BERKELEY LAB

LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Data Elevator – Low-contention Data Movement in Hierarchical Storage Systems

Bin Dong, **Suren Byna**, John Wu, Prabhat, Hans Johansen, Jeffrey Johnson, Noel Keen

Lawrence Berkeley National Laboratory

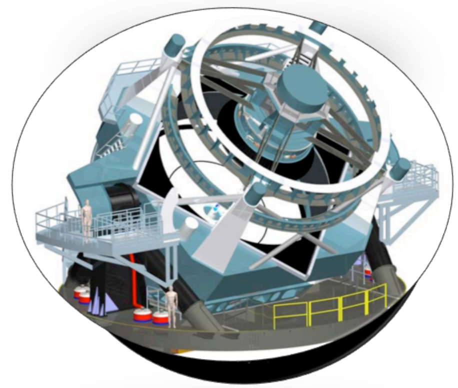
Contact: SByna@lbl.gov

HiPC 2016, Dec 20th, 2016

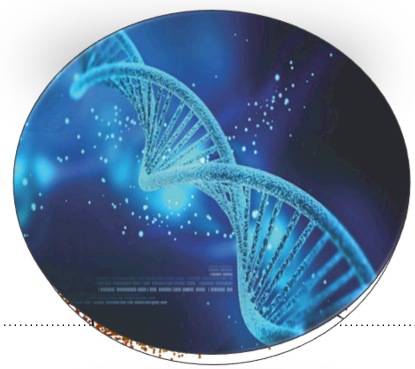
Data-driven science

- Simulations
  - Multi-physics (FLASH) – 10 PB
  - Cosmology (NyX) – 10 PB
  - Plasma physics (VPIC) – 1 PB
- Experimental and Observational data
  - High energy physics (LHC) – 100 PB
  - Cosmology (LSST) – 60 PB
  - Genomics – 100 TB to 1 PB
- Scientific applications rely on efficient access to data
  - Storage and I/O are critical requirements of High Performance Computing (HPC)

FLASH LHC



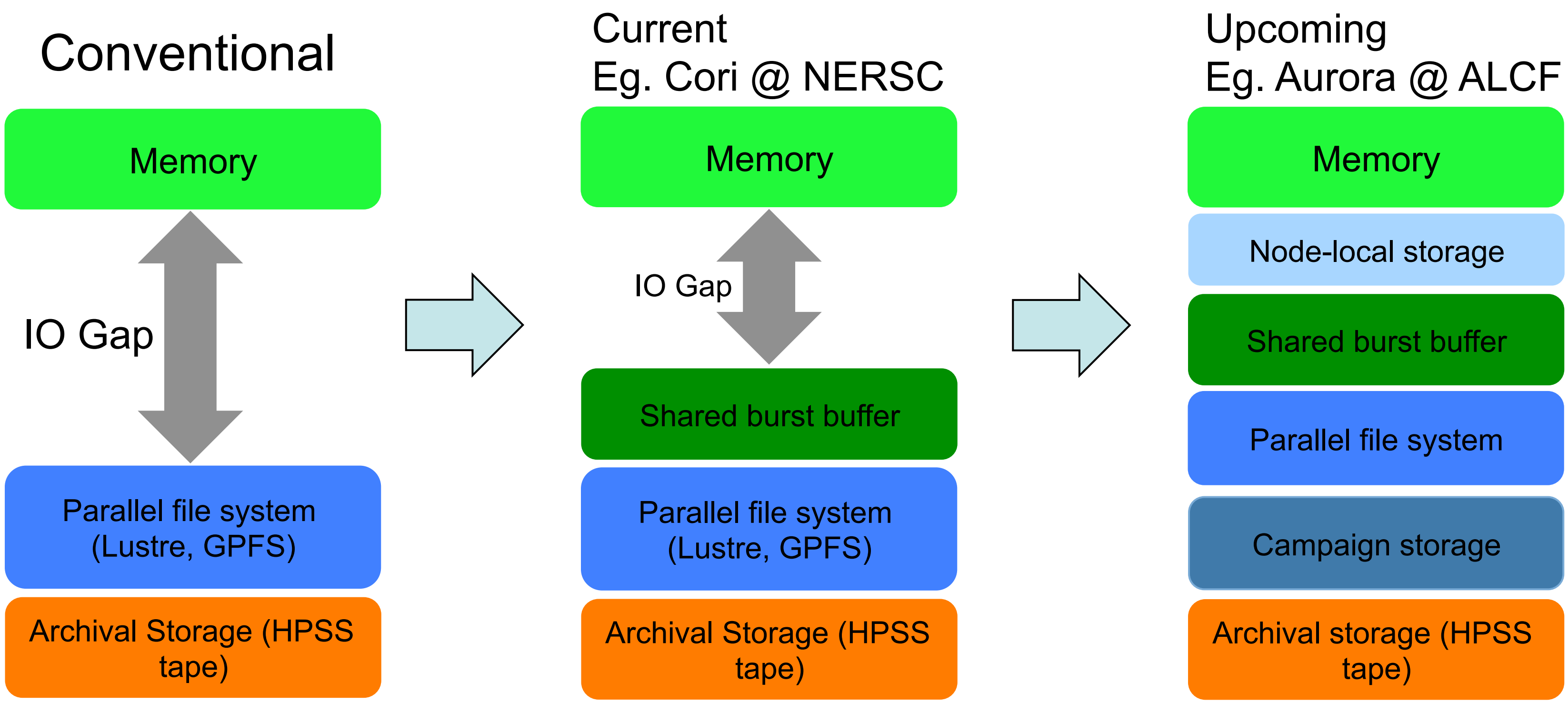
NyX LSST



VPIC Genomics

BERKELEY LAB

Storage system transformation in HPC

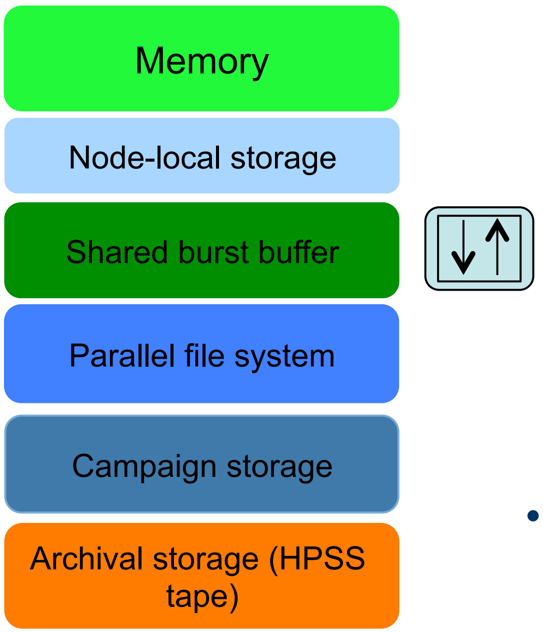


- IO performance gap in HPC storage is a significant bottleneck because of slow disk-based storage
- SSD and new memory technologies are trying to fill the gap, but increase the depth of storage hierarchy

Challenges of deep storage hierarchy

- **Modes of moving data between the layers**
  - Offline: Stage out after writing the data
    - E.g., Cray DataWarp provides ***stage_in*** and ***stage_out*** commands
  - In applications: API for moving data
    - Cray DataWarp provides an API for moving data in and out of burst buffers
  - Transparent caching
    - Burst buffer servers move data transparently
- **Challenges**
  - *Inefficiency* : Existing methods for staging in/out data to/from burst buffers (BB) compete for resources on BB servers
  - *Burden on users* : Users or applications have to explicitly make the data movement decisions, which could lead to inefficiency
  - *Limited to one level*: Transparent caching is aware of a single level storage

Our solution: Data Elevator for moving data

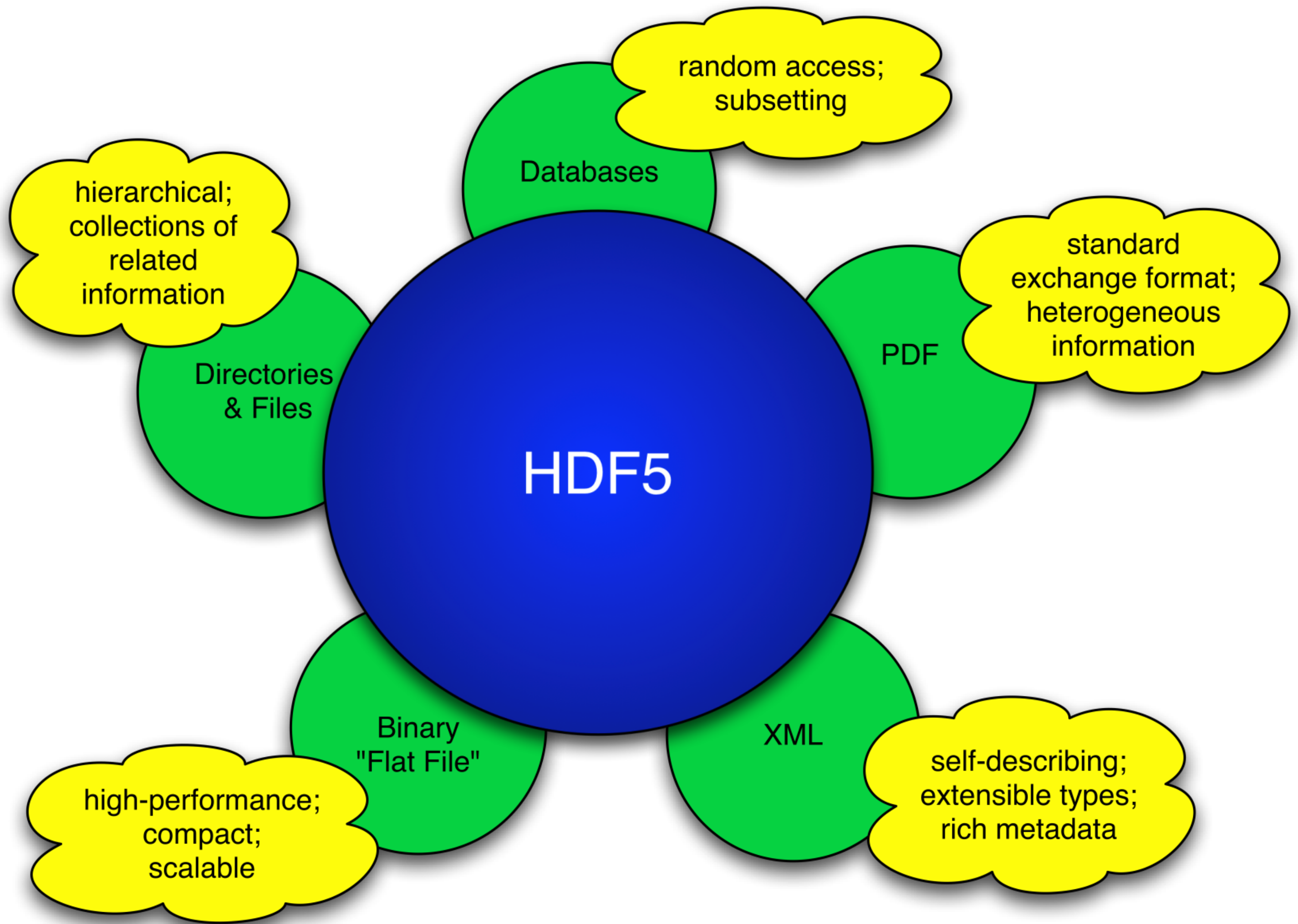


- **Contributions**
  - Low-contention data movement library for hierarchical storage systems
  - Offload of data movement task to a few compute nodes or cores
  - Data Elevator on NERSC's Cori system
    - With a couple of science applications, demonstrated that Data Elevator is **4X** faster than Cray DataWarp ***stage_out*** and **4X** faster than writing data to parallel file system
- **Benefits of using Data Elevator**
  - **Transparent data movement**: Applications using **HDF5** specify destination of data file and the Data Elevator transparently moves data from a source to the destination
  - **Efficiency**: Data Elevator reduces contention on BB
  - **In transit analysis**: While data is in a faster storage layer, analysis can be done in the data path

Background – HDF5

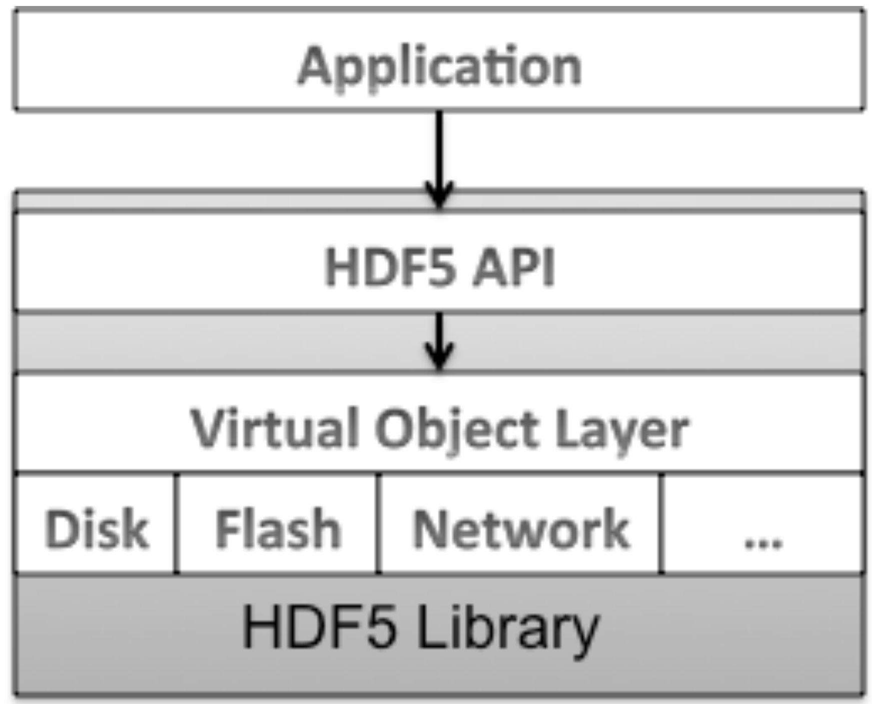
- **HDF5** – Hierarchical Data Format, v5 developed and maintained by The HDF Group
  - First version of HDF5 released in 1998
- Open file format
  - Designed for high volume or complex data
  - Parallel I/O library
- Open source software
  - Works with data in the format
- A data model
  - Structures for data organization and specification

HDF5 is like …


random access;
subsetting
Databases
hierarchical;
collections of
related
information
Directories
& Files
standard
exchange format;
heterogeneous
information
PDF
HDF5
Binary
"Flat File"
XML
high-performance;
compact;
scalable
self-describing;
extensible types;
rich metadata

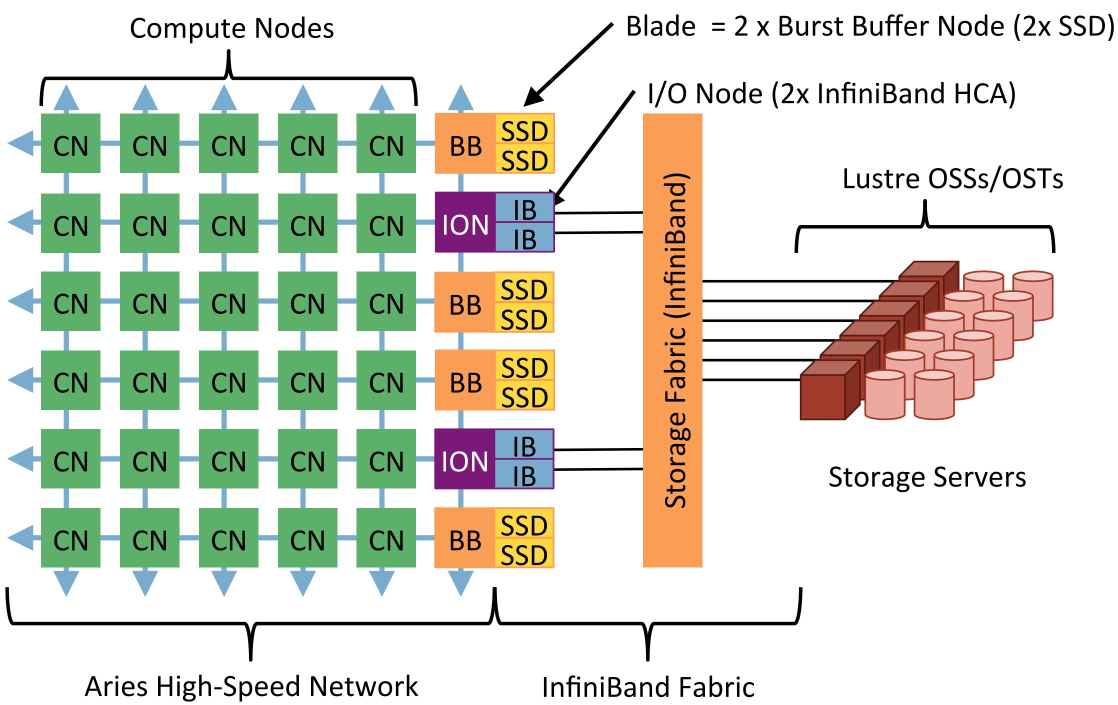
Background – HDF5 Virtual Object Layer (VOL)

- **Data Elevator uses VOL for intercepting HDF5 calls**
- **VOL**
  - Abstract HDF5 object storage to enable developers to easily use HDF5 on novel storage systems
  - Binary instrumentation approach allows intercepting HDF5 calls without code changes
  - Allows all HDF5 applications to migrate to future storage systems and mechanisms with no source code modifications



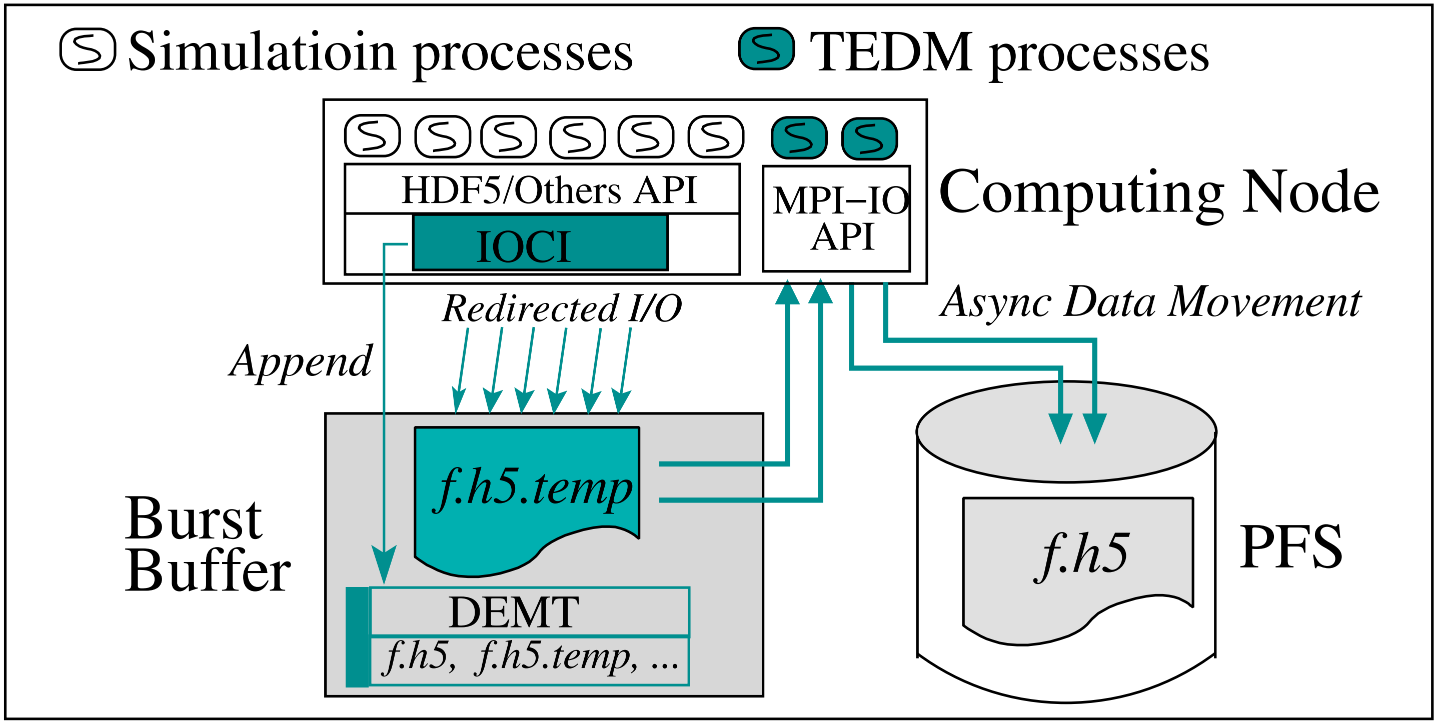
BERKELEY LAB

Background – Burst buffer on Cori system



- **Fact sheet**
  - Cori is a Cray XC40 system
  - 144 burst buffer nodes
  - Each server has two Intel P3608 3.2 TB NAND Flash SSDs
  - Cray DataWarp® manages the burst buffer
    - *stage_in* and *stage_out* commands
    - BB API for programmatically move data
      - Allows async data transfers

Data Elevator design

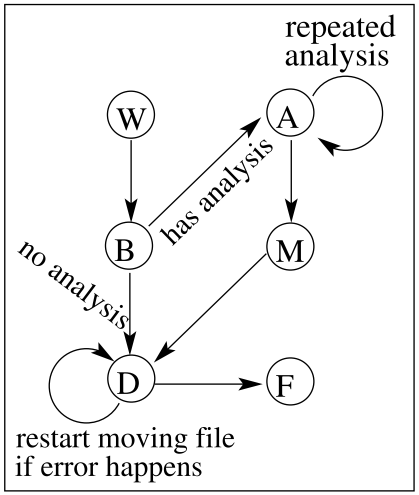


- **Implementation challenges**
  - Transparently intercepting I/O calls
  - Moving data between storage layers efficiently w/ low contention
- **Solutions**
  - IOCI – IO Call Interceptor library - VOL
  - Transparent & Efficient Data Mover processes – Concurrent MPI job

Metadata for managing the state of data

- Metadata Table to manage the data movement status
  - Data written to BB
  - Data is written to BB
  - Request to analyze data and start analysis
  - All data reads are done
  - Data is being written to PFS
  - Data is moved to PFS

Status	Description
W	Start writing to BB
B	Finish writing to BB
A	Start analysis
M	Finish analysis
D	Start moving to PFS
F	Finish moving to PFS



Optimizations

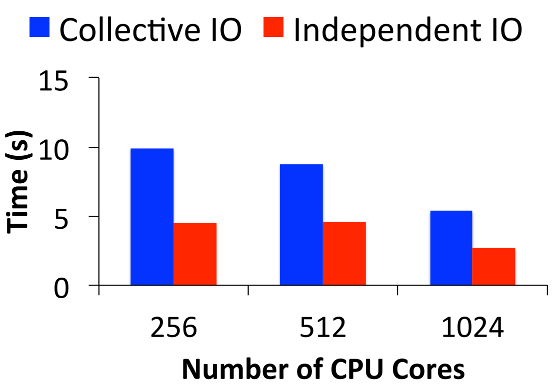
- Scalable and low-contention parallel data movement
  - Data Elevator processes run on compute nodes
    - Allows scaling up or down the number of data movement processes
    - BB server resources are entirely used for I/O
- Overlapping data reads from BB and writes to PFS
  - Data is written to file system in chunks
  - Allows reading data from BB and writing to PFS can be overlapped
- Stripe size alignment
  - Parallel file systems, such as Lustre, provide striping optimizations
    - Stripe size, stripe count, alignment, etc.
- In transit analysis, while data is in a burst buffer level
  - Analysis jobs can poll the metadata table for availability of data

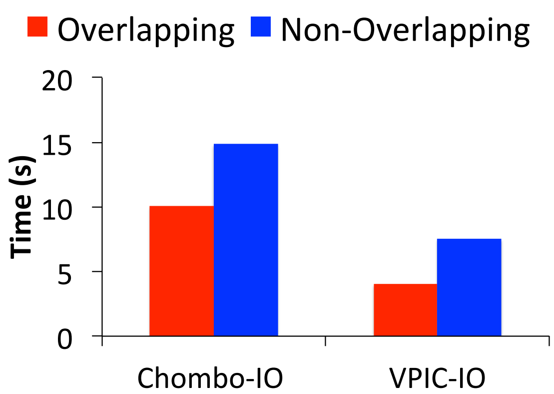
Experimental set up

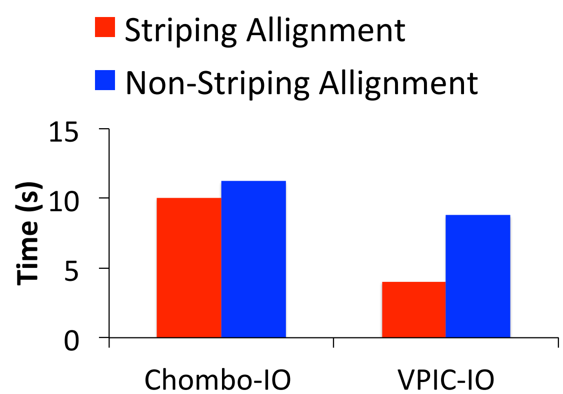
- Platform – NERSC Cray XC40 system, Cori
- Benchmarks and applications
  - VPIC – Plasma Physics code simulating magnetic reconnection (solar weather)
  - CAMR – Climate Adaptive Mesh Refinement code simulating climate at high resolutions (1km resolution)
- Metrics
  - **End-to-end execution time** – Total execution time of the application, including I/O and data movement
  - **End-to-end data movement time** – Time to move data from memory to Lustre file system (final destination of the data)

Data Elevator – Tuning space exploration

- MPI-IO Collective (two-phase) vs. Independent modes
- Overlapping BB reads w/ PFS writes
- Striping alignment on PFS

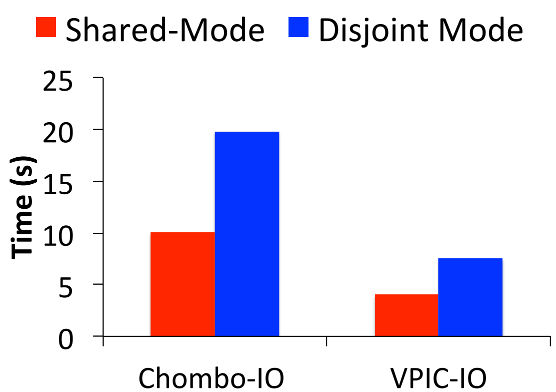


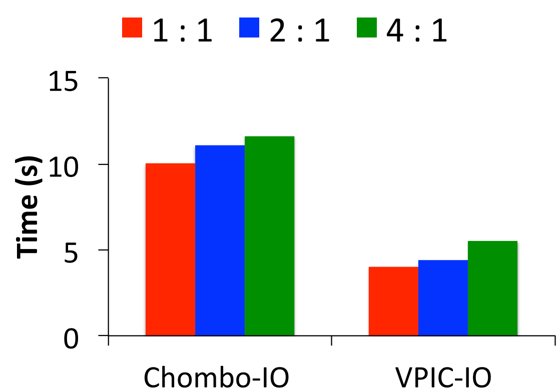


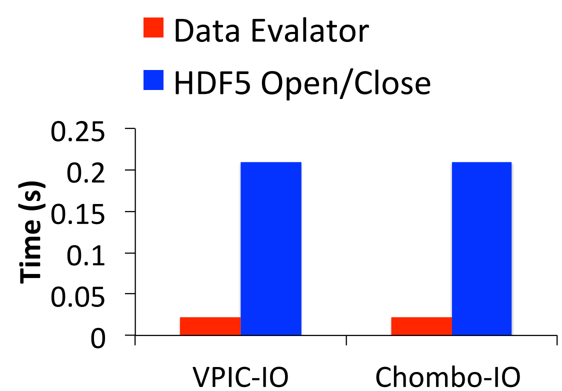


Data Elevator – Tuning space exploration

- Sharing compute cores vs. dedicated Data Elevator nodes
- Data Elevator size
- Metadata overhead

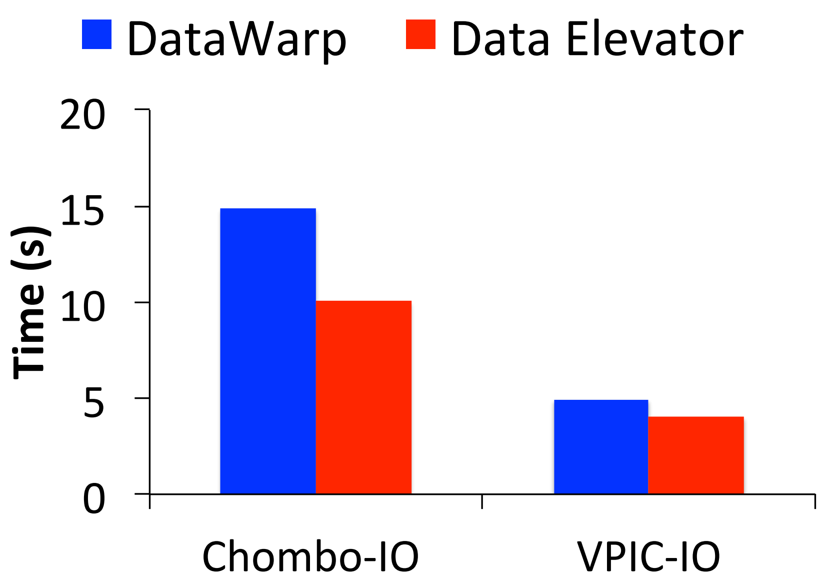


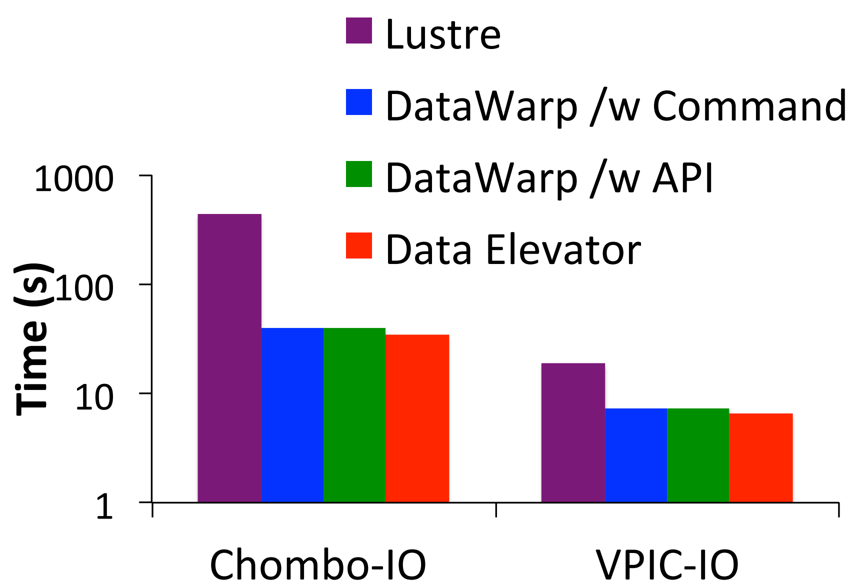




Performance comparison with benchmarks

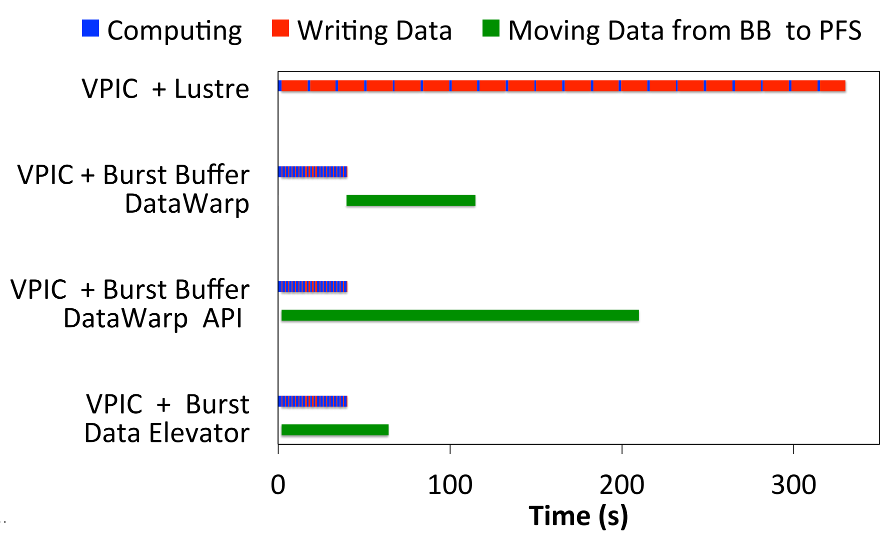
- Staging out performance – Cray DataWarp vs. Data Elevator
  - 1K procs
  - DataWarp – 144 nodes
  - Data Elevator – 64 processes
  - Data Elevator is faster than DataWarp by 14% to 22%



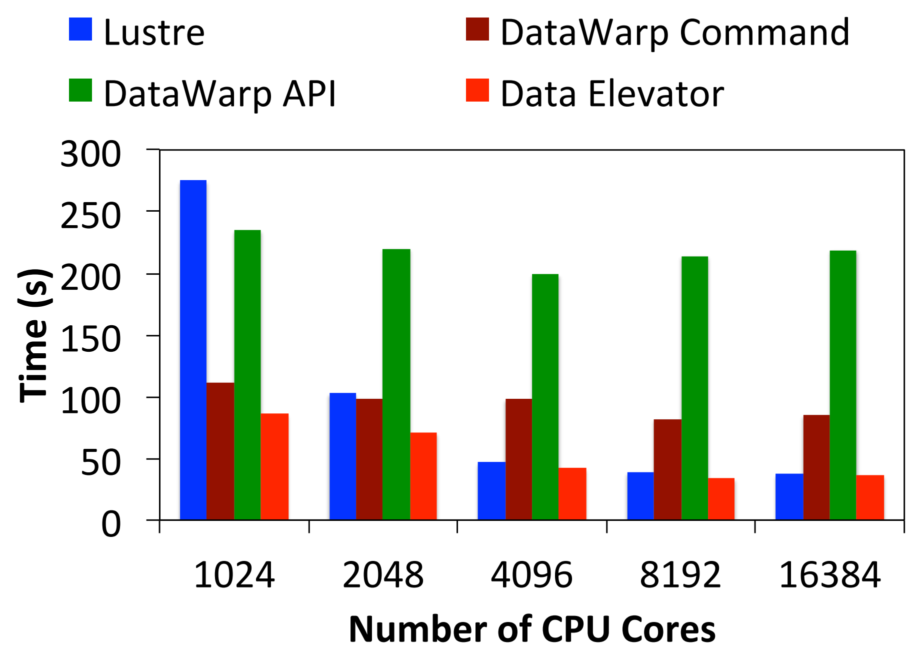


Performance with Plasma physics simulation

- Total execution time of running VPIC code for 20 time steps, writing a file at the end of each time step – data write intensive workload
- Data Elevator
  - 1.7X faster than PFS
  - 1.8X faster than DataWarp ***stage_out*** command
  - 4.2X faster than DataWarp API

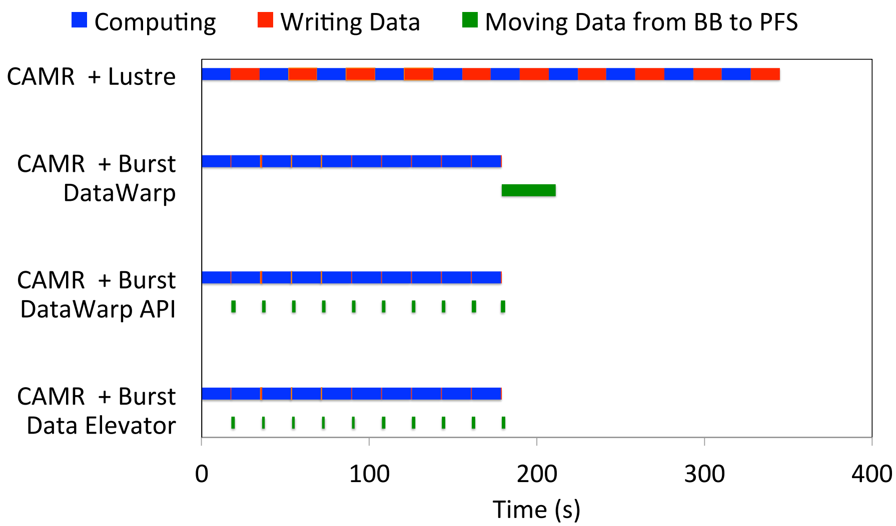


1K cores

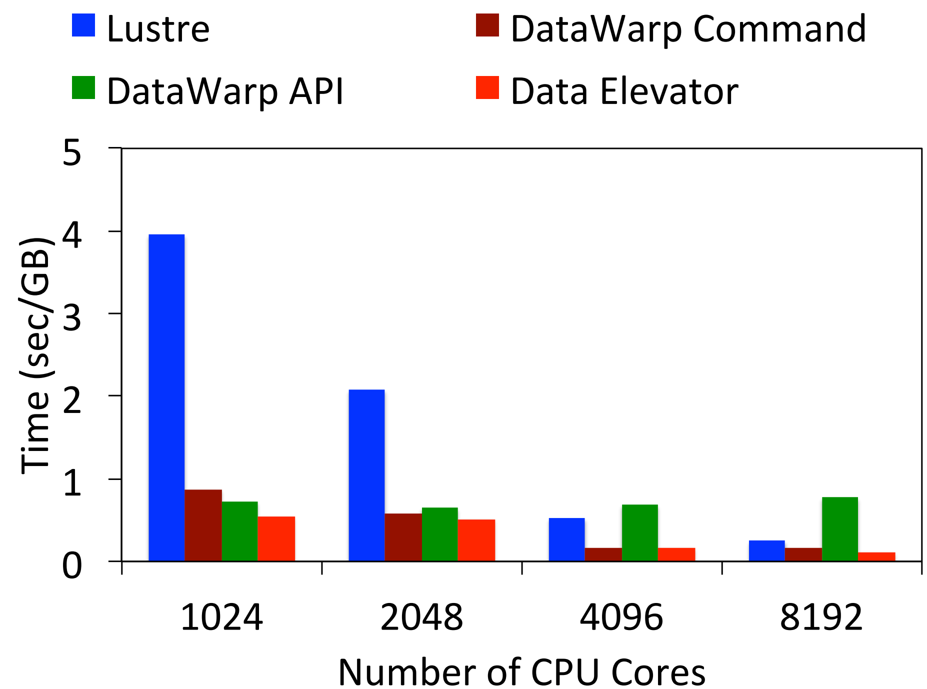


Performance with climate simulation

- Total execution time of running CAMR code for 20 time steps, writing a file at the end of each time step – compute intensive workload
- Data Elevator
  - 4X faster than PFS
  - 1.2X faster than DataWarp ***stage_out*** command
  - 3.3X faster than DataWarp API

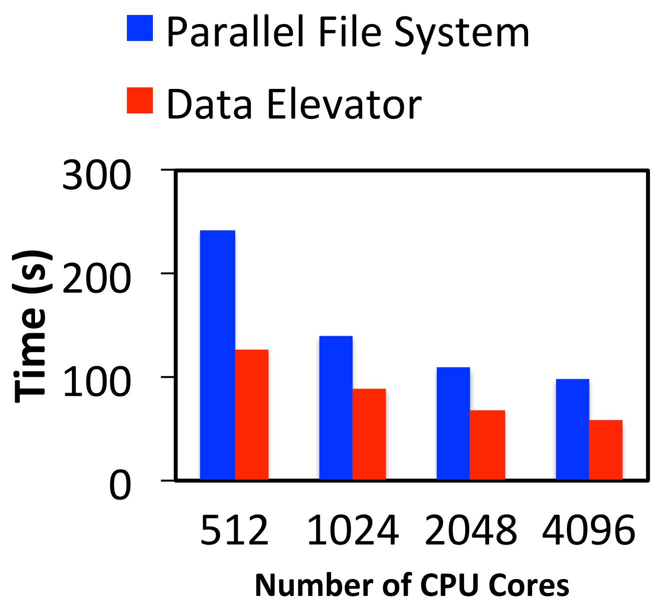


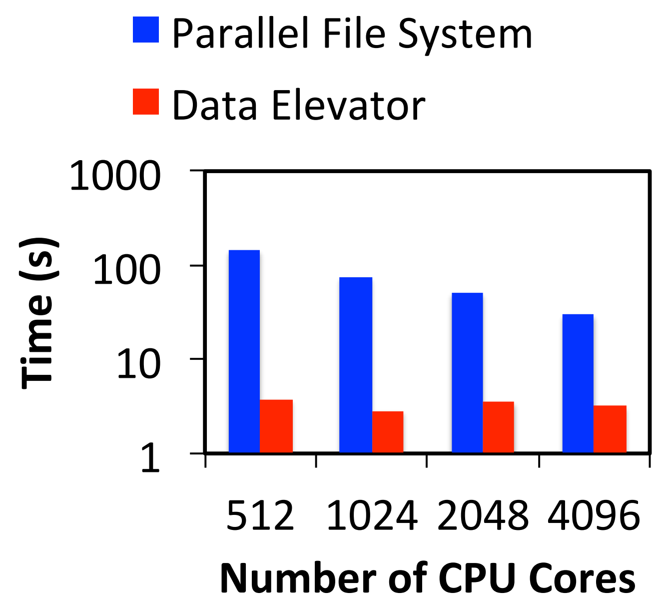
1K cores



Performance of *in transit* data analysis - Querying

- Querying data while it is in BB
  - Indexing is 2X faster
  - Querying is 6.5X faster





Conclusions

- Moving data in hierarchical storage needs to be:
  - Efficient and cause low contention on BB servers
  - Transparent transfers without burden on users and/or app developers
- Data Elevator achieves these goals
  - … for writing data to PFS
- Future work
  - Caching and prefetching for data reads
  - More than two levels (node-local, campaign, and archival storage layers)
  - Tuning of energy consumption and compute node efficiency

Contact: Suren Byna, LBNL, (SByna@lbl.gov)

Thanks to: U.S. DEPARTMENT OF ENERGY | Office of Science, NERSC, The HDF Group, BERKELEY LAB