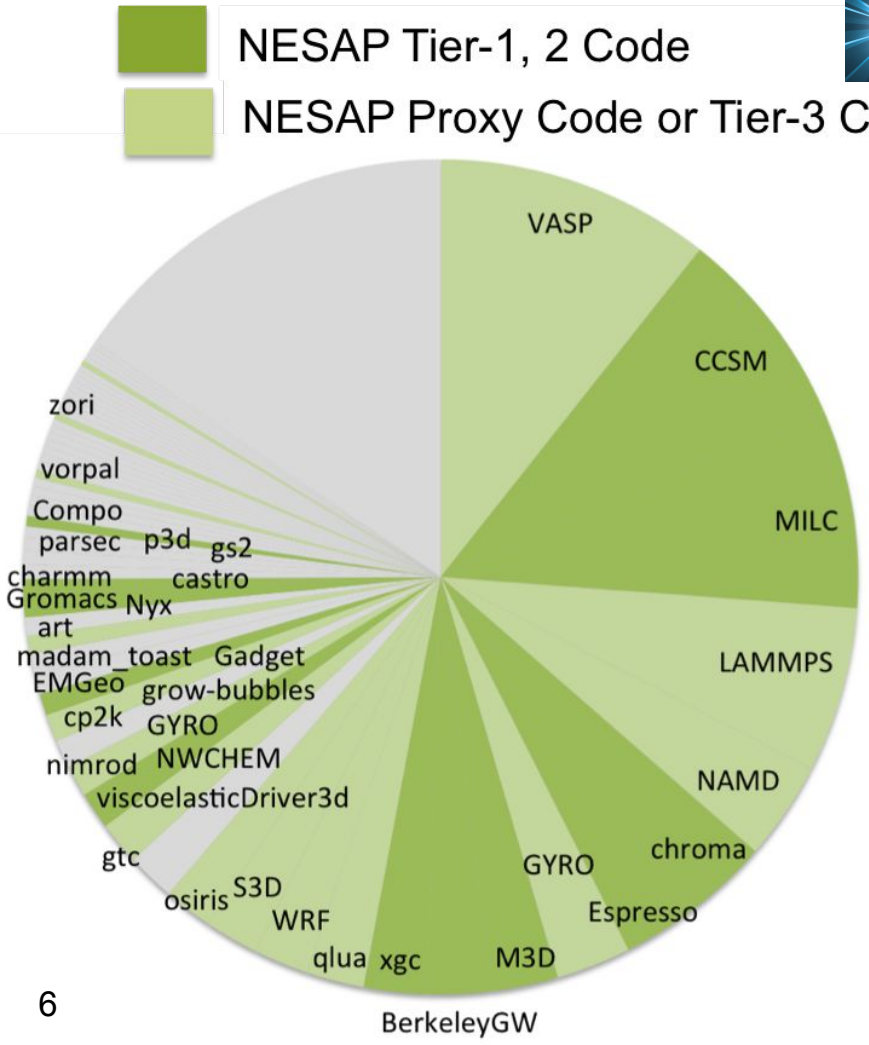
- Began in Fall 2014
- Goal: Prepare DOE Office of Science users for manycore
- Partner closely with ~20 application teams (and additional 20 teams at lower level) and apply lessons learned to broad NERSC user community.
- These 20 codes represent ~50% of NERSC hours used

NESAP activities include:



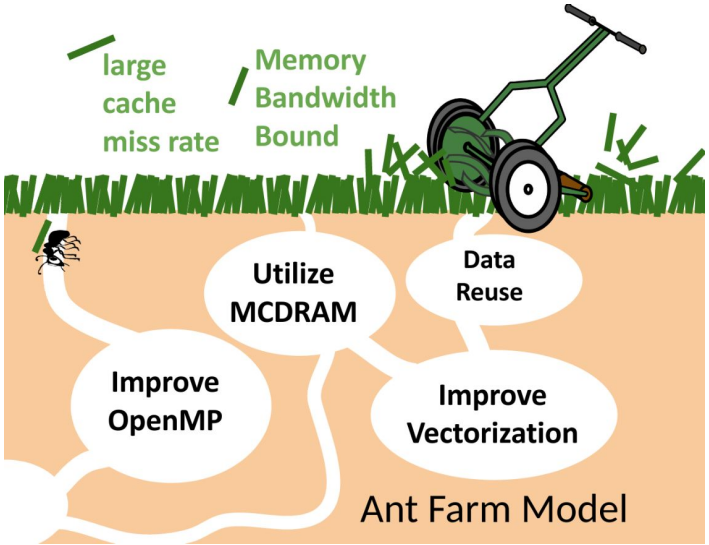
Breakdown of Application Hours Edison at Start of NESAP





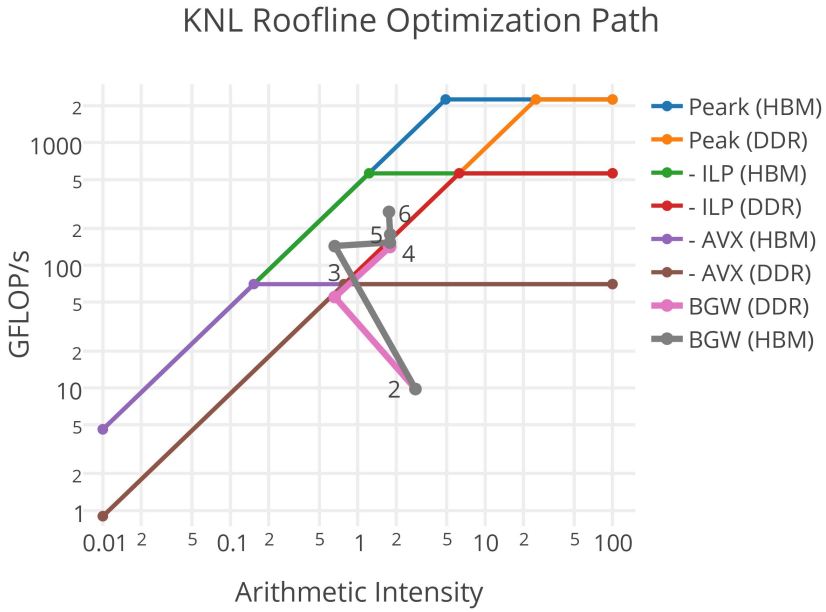
Optimization Challenge

- Energy-efficient processors have multiple hardware features to optimize against.
  - Many (heterogeneous) cores
  - Bigger vectors
  - New Instruction Set Architecture (ISA)
  - Multiple memory tiers
- It is easy for users to get bogged down in the weeds:
  - How do you know what KNL hardware feature to target?
  - How do you know how your code performs in an absolute sense and when to stop?



Optimization Strategy and Tools

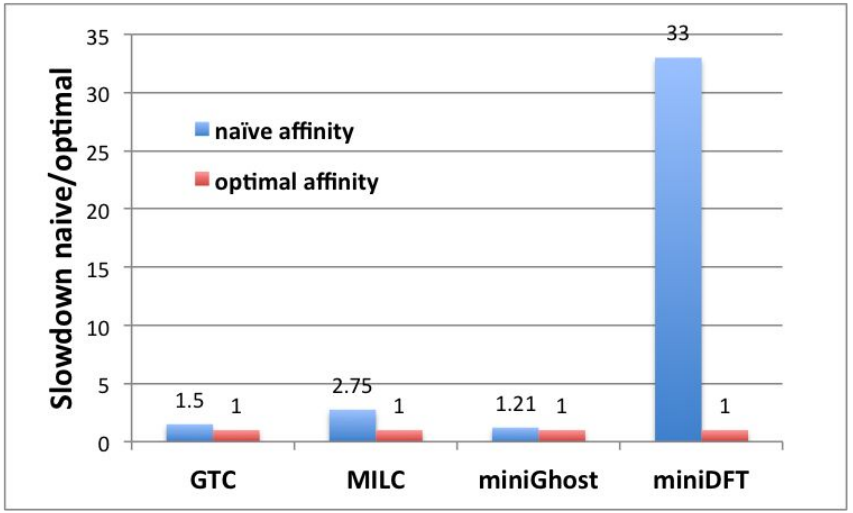
- Cori KNL uses same Aries interconnect and dragonfly topology as Edison and Cori Haswell
- **Focus on single-node KNL optimization**
- Use roofline as an optimization guide
  - Understand the theoretical peak
  - Guidance for effectiveness of bandwidth or CPU optimization
- Data collection with Intel VTune, SDE, and Vector Advisor tools



2 - addition of OpenMP
3 - loop reordering for vector code generation
4 - cache blocking
5,6 - hyperthreading and refined vectorization

Running on KNL Efficiently is More Complicated

- Getting the optimal process and thread affinity is critical
- Core specialization
- Broadcasting executables
- Using memory modes
- Using hugepages



Affinity: “-c --cpu_bind” flags are essential

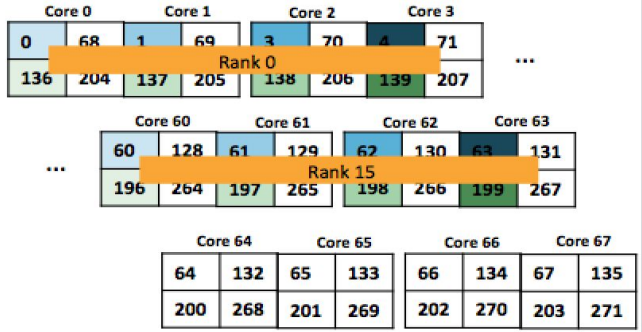
Sample job script to run under the quad,cache mode

Sample Job script (MPI+OpenMP)

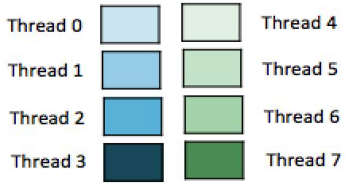
```
#!/bin/bash -l
#SBATCH –N 1
#SBATCH –p regular
#SBATCH –t 1:00:00
#SBATCH -C knl,quad,cache

export OMP_PROC_BIND=true
export OMP_PLACES=threads
export OMP_NUM_THREADS=8
srun –n16 –c16 --cpu_bind=cores ./a.out
```

Process affinity outcome



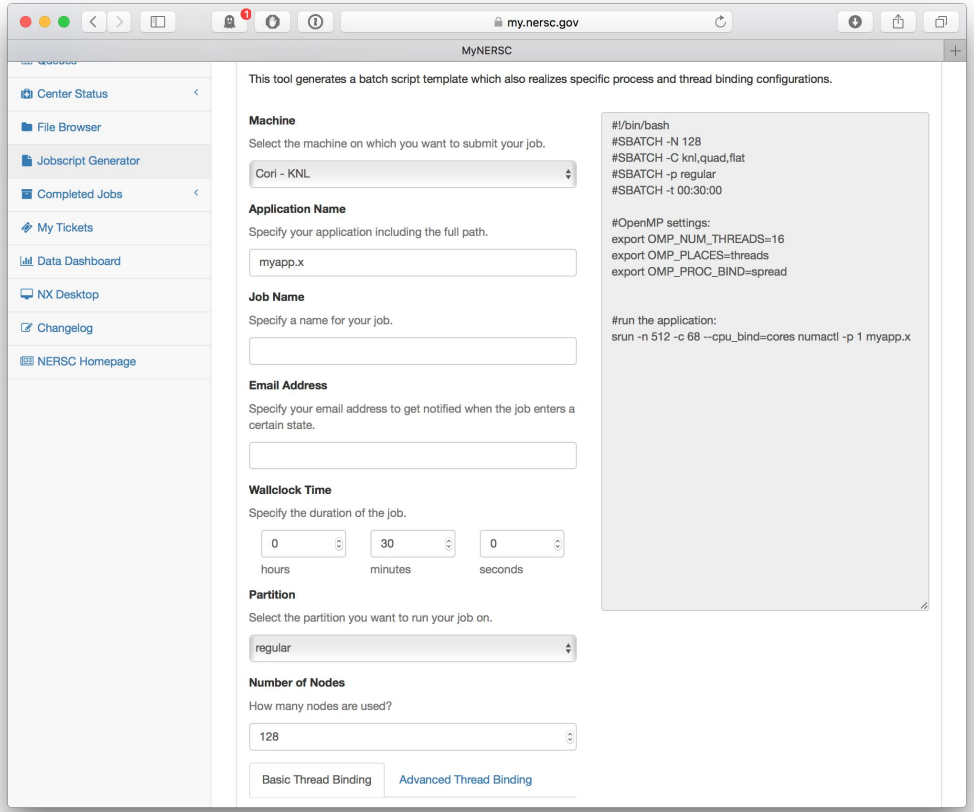
With the above two OpenMP envs, each thread is pinned to a single CPU on the cores allocated to the task. The resulting process/thread is shown in the right figure.



- 19 -

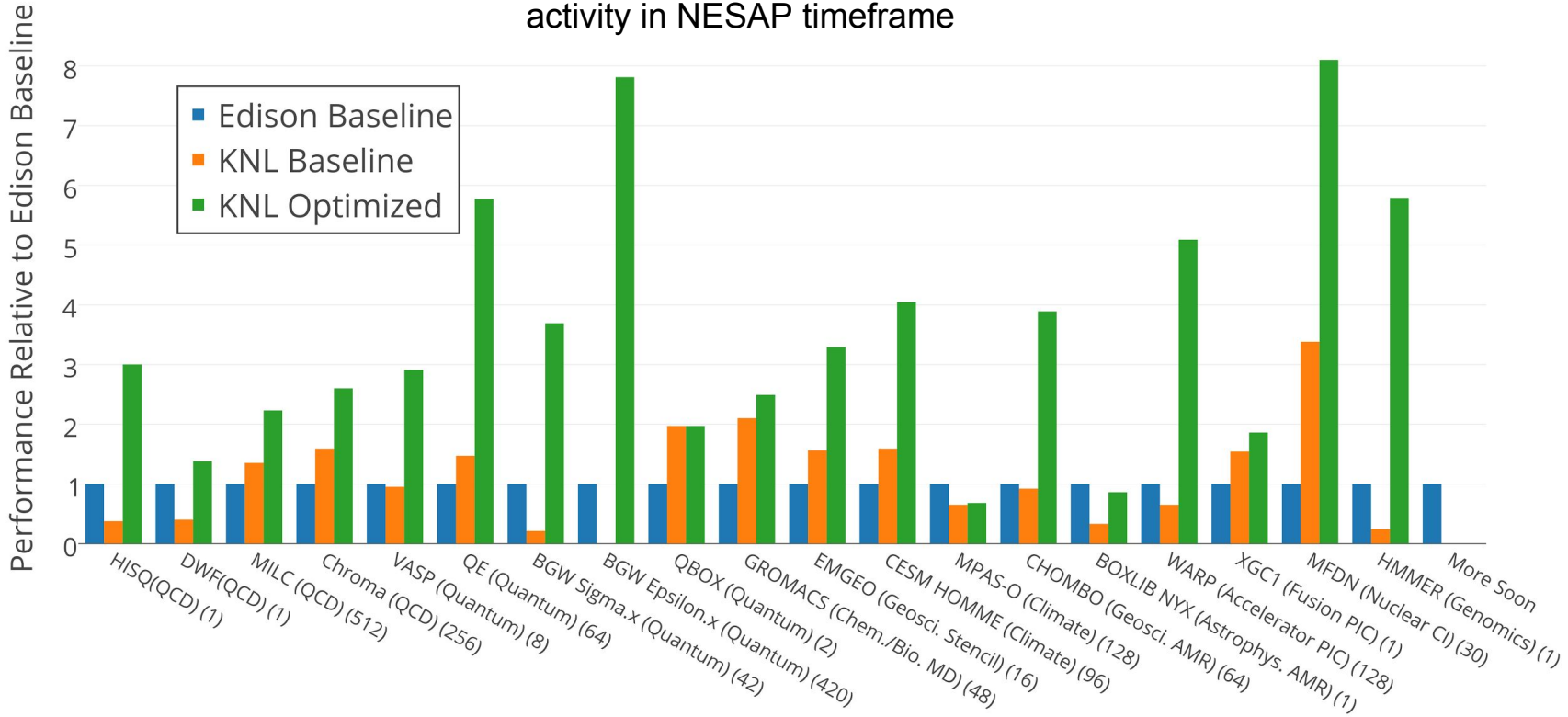
Job Script Generator Lowers Barrier to Entry

- Choices of Edison, Cori Haswell, and KNL
- Choices of KNL modes
- Hybrid MPI/OpenMP
- We also provide pre-built binaries for users to check affinity

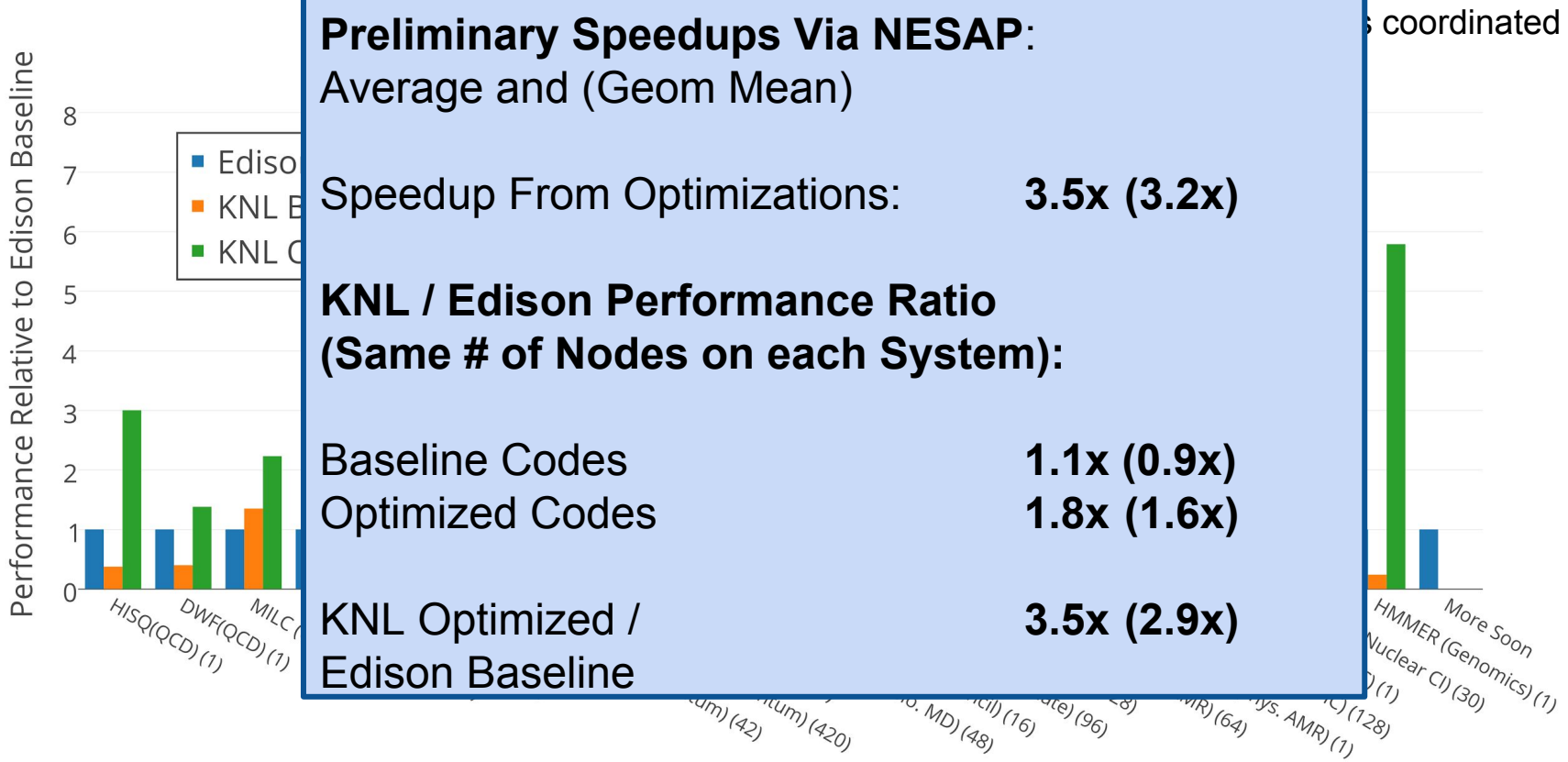


Preliminary NESAP Code Performance on KNL

*Speedups from direct/indirect NESAP efforts as well as coordinated activity in NESAP timeframe



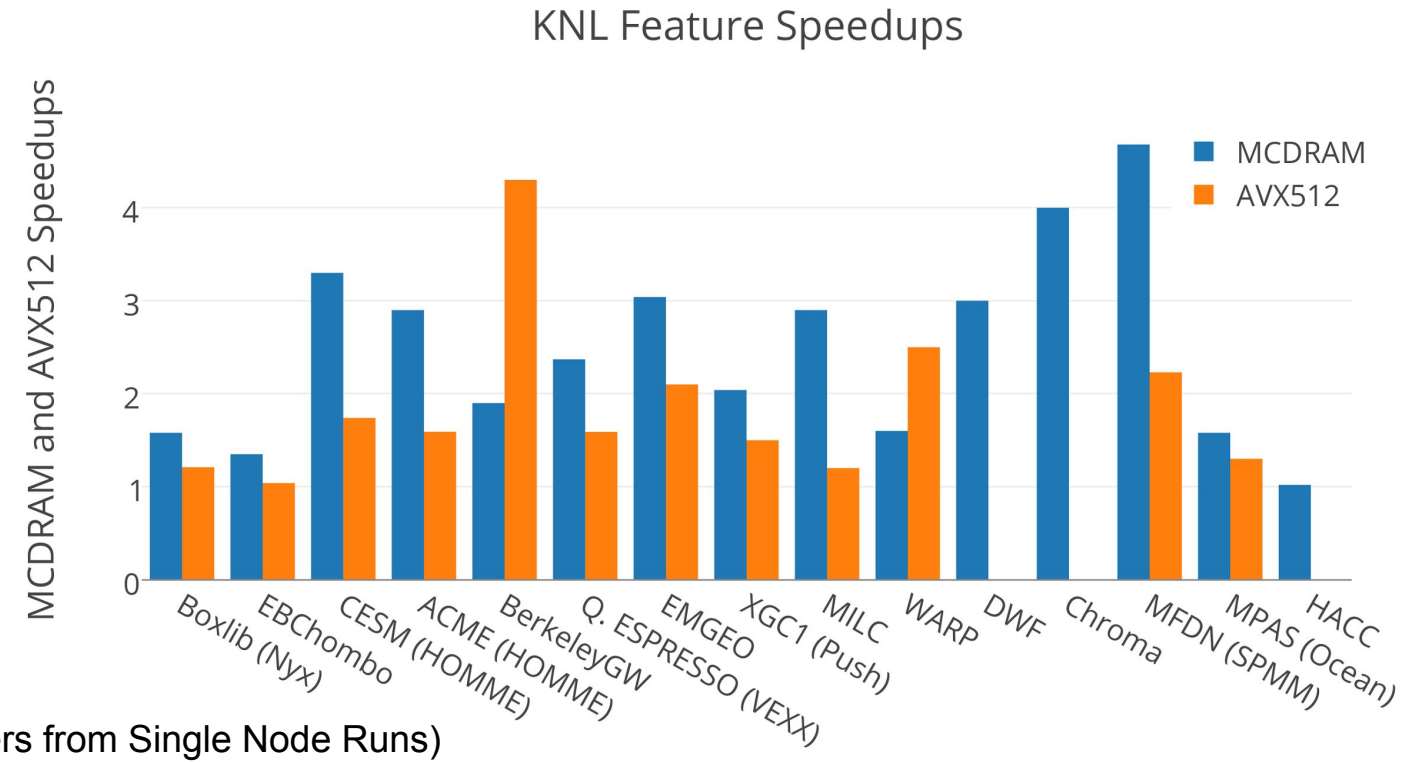
Preliminary NESAP Code Performance on KNL



Preliminary Speedups Via NESAP:
Average and (Geom Mean)

Speedup From Optimizations:	**3.5x (3.2x)**
KNL / Edison Performance Ratio (Same # of Nodes on each System):	
Baseline Codes	**1.1x (0.9x)**
Optimized Codes	**1.8x (1.6x)**
KNL Optimized / Edison Baseline	**3.5x (2.9x)**

Speedups from KNL Architecture Features



(Numbers from Single Node Runs)

Application Readiness Activities

- 14 Dungeon Sessions Charged
  - 3 Hack-a-Thons
- 13 NERSC Led Training Sessions (some multi-day)
- 9 Vendor Led Training Sessions
- 20+ Papers/Chapters in Publication (http://goo.gl/0NfGnd)
- Many external presentations: ISC, SC, APS, IXPUG, DOE Portability Workshop

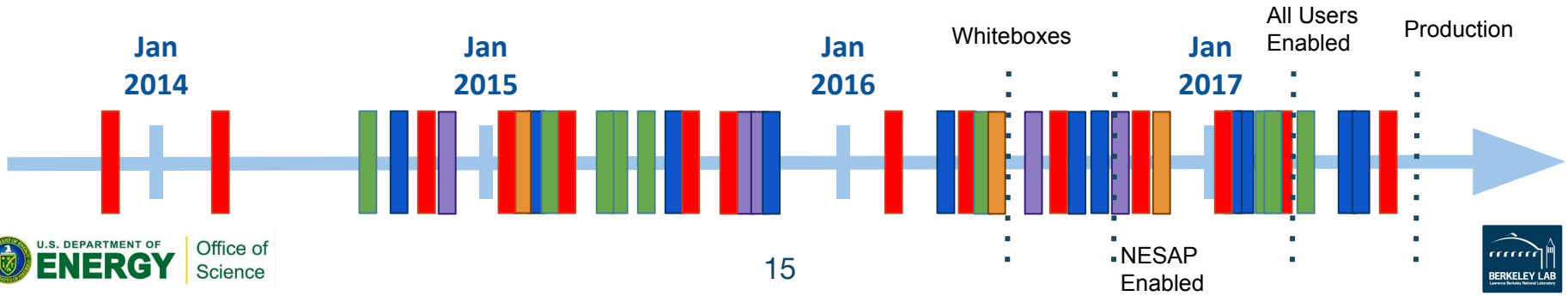
NERSC Perf.Training

Vendor/Tools Training

Dungeon Session

Hackathon

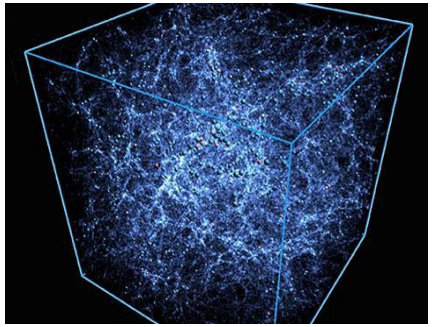
Community Workshops (IXPUG, C++, Mat. Sci., Accelerators)



What Works Well (and We Will Continue)

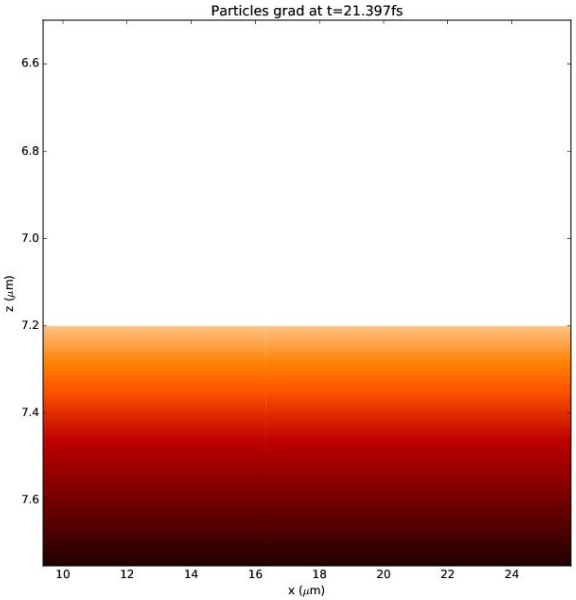
- Stimulate direct interactions between application teams and vendors (example: dungeon sessions and workshops).
- Influence vendor development of software features.
- Document extensively of lessons learned about tools and architectures and performance case studies.
- Continue the Postdoc program for the success of NESAP teams. Train new generations, with positive return to entire community.
- Engage the general user community, help more users and code teams through outreach, training, and incentive programs such as the Large Scale Science Program.

Optimizations Pay Off: Sample Science Stories you will hear later in the day!

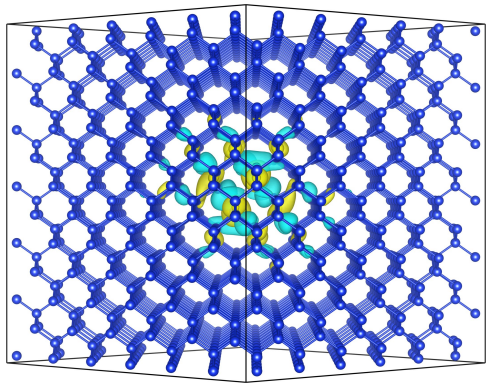


Galaxies Cosmology Scaling Runs

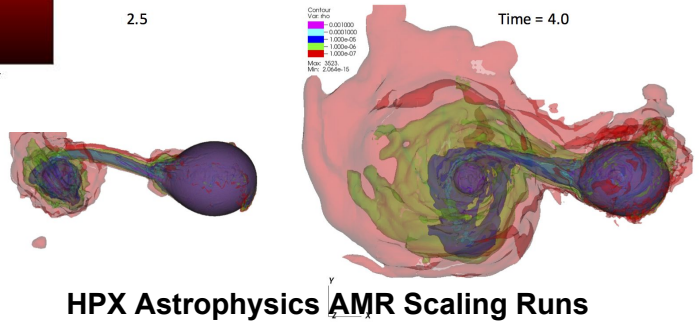
Deep Learning for Climate



Large Scale Particle in Cell Plasma Simulations



Materials Properties Scaling Runs



HPX Astrophysics AMR Scaling Runs

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB