Preparing NERSC Users for Cori, a Cray XC40 System with Intel Many Integrated Cores

Yun (Helen) He[1], Brandon Cook[1], Jack Deslippe[1], Brian Friesen[1], Richard Gerber[1], Rebecca Hartman-Baker[1], Alice Koniges[1], Thorsten Kurth[1], Stephen Leak[1], Woo-Sun Yang[1], Zhengji Zhao[1], Eddie Baron[2], Peter Hauschildt[3]

1 NERSC/Lawrence Berkeley National Laboratory
2 University of Oklahoma
3 Hamburger Sternwarte

Cray User Group Meeting 2017, Redmond, WA, May 7-11

Outline

- NERSC and Cori
- NESAP
- Heterogeneous Programming Environment
- Recommendations for Jobs
- Other Considerations
- System Issues Affecting Users
- Selected User Stories
- Summary

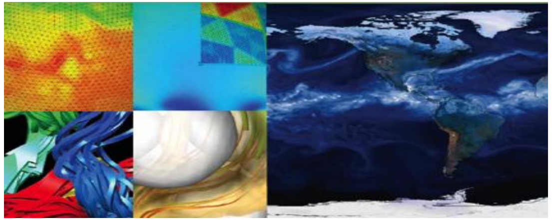
NERSC and Cori

NERSC: the Mission HPC Facility for DOE Office of Science Research

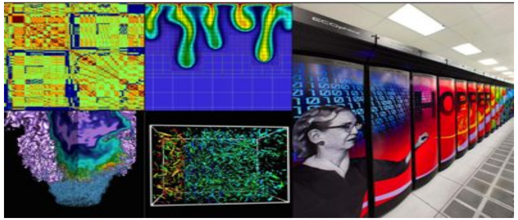
NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

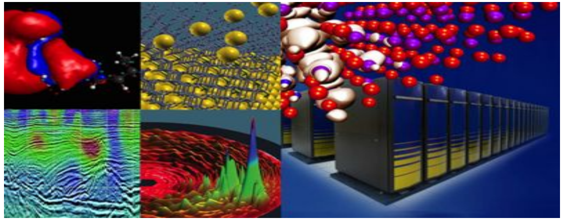
Largest funder of physical science research in U.S.



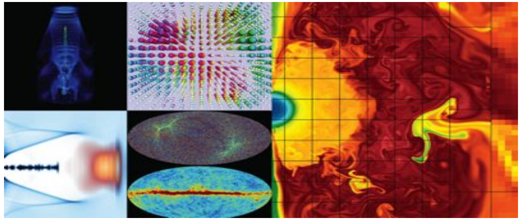
Bio Energy, Environment



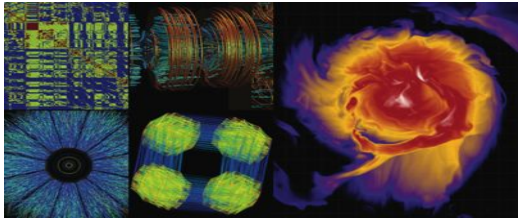
Computing



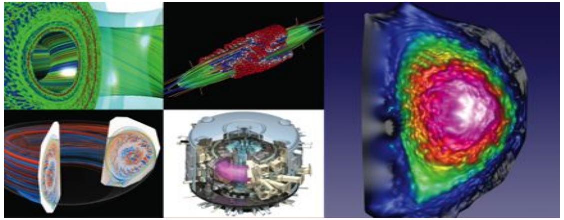
Materials, Chemistry, Geophysics



Particle Physics, Astrophysics



Nuclear Physics



Fusion Energy, Plasma Physics

6,000 users, 700 projects, 700 codes, 48 states, 40 countries, universities & national labs

The NERSC "Cori" System

- Cori will support the broad Office of Science research community & begin transitioning the workload to more energy-efficient architectures
- Cray XC system with **9,688 Intel Knights Landing** compute nodes
  - 68 cores per node, 4 hardware threads per core
  - Larger vector units (512 bits) with more complex instructions
  - 96 GB DRAM, 16 GB on-package MCDRAM
- **2,388 Intel Xeon Haswell** compute nodes: 32 cores/node
- Cori KNL nodes are integrated with Haswell nodes on Aries network as **one system**

Preparing Users for KNL

- A **simple recompile will yield lower performance** on KNL than on Haswell for most codes
- 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**

NESAP - NERSC Exascale Science Applications Program

NERSC Exascale Scientific Application Program (NESAP)

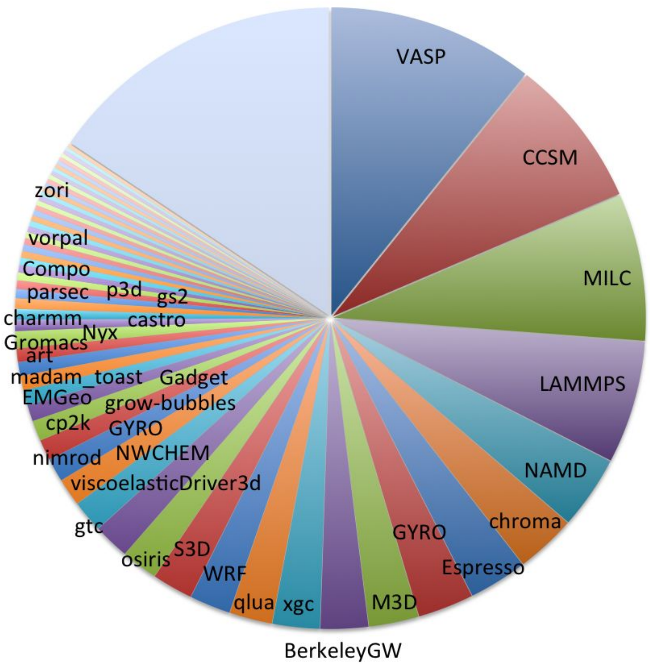
- Began in Fall 2014
- Goal: Prepare DOE Office of Science users for manycore
- Partner closely with ~20 application teams and apply lessons learned to broad NERSC user community

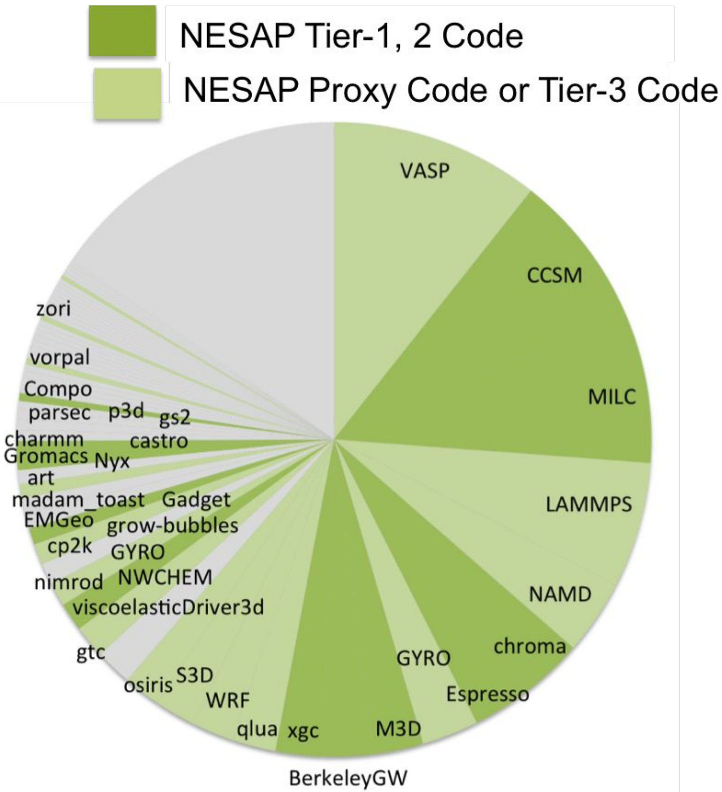
NESAP activities include:

- Close interactions with vendors
- Developer Workshops
- Early engagement with code teams
- Postdoc Program
- Leverage community efforts
- Training and online modules
- Early access to KNL

NESAP Code Coverage (~60% NERSC hours)

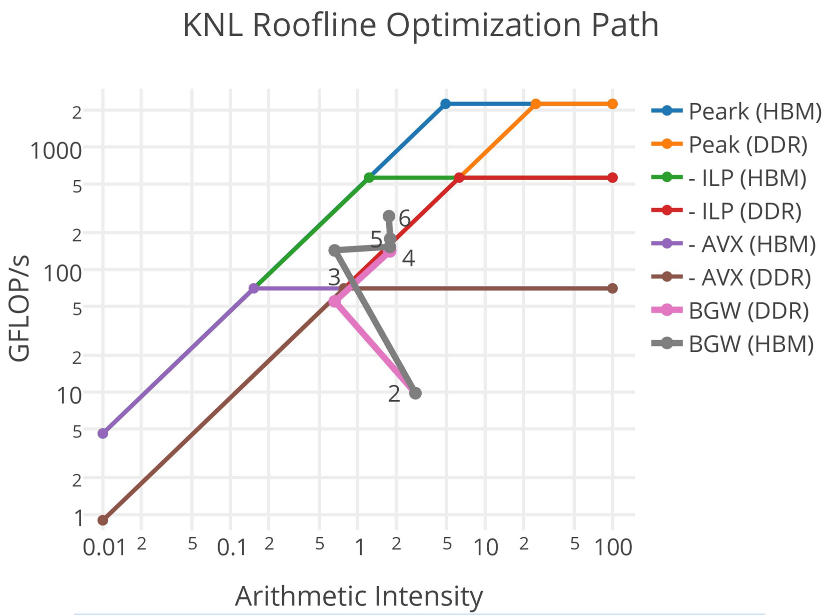
Breakdown of Application Hours
Edison at Start of NESAP





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

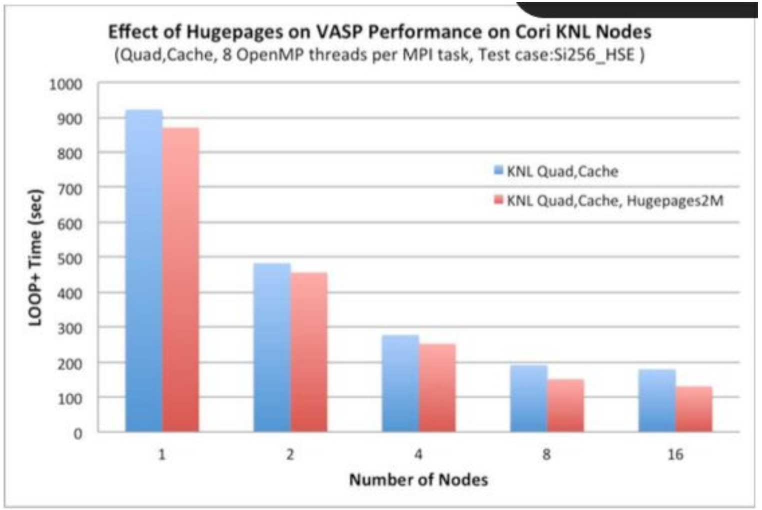


2 - addition of OpenMP
3 - loop reordering for vector code generation
4 - cache blocking
5,6 - hyperthreading and refined vectorization

Optimization Tips

- Hybrid MPI/OpenMP
  - and nested OpenMP (e.g. within MKL)
- Vectorization, locality more important than with Xeon
  - OpenMP SIMD directives
  - Reconsider loop nest order
  - Cache blocking (no L3 cache)
- Hyperthreading worth investigating
- Hugepages
  - Also best to reboot nodes often as hugepages availability will decrease over time (fragmentation)

VASP with Hugepages

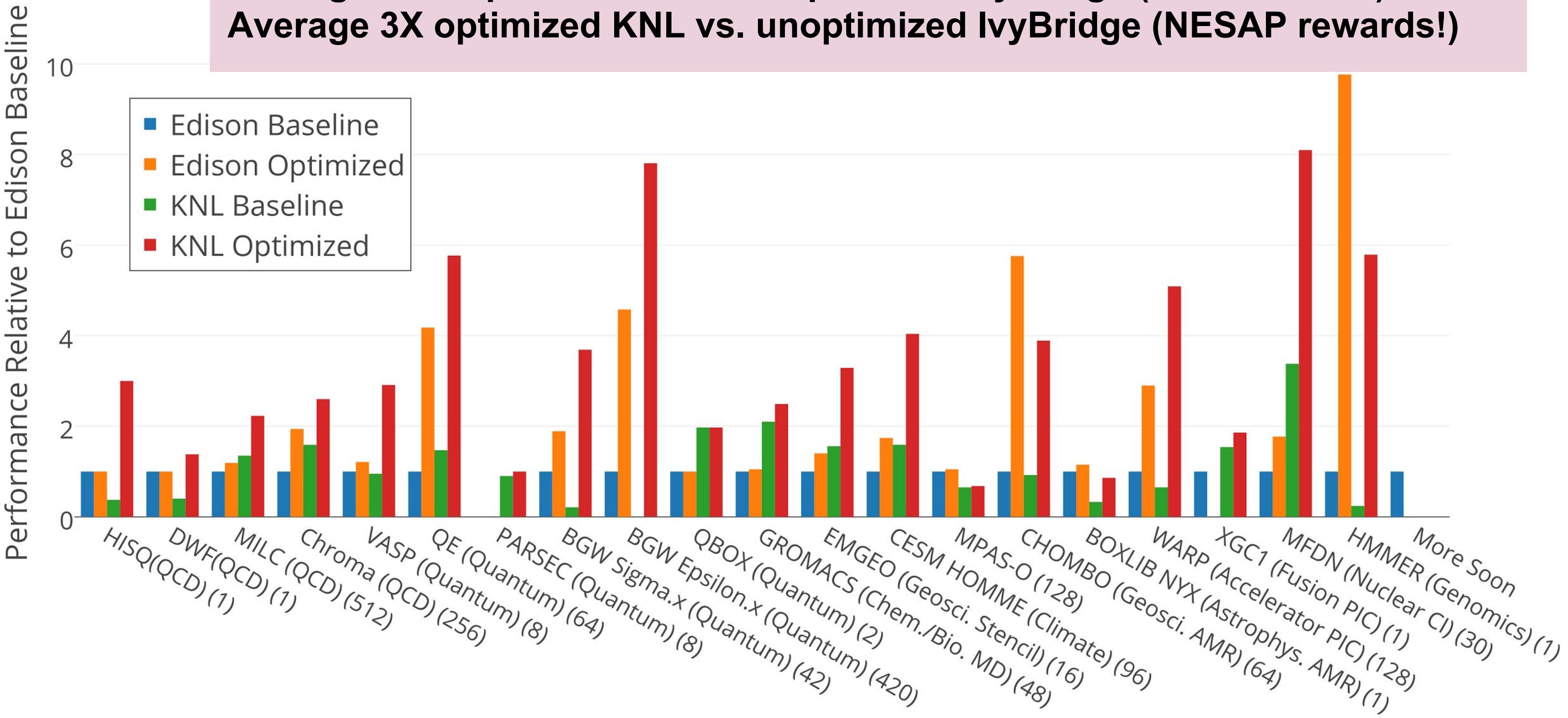


Training and Outreach

- Training and web pages are provided with examples, recipes, and general recommendations for running on KNL
- Participation at outreach events and community involvement such as IXPUG and OpenMP standard.

NESAP Results

Average 1.5X optimized KNL vs. optimized IvyBridge (node to node)
Average 3X optimized KNL vs. unoptimized IvyBridge (NESAP rewards!)



Heterogeneous (Xeon, KNL) Programming Environment

Building Software for a Heterogeneous System

- CLE6.0+ version required to support KNL
- Haswell binary can run on KNL, but not vice versa
- Only rebuild libraries and application packages that can take advantage of KNL architecture

Cross Compilation

- NERSC configuration: minimal OS image on compute nodes
  - compute nodes do not have full build environment
- **Applications are cross-compiled** from Haswell login nodes
  - % module swap craype-haswell craype-mic-knl
  - Note: can also build a binary to target both Haswell and KNL with "-axMIC-AVX512,CORE-AVX2" (Intel compilers)
- But configure scripts often need to run small test program:
- Solution: **configure for Haswell, build for KNL**
  - % module load craype-haswell
  - % ./configure CC=cc F77=ftn CXX=CC …
  - % module swap craype-haswell craype-mic-knl
  - % make

Recommendations for Jobs

Job Scripts Recommendations (1)

- Native Slurm is used to launch batch jobs on Cori
- NERSC recommendations for jobs on KNL:
  - Use Hybrid MPI/OpenMP
  - **Set process affinity explicitly** - default process and threading binding does not work well when using 64 (of available 68) cores in most cases
    - Use “-c” option for Slurm “srun” when launching jobs
    - Use OMP_PROC_BIND and OMP_PLACES
  - Use MCDRAM (numactl -m, srun --mem_bind=preferred,map_mem:1 …)
  - Use core specialization (#SBATCH -S 2)
  - **Use “sbcast” for large jobs**
- NERSC provides binaries that report affinity
  - Use these to check how your script options will affect affinity

Job Scripts Recommendations (2)

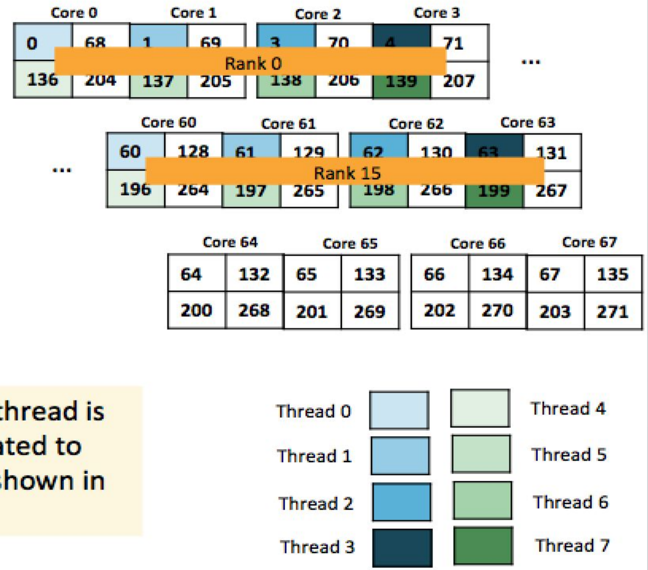
NERSC

Sample job script to run under the quad,cache mode

Sample Job script (MPI+OpenMP)

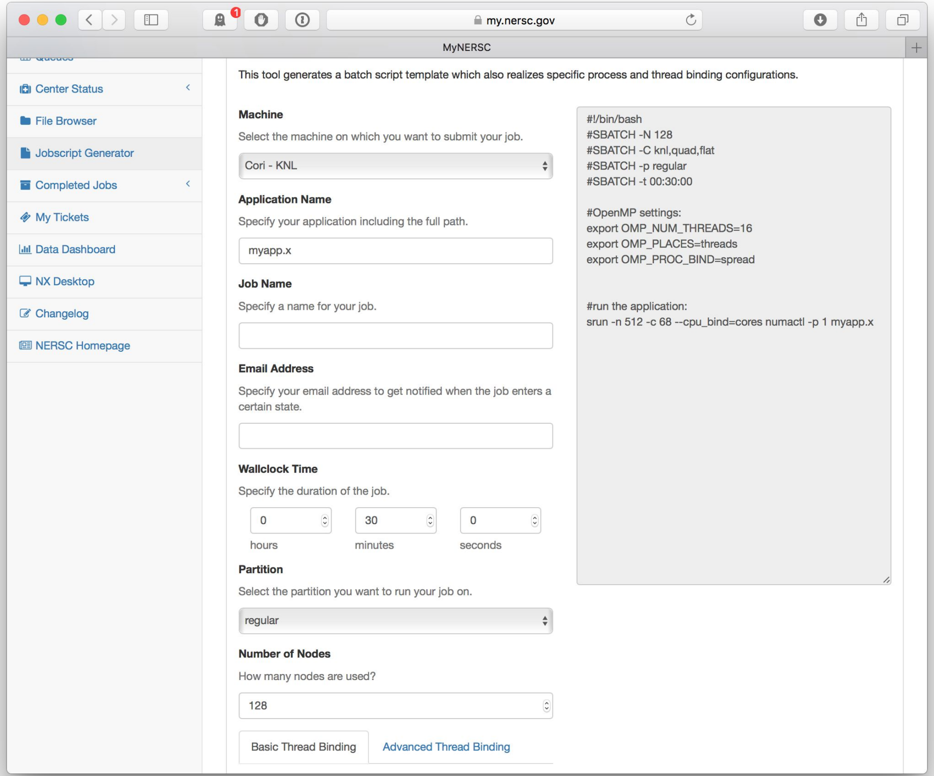
```
#!/bin/bash -l
#SBATCH –N 1
#SBATCH –p regular
#SBATCH –t 1:00:00
#SBATCH -C knl,quad,cache

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.

Job Scripts Generator



Other Considerations

KNL Default Mode Selections

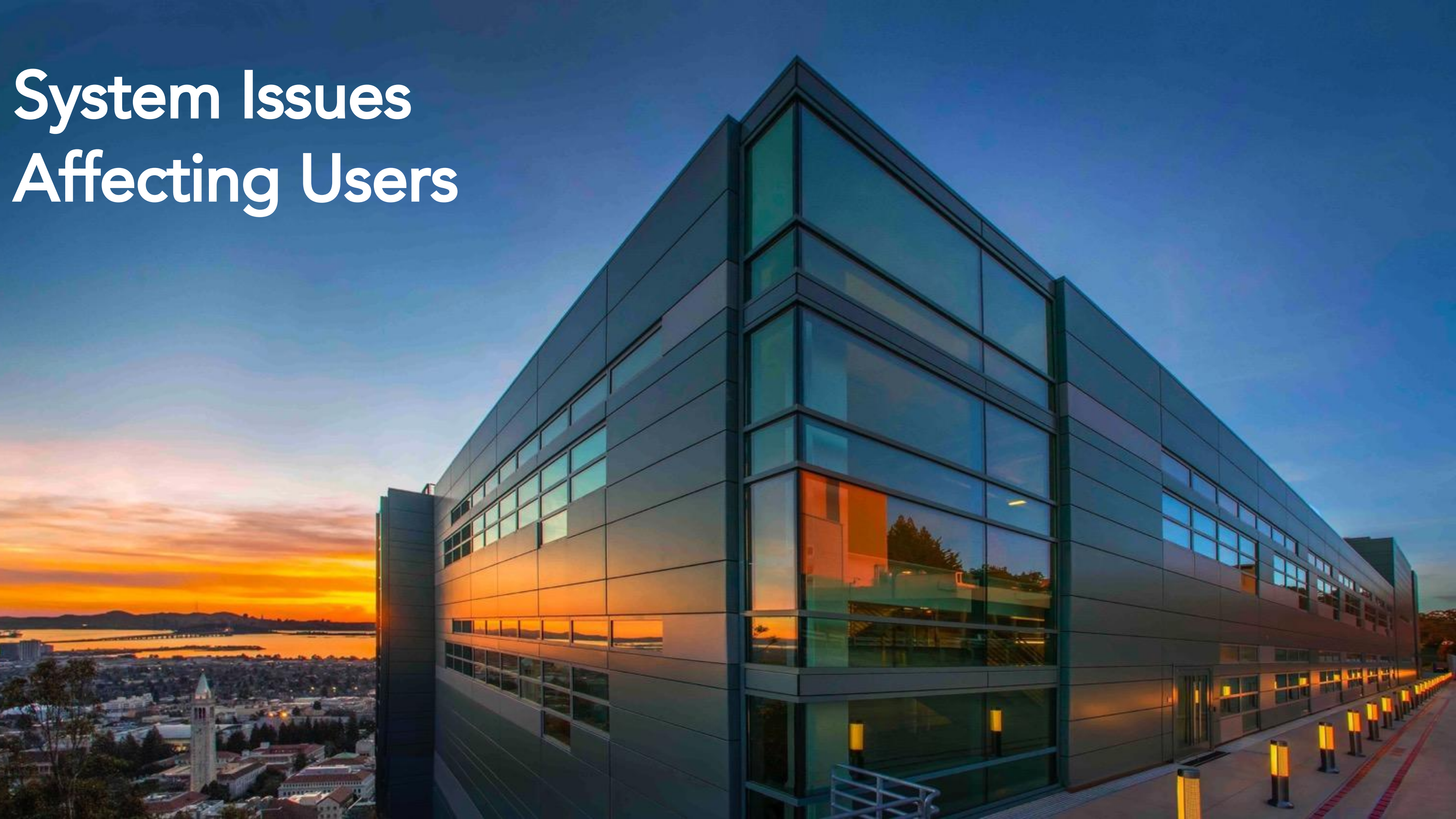
- KNL nodes can be booted with different clustering (quad, snc, etc), MCDRAM (flat, cache) .. which to use?
  - SNC adds significant complexity, minimal performance benefit
  - **Recommendation: “quad” mode as default**
- NERSC, APEX benchmarks, other applications mostly show <5% performance benefit of “flat” compared to “cache” MCDRAM mode
  - “quad,flat” offers a few percent speedup for memory footprint <16 GB
  - Flat mode (with >16GB/node) requires additional programming effort
- **We set 6,600+ nodes to “quad,cache”**, with mode changes disallowed
- **~3,000 nodes allowed to reboot** into new mode at job launch (by users)

KNLEAP (KNL Early Access Program) Gating

- Non-NESAP users given limited access to KNL nodes - up to 2 hrs, 512 nodes
- Users can apply to run larger or longer via KNLEAP
- **KNLEAP form guides users** through KNL features, application performance, testing process and thread topologies - not just blindly running same code on KNL

- Users need to provide time to solution with thread scaling runs.
- KNLEAP form provides live plotting

System Issues Affecting Users



Issues we have faced

- KNL Node Reboot Time (~ 40 min)
- High Speed Network (HSN) Quiesces
- Runtime Variations
- Cache conflicts due to direct-mapping of MCDRAM in cache mode - Zone Sort

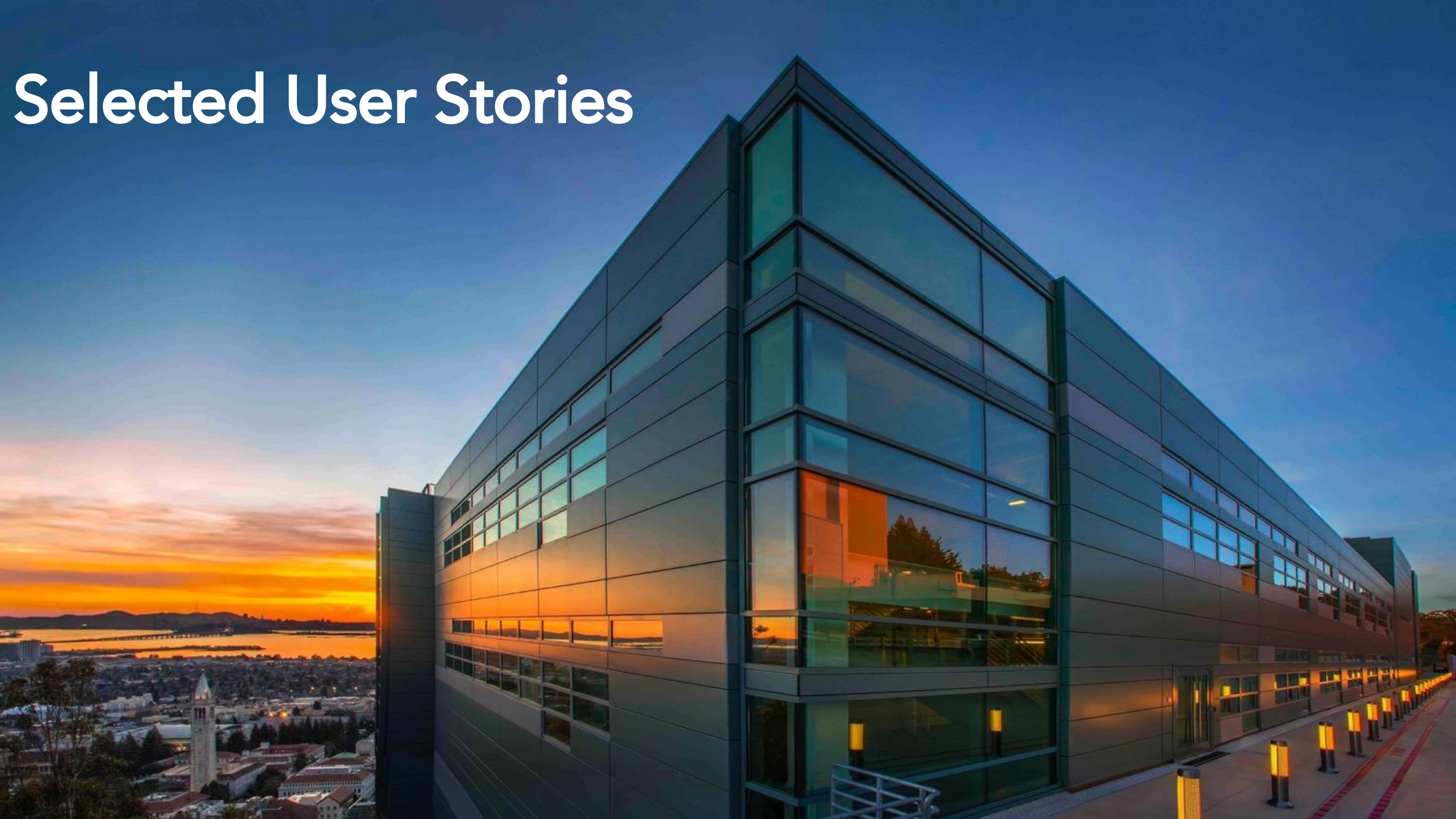
Runtime Variations

- Runtime variations of **30%** are seen even with perfect load balancing
- Some possible sources are
  - Network inteferences from other user jobs => **prime suspect**
  - MCDRAM in cache mode is direct-mapped cache
  - Turbo mode being turned on
  - HSN quiesces
  - File system IO load variations
  - More MPI synchronizations for larger jobs
- Cross-checking to find aggressive user applications or IO load (ongoing)
- Investigating different packet routing strategies and topology-aware scheduling
  - Slurm flag: --switch=N@HH:MM:SS
- **Not solved/understood yet**

Zone Sort

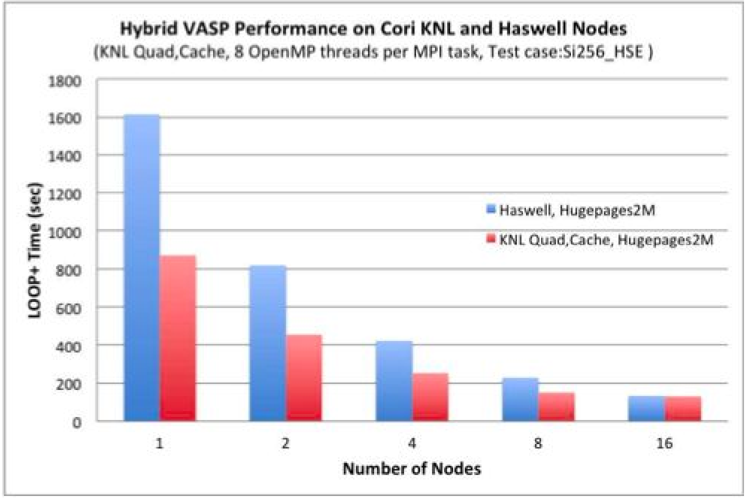
- MCDRAM in Cache mode is direct-mapped cache
  - Physical memory addresses modulo cache size map to same cache line
  - => so **cannot simultaneously be in cache**
- List of free memory pages fragments over time
  - Does not remain a small contiguous block
  - Increasing likelihood of mapping to same cache line
  - **Becomes worse as node stays up longer**
- “Zone Sort” kernel module from Intel sorts list of free pages
  - **We run it before each Slurm job step**
  - Users can also run this more frequently as wish
  - Appears to significantly reduce direct-mapped cache conflicts

Selected User Stories



VASP

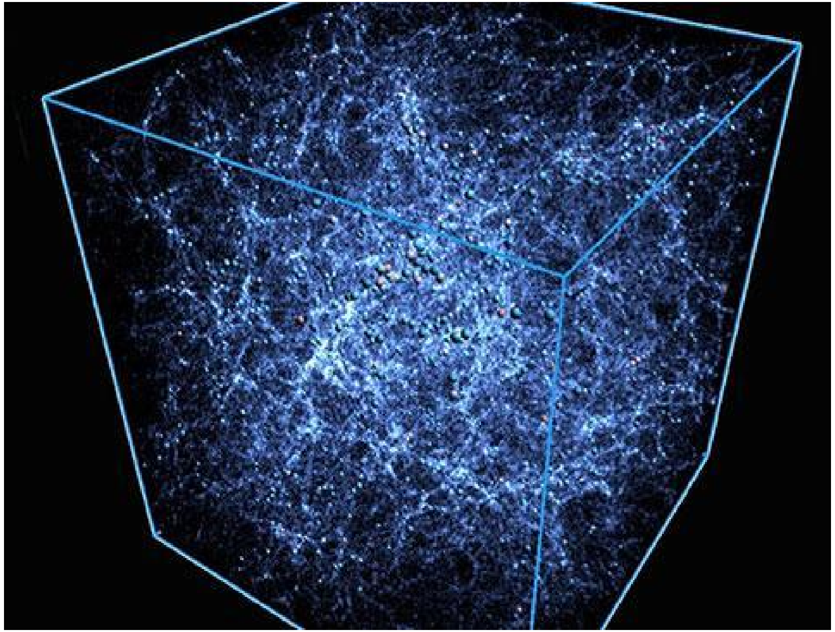
- Computational materials science code
- **Top ranking code at NERSC, >10% computing cycles, 850 registered users**
- Beta-testing program collaborating with code developers resulted in optimized hybrid MPI/OpenMP code now performing better on KNL than on Haswell
- NERSC provided comprehensive user guide; hosted VASP workshop
- Details in Zhao *et.al.* CUG2017 paper (Thursday PM, session 27A)



Cosmology Scaling Runs

3-Pt Correlation On 2B Galaxies Recently Completed on Cori

- NESAP For Data Prototype (Galactos)
- First anisotropic, 3-pt correlation computation on 2B Galaxies from Outer Rim Simulation
- Solves an open problem in cosmology for the next decade (LSST will observe 10B galaxies)
- Can address questions about the nature of dark-energy and gravity
- **Novel $O(N^2)$ algorithm** based on spherical harmonics for 3-pt correlation



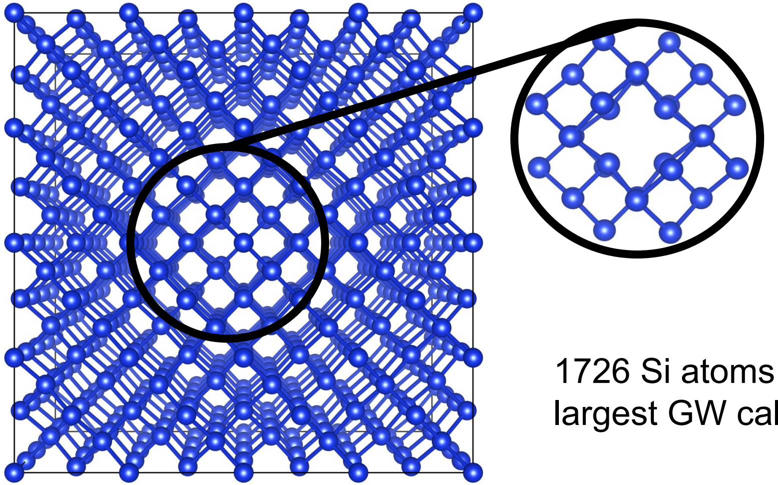
Scale:

- **9600+ KNL Nodes** (Significant Fraction of Peak)

Materials Properties Scaling Runs

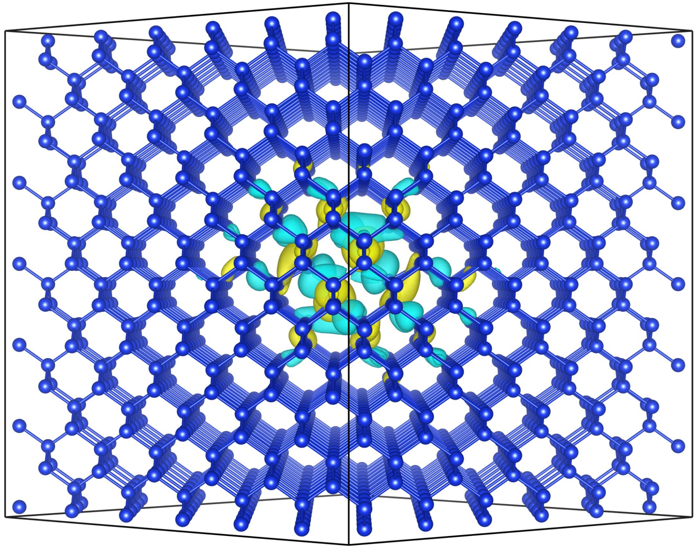
Defect States in Materials:
- Important material properties of, for examples, transistors and photovoltaics are often determined by the effects of defects.
- However, accurately studying defect properties require extremely large calculations to isolate defect states
- Featured in **one of 5 BES Material Software Centers**

Scale:
- Simulated on Cori with up to **9600 KNL Nodes**
- **Near Perfect Strong and Weak Scaling**.
- Large percentage of peak performance obtained (> 10 PFLOPS)



Schematic of di-vacancy defect and localized defect orbital in crystaline Silicon.

1726 Si atoms (~7K electrons) is largest GW calculation published



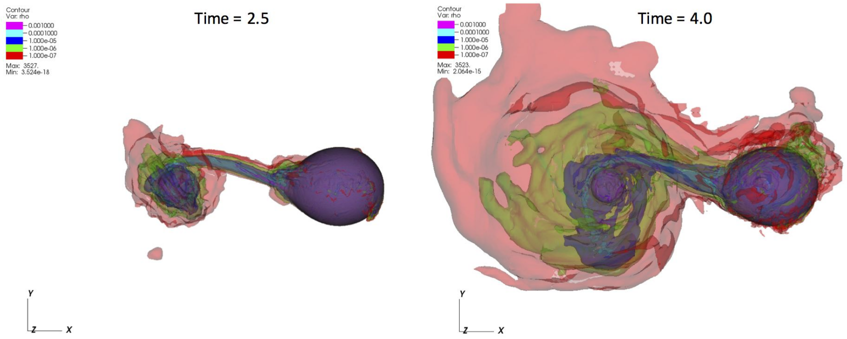
HPX Astrophysics AMR Scaling Runs

Cori can Support Billions of Light-Weight HPX Tasks:

- Modeled the creation of an accretion disk in a double-white-dwarf system which spans 17 orders of magnitude in astrophysical densities
- In-situ performance adaptation framework APEX enables irregular applications
- Same HPX C++ code, OctoTiger, runs on laptop to full Cori without significant changes or optimization
- Understanding binary systems and stellar mergers is critical to explaining dark matter/energy mysteries

Scale:

- Demonstrated the scalability of HPX asynchronous runtime on 655,520 Cori KNL cores (9,640 nodes)
- Achieved nearly perfect parallel efficiency



Images (density log-scale) show the early stages of the formation of the mass-transfer stream and accretion disk in a binary star system.

Summary

- Although most applications can run unmodified on KNL, users must explore thread scaling, vectorization, and MCDRAM usage to achieve optimal performance
- This paper summarized various aspects of the NERSC efforts in preparing users for the Cori KNL system
  - System configuration considerations
  - Programming
  - Performance optimization
  - User facing run time
- We hope lessons are helpful for other sites deploying a KNL system