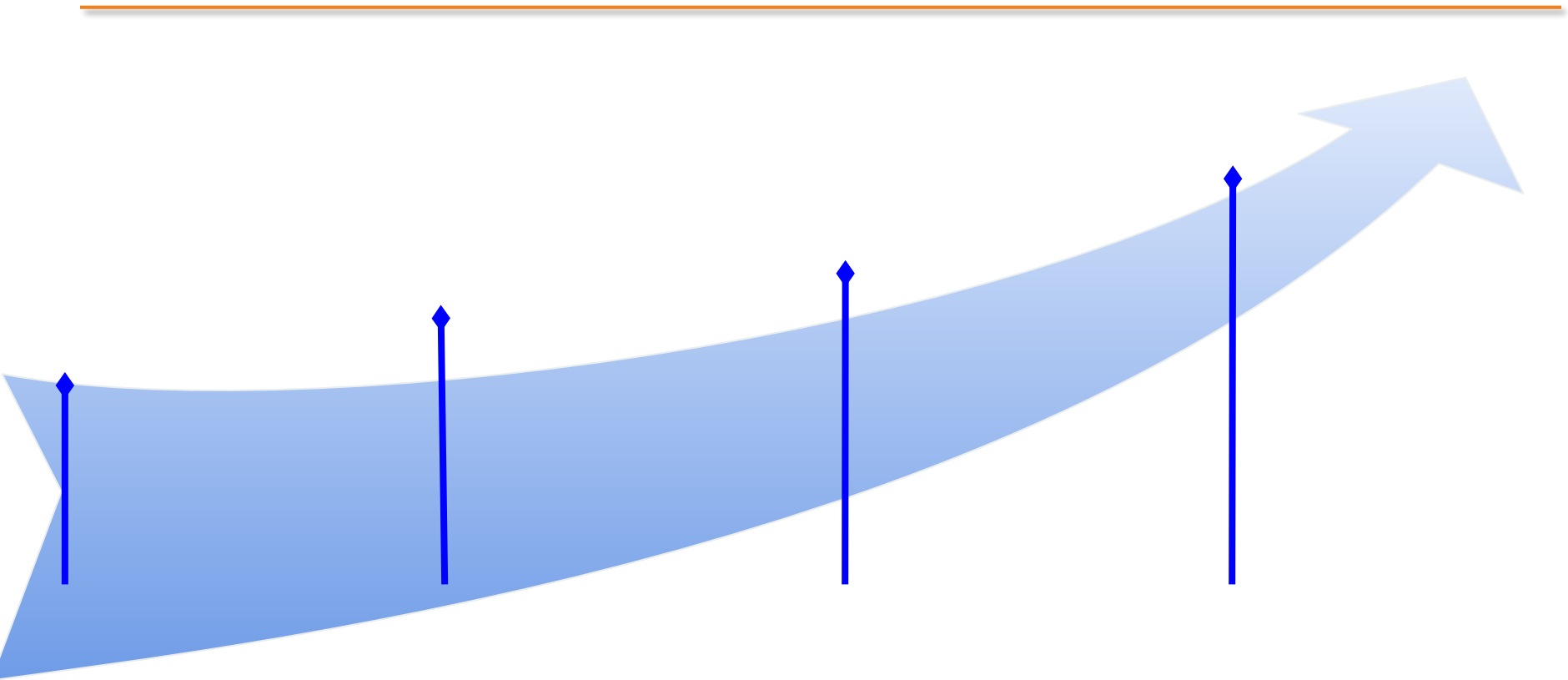


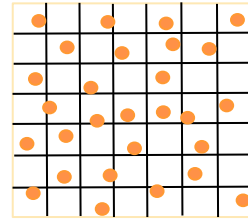


NERSC is the mission HPC facility





NERSC-9 Project Major Scope Items







NERSC-9: A System Optimized for Science

**From the start NERSC-9 had requirements of
simulation and data users in mind**

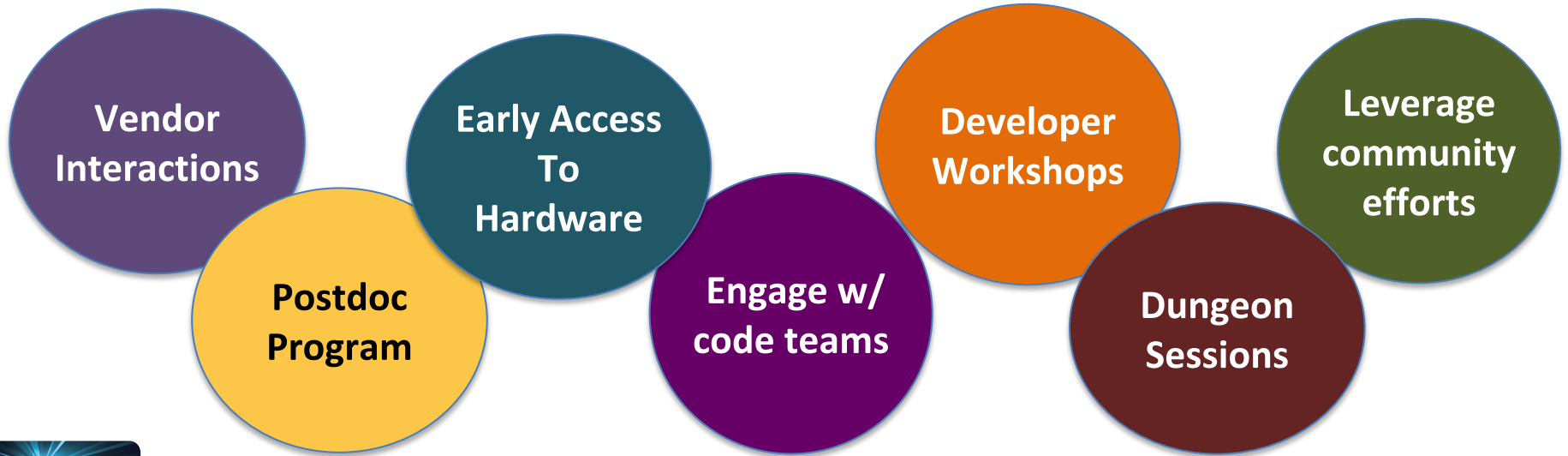






NERSC Exascale Scientific Application Program (NESAP)

- Prepare DOE SC users for advanced architectures like Cori
- Partner closely with ~20 application teams and apply lessons learned to broad NERSC user community.







Transitioning From KNL to AMD Processors

Codes optimized on Xeon Phi (KNL) will run well on Perlmutter

Many KNL architecture features are present on Perlmutter CPUs

Many-Core

MPI+OpenMP Programming Model Will Continue

Easier Onramp to “Many-Core” with Perlmutter CPUs than with KNL

More Traditional Cores

Single Memory Technology





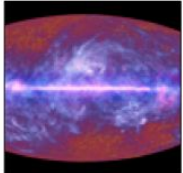




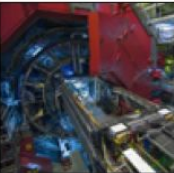
NERSC already supports a large number of users and projects from DOE SC's experimental and observational facilities



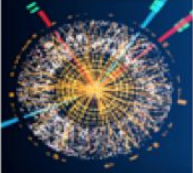
Palomar Transient Factory Supernova



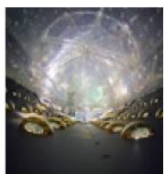
Planck Satellite Cosmic Microwave Background Radiation



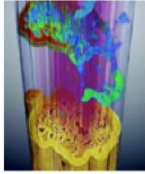
Alice Large Hadron Collider



Atlas Large Hadron Collider



Dayabay Neutrinos



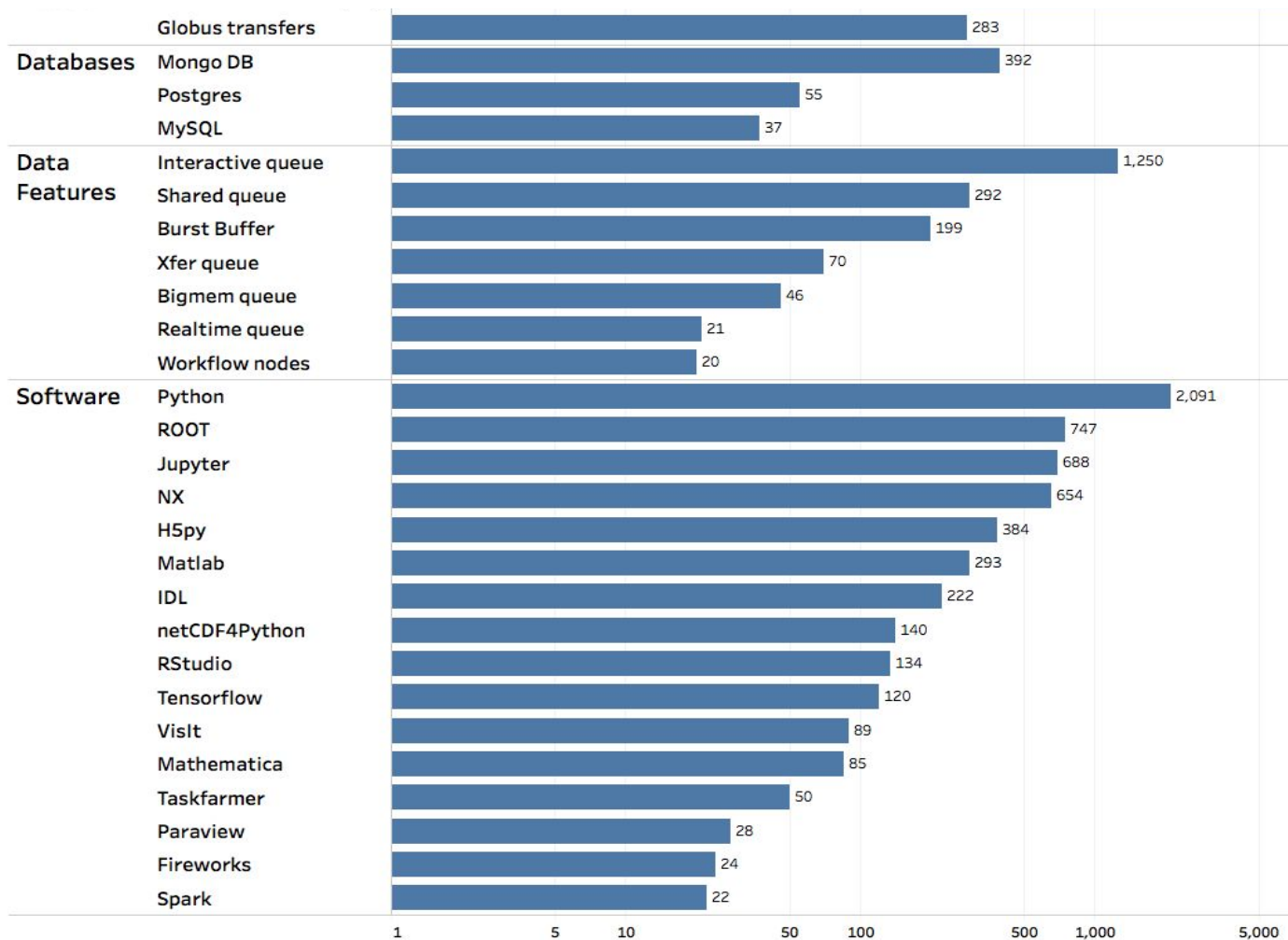
ALS Light Source



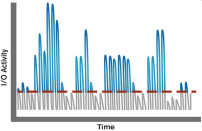
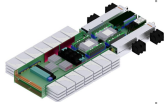
LCLS Light Source



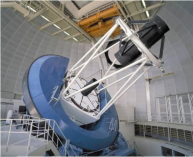



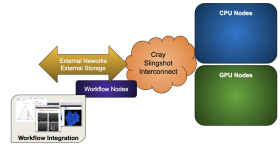
Joint Genome Institute Bioinformatics

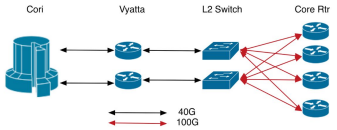
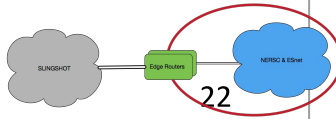


Data Features	Cori experience	N9 enhancements
---------------	-----------------	-----------------

<h2>I/O and Storage</h2>	<p>Burst Buffer</p> 	<p>All-flash file system: performance with ease of data management</p> 
--------------------------	--	--

<h2>Analytics</h2> <ul style="list-style-type: none"> - Production stacks - Analytics libraries - Machine learning 	<p>User defined images with Shifter NESAP for data</p>  <p>New analytics and ML libraries</p> 	<p>Benchmark Production Analytics workflows. Data apps in NESAP at outset</p>  <p>Optimised analytics libraries and deep learning application benchmarks</p>
---	---	---

<h2>Workflow integration</h2>	<p>Real-time queues</p> 	<p>SLURM co-scheduling Workflow nodes integrated</p> 
-------------------------------	---	--

<h2>Data transfer and streaming</h2>	<p>SDN</p> 	<p>Slingshot ethernet-based converged fabric</p> 
--------------------------------------	--	---













GPU Programming Models

We will support and engage our user community where their apps are:

CUDA: MILC, Chroma, HACC ...

CUDA FORTRAN: Quantum ESPRESSO, StarLord (AMREX)

OpenACC: VASP, E3SM, MPAS, GTC, XGC ...

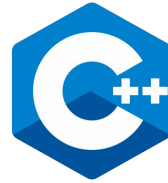
Kokkos: LAMMPS, PELE, Chroma ...

Raja: SW4

Engaging around Performance Portability



NERSC is working with PGI/NVIDIA to enable OpenMP GPU acceleration



NERSC Hosted Past C++ Summit and ISO C++ meeting on HPC.



NERSC Will Pursue Membership in 2018

Performance Portability / Measurements / Measurement Techniques

speed and vector/instruction-sets)

Performance Portability
Introduction
Office of Science Facilities ▾
Performance Portability ^
Overview
Definition
Measurements ▾
Measurement Techniques
Collecting Roofline on KNL
Collecting Roofline on GPUs
Strategy
Approaches ▾
Case Studies ▾
Summary

- The application or algorithm may be fundamentally limited by *different* aspects of the system on different HPC system.

As an example, an implementation of an algorithm that is limited by memory bandwidth may be achieving the best performance it theoretically can on systems with different architectures but could be achieving widely varying percentage of peaks FLOPS on the different systems.

Instead we advocate for one of two approaches for defining performance against expected or optimal performance on the system for an algorithm:

1. Compare against a known, well-recognized (potentially non-portable), implementation.

Some applications, algorithms or methods have well-recognized optimal (often hand-tuned) implementations on different architectures. These can be used as a baseline for defining relative performance of portable versions. Our Chrome application case study shows this approach. See

Table of contents
Measuring Portability
Measuring Performance

1. Compare against a known, well-recognized (potentially non-portable), implementation.
2. Use the roofline approach to compare actual to expected performance

NERSC is leading development of performanceportability.org



Doug Doerfler Leading Accepted Performance Portability Workshop at SC18. and 2019 DOE COE Perf. Port. Meeting

