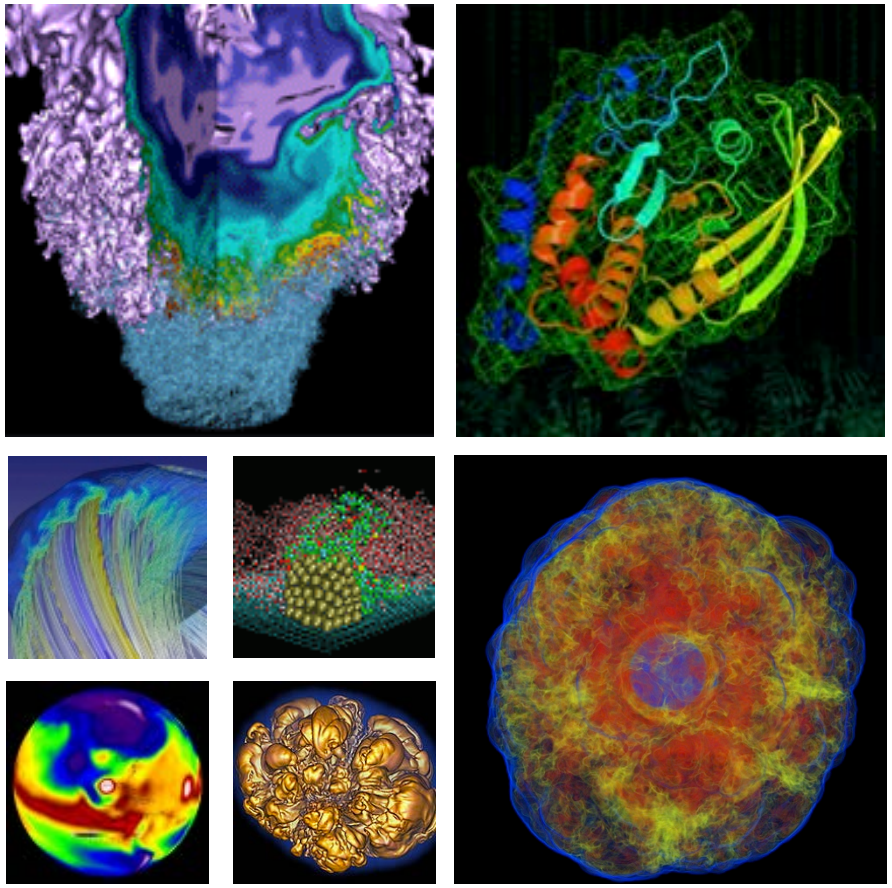
Performance Analysis of Emerging Data Analytics and HPC workloads



Christopher Daley
Sudip Dosanjh, Prabhat, Nicholas Wright

PDSW-DISCS 2017
November 13, 2017

Introduction

- **The National Energy Research Scientific Computing Center (NERSC) is the primary computing facility for the Office of Science in the U.S Department of Energy (DOE)**
- **The NERSC Cori supercomputer contains different compute nodes for compute and data workloads**
- **In this presentation, we analyze representative applications to understand whether this is the right architectural approach**
- **We also consider the benefits of a many-core processor architecture and a Burst Buffer**

The two partitions of Cori supercomputer

Cori-P1: Data partition

Optimized for latency and single-thread performance

- 2,388 compute nodes
- 2 * Intel Xeon E5-2698 v3 (Haswell) processors per compute node
- 2.3 GHz
- 32 cores per node
- 2 HW threads per core
- 256-bit vector length

Cori-P2: Compute partition

Optimized for throughput and performance per watt

- 9,688 compute nodes
- 1 * Intel Xeon-Phi 7250 (KNL) processor per compute node
- 1.4 GHz
- 68 cores per node
- 4 HW threads per core
- 512-bit vector length

The two partitions of Cori supercomputer

Cori-P1: Data partition

Optimized for latency and single-thread performance

- 128 GB DDR4 memory ~115 GB/s memory bandwidth

Cori-P2: Compute partition

Optimized for throughput and performance per watt

- 96 GB DDR4 memory ~85 GB/s memory bandwidth
- 16 GB MCDRAM memory ~450 GB/s memory bandwidth

The two partitions of Cori supercomputer

Cori-P1: Data partition

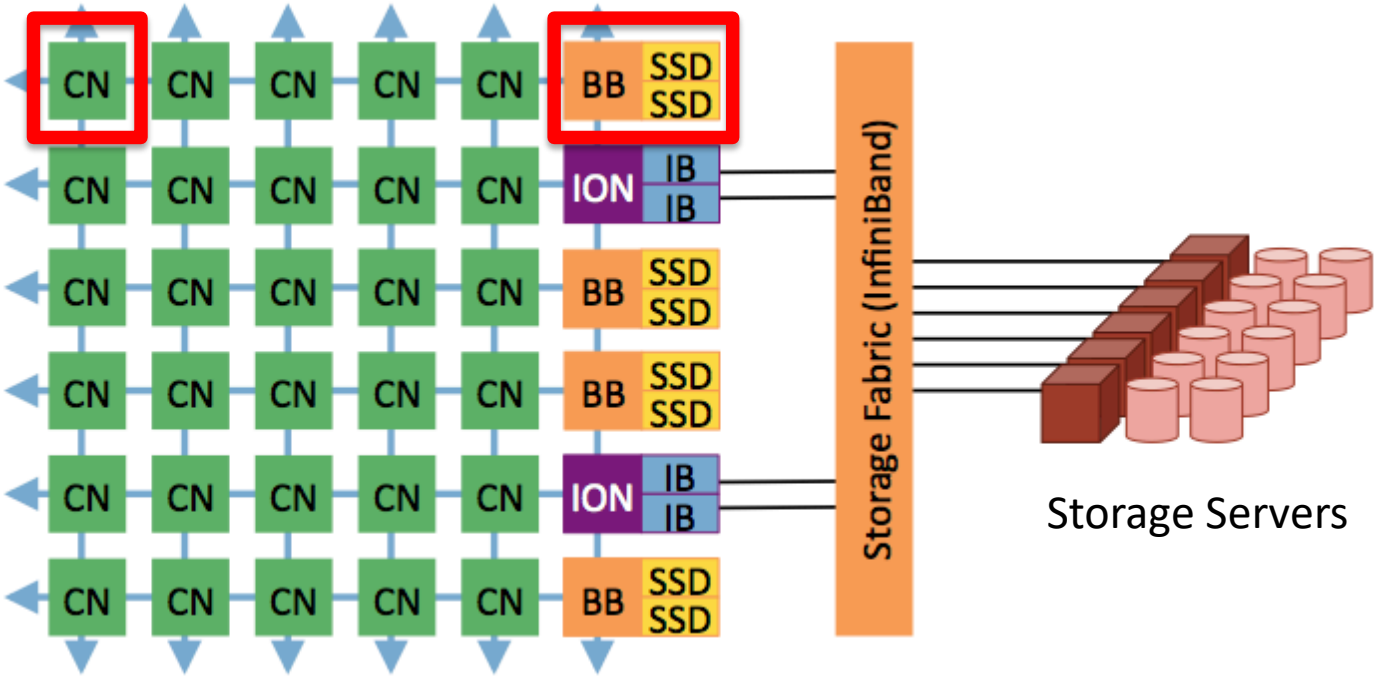
Optimized for latency and single-thread performance

Cori-P2: Compute partition

Optimized for throughput and performance per watt

- Cray Aries high-speed network
- 28 PB Lustre Scratch file system
  ~700 GB/s I/O performance
- 1.5 PB Cray DataWarp Burst Buffer (BB)
  ~1.5 TB/s I/O performance

Cori system architecture overview



The user job submission script chooses

- Compute node type (Haswell or KNL)
- Number of Burst Buffer nodes - through a capacity parameter

Compute and data workload

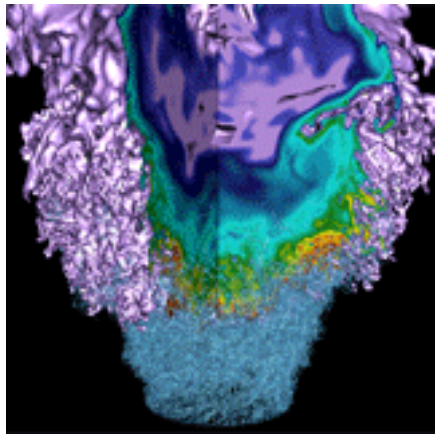
Applications represent the A) simulation science, B) data analytics of simulation data sets and C) data analytics of experimental data sets workload at NERSC

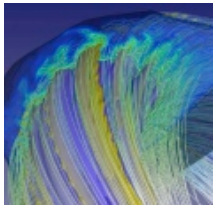
	Application	Purpose	Parallelization	Nodes	Mem/node (GiB)
A	Nyx	Cosmology simulations	MPI+OpenMP	16	61.0
A	Quantum Espresso	Quantum Chemistry simulations	MPI+OpenMP	96	42.4
B	BD-CATS	Identify particle clusters	MPI+OpenMP	16	5.7
B	PCA	Principle Component Analysis	MPI	50	44.7
C	SExtractor	Catalog light sources found in sky survey images	None	1	0.6
C	PSFEx	Extract Point Spread Function (PSF) in sky survey images	Pthreads	1	0.1

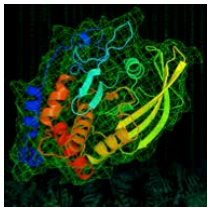
Two sets of performance experiments

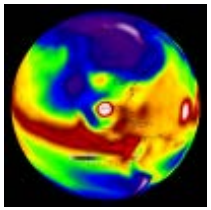
1. **Analysis of baseline application performance**
   - Breakdown of time spent in compute, communication and I/O
   - Comparison of performance on Cori-P1 and Cori-P2
2. **Case studies considering how to better utilize technology features of Cori-P2 without making any code modifications**
   - Strong scaling problems to better utilize the high bandwidth memory on KNL
   - Making use of many small KNL cores
   - Accelerating I/O with a Burst Buffer

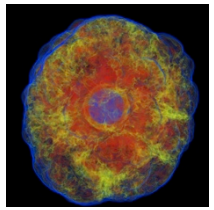
Baseline application performance

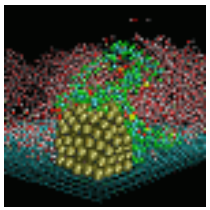




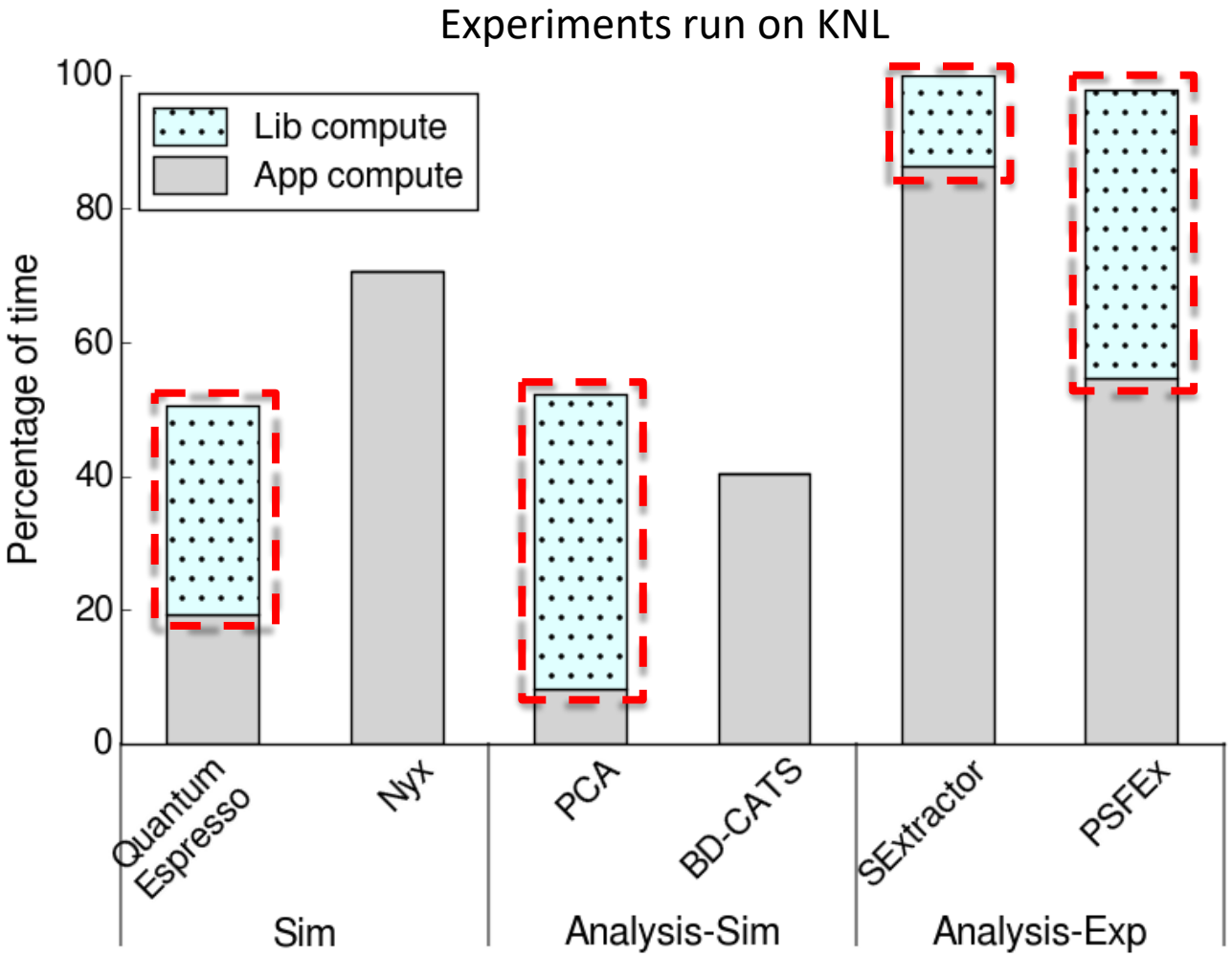






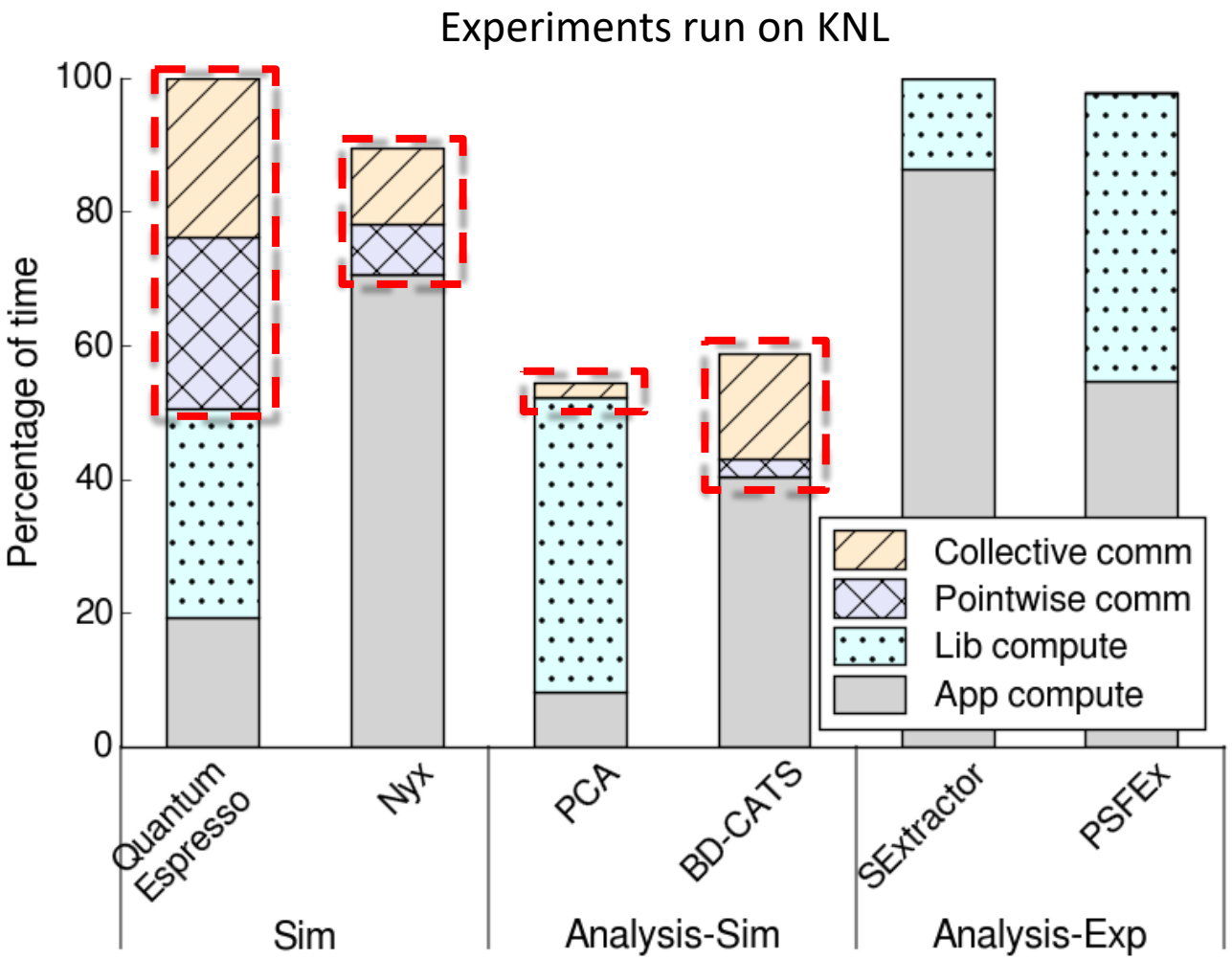


Observation #1: Common math libraries



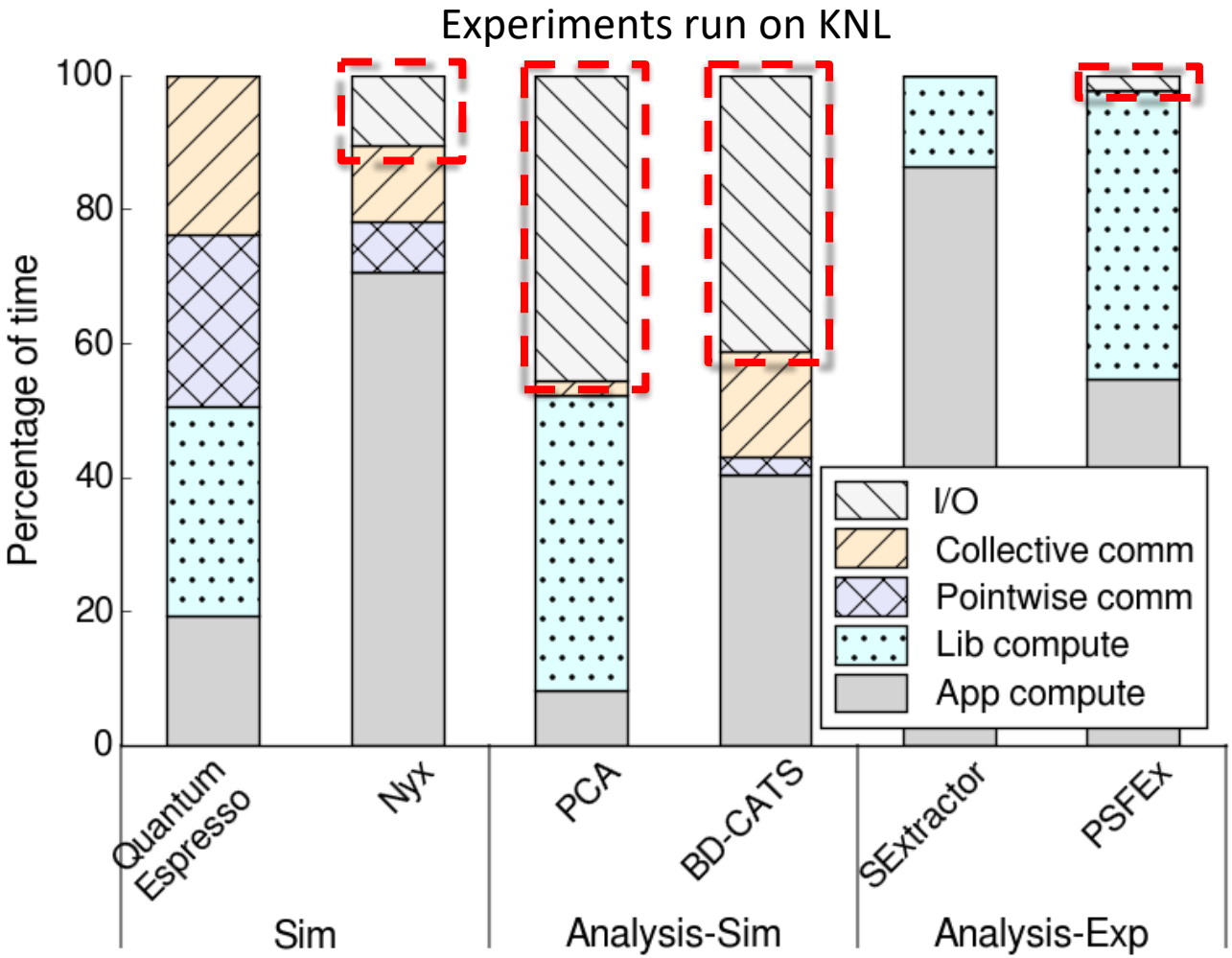
Four of the six applications use BLAS, LAPACK or FFTW libraries (through Intel MKL)

Observation #2: Significantly different network requirements



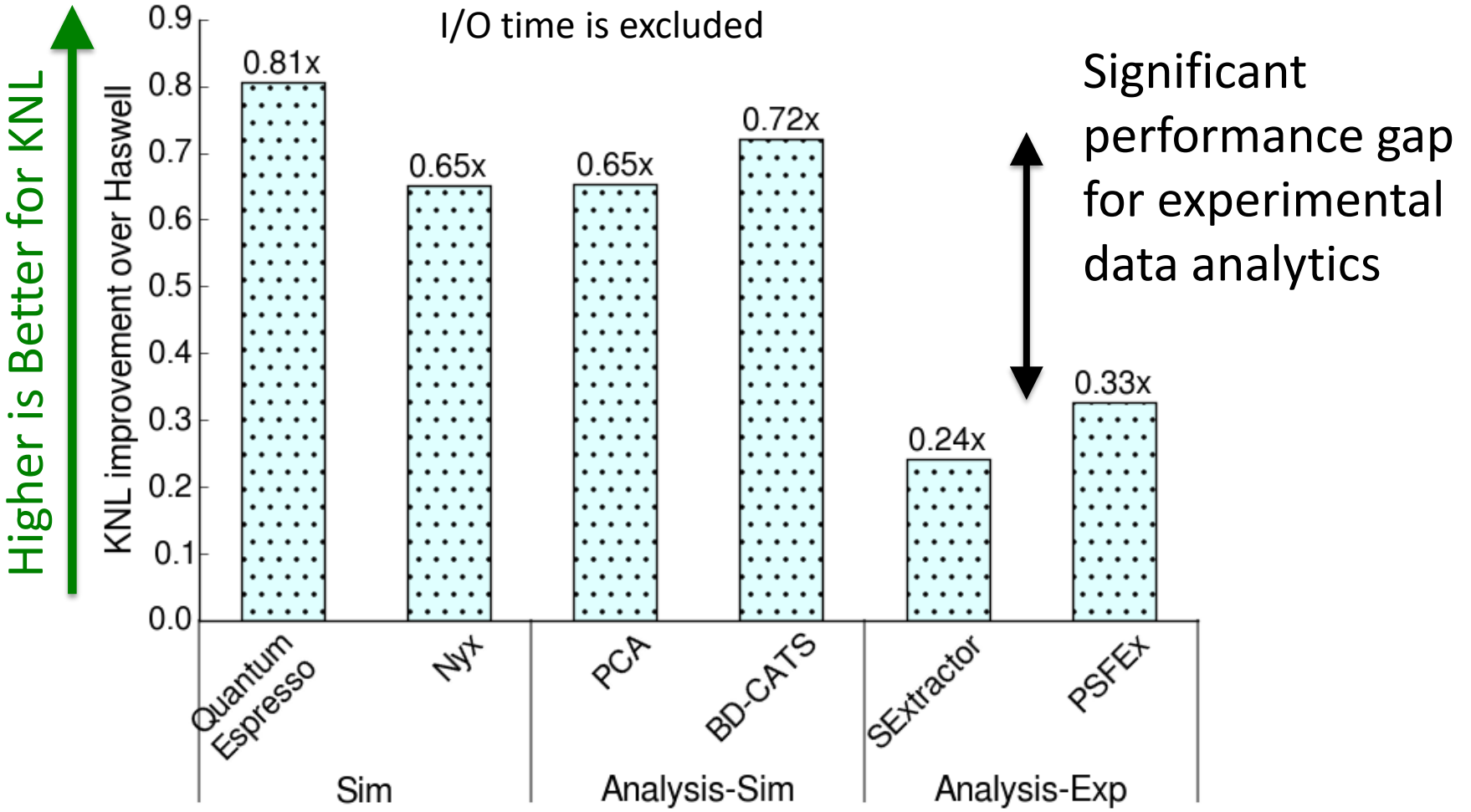
0 – 50% of time in MPI communication

analytics applications spend more time in I/O



PCA and BD-CATS spend more than 40% of time in I/O

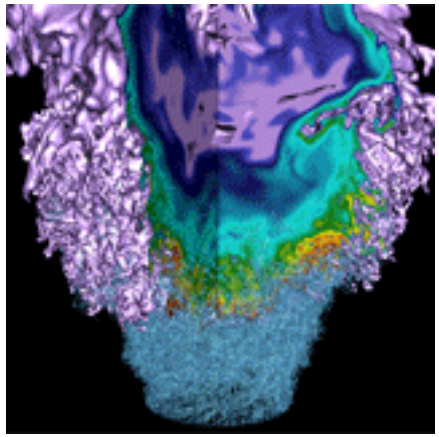
Base configurations perform worse on KNL nodes than Haswell nodes

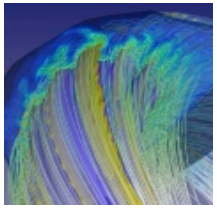


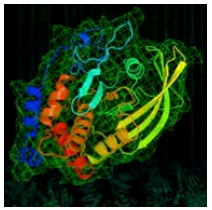
Baseline performance summary

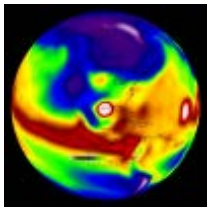
- **The same math libraries are used in compute and data workloads**
- **There are significant differences in the network requirements of applications**
- **Simulation data analytics applications spend much more time in I/O than the other applications**
- **All baseline configurations perform worse on a KNL node than a 2-socket Haswell node**
  - Experimental data analytics applications have the worst relative performance

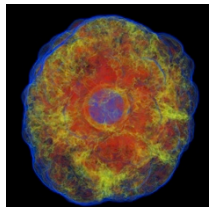
Optimizing the application configurations

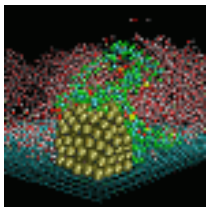












3 optimization use cases

1. **Strong scaling the PCA application so that it fits in the memory capacity of MCDRAM**
2. **Running high throughput configurations of SExtractor and PSFEx per compute node**
3. **Using the Cori Burst Buffer to accelerate I/O in Nyx, PCA and BD-CATS**

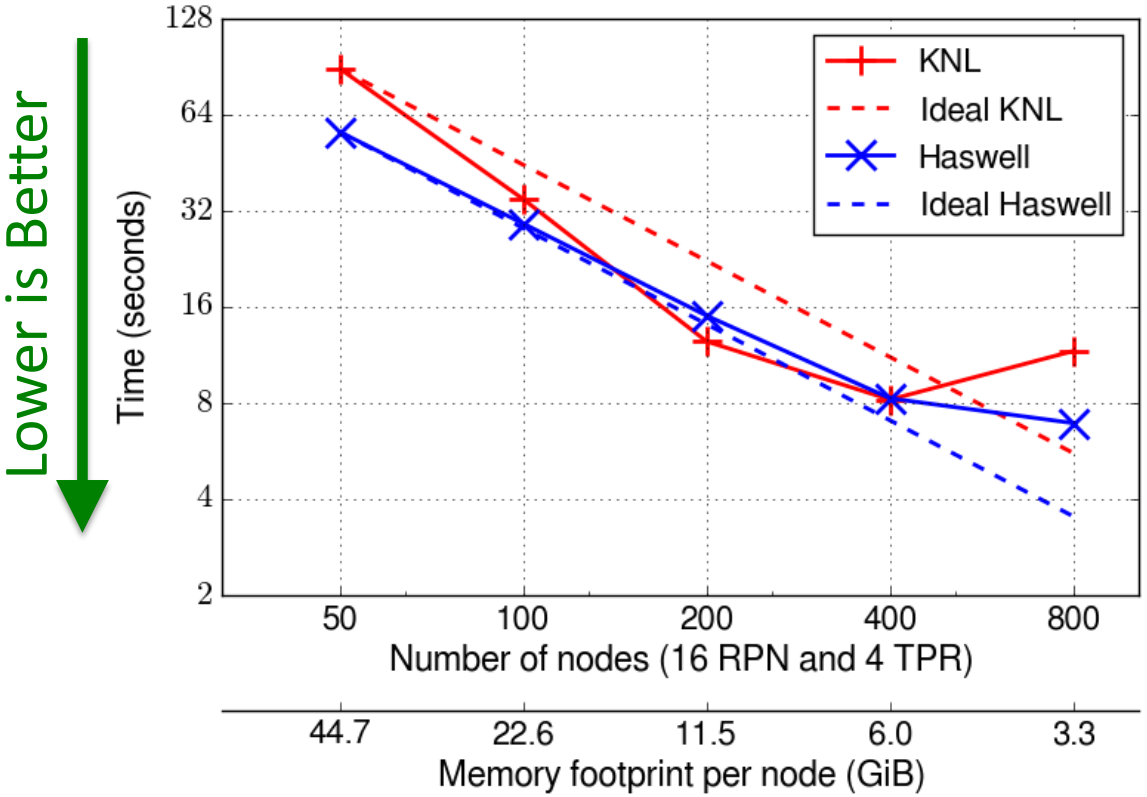
applications to fit in MCDRAM memory capacity

- **PCA has a memory footprint of 44.7 GiB per node**
- **Most of the compute time is spent in a matrix-vector multiply (DGEMV) kernel**
  - Performs best when data fits in the memory capacity of MCDRAM

Kernel	GFLOP/s/node larger than MCDRAM	GFLOP/s/node smaller than MCDRAM	Performance improvement
Matrix-matrix multiply (DGEMM)	1561	1951	1.2x
Matrix-vector multiply (DGEMV)	20	84	4.2x

Use case #1: Strong-scaling PCA significantly improves performance

I/O time is excluded



Super-linear speedup on KNL as more of PCA's 2 matrices fit in MCDRAM

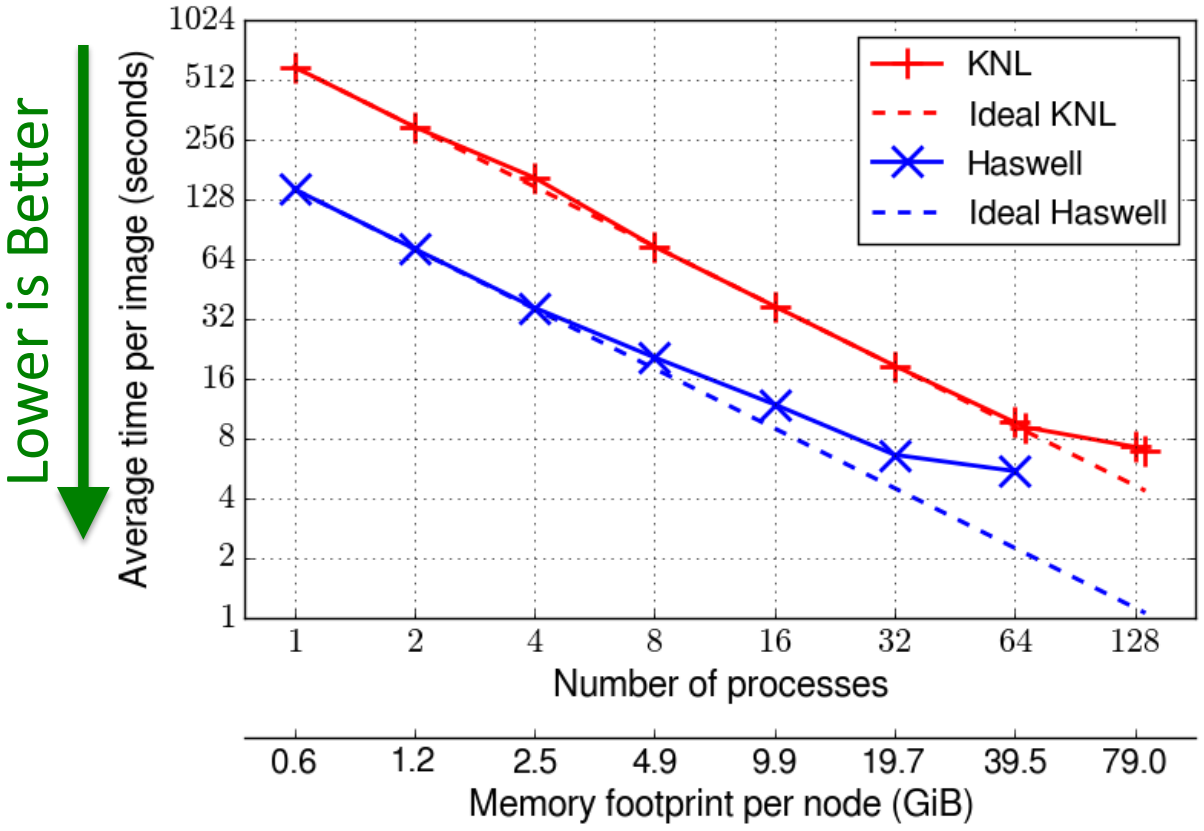
PCA runs faster on KNL than Haswell at 200 nodes

Use case #2: Using many small cores of KNL

- **The experimental data analytics applications perform poorly on the KNL processor architecture**
  - The node-to-node performance relative to Haswell is 0.24x (SExtractor) and 0.33x (PSFEx)
- **Both applications are embarrassingly parallel**
  - Trivial to analyze different images at the same time
- **We consider whether we can launch enough tasks on the many small cores to improve the relative performance**

node needed to be competitive with Haswell

Plot shows SExtractor application
I/O time is excluded



~3x improvement in node-to-node performance

SExtractor:
0.24x to 0.75x

PSFEx:
0.33x to 1.02x

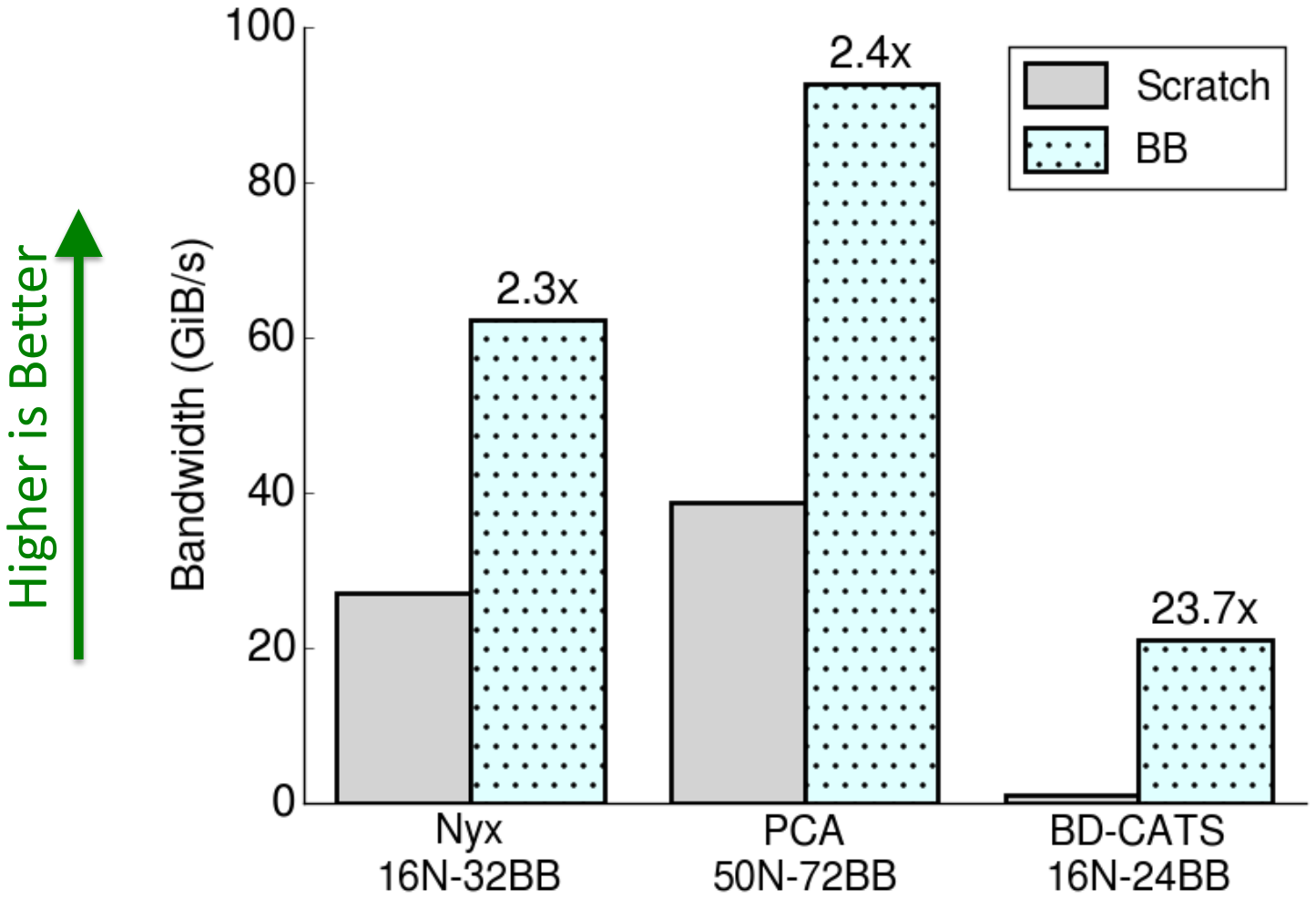
Overview of the I/O from the top 3 applications

	Application	I/O time (%)	API	Style	Overview
A	Nyx	10.6%	POSIX	N:M	Large sequential writes to checkpoint and analysis files (1.2 TiB)
B	PCA	45.6%	HDF5 - ind. I/O	N:1	Large sub-array reads from input file (2.2 TiB)
B	BD-CATS	41.3%	HDF5 - coll. I/O	N:1	Large sub-array reads from input file (12 GiB) and writes to analysis file (8 GiB)

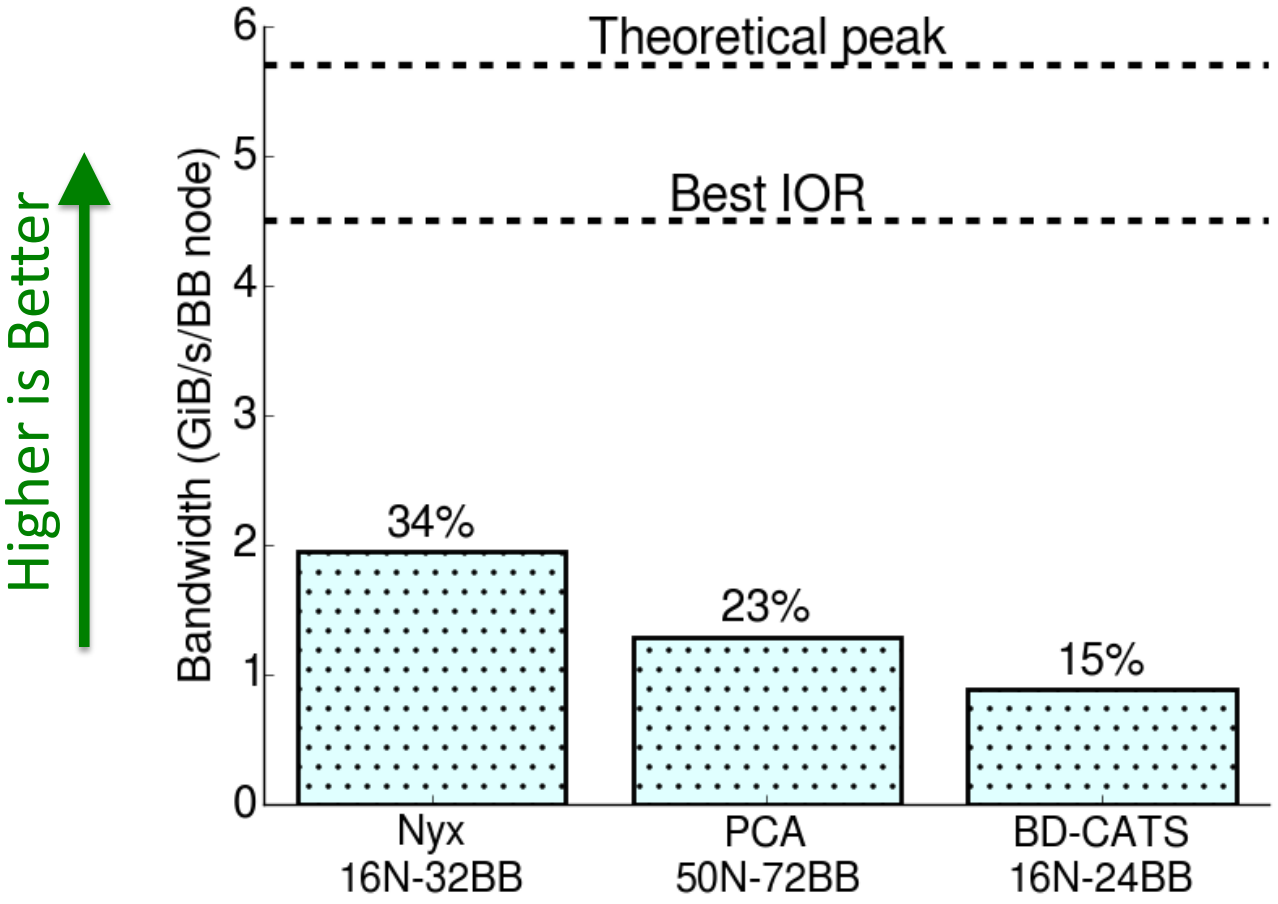
A = Simulation science
B = Data analytics of simulation data sets

No fine-grained non-sequential I/O in any of the 6 applications

Use case #3: The Burst Buffer improves performance for every application



shows satisfactory usage over a broad workload



Higher fractions of peak would be possible by using more compute nodes than Burst Buffer nodes

Conclusions

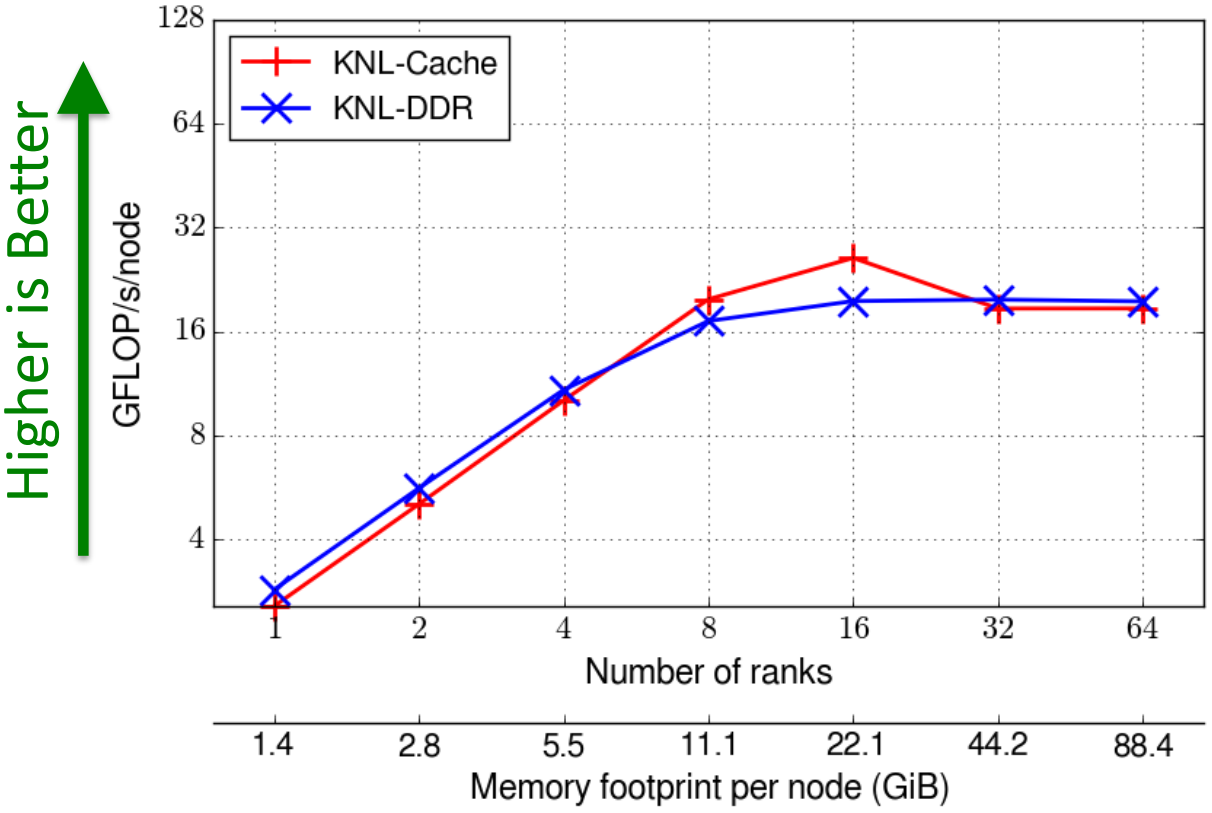
- **All baseline configurations perform worse on a KNL node than a 2-socket Haswell node (Many-core is hard!)**
  - High throughput configurations of experimental data analytics improve node-to-node performance by 3x
  - Strong-scaling an application can improve the use of MCDRAM, e.g. PCA application ran faster on KNL than Haswell at the optimal concurrency
- **The Burst Buffer improves I/O performance by a factor of 2.3x – 23.7x**
- **There is evidence that the same architectural features can benefit both compute and data analytics workloads**

Thank you.

This work was supported by Laboratory Directed Research and Development (LDRD) funding from Berkeley Lab, provided by the Director, Office of Science and Office of Science, Office of Advanced Scientific Computing Research (ASCR) of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Use case #1: Single-node DGEMV on KNL - **PCA matrix size**

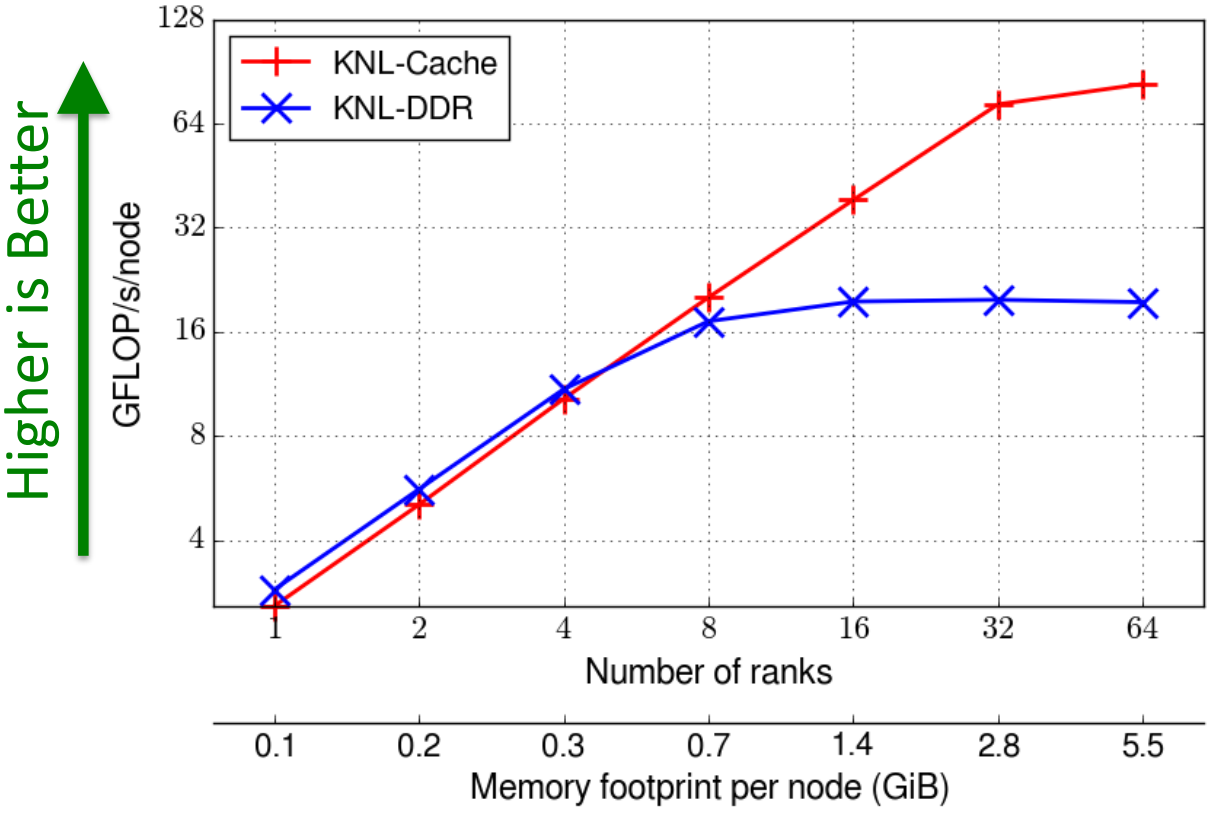
Matrix size of 1.38 GiB (3969 x 46715)
DGEMV kernel replicated by each MPI rank



- **1-node DGEMV does not scale beyond 16 MPI ranks**
- **50x performance deficit to DGEMM FLOP/s/node**

Use case #1: Single-node DGEMV on KNL - **small matrix size**

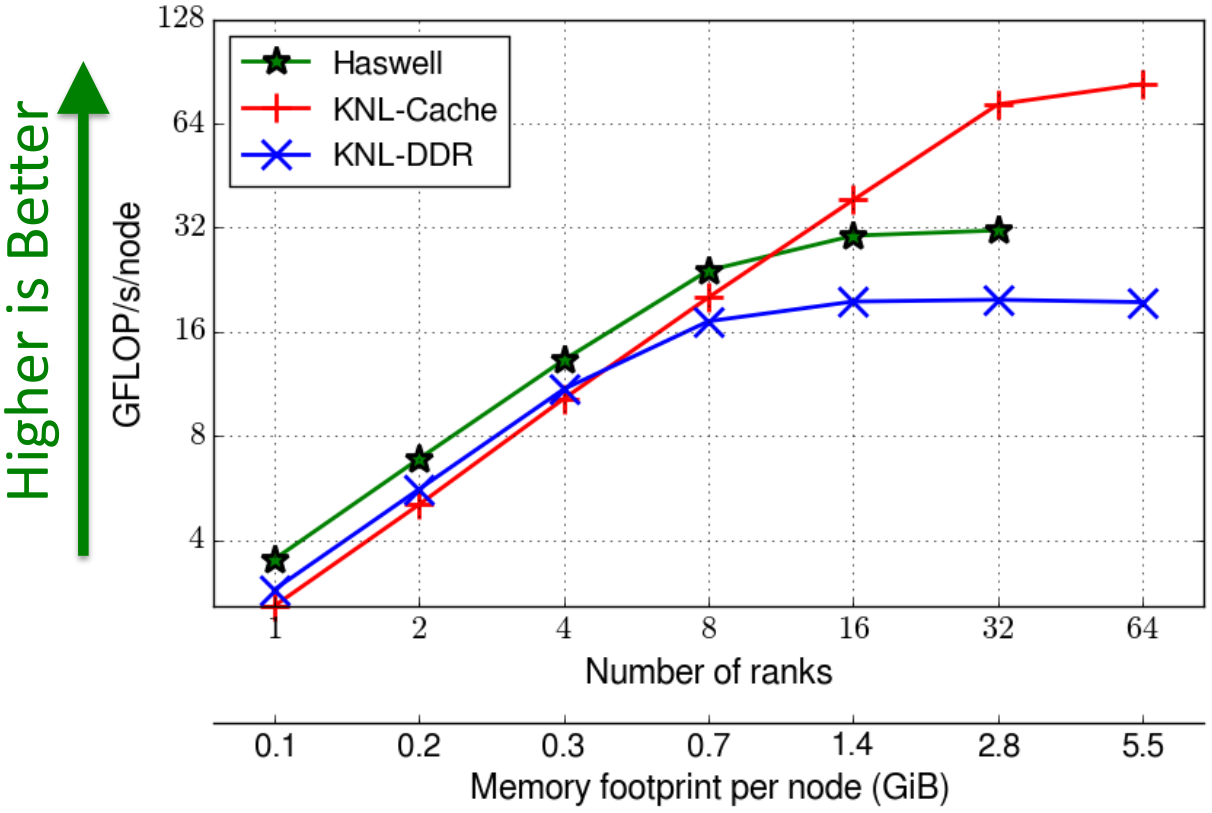
Matrix size of 0.09 GiB (249 x 46715)
DGEMV kernel replicated by each MPI rank



- **This time DGEMV scales to 64 MPI ranks because aggregate matrix size < MCDRAM capacity**
- **4.2x performance gain compared to using DDR memory**

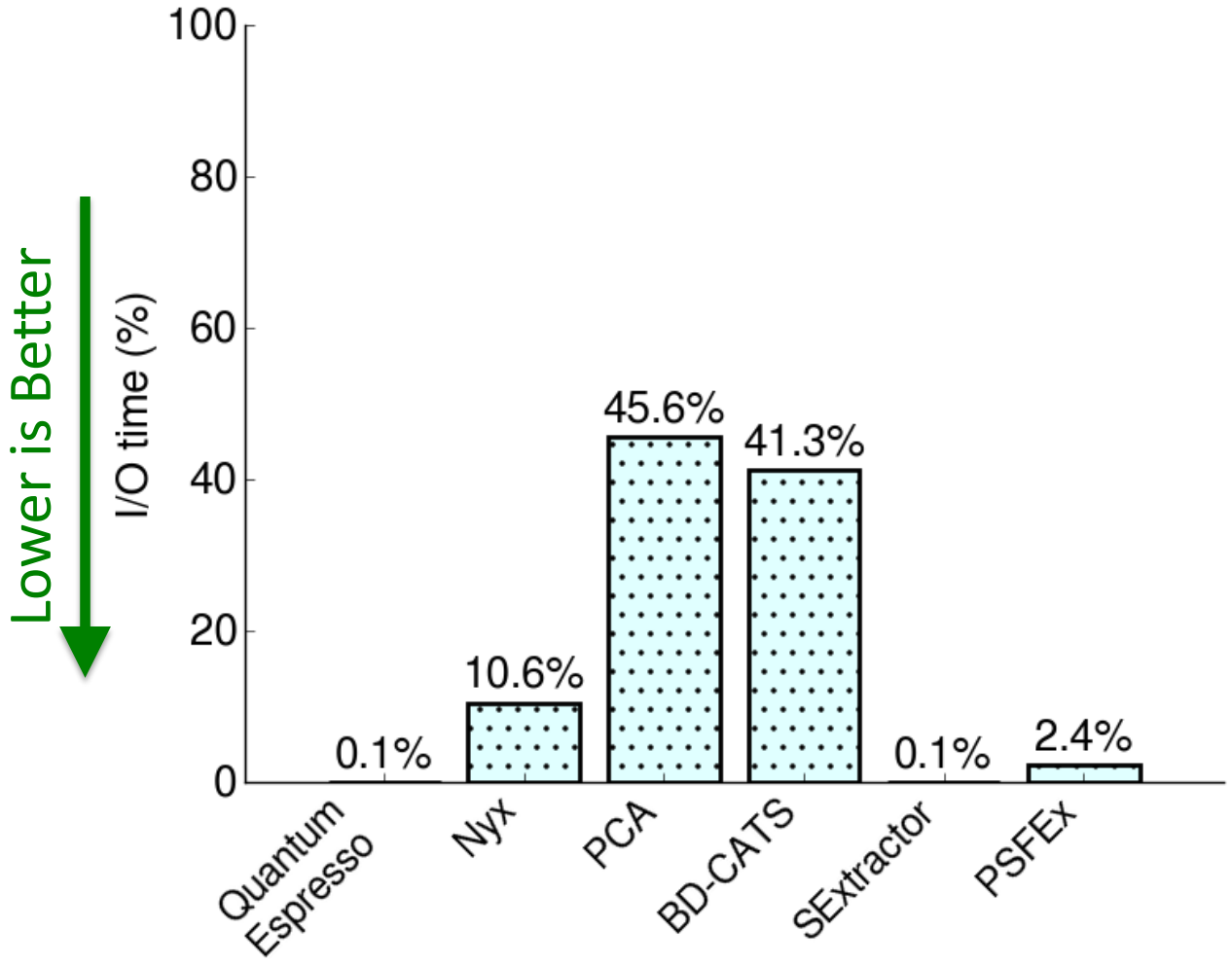
Use case #1: Single-node DGEMV on KNL and Haswell - **small matrix size**

Matrix size of 0.09 GiB (249 x 46715)
DGEMV kernel replicated by each MPI rank



- **The DGEMV kernel runs 2.7x faster on KNL than Haswell**

Three applications spend more than 10% of time in I/O



The applications perform structured I/O with different I/O motifs

- **Nyx**
  - Flexible N:M I/O using the POSIX API
  - Writes a checkpoint data set of size 157 GiB and a plot file data set of size 89 GiB every single step; total of 1.2 TiB.
- **PCA**
  - Single shared file I/O using the HDF5 API and independent access mode
  - Simple file layout containing a single 2D HDF5 datasets; processes read a unique sub-array from the dataset
  - Reads 2.2 TiB and process 0 writes 1 GiB
- **BD-CATS**
  - Single shared file I/O using the HDF5 API and collective access mode
  - Simple file layout containing 6 1D HDF5 datasets; processes read a unique sub-array from each dataset
  - Reads 12 GiB and writes 8 GiB

Overview of the I/O from the 3 applications spending most time in I/O

The I/O styles include shared file I/O and file per process (technically N:M)

	Application	I/O time (%)	API	Style	Data sets	I/O	Nodes	Node mem (%)
A	Nyx	10.6%	POSIX	N:M	5 checkpoint, 5 analysis	1.2 TiB (out)	16	10% (checkpoint), 6% (analysis)
B	PCA	45.6%	HDF5 - ind. I/O	N:1	Input	2.2 TiB (in)	50	47%
B	BD-CATS	41.3%	HDF5 - coll. I/O	N:1	Input, analysis	12 GiB (in), 8 GiB (out)	16	0.8% (input), 0.5% (analysis)

Cori System Architecture Overview