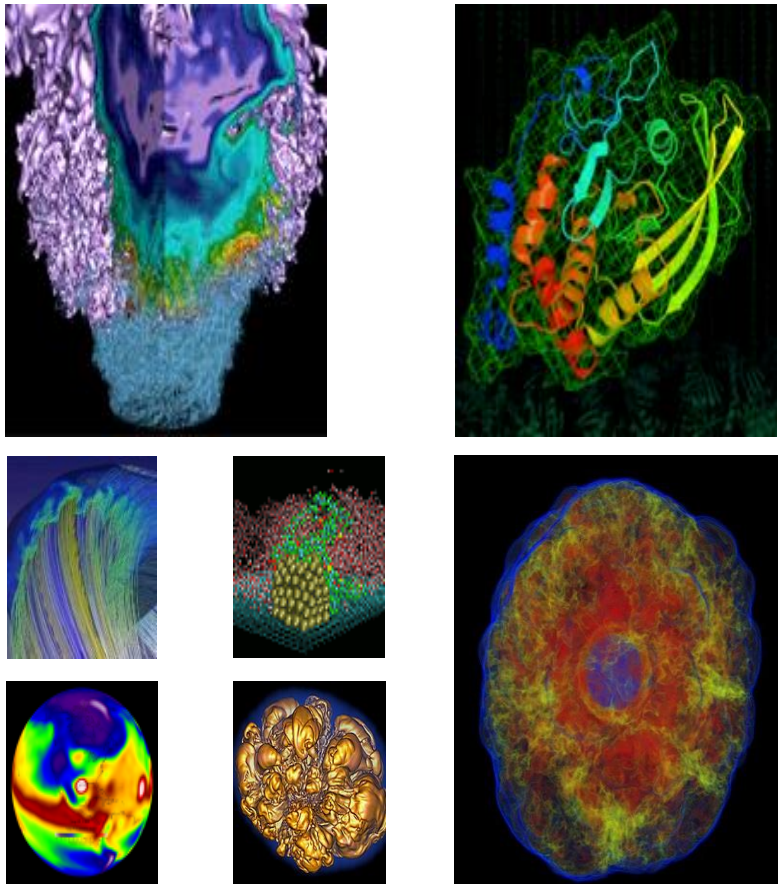
Scientific Data Stores Outside HEP



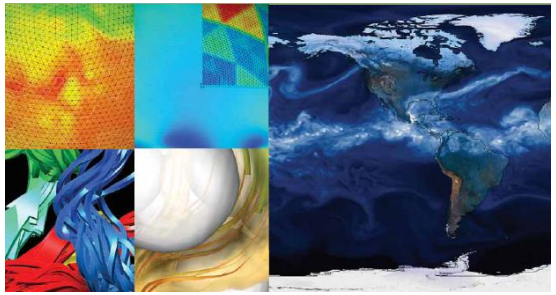
Wahid Bhimji
(Contributions from Joaquin Correa, Lisa Gerhardt and the NERSC Data and Analytics Services Group)

Atlas Software TIM
Nov 9th, 2015

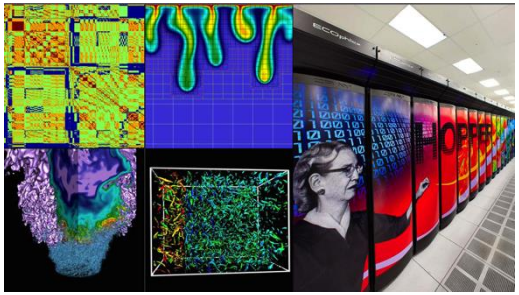
NERSC is the Production HPC & Data Facility for DOE Office of Science Research

U.S. DEPARTMENT OF ENERGY | Office of Science

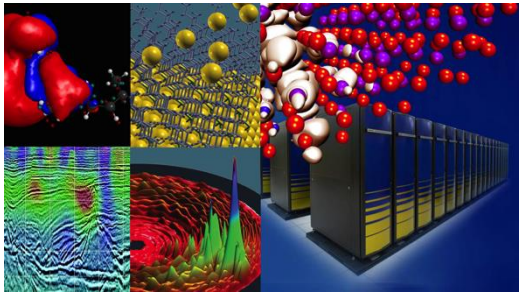
Largest funder of physical science research in U.S.



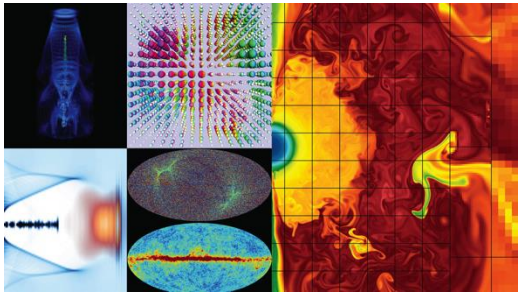
Bio Energy, Environment



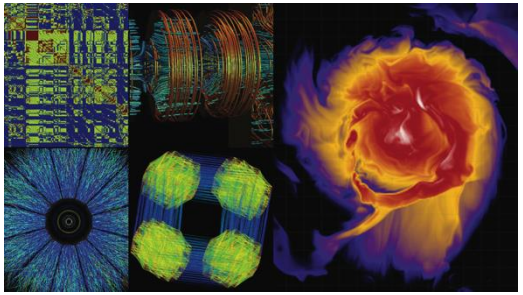
Computing



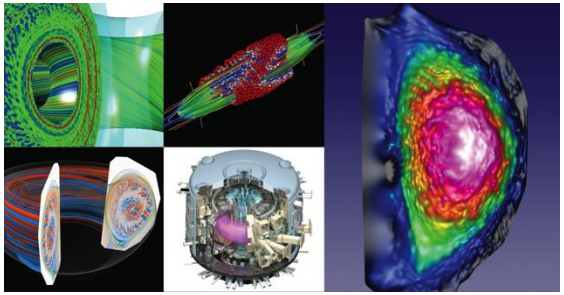
Materials, Chemistry, Geophysics



Particle Physics, Astrophysics



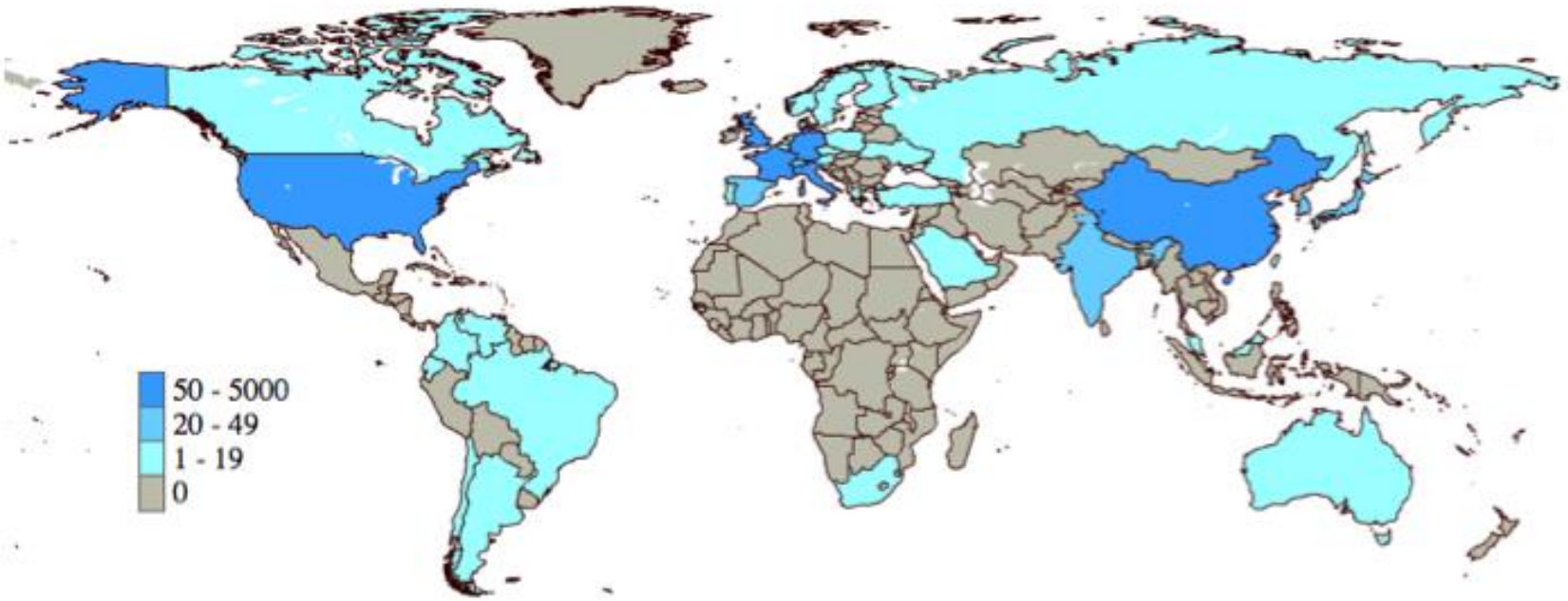
Nuclear Physics



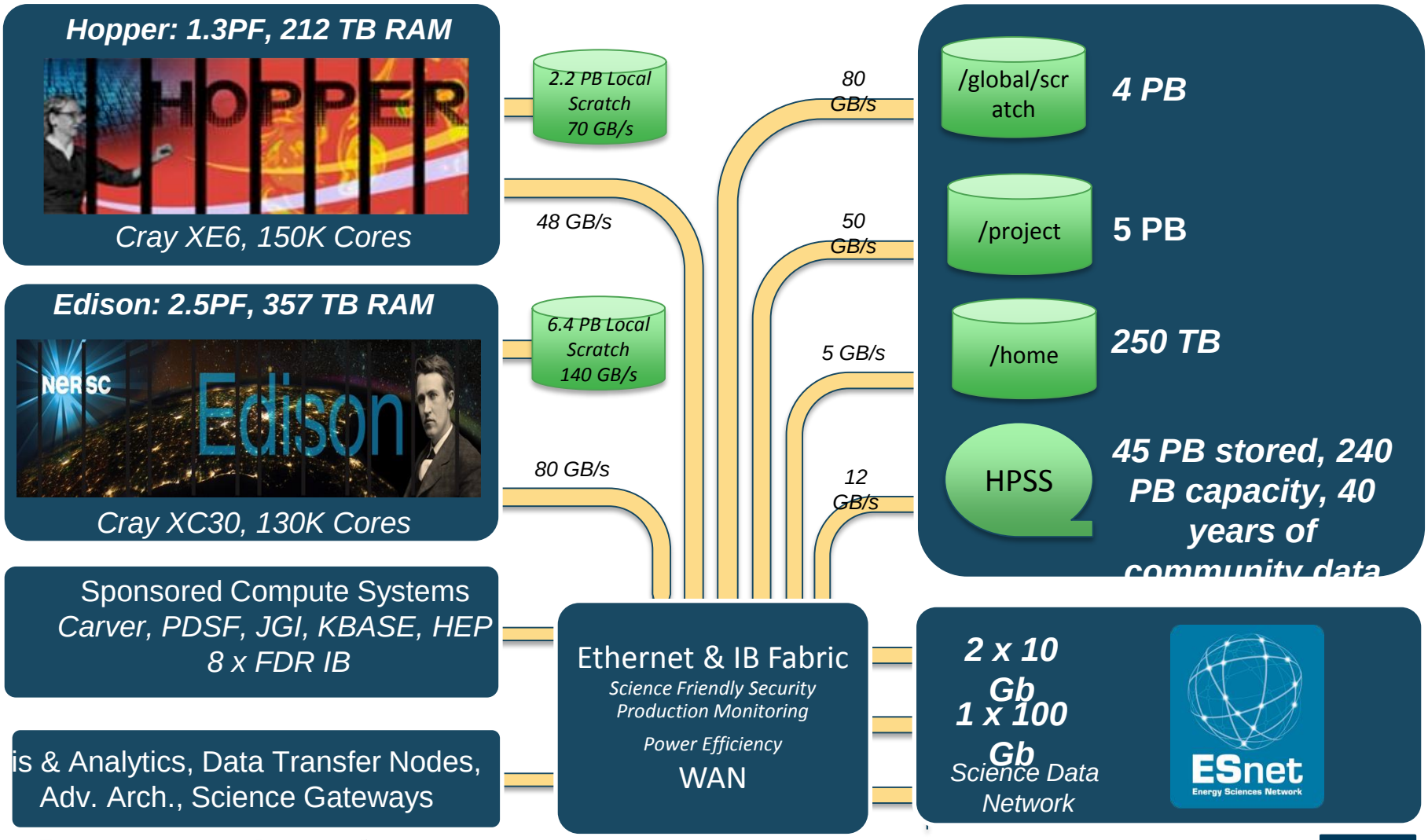
Fusion Energy, Plasma Physics

We support a broad user base

- **5000 users, and we typically add 350 per year**
- **Geographically distributed – worldwide**

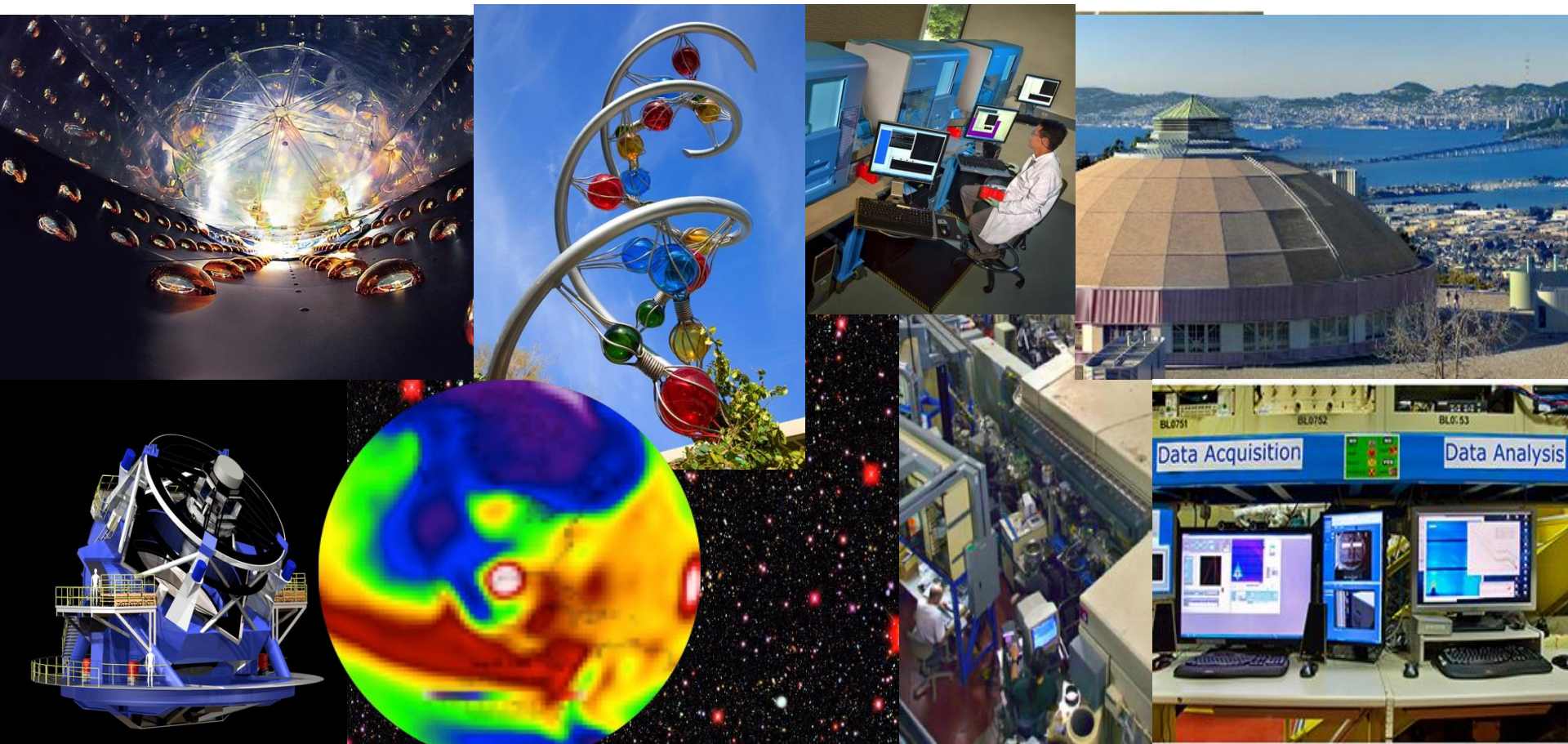


NERSC Systems

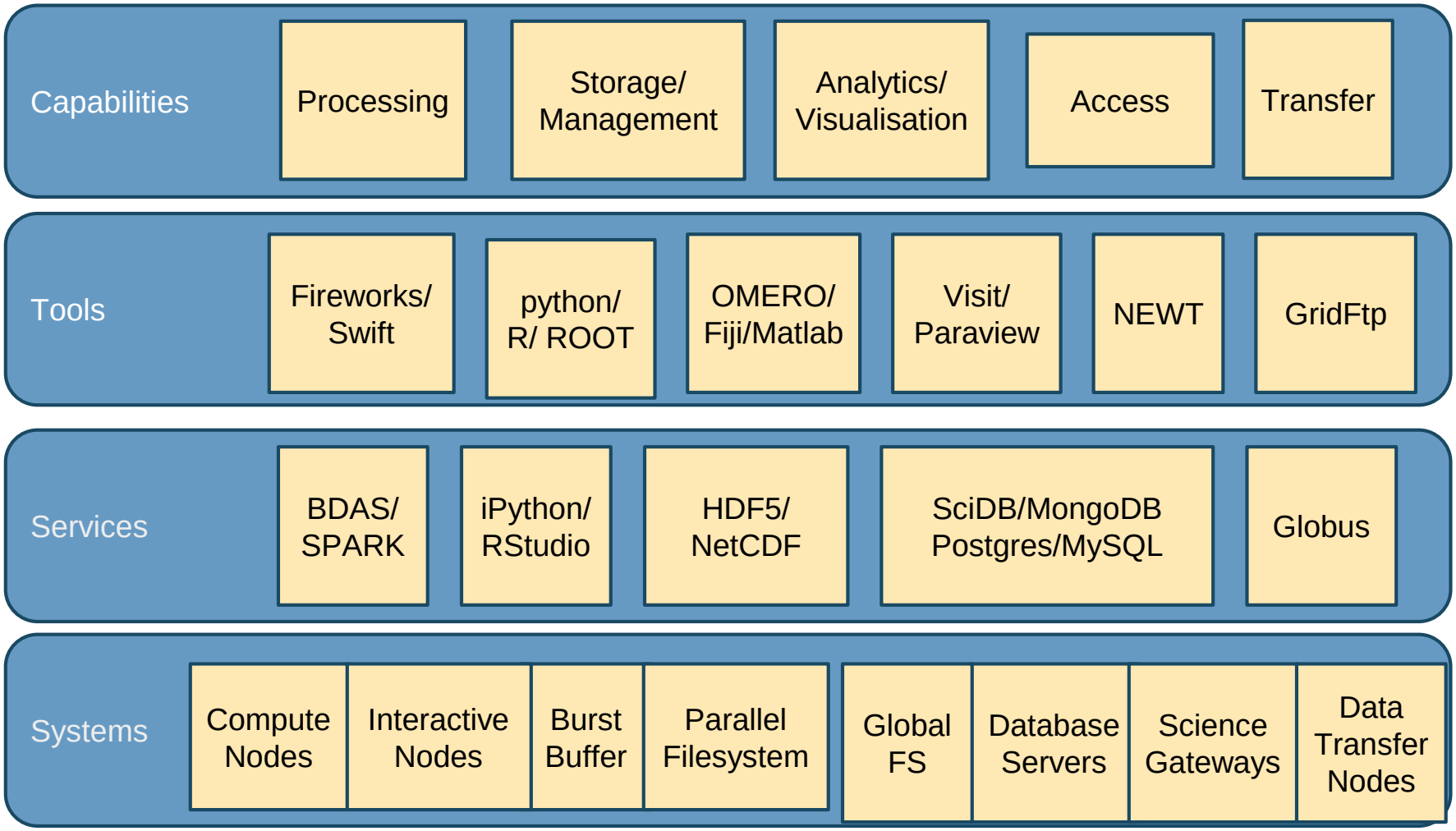


NERSC Data and Analytics

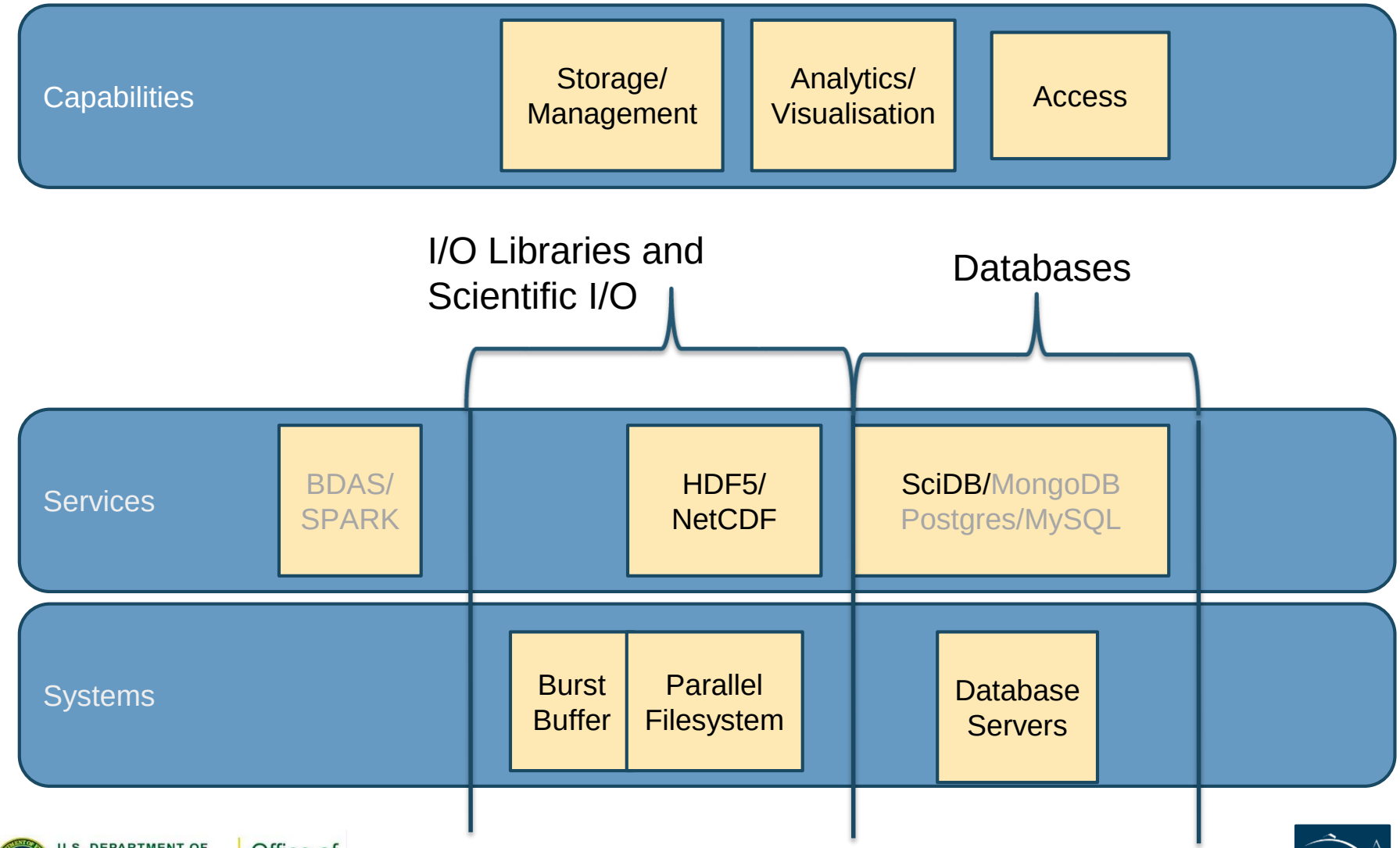
- DAS group - relatively new at NERSC (1.5 yrs) - supports data-intensive science across DoE science



NERSC Data and Analytics Stack

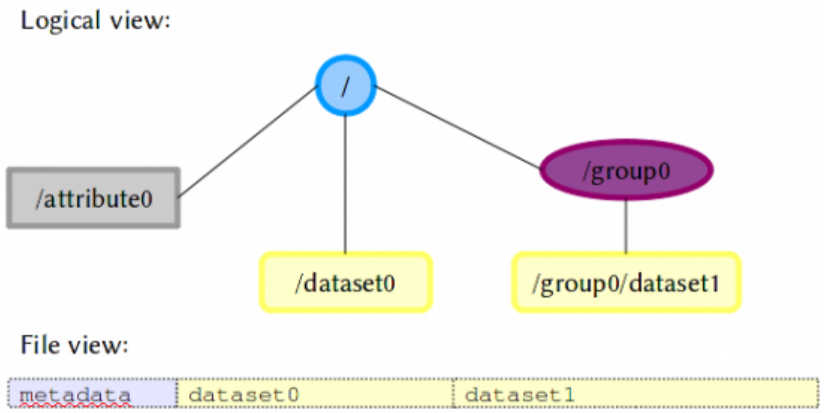


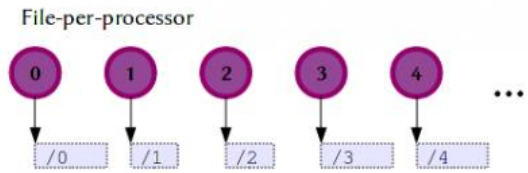
Data Stores

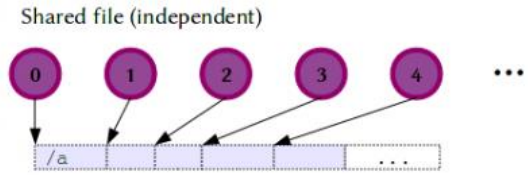

Capabilities
Storage/ Management
Analytics/ Visualisation
Access
I/O Libraries and Scientific I/O
Databases
Services
BDAS/ SPARK
HDF5/ NetCDF
SciDB/MongoDB Postgres/MySQL
Systems
Burst Buffer
Parallel Filesystem
Database Servers

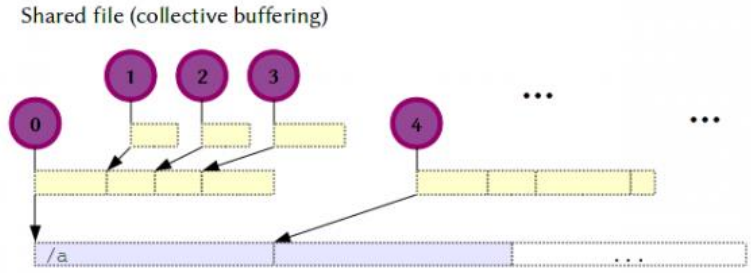
HDF5

- **HDF5 (Hierarchical Data Format v5)** probably the most widely-used / general purpose IO library:
  - Data model allows for complex data objects.
  - Portable file format across machines.
  - Software libraries; performance features and tools.
- **Hierarchical:** *groups* and *datasets* like folders and files
- **Datasets:** Have *headers* (that describe type, dimensionality, and layout) and *data array*
- ***Attribute***: Name, Value pairs
- **Parallel HDF5** supports MPI and MPI-I/O
  - Write one shared file in parallel
  - Using collective buffering parallel I/O can reach file-per-process performance



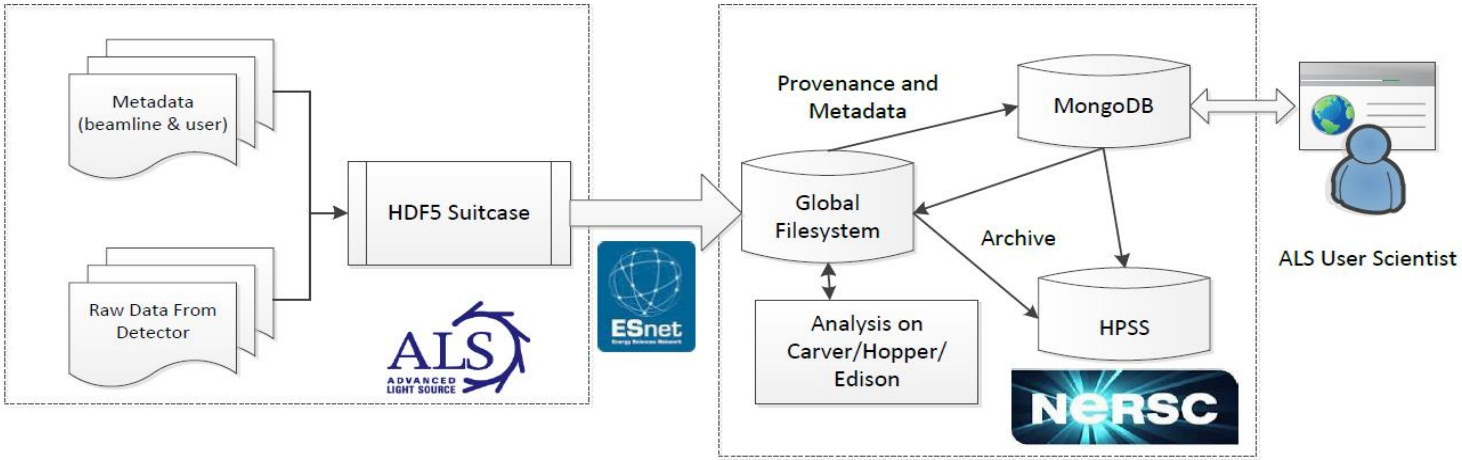






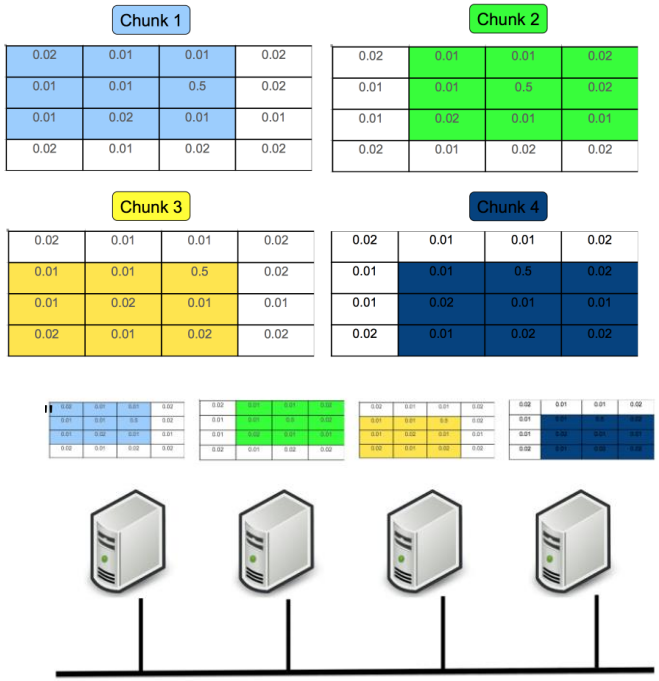
HDF5 – some use cases

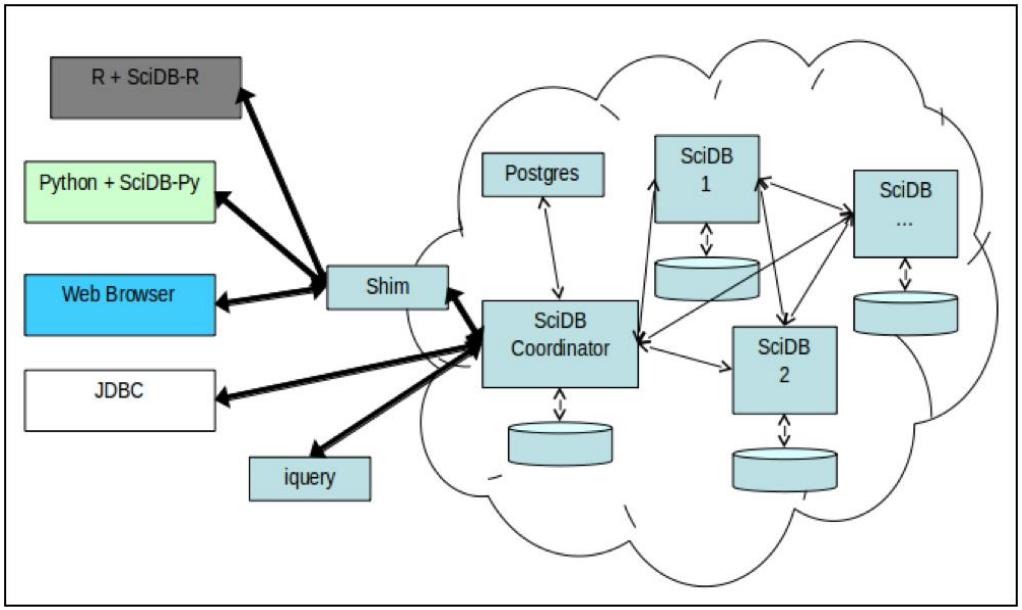
- Widespread use of HDF5 in large-scale massively parallel simulations
- Also allows for statistical analysis on large datasets
  - E.g Clustering of trillion particle simulation (Single 24TB HDF5 file) in 30 minutes (on 100k cores) [BD-CATS]
  - And *k*-Nearest Neighbors (KNN) on TB-sized O(100 billion records) simulation and experimental datasets in < 1 min [submitted to ICLR]
- And HEP-like experimental pipelines: e.g SPOT suite for Advanced Light Source



SciDB

- Open-source Array Database– supported by *Paradigm4*
- Shared nothing: scalable to 1000s of processors and petabytes of data
- SQL-like Array Query Language and friendlier Array Functional Language
- Optimised functions for e.g. Stats and Linear Algebra (ScaLAPACK),
- Extensible with User Defined Functions

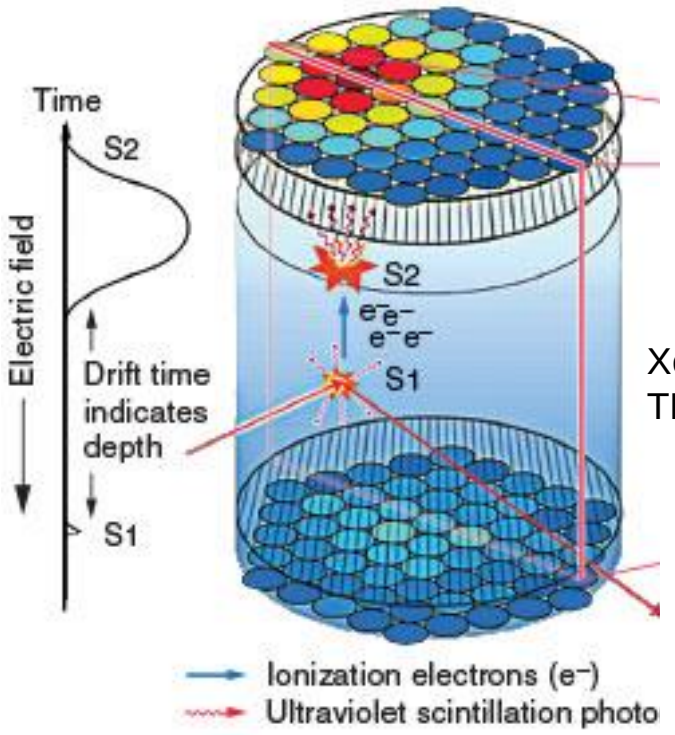




SciDB Use Case - LUX

Gerhardt et. al. *Accelerating Scientific Analysis with SciDB* (CHEP 2015)

- Signal is scintillation photon (S1) followed by ionization electron (S2)
- In SciDB stored as an EventArray and PulseArray
- To identify signal can:
  - Filter these arrays to find EventArray with 2 pulses and S1 and S2 within a 'cut' on photon counts
  - Then 'cross_join' to form table of events and properties
- Process 83m events (LUX 2013 Data) in 90s
  - On NERSC test-bed: 16 8-core Intel Xeon X5550 (2.67 GHz) nodes, 24 GB of RAM
  - Fast and less cumbersome than running on ~1m files in 10+TB Dataset



SciHDF5

S. Floratos, S. Blanas, S. Byna, B. Dong, Prabhat, K. Wu, Ohio State University and LBL

Combine benefits of

- high-level query language and optimized execution engine
- Standard file format for data exchange

User view

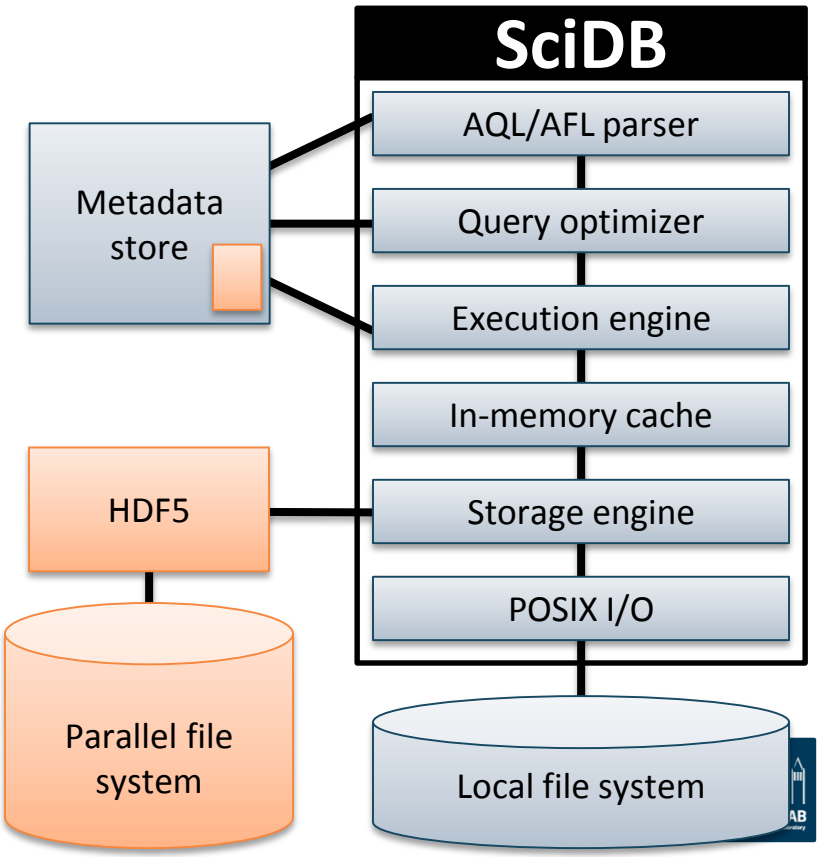
1

CREATE ARRAY USING HDF5 creates a SciDB object that points to a dataset in HDF5

2

Queries touching this SciDB object read/write data directly from the HDF5 array as needed

Architectural view



U.S. DEPARTMENT OF ENERGY | Office of Science

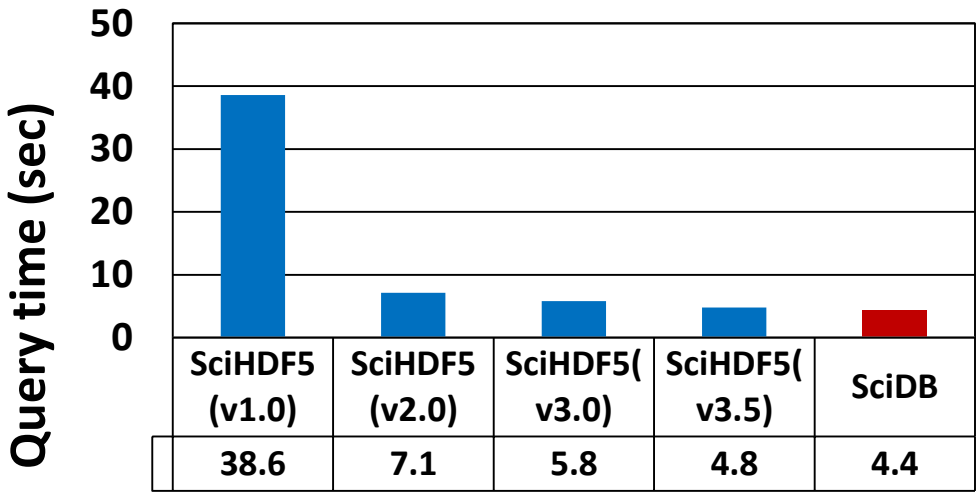
SciHDF5 - Preliminary results

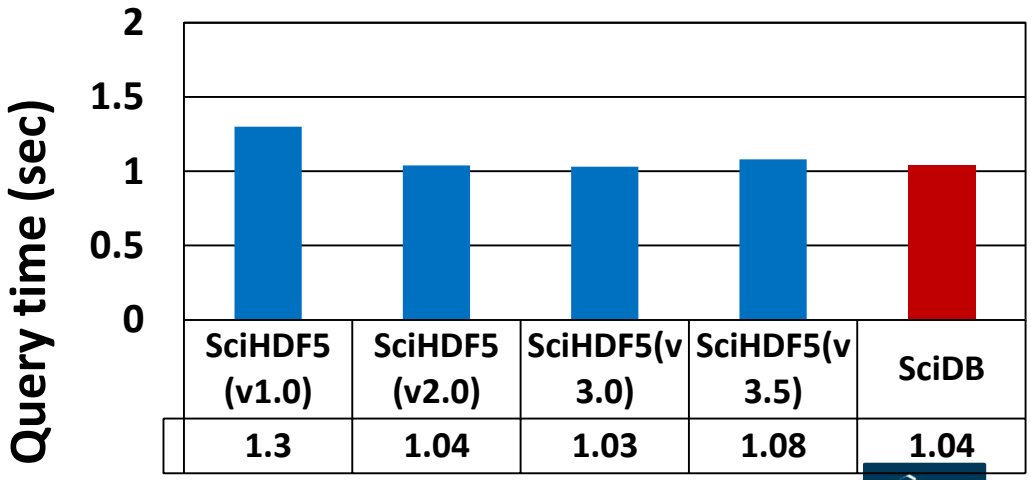
S. Floratos, S. Blanas, S. Byna, B. Dong, Prabhat, K. Wu, Ohio State University and LBL

- Dataset: 10GB array, random data
- Storage: 8MB stripe size, across 16 OSTs
- SciDB: 8 SciDB instances, 1 node, 8MB chunks

- Aggregation query: average of the entire array
- High-selectivity query: sum a 1000 x 1000 array subset

- Aproaching native SciDB performance
- Currently prototyping idea, work needed to deliver in production.





We currently deploy separate Compute Intensive and Data Intensive Systems

Compute Intensive

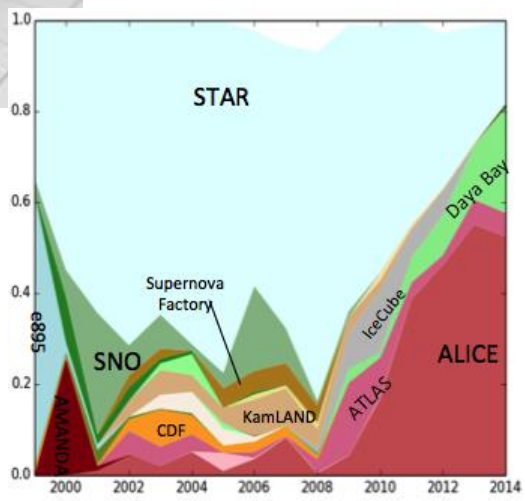
Data Intensive

PostgreSQL

SciDB

mongoDB

Genepool



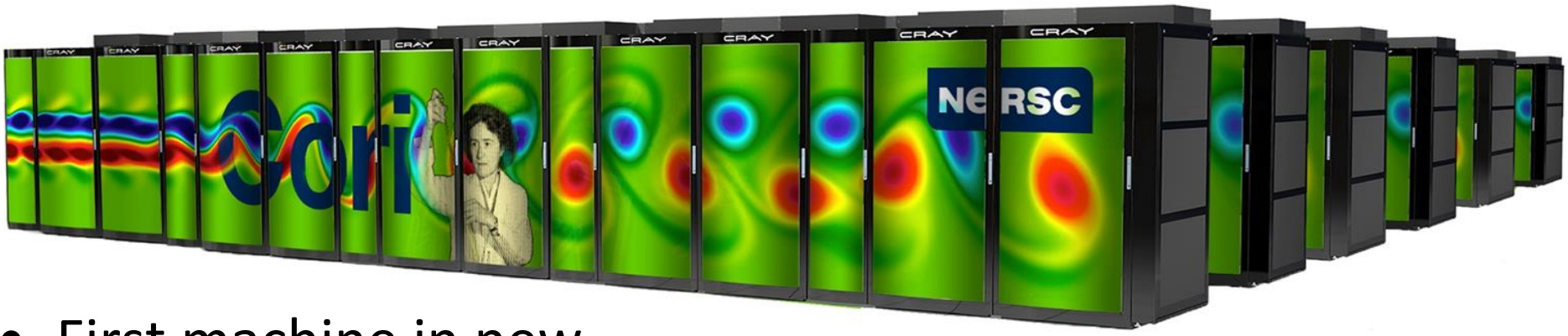
PDSF

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

The Cori System

- Cori will cater for HPC **and** HTC, support existing users of HPC (e.g. Edison) and HTC (e.g. Carver) but also enable **new** data workflows



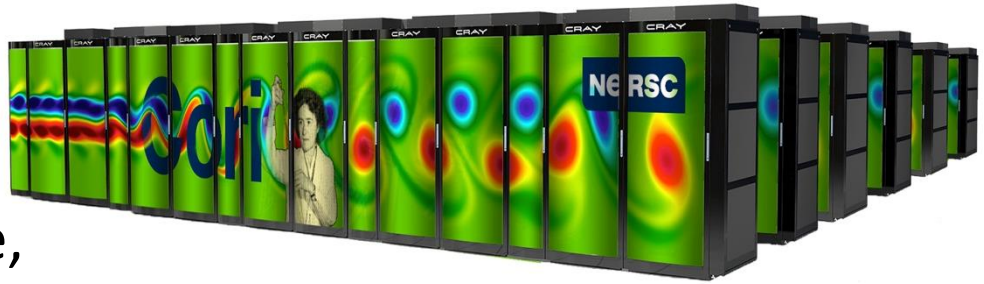
- First machine in new Computational Research and Theory (CRT) building

U.S. DEPARTMENT OF ENERGY | Office of Science

Cori Overview

- **Phase 2 (coming mid-2016) - over 9,300 'Knights Landing' compute nodes -** See talk by Katie Antypas tomorrow

Phase 1 (installed now):

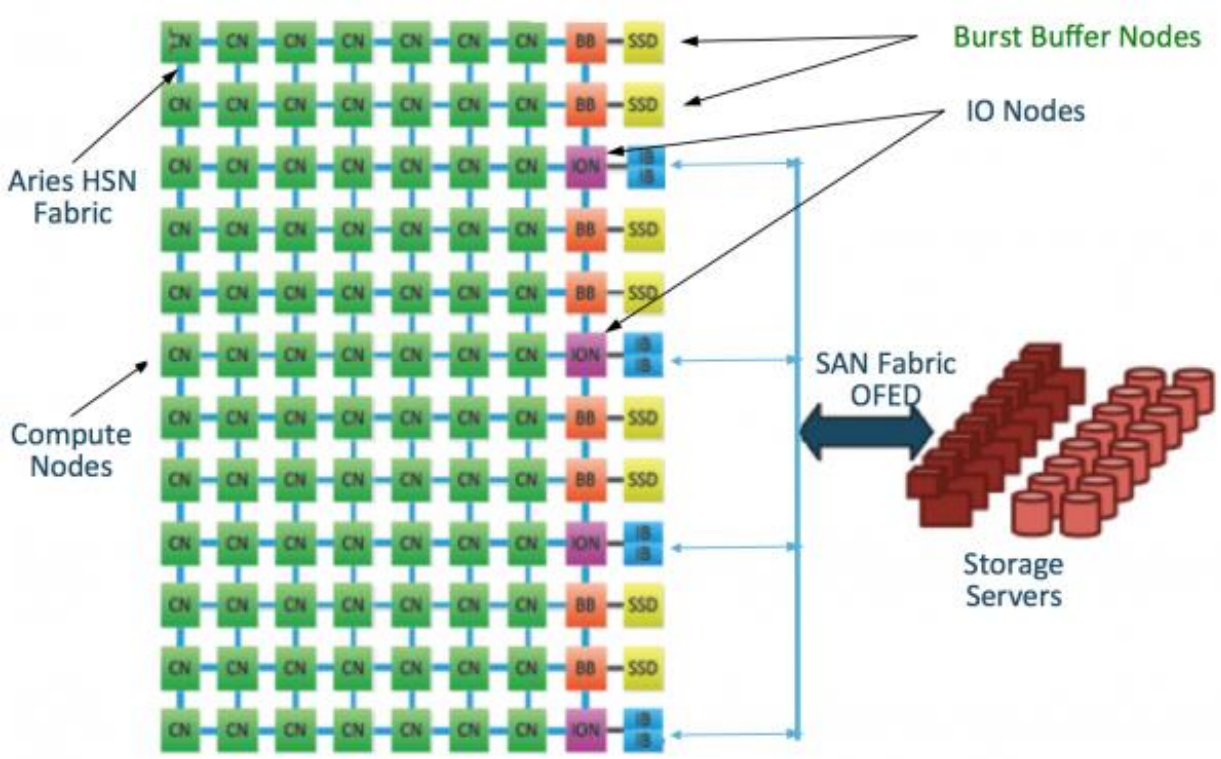


- 'Data partition'
- 1630 Compute Nodes
- Two Haswell processors/node,
  - —16 cores/processor at 2.3 GHz
  - —128 GB DDR4 2133 Mhz memory/ node(some 512 /768 GB)
- —Cray Aries high-speed "dragonfly" topology interconnect
- —High-performance networking external to the system too
- —SLURM batch system – more queues for 'realtime' (e.g. for experiment processing), 'shared' (e.g. serial), killable

- **Lustre File system (also installed now):** 28 PB capacity, >700 GB/sec peak performance

NVRAM based – 'Burst Buffer'

https://www.nersc.gov/users/computational-systems/cori/burst-buffer/



- ~1.5PB capacity, ~1.5TB/s for full Cori System
- Half with Phase 1 (144 nodes with 2x3.2TB SSD modules
- Available via SLURM batch system integration with Cray 'Data Warp' Software

IO improvements: high bandwidth reads and writes, e.g. checkpoint/restart; **high IOP/s**, e.g. non-sequential table lookup; **out-of-core** applications

Workflow performance improvements: coupling applications, using the BB as interim storage; Optimizing node usage by changing node concurrency part way through a workflow (using a persistent BB reservation)

Analysis and Visualization: In-situ / in-transit; Interactive (using a persistent BB reservation)

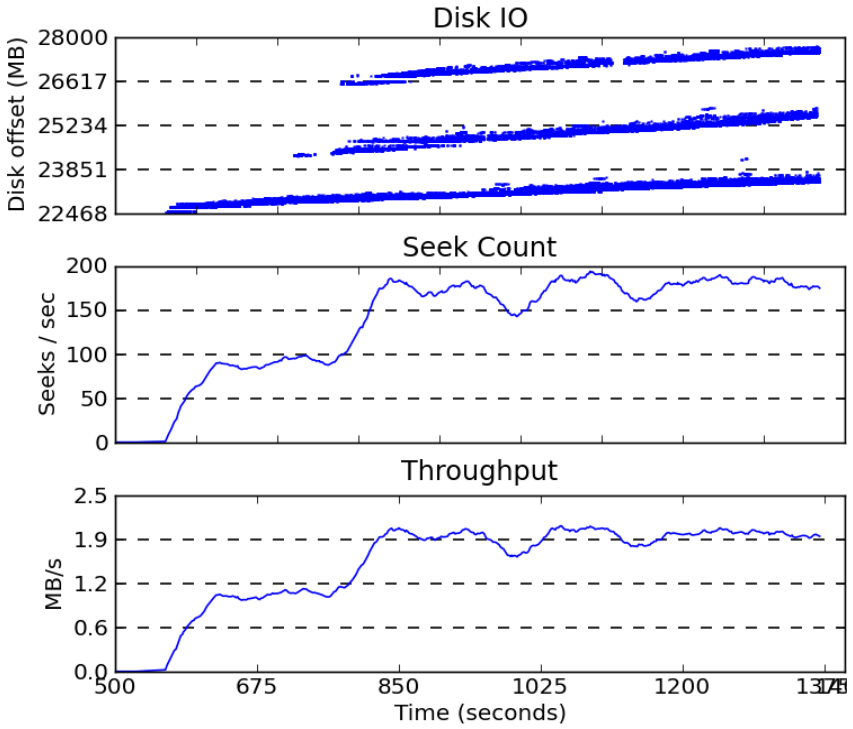
Burst Buffer – early user program

- *starting soon:* 28 proposals across science domains, e.g:

Project	DoE office	BB data features
Nyx/Boxlib cosmology simulations (*Ann Almgren, LBNL*)	HEP	I/O bandwidth with BB; checkpointing; workflow application coupling; in-situ analysis.
Phoenix: 3D atmosphere simulator for supernovae (*Eddie Baron, U. Oklahoma*)	HEP	I/O bandwidth with BB; staging intermediate files; workflow application coupling; checkpointing.
Chombo-Crunch + Visit for carbon sequestration (*David Trebotich, LBNL*)	BES	I/O bandwidth with BB; in-situ analysis/visualization using BB; workflow application coupling.
Sigma/UniFam/Sipros Bioinformatics codes (*Chongle Pan, ORNL*)	BER	Staging intermediate files; high IOPs; checkpointing; fast reads.
XGC1 for plasma simulation (*Scott Klasky, ORNL*)	Fusion	I/O bandwidth with BB; intermediate file I/O; checkpointing.
PSANA for LCLS (*Amadeo Perazzo, SLAC*)	BES/BER	Staging data with BB; workflow management; in-transit analysis.

Burst Buffer - performance

- How ATLAS once was: (CHEP Taipei 2010)
  - –Old unordered AODs
  - –SATA HDD tops out at 100-200 IOPS
  - –Various improvements to ATLAS file layout made SSDs not cost effective then
- NERSC Burst Buffer
  - –Each 3.2 TB module has around 80k IOPS and we have 288 of those
  - –E.g. 4.5k processes write/read of 4.5 TB file in 4k transfers with 1G seek: O(10M) total IOPS



Conclusions

- Other science domains face similar challenges to HEP and employ similar solutions
  - –But with different tools
- HEP has been at the forefront of science data management for a while
  - –Other domains are not using (or able to use or even aware of) many of its tools
  - –They have a variety of database, I/O, storage solutions – touched on a few here
- Some more technical interchange is desirable

Extra Slides

Burst Buffer Software Development Timeline

Phase 3

- BB-node functionality: In Transit, filtering

Phase 2

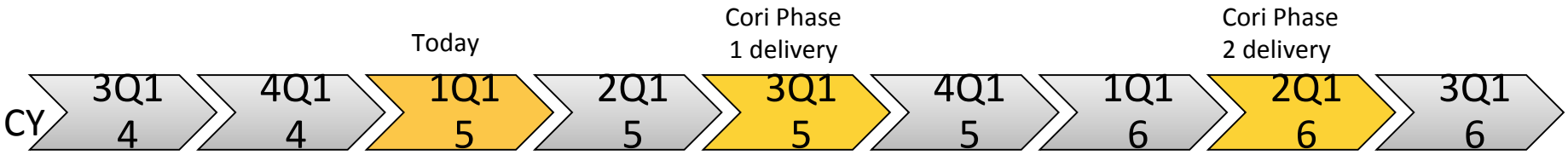
- Usability enhancements: Caching mode

Phase 1

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

Phase 0

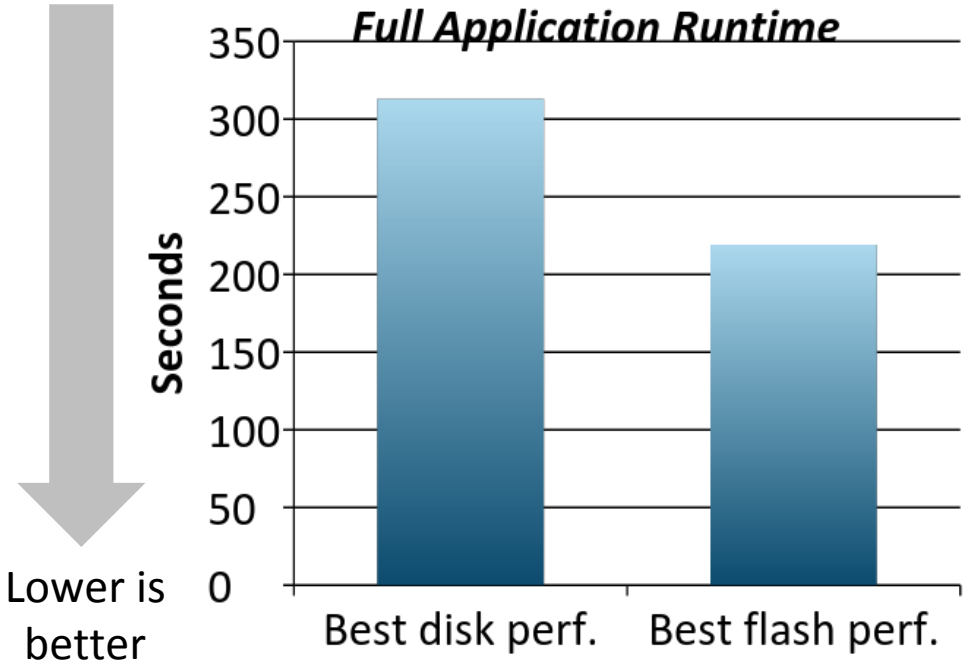
- Static mapping of compute to BB node
- User responsible for migration of data

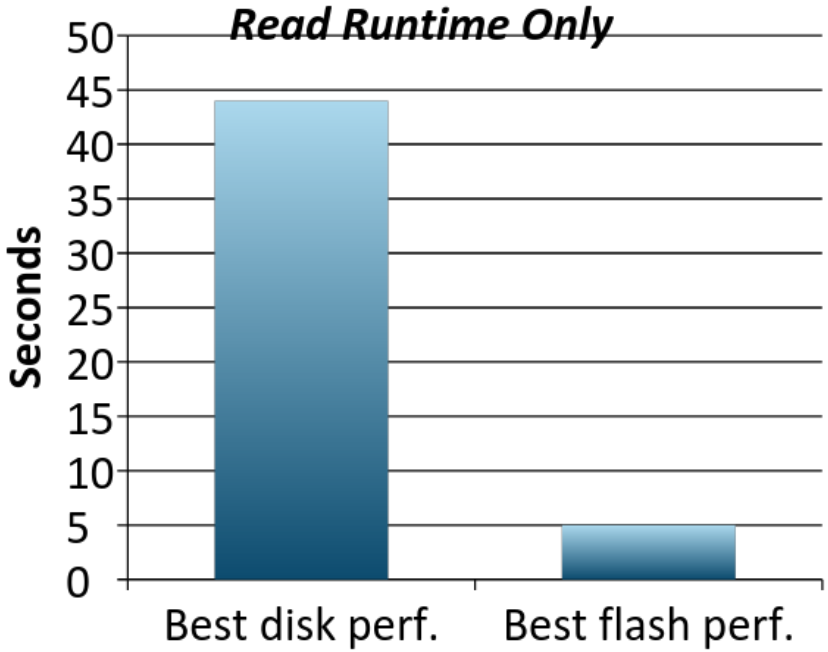


Outline

- **Introduction to NERSC**
- **Scientific Data and Analysis at NERSC**
- **Some data store technologies and users**
  - HDF5
  - SciDB (…and SciHDF5)
  - Burst Buffers

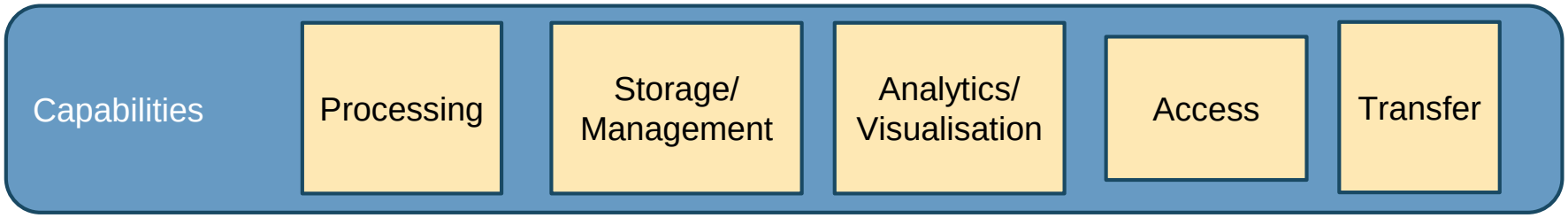
TomoPy performance comparison between flash and disk file systems





- **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)**

Top Data Management Challenges



1. **Streaming Data from LCLS/ALS to Cori** - Making use of 100Gbps links, Software Defined Networking
2. **Interactive query of LUX/LZ data** - O(10)TB/s, time-series, aggregations, statistics
3. **Scalable data management backend for LSST** - O(100) TB, search, spatial subsetting, full-sky analytics
4. **In-situ, In-transit analysis for VPIC/H3D** - O(1T) particles, statistical summary, visualization
5. **Seamless on-site data movement for climate simulations** - Burst Buffer, Lustre, HPSS, workflows
6. **Business analytics on streaming sensor data for NERSC operations** - Fault detection, workload prediction, removing bottlenecks
7. **Reproducible science for LHC /experimental particle physics** - Collaborative analysis, data sharing
8. **Fluent data organisation for all NERSC Users** - Immediate access to users file information and metadata
9. **Seamless off-site data movement across centres for cosmology simulations -** Virtual data facility ; data federations

MongoDB

- **'NoSQL', document-oriented database**
- JSON-like documents (key: value)
- Queries are javascript expressions
- Memory-mapped files – queries can be fast
- Though not configured for very frequent/ high-volume writes or very many connections
- **Good For:**
  - Un-Structured Data ('Schema-less')
  - Mid-Size to Large, e.g. 10 GB of Text

Materials Project Gateway