SC19

HDF5 BOF

Hosted by

The HDF Group

LBNL

https://hdfgroup.org/sc19

BERKELEY LAB
Lawrence Berkeley National Laboratory

The HDF Group

Agenda

- HDF5 Roadmap and community engagements
- Opening HDF5 with performance optimizations for new architectures
  - Virtual File Drivers
  - Virtual Object Layer (VOL) and VOL connectors
- Exascale I/O and EOD
- Community presentations
  - Use case details and how HDF5 is used
  - What's working with HDF5?
  - What improvements are needed to HDF5 to facilitate your use case?
- Open discussion

HDF5 Roadmap and Community Engagement

HDF5 Roadmap

- Focus on:
  - Performance
  - Enabling applications to access data in the Cloud and Object Store
  - Facilitating convergence of HDF5 HPC and Cloud applications
  - Enabling ML with data in HDF5

- Upcoming releases
  - 1.12.0 by 12/31/2019
    - **Webinar on 1.12.0 features on 12/6/2019 @11.00am Central**
    - 1.12.* every six months
  - 1.10.6 by 12/31/2019
    - 1.10.* every six month
  - 1.8.22 Summer 2020
    - *We plan to drop 1.8.**
    - Start migrating now!

Community Engagement

- HDF5 Source https://bitbucket.hdfgroup.org and Issues database JIRA https://jira.hdfgroup.org are open
  - Requires free registration at https://hdfgroup.org
- Watch for THG webinars and training announcements
- HDF5 European Workshop for Science and Industry @ESRF in Grenoble, France, September 2019
  - h5py coding camp
- Plan to have a Workshop in Washington, DC in 2020

Community Engagement

- Collaboration with Unidata and netCDF-4 projects
  - See “PIO and NetCDF” presentation
    - Enabling scalability by doing I/O by the subset of processes
  - We will be working with netCDF-4 developers on enabling parallel compression in netCDF-4
- Collaboration with HDFql team
  - See “HDFql – An easy way to manage HDF5 data”
    - Reads and writes HDF5 data; supports parallel I/O and OpenMP for post-processing
    - Data and metadata querying capabilities
    - Available on major platforms and supports C, C++, Jaba, Python, C#, Fortran, R
- HDF5 C++ Users Group
  https://www.hdfgroup.org/2019/01/hdf5-c-webinar-followup/

Opening up HDF5 with performance optimizations for new architectures

HDF5 VFDs and VOL connectors

HDF5 Library Architecture (1.12.0)

HDF5 API and language bindings

Virtual Object Layer (VOL)

Pass-through VOL connectors

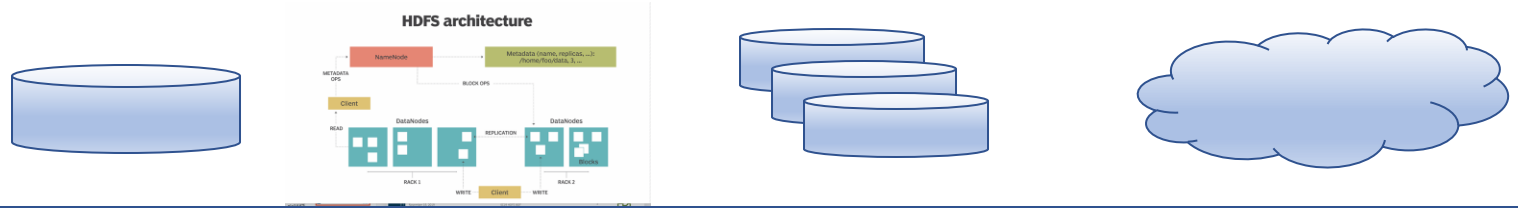
HDF5 Core Library

Native Connector

VFDs

POSIX | MPI I/O | SWMR | .... | S3 | HDFS

REST | DAOS | .... | Data Elevator | Hermes



BERKELEY LAB
Lawrence Berkeley National Laboratory

The HDF Group

HDF5 Virtual File Drivers

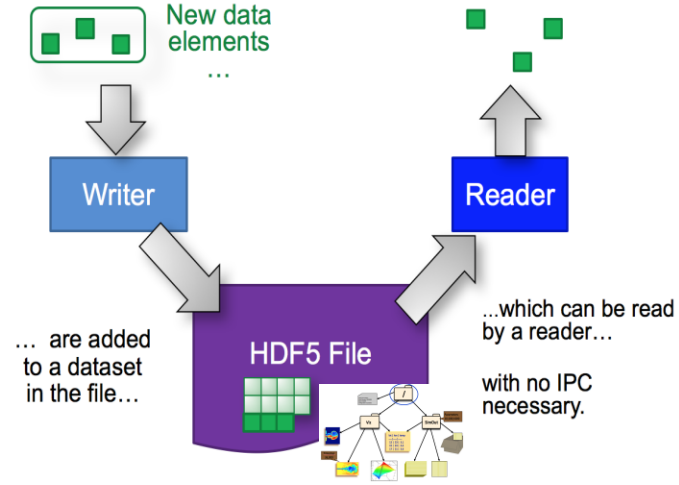
What is new?

NEW VFD Drivers

- S3 (1.12.0)
  - Read-only; enables access to HDF5 file stored in S3 bucket
  - Optimized read-only S3 (1.12.*)
  - Read-write S3
- HDFS
  - Enables access to HDF5 file stored on HDFS (1.12.0)
- Splitter and Mirror VFDs
  - Makes HDF5 data available on a remote system for reading while creating a file on a local system
  - https://www.hdfgroup.org/2019/10/webinar-followup-new-hdf5-features-coming-in-2020-2021/

VFD SWMR (Single Writer/ Multiple Reader)

- Planned for 2021
- Solid prototype is available now
- Problem
  - Address limitations of the current SWMR implementation
    - Allow *modifications of file structure* (i.e., writer can add or delete groups, datasets, and attributes)
    - Support VL datatypes
    - Relax the order in which writer and readers come in
    - Remove POSIX semantics requirement for write atomicity
      - NFS support
    - *Guarantee maximum time from write to availability for read under assumption that FS is fast enough.*
    - Remove performance penalty for applications and minimize maintenance cost for the feature



HDF5 VOL Connectors

https://bitbucket.hdfgroup.org/projects/HDF5VOL

Virtual Object Layer (VOL)

- Abstraction Layer within HDF5 Library
  - Redirect I/O operations into VOL “connector”, immediately after an API routine is invoked
- VOL Connectors
  - Implement “storage” for HDF5 objects, and “methods” on those objects
    - Dataset create, write / read selection, query metadata, close, …
  - Able to be transparently invoked from a dynamically loaded shared library, without modifying application source code (or even rebuilding the app binary)
  - Can be stacked, to allow many types of connector to be invoked
    - “Pass-through” and “Terminal” connector types

HDF5 API

Operations on a Container

All Other HDF5 Operations

Virtual Object Layer (VOL)

Pass-through

Async.

Independent Metadata

Data Elevator

Python Adapter

....

Terminal

Native

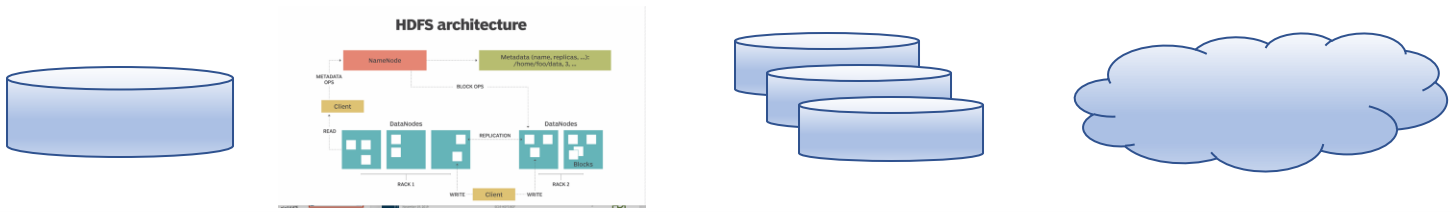
REST

DAOS

Hermes

....

HDF5 Library Infrastructure



Asynchronous HDF5 Operations VOL Connector

- Allows asynchronous operations for HDF5 applications:
  - Implicit: For unmodified applications; transparently invoked by setting environment variable; conservative asynchronous behavior
  - Explicit: For applications that want more control of async operations; can extract more performance using async operations that return "request tokens" to app
- Pass-through VOL connector
  - Allows asynchronous operations for *any* underlying HDF5 VOL connector
- Implemented w/background thread, using Argobots
- Can give up to 9x I/O performance improvement to applications
- See Tang's PDSW Paper:
  - https://sc19.supercomputing.org/proceedings/workshops/workshop_files/ws_pdsw109s2-file1.pdf

Independent Metadata Modification (IMM) VOL Connector

- Allows HDF5 metadata operations to be performed independently
  - Currently, all HDF5 metadata modification operations must be collective
    - Dataset / group creation & deletion, attribute create, write, etc.
  - Each metadata modification is "voted on" by other HDF5 processes writing to that file using non-blocking communication channels, then committed to the file
- IMM is a pass-through VOL connector
  - Allows IMM operations for *any* underlying HDF5 VOL connector
- Connector is extendible to multiple comm. channels: MPI, ZeroMQ, POSIX, etc.
- Async and IMM connectors demonstrate the power of pass-through VOL connectors to modify behavior of HDF5 library

H5Py VOL

- Contributed by PierLauro Sciarelli pierlauro.sciarelli@bsc.es
- H5PyVol allows implementation and binding of Python VOL
  - HDF5 plugin: a shared library bridging C and Python calling back and forth from the HDF lib to virtual object layers
  - Python package: provides a high level generic abstract data model to be filled for concrete implementation of VOLs.
- H5PyVOL has already been used to bind HDF5 basic operations to dataClay – Ceph Rados – Minio – OpenIO – Openstack Swift
- Can be used to interface HDF5 with any kind of external data store (object store, key-value, database, custom file format, etc...).

H5Py VOL FAQ

Q: Is H5PyVOL usable **JUST** with h5py (Python HDF5 library)?

NO: it is language-independent and works with any HDF5 application

Q: Can it be used with MPI?

YES: both applications-side and VOL-side

Q: Which operations are currently supported?

All basic r/w operations on File, Group and Dataset objects

Q: Which version of HDF5 is supported?

HDF5 1.12.0-alpha0 (H5PyVOL is aligned with HDF5 develop branch)

Q: Is it open source?

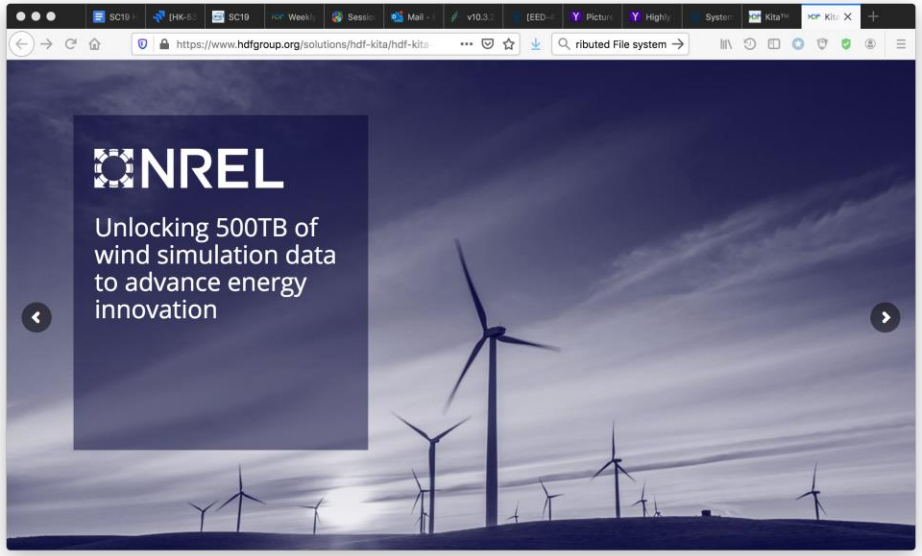
YES, it is free and open source: github.com/pierlauro/h5pyvol

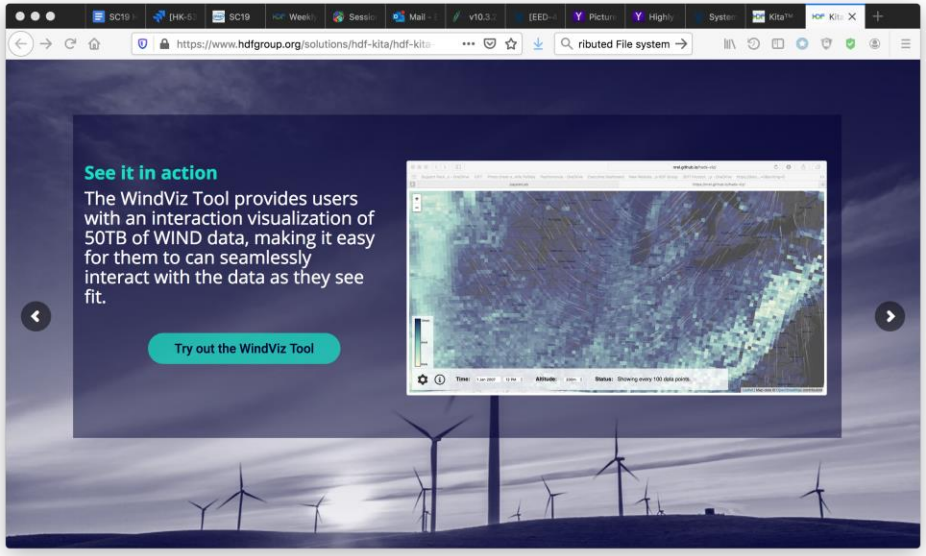
Q: When is the first stable version planned for?

Early 2020

HDF5 REST VOL Connector

- REST VOL works with Highly Scalable Data Server (HSDS) to access unlimited amount of data stored in S3
  - REST VOL https://bitbucket.hdfgroup.org/projects/HDF5VOL/repos/rest/browse
  - HSDS https://github.com/HDFGroup/hsds





HDF5 DAOS VOL Connector

- Want to learn about first production ready HDF5 VOL connector to Object Store?

  3nd annual DAOS User Group Meeting

  November 20 from 3:00 to 5:00 pm MT.

  Mt. Columbia Room at the Grand Hyatt Denver, 1750 Welton Street, Denver.

- HDF5 DAOS VOL connector will be released by 12/31/19

New HDF5 Features for Exascale and Experimental and Observational Data

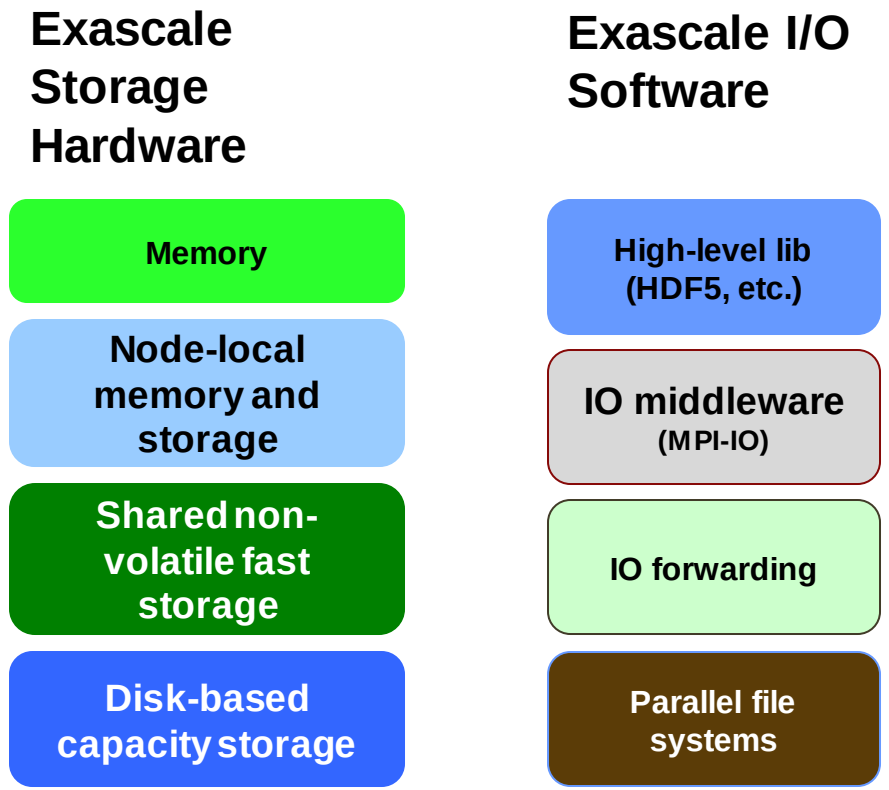
Suren Byna

Quincey Koziol, Scot Breitenfeld, John Mainzer, Venkat Vishwanath

ECP ExaIO, EOD-HDF5, and LCLS-LBNL Analytics (LLana) project teams

Exascale I/O architectures and software

- Exascale storage hardware
  - Deepening hierarchy with:
    - Fast node-local storage and storage-class memory
    - Shared SSD-based storage layer
    - Disk-based capacity storage
- I/O software
  - High-level self-describing I/O libraries (HDF5, etc.)
  - Middleware (MPI-IO) and optimization layers
  - File systems (Lustre and Spectrum Scale/GPFS) and object storage (Intel DAOS)
- Challenges
  - Heterogeneity of storage devices and distributed
  - Disparity of I/O software stack
  - Overhead of managing metadata in self-describing formats
  - Obtaining sustained I/O performance on exascale storage

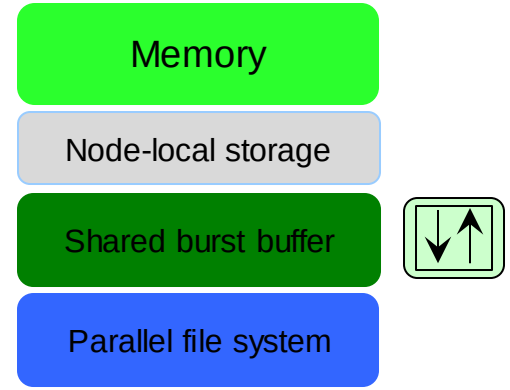


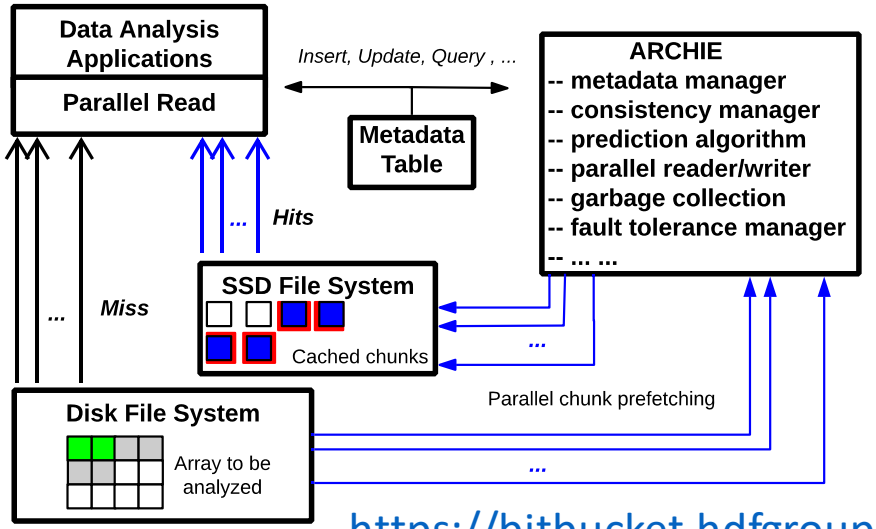
ECP HDF5 Features

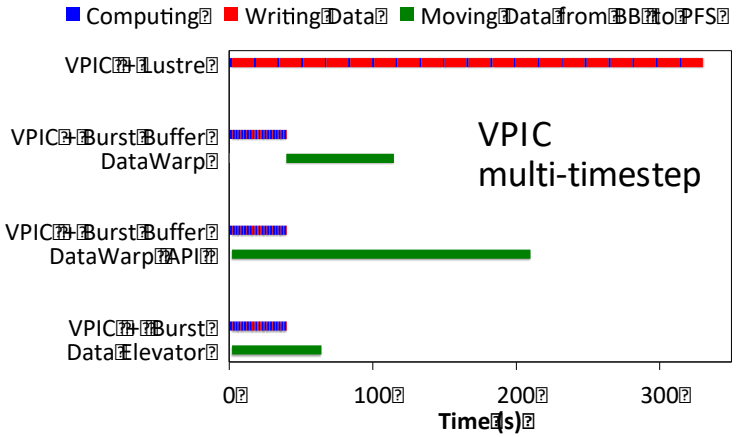
- VOL (Presented earlier)
- HDF5 Data Elevator VOL connector
- Asynchronous I/O
- Topology-aware I/O
- ECP application use cases

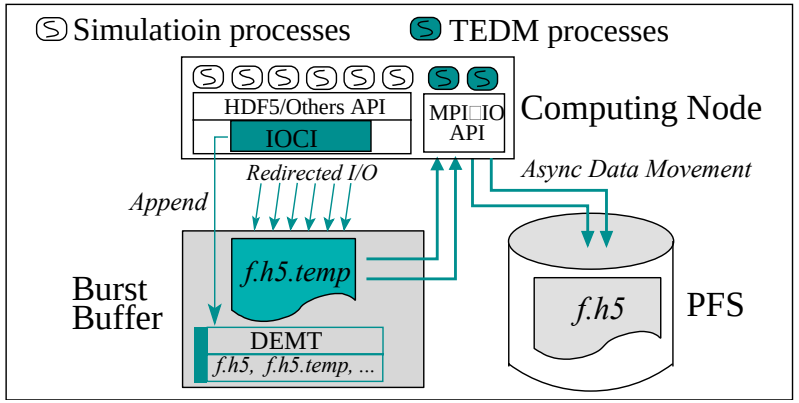
Burst buffer usage - HDF5 Data Elevator

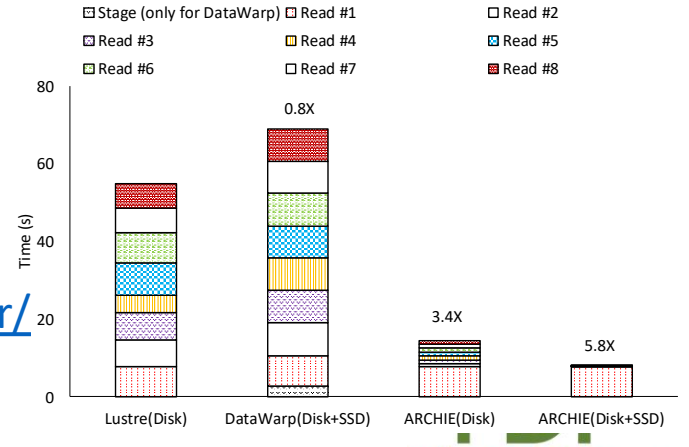
- Data Elevator VOL connector
  - Transparent data movement in storage hierarchy - writes and reads
  - Intercepts file opens, write, read, and close function calls and places data in burst buffers temporarily; DE moves data asynchronously
  - Prefetches predicted chunks of data to burst buffer or memory
  - In situ data analysis capability using burst buffers
  - Phase 2 plan includes extending capabilities of Data Elevator for node-local storage











https://bitbucket.hdfgroup.org/projects/HDF5VOL/repos/dataelevator/

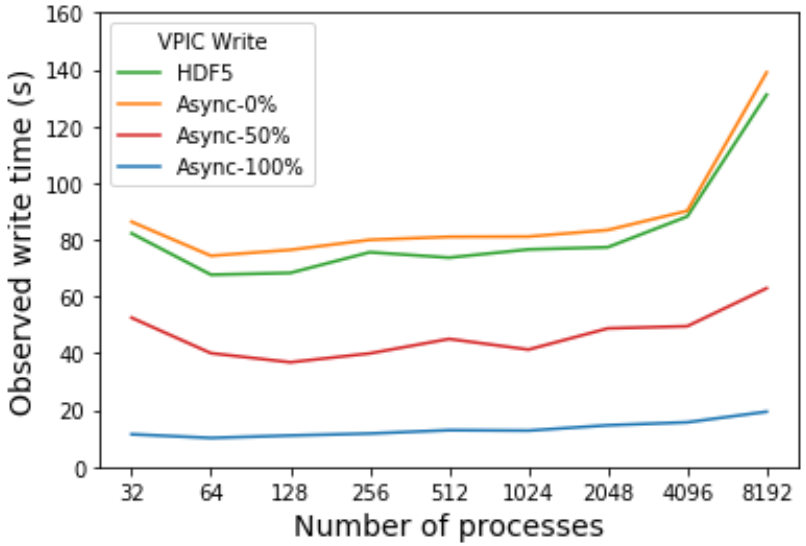
BERKELEY LAB
Lawrence Berkeley National Laboratory

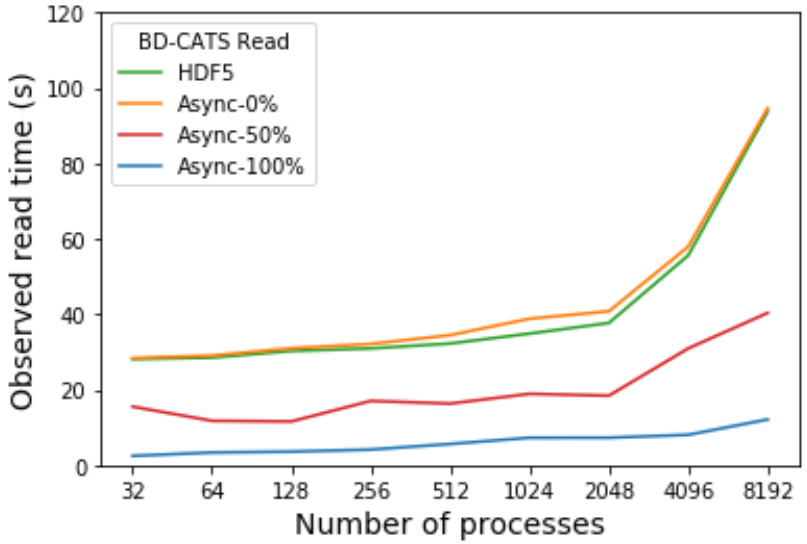
ECP EXASCALE COMPUTING PROJECT

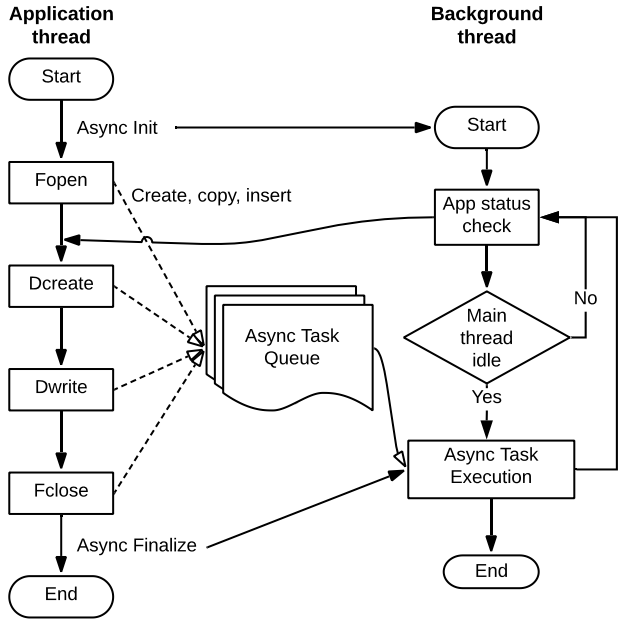
The HDF Group

Asynchronous I/O with HDF5

- Asynchronous I/O allows an application to overlap I/O with other operations
- The asynchronous I/O feature has been implemented as a VOL (Virtual Object Layer) connector, without requiring major change the HDF5 library



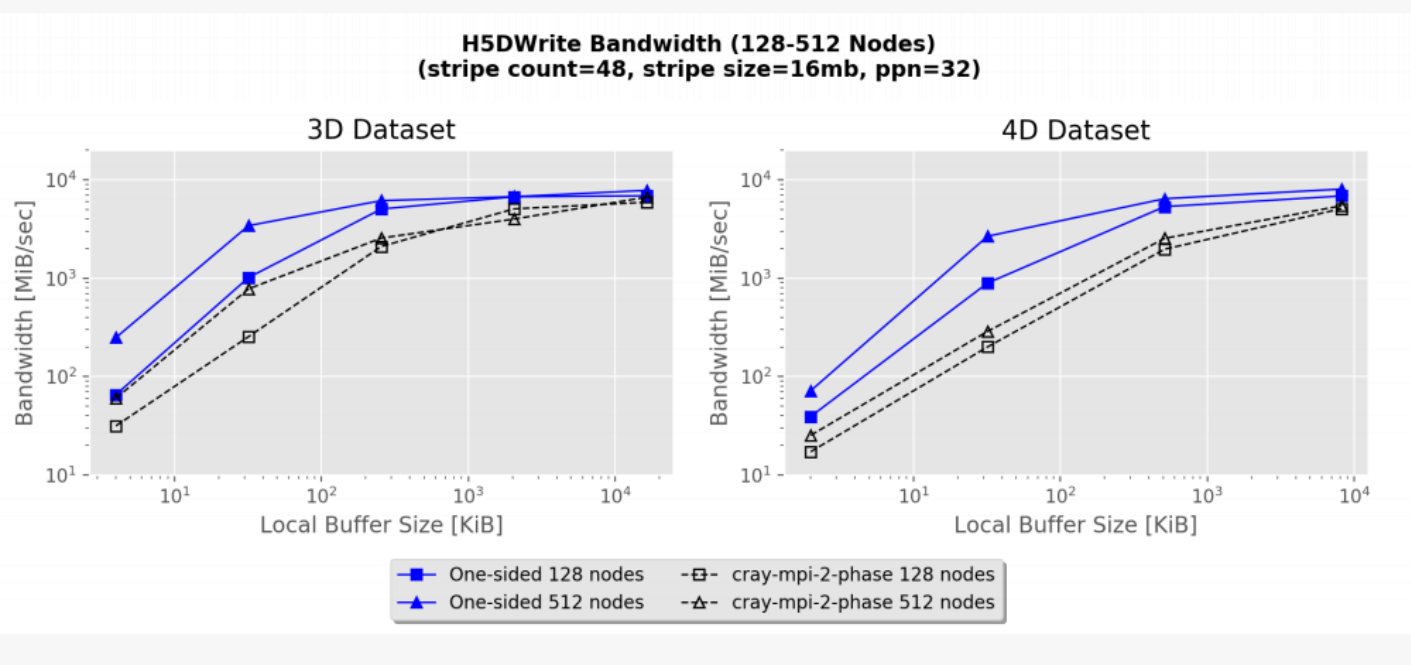




https://bitbucket.hdfgroup.org/projects/HDF5VOL/repos/async/

HDF5 Topology-aware I/O

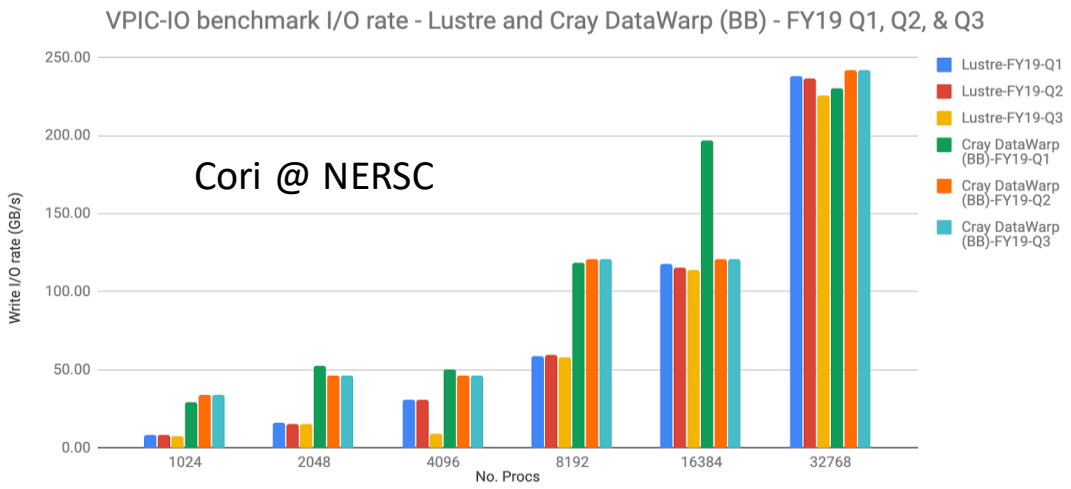
- Taking advantage of the topology of compute and I/O nodes and network among them improves overall I/O performance
- Developing topology-aware data-movement algorithms and collective I/O optimizations within a new HDF5 virtual file driver (VFD)
- In Phase 2, system-aware I/O will use node-local storage for data aggregation



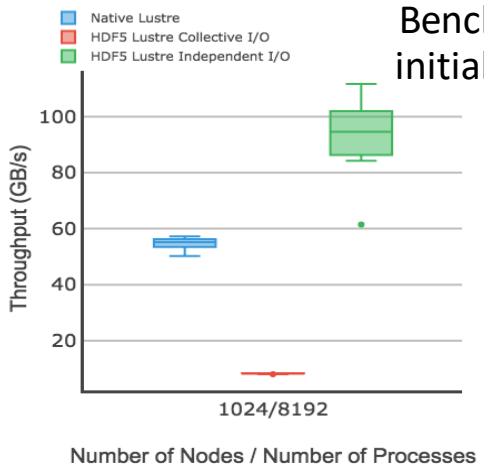
Performance comparison of the new HDF5 VFD, using one-sided aggregation, with the default binding to Cray MPICH MPI-IO. Data was collected on Theta using an I/O benchmarking tool (the HDF5 Exerciser),

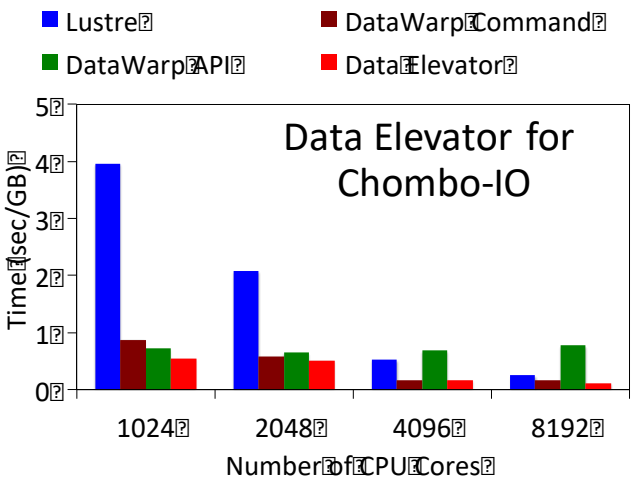
Implementation: **CCIO** branch
https://bitbucket.hdfgroup.org/projects/HDFFV/repos/hdf5/

HDF5 performance and ECP application use cases

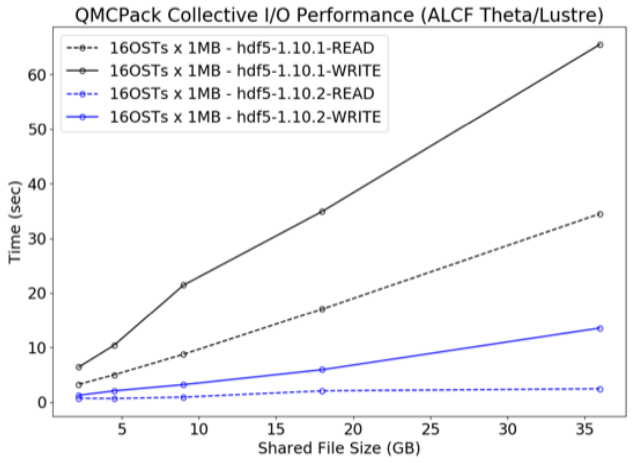


AMReX I/O Benchmark initial tuning



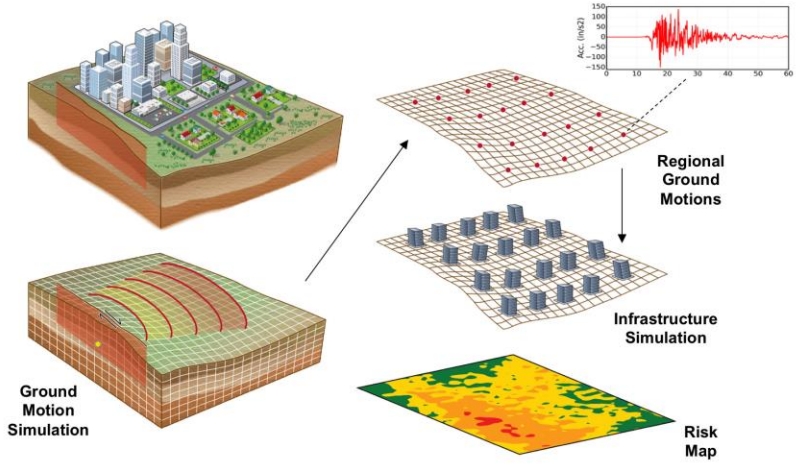


QMCPACK



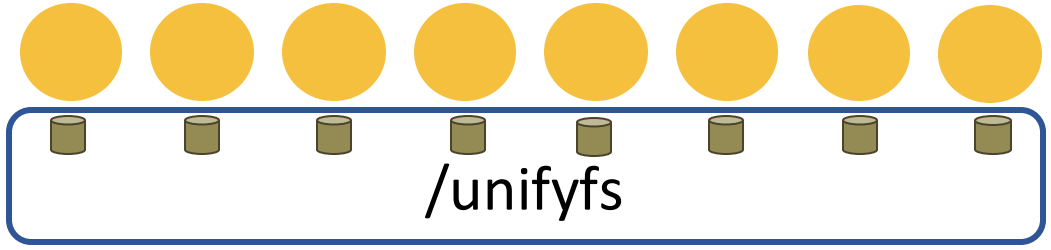
EQSim

Building SW4's I/O infrastructure w/ HDF5



Ongoing activity in the ECP ExaIO project

- Sub-filing: Reduces contention in accessing a single shared file
- Fine tuning asynchronous I/O
- Improving Data Elevator for using node-local storage
- Collaboration w/ UnifyFS that provides single namespace for node-local storage
  - https://github.com/LLNL/UnifyFS



Experimental and observational data (EOD) management features in development

- Handle multiple producers and multiple consumers of data (MWMR)
- Support remote streaming synchronization
- Extend data models to support changes in data and data schemas
- Manage and search metadata and provenance - MIQS paper @ SC19
- Support for sparse, variable length, streaming, KV modes of data
- Develop a VOL for reading XTC2 format data
- Target science drivers
  - LCLS / LCLS-II, ALS, NCEM, (NIF), (LSST)

Multi-writer and multi-reader (MWMR)

- Multiple producers and multiple consumers
  - Use case: Multiple producers of LCLS-2 data at SLAC need to write to the same HDF5 file, while the same file is being analyzed by multiple readers
  - A full functionality Single Writer and Multiple Readers (SMWR) implementation is in progress
  - Exploring consistency and coherence models of MWMR in existing tools, such as TileDB
  - Designing a solution that allows independent metadata updates from different writers

Thank you!