ExaIO: Delivering Efficient Parallel I/O on Exascale Computing Systems with HDF5 and UnifyFS

PI: Suren Byna (Lawrence Berkeley Lab)

Co-PIs: Scot Breitenfeld (The HDF Group), Kathryn Mohror (LLNL), Sarp Oral (ORNL), and Venkat Vishwanath (ANL)

June 1st, 2022

NNSA National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

2.3.4.15. ExaIO Team

- Team members
  - **HDF5**: Suren Byna[1], Scot Breitenfeld[3], Venkat Vishwanath[2], Houjun Tang[1], Jean Luca Bez[1], Huihuo Zheng[2], Neil Fortner[3], Dana Robinson[3], Jordan Henderson[3], Neelam Bagha[3], Michela Becchi[6], John Ravi[6]
  - *Alumni: Quincey Koziol[1], Qiao Kang[1], Jerome Soumagne[3], John Mainzer[3], Richard Warren[3], Elena Pourmal[3]*
  - **UnifyFS**: Kathryn Mohror[4], Sarp Oral[5], Adam Moody[4], Cameron Stanavige[4], Michael Brim[5], Seung-Hwan Lim[5], Ross Miller[5], Swen Boehm[5]

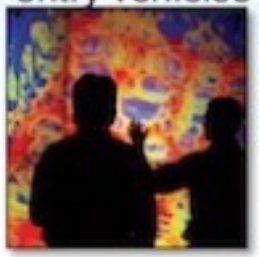
1. LBNL
2. ANL
3. The HDF Group
4. LLNL
5. ORNL
6. NCSU

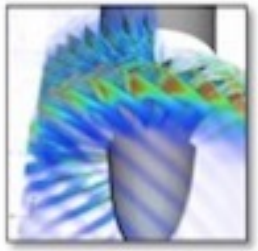
Exascale applications target US national problems in 6 strategic areas

National security

Stockpile stewardship

Next-generation electromagnetics simulation of hostile environment and virtual flight testing for hypersonic re-entry vehicles





Energy security

Turbine wind plant efficiency

High-efficiency, low-emission combustion engine and gas turbine design

Materials design for extreme environments of nuclear fission and fusion reactors

Design and commercialization of Small Modular Reactors

Subsurface use for carbon capture, petroleum extraction, waste disposal

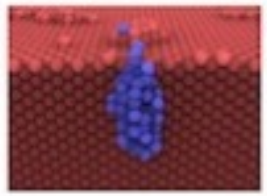
Scale-up of clean fossil fuel combustion

Economic security

Additive manufacturing of qualifiable metal parts

Reliable and efficient planning of the power grid

Seismic hazard risk assessment



Scientific discovery

Find, predict, and control materials and properties

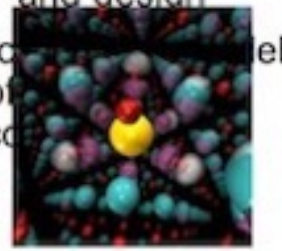
Cosmological probe of the standard model of particle physics

Validate fundamental laws of nature

Demystify origin of chemical elements

Light source-enabled analysis of protein and molecular structure and design

Wh el
o
c



Earth system

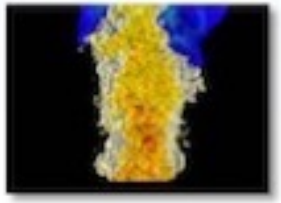
Accurate regional impact assessments in Earth system models

Stress-resistant crop analysis and catalytic conversion of biomass-derived alcohols

Metagenomics for analysis of biogeochemical cycles, climate change, environmental remediation

Health care

Accelerate and translate cancer research

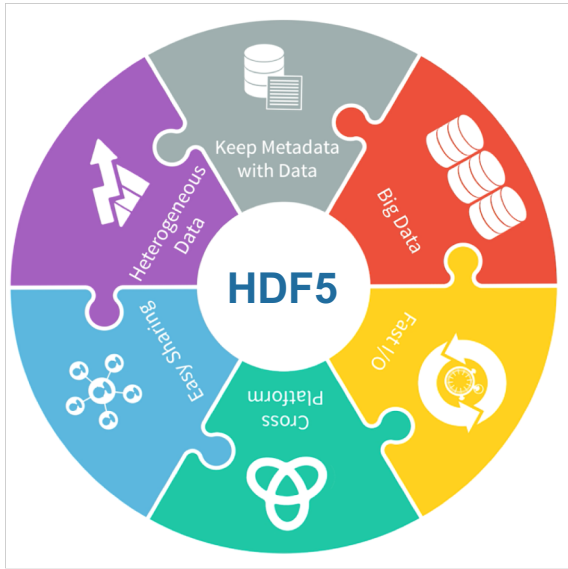


ECP EXASCALE COMPUTING PROJECT

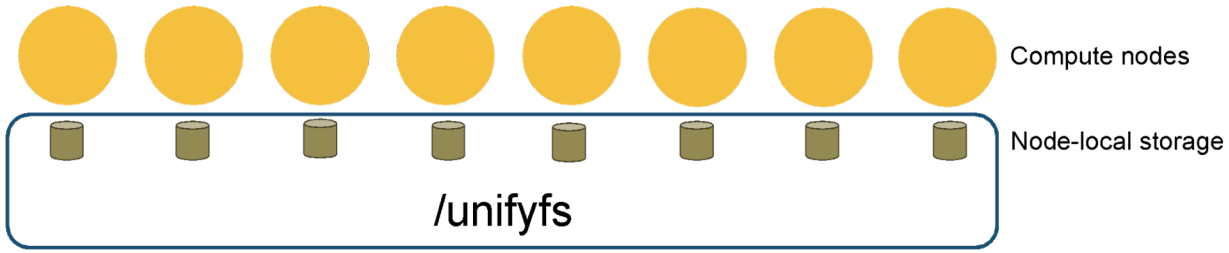
Slide from Doug Kothe

ExaIO Project – Enhancing HDF5 and Developing UnifyFS

- **HDF5: Parallel I/O API, library, and file format**
- HDF5 is a self-describing file format, API, and tools designed to store, access, analyze, share, and preserve diverse, complex data in continuously evolving heterogeneous computing and storage environments



- **UnifyFS: A file system for burst buffers**
- UnifyFS presents a shared namespace across distributed storage to read/write files **easy** and **fast**



ExaHDF5 mission - Applications, features, and tuning

- Many ECP Apps have a dependency on HDF5-based I/O
  - 17 critical, 11 important, 8 interested

Applications	Support ECP apps and ST tools achieve performant I/O with HDF5
New Features	Develop features that make HDF5 ready for exascale architectures
Tuning & Maintenance	Tune existing HDF5 capabilities to perform well at large scale

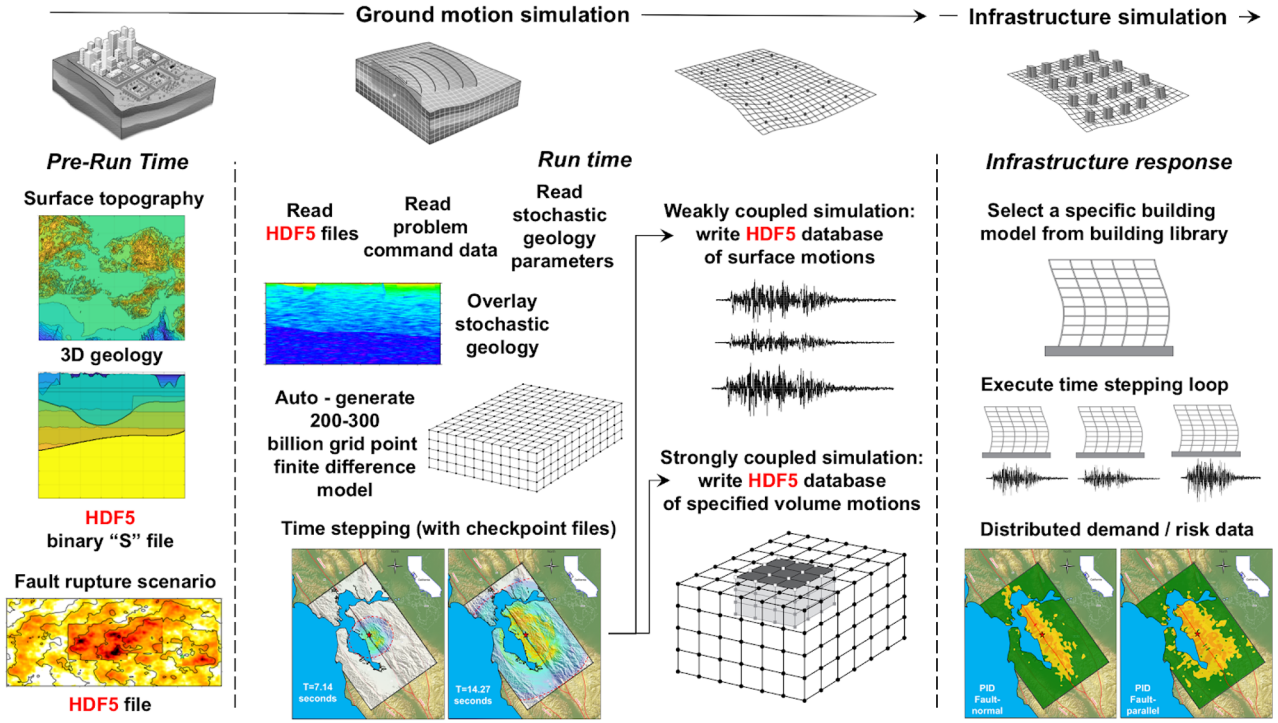
HDF5 capability integration - Active AD team interactions

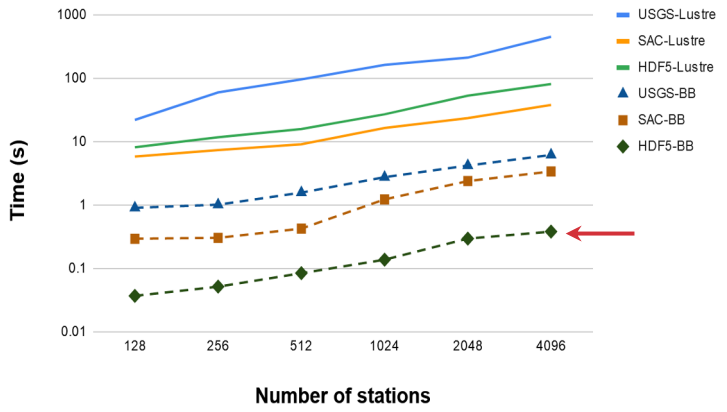
ECP AD team	Type of engagement	Status	ExaIO POC(s) / ECP team POC(s)
EQSIM	Development of I/O framework based on HDF5	Implemented most of the components, tuning at large scale	Suren Byna, Houjun Tang / Houjun Tang
AMReX	Development of HDF5 I/O	Implemented HDF5 I/O, adding compression	Suren Byna, Houjun Tang / Ann Almgren, A. Myers
QMCPACK (KPP-3)	File close performance issue	Improved performance	Venkat Vishwanath / Ye Luo, Paul Kent
ExaSky - Nyx	Integrated I/O in AMReX, adding compression support	Developed a new file layout and adding compression	Houjun Tang / Zarija Lucic
Subsurface simulation	I/O performance tuning	Improved performance	Suren Byna, Houjun Tang / Brian van Straalen
FLASH-X	Implemented async I/O routines	Testing performance at large scale	Houjun Tang / Rajeev Jain
ExaSky – HACC	I/O performance tuning	Tuning performance - subfiling	Scot Breitenfeld / Salman Habib
WarpX / OpenPMD	Tuning HDF5 I/O performance of OpenPMD	Tuned I/O performance by 10X for a benchmark; more potential for performance improvement	Suren Byna, Jean Luca Bez / Junmin Gu and Axel Huebl
E3SM	Improving HDF5 performance	Identified multi-dataset API improves performance; tuning further	Suren Byna, Qiao Kang / Jayesh Krishna, Danqing Wu
Lattice QCD, NWChemEx, CANDLE	I/O using HDF5	Initial communications w/ the AD teams	Suren Byna, Venkat Vishwanath / Chulwoo (LQCD), Ray Bair (NWChem), Venkat (CANDLE)
ExaLearn	I/O for ML applications	Performance evaluation and testing cache VOL	Suren Byna, Huihuo Zheng / Peter Nugent

HDF5 Applications: EQSIM

- A framework for regional-scale earthquake fault-to-structure simulations
- I/O and data management challenges
  - Easy-to-use data and file format for EQSIM workflows
  - Increased volume of data
  - Compressing checkpoint and multiple data products
- HDF5 benefits for EQSIM
  - Using HDF5 files reduced input time from hours to minutes for a 3600-node run on Summit
  - HDF5's self-describing format and portability allows convenient data sharing among scientists
  - Improved I/O performance for both input and output data
  - Reduced number of time-history files from thousands to 1 per simulation
    - Transparent compression capability allows saving and analyzing more data pain-free

Application POCs: D. McCallen, H. Tang, and N. Petersson





Config	CR	HDF5 File Size
Default	1	(76 TB)
zfp-acc=1e-2	261	293 GB
zfp-acc=1e-1	494	155 GB

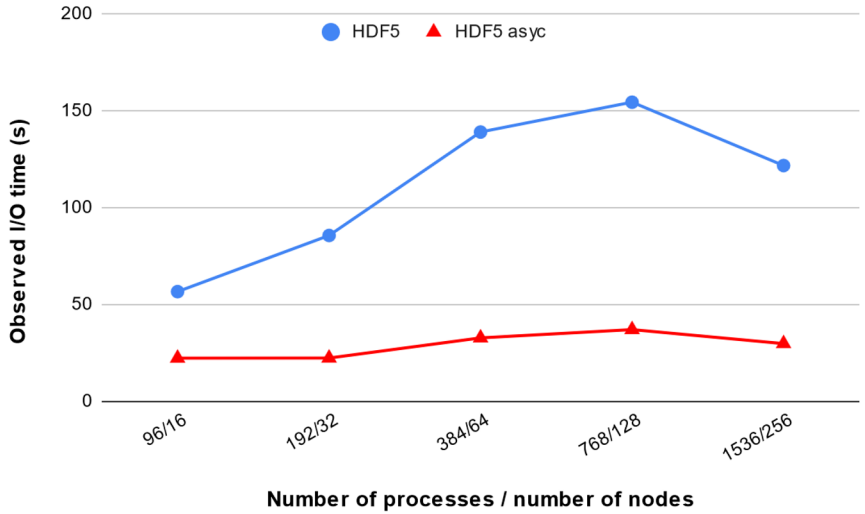
Compared to SAC format that generates file-per-process
Cori - HDF5 is **5X to 9X** faster using burst buffers
Summit - HDF5 output is 20% faster

Used ***ZFP*** as a HDF5 filter
13X performance improvement
Compression ratio **> 260**

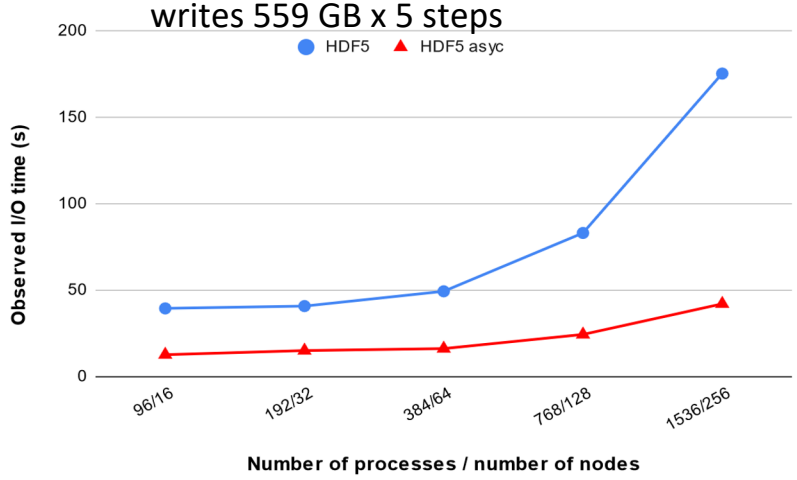
HDF5 Applications: Nyx and Castro with AMReX

- AMReX - block-structured AMR framework for solving systems of PDEs on exascale architectures
- Supports five ECP AD projects - WarpX, ExaStar, Pele, ExaSky, and MFIX-Exa
- I/O is based on native binary format and HDF5 file format
- ExaIO team is developing and tuning the HDF5 I/O
  - Integrated HDF5 I/O framework in AMReX
  - Upgraded HDF5 I/O with asynchronous I/O that effectively overlaps I/O latency with computations → **~4X speedup for 5 time steps**
  - Work in progress
    - Updating file layout for achieving better compression of data

NyX workload, single refinement level, writes 385 GB x 5 steps



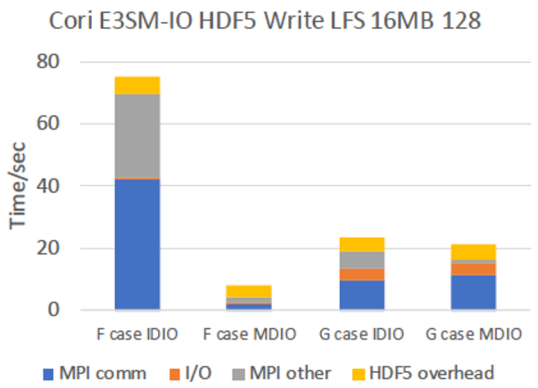
Castro workload, three refinement levels, writes 559 GB x 5 steps

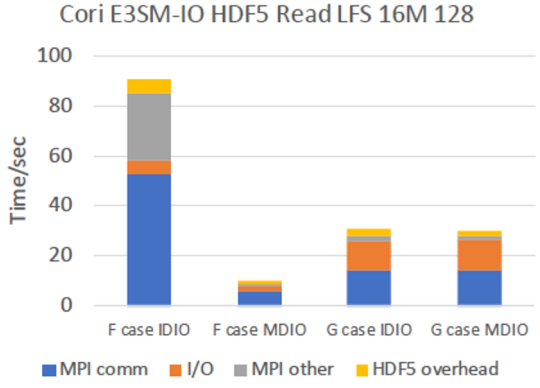


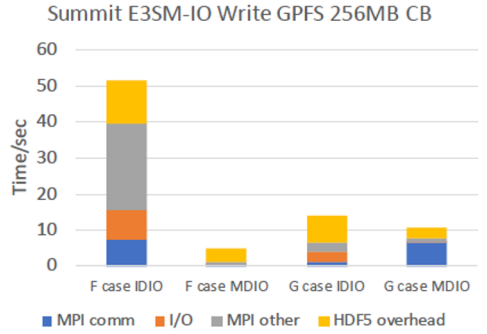
Application POCs: A. Almgren, A. Myers, Z. Lucic, J. Sexton, and K. Gott

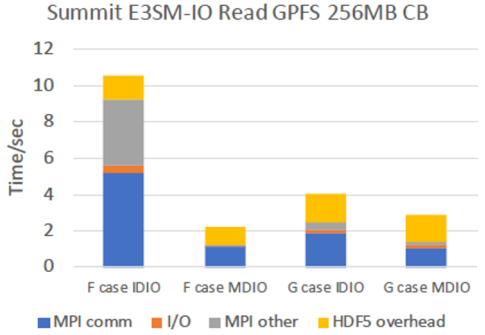
HDF5 Applications: E3SM

- E3SM - A large-scale climate simulation model
- E3SM I/O uses PIO library, built using multiple file formats
  - NetCDF-4 (which uses HDF5 internally) and PnetCDF
  - NetCDF-4 I/O has been suffering from poor I/O performance
- Benchmark with HDF5 API (without NetCDF-4)
  - Collaboration with DataLib and E3SM teams
  - Two versions of HDF5 benchmark that maintains canonical ordering of data from the application
    - Using regular write/read API
    - Using multi-dataset API that allows reading/writing multiple requests with a single API call
  - Multi-dataset API and further tuning on file system shows **up to 10X** improvement for the F case
  - Working on integrating the multi-dataset API branch in HDF5
  - Exploring further optimizations - DataLib team's log-based VOL





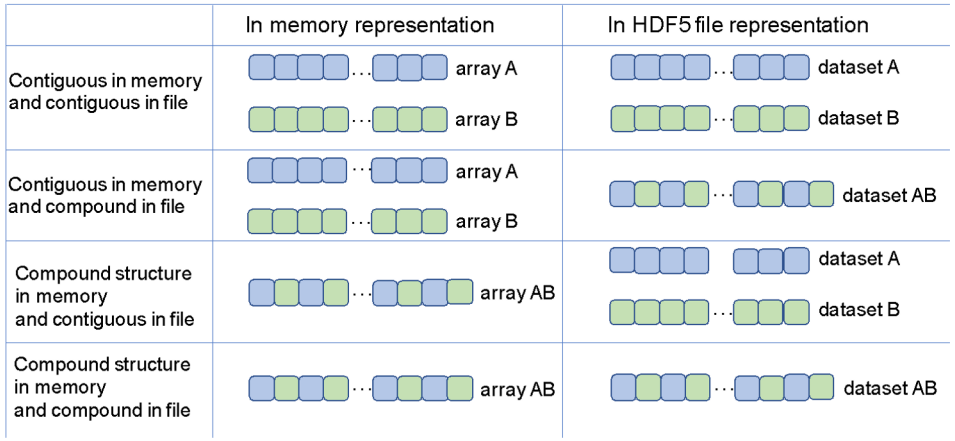


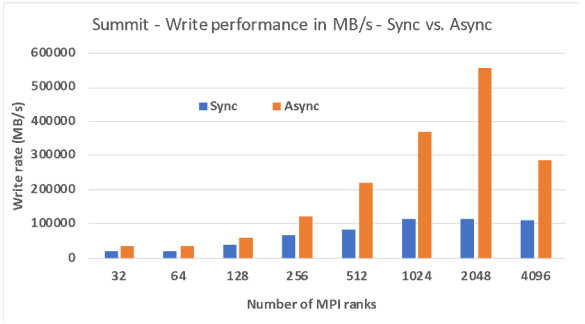


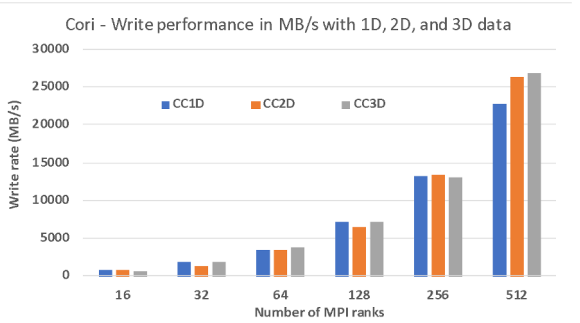
Representative I/O benchmarks / kernels - **h5bench**

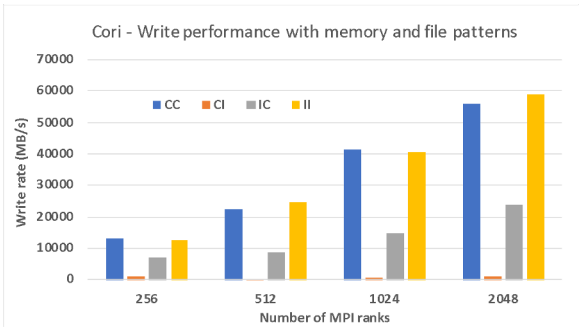
- h5bench - HDF5 I/O kernel suite for exercising common parallel I/O patterns to compare various HDF5 features
- Exercises I/O operations (read, write, streaming append, modify), data locality, file layout, I/O modes (synchronous and asynchronous), MPI-IO tuning options (collective buffering), file system configurations (alignment, striping, etc.)
- Metadata stress tests
- Application kernels
  - AMReX (Nyx and Castro configurations)
  - OpenPMD (WarpX configuration)
  - *E3SM I/O kernel*
  - *More HDF5 benchmarks from the community*

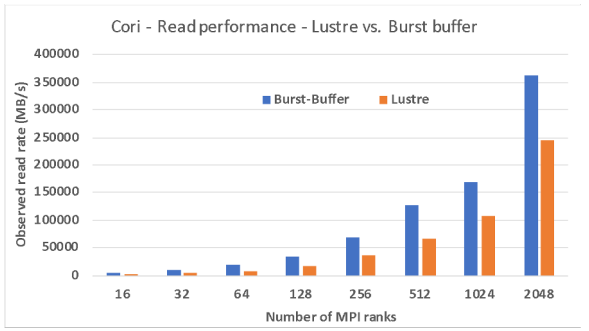
https://github.com/hpc-io/h5bench











Exascale readiness - Summary of ExaIO HDF5 features and status

HDF5 component	Development status	Impact (*Apps*)	Systems used for testing
Virtual Object Layer (VOL) framework	Integrated in the HDF5 maintenance releases (1.12.x) VOL 2.0 is in 1.13.0 pre-release	Enables using HDF5 on novel current and future storage systems easily (*ExaIO, DataLib, ADIOS, and others*)	Summit, Cori, Theta, Spock, and other testbeds
Asynchronous I/O	Released v.1.0	Allows overlapping I/O latency with compute phase (*EQSIM, AMReX apps, external*)	Summit, Cori, Theta, Spock, Perlmutter, and other testbeds
Cache VOL	Released v.1.0	Allows using node-local memory and/or storage for caching data (*On systems w/ node-local memory/storage resources, ML apps*)	Summit, Theta, and Cori
GPU I/O	Developed pluggable VFD in HDF5 (in 1.13.0 pre-release) GPU I/O VFD v.1 is released	GPU I/O VFD allows using NVIDIA's GPU Direct Storage (GDS) (*Apps on GDS enabled GPUs, pluggable VFD allows developing new VFDs*)	Tested on NVIDIA systems and a local server (Dependencies: GPUs that are GDS compatible and NVIDIA GDS driver installation)
Subfiling	Selection I/O has been implemented and integrated in HDF5 Implementation in progress	Allows writing/reading multiple subfiles (instead of single shared file) (*Testing w/ h5bench*)	Testing on Summit and Cori
Multi-dataset I/O API	A prototype available; design updates in progress	Allows writing multiple HDF5 datasets with a single write/read call (*E3SM*)	Prototype was tested on Summit and on Cori with E3SM F and G case configurations
h5bench	Released v.1.1	Allows testing a diverse set of I/O patterns and app kernels with various HDF5 features (*Broad*)	Summit, Theta, Perlmutter
Parallel compression	Released in HDF5 maintenance	Evaluating performance and tuning as needed (*EQSIM, AMReX applications, and others*)	Evaluating performance on Summit and Cori with EQSIM checkpointing using ZFP compression

Green: In HDF5 library internal

Blue: External plugins / connectors

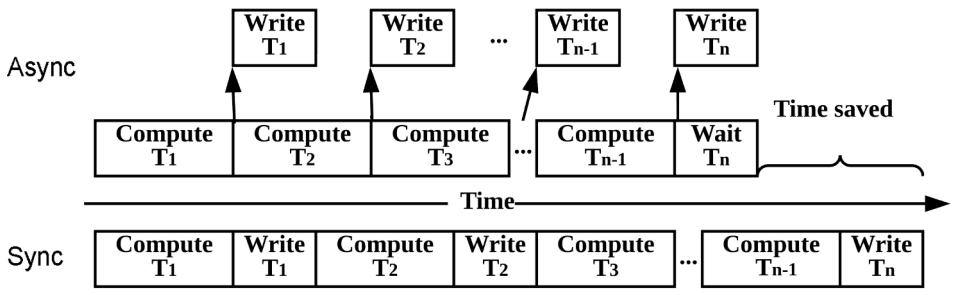
Features: Asynchronous I/O

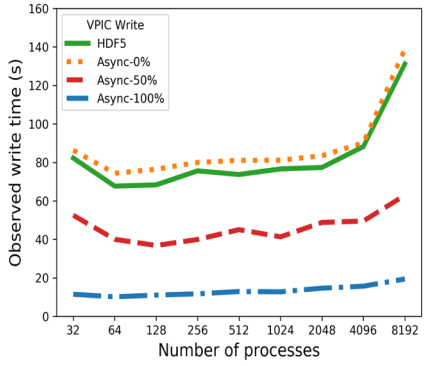
- Pass-through VOL connector with background threads performing I/O operations, using Argobots
- Two modes
  - Implicit: For unmodified applications by setting env. variable
  - Explicit: For applications that want more control of async operations, such as when to trigger async I/O
- Built and tested on all major platforms and exascale testbeds
  - Summit, Spock, Cori, Perlmutter, and other testbeds

Application integration status

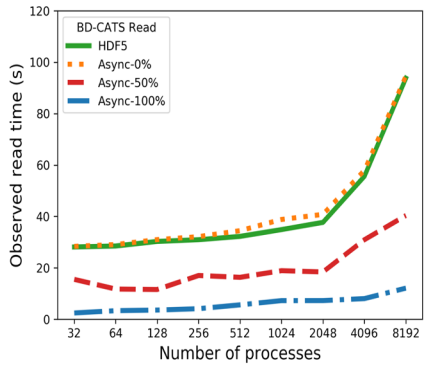
Application	Status
Nyx and Castro (via AMReX)	Integrated in AMReX codebase
FLASH-X	Prototype code developed and performance tuning in progress
EQSIM	Developed code to integrate asynchronous I/O for checkpointing, testing is in progress
OpenPMD	Code development is in progress
A NASA Ames application	*An external user integrated async I/O; testing and tuning performance*

E4S integration:
- Spack package and CI are available





VPIC-IO on Cori



BD-CATS-IO on Cori

Weak scaling

A FLASH-X configuration on Summit

# nodes	Async I/O speedup	App speedup
1	7.87	1.03
2	14.71	1.14
3	19.50	1.21
4	20.35	1.23
5	13.32	1.26
6	9.31	1.24
7	6.61	1.27

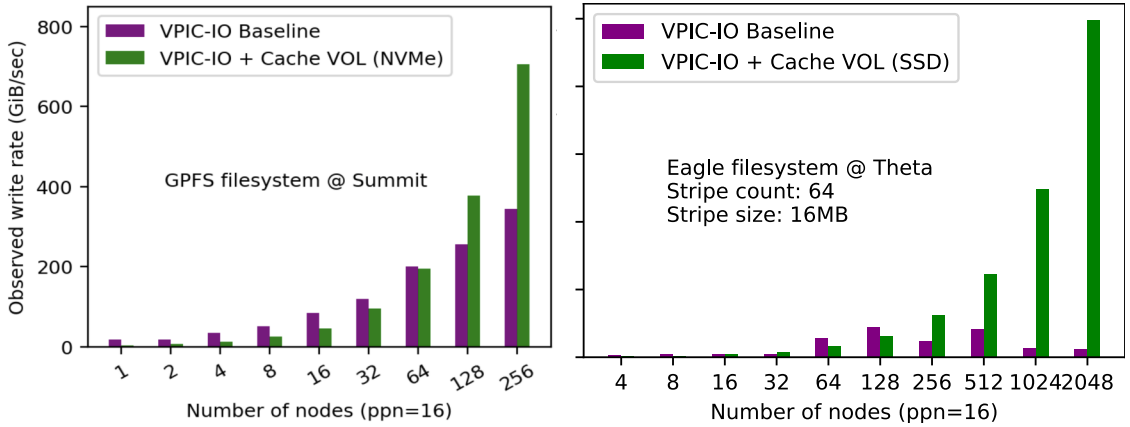
Strong scaling

https://github.com/hpc-io/vol-async

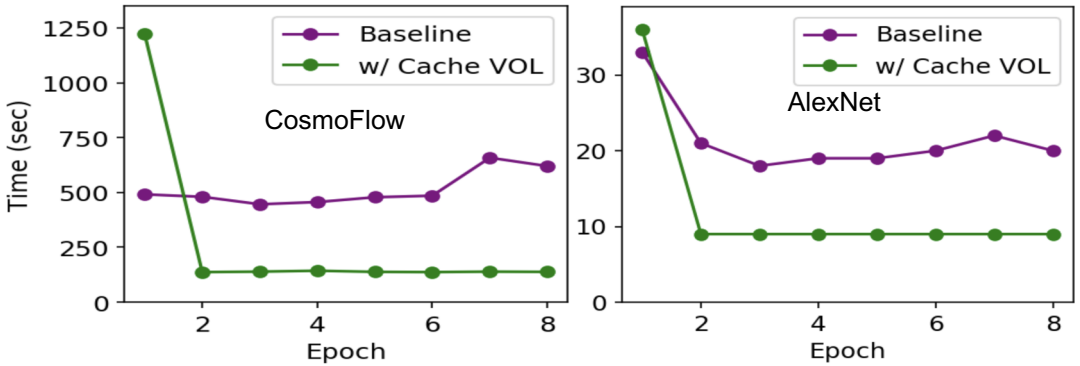
Features: Caching with node-local memory and storage

- Use node-local memory and storage to reduce the performance gap between memory and long-term storage
- Developed "Cache" VOL connector
  - Node-local memory
  - Node-local storage, including "remote" node-local
  - Shared burst buffer storage layer
- Stacked *cache* and *asynchronous* I/O VOL connectors
  - Cache VOL focuses on using node-local memory and storage locations ("space-shifting" operations)
  - Async I/O VOL to perform data movement and HDF5 file operations asynchronously ("time-shifting" operations)
- Implicit VOL -- no code changes needed and environment variable set

E4S integration:
- Spack package to be committed and CI is available



Improvement of h5bench write bandwidth with Cache VOL on Summit with GPFS file system and Theta with Lustre file system. The data, 32MB per process, were cached first on the node-local storage, NVMe / SSD, and moved to the parallel file system asynchronously. 16 MB alignment was set on Summit for optimal HDF5 performance.

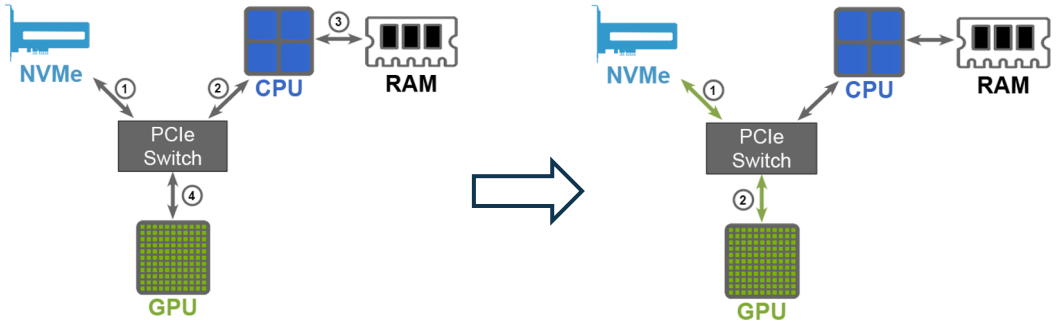


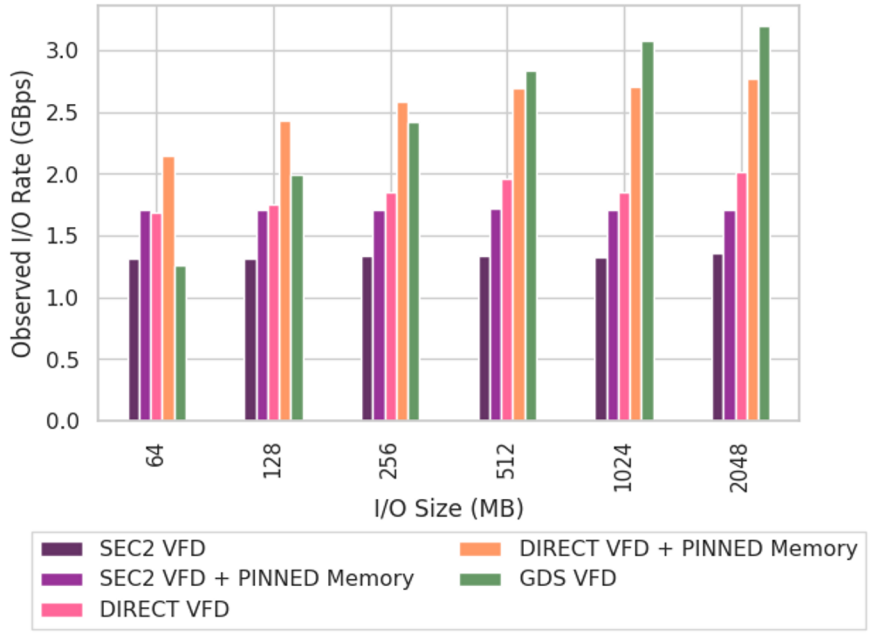
Cache VOL reduces the training time by **2x** for read intensive deep learning applications: CosmoFlow and AlexNet with TensorFlow. The datasets are loaded from a single HDF5 file through h5py and tf.data pipeline. The size of the dataset is 8 TB for CosmoFlow and 180 GB for AlexNet. Experiments were performed on **128 A100 GPUs @ Theta**.

https://github.com/hpc-io/vol-cache

Features: GPU I/O

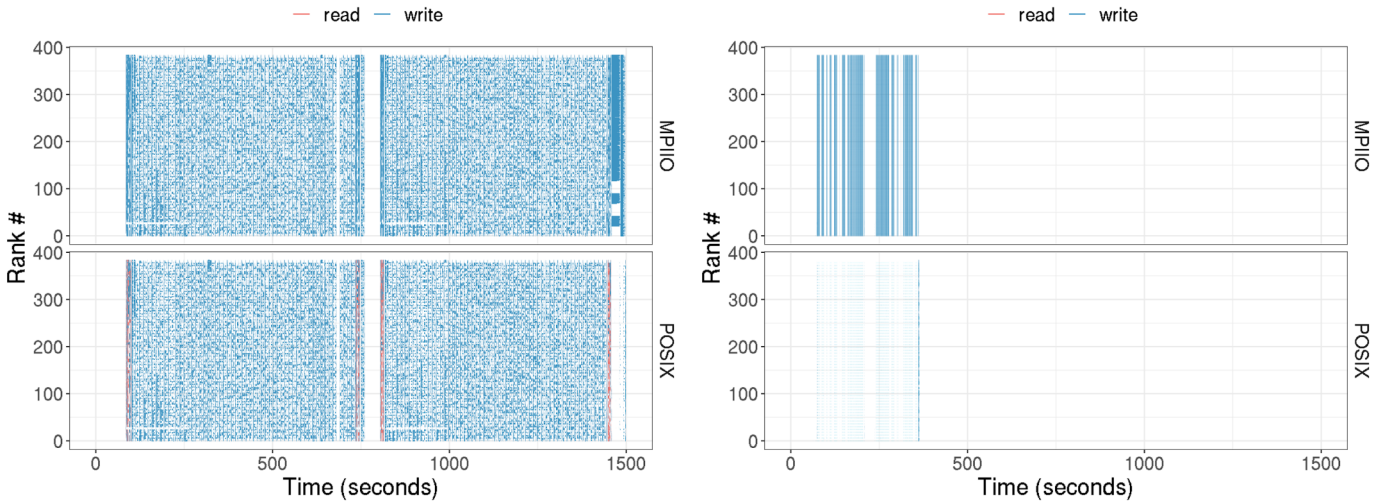
- File I/O to move data between GPUs and storage devices becomes critical
- HDF5 team efforts (with contingency funding):
  - Developed pluggable Virtual File Driver (VFD) infrastructure
  - VFD for NVIDIA's GPU Direct Storage (GDS)
    - Performance benefits with larger data sizes
    - Integrated into HDF5 (https://github.com/hpc-io/vfd-gds)
  - Asynchronous data movement between GPU and CPUs, and between CPU and storage
    - Testing h5bench read/write patterns with GPU memory
    - Initial results show significant benefit when overlapping write time and transfers between CPU and GPU
    - Designing integration with HDF5 using async and cache VOL connectors
    - More testing on GPUs from more vendors
    - Considering RAJA, Kokkos, HIP, Sycl, One API, etc.



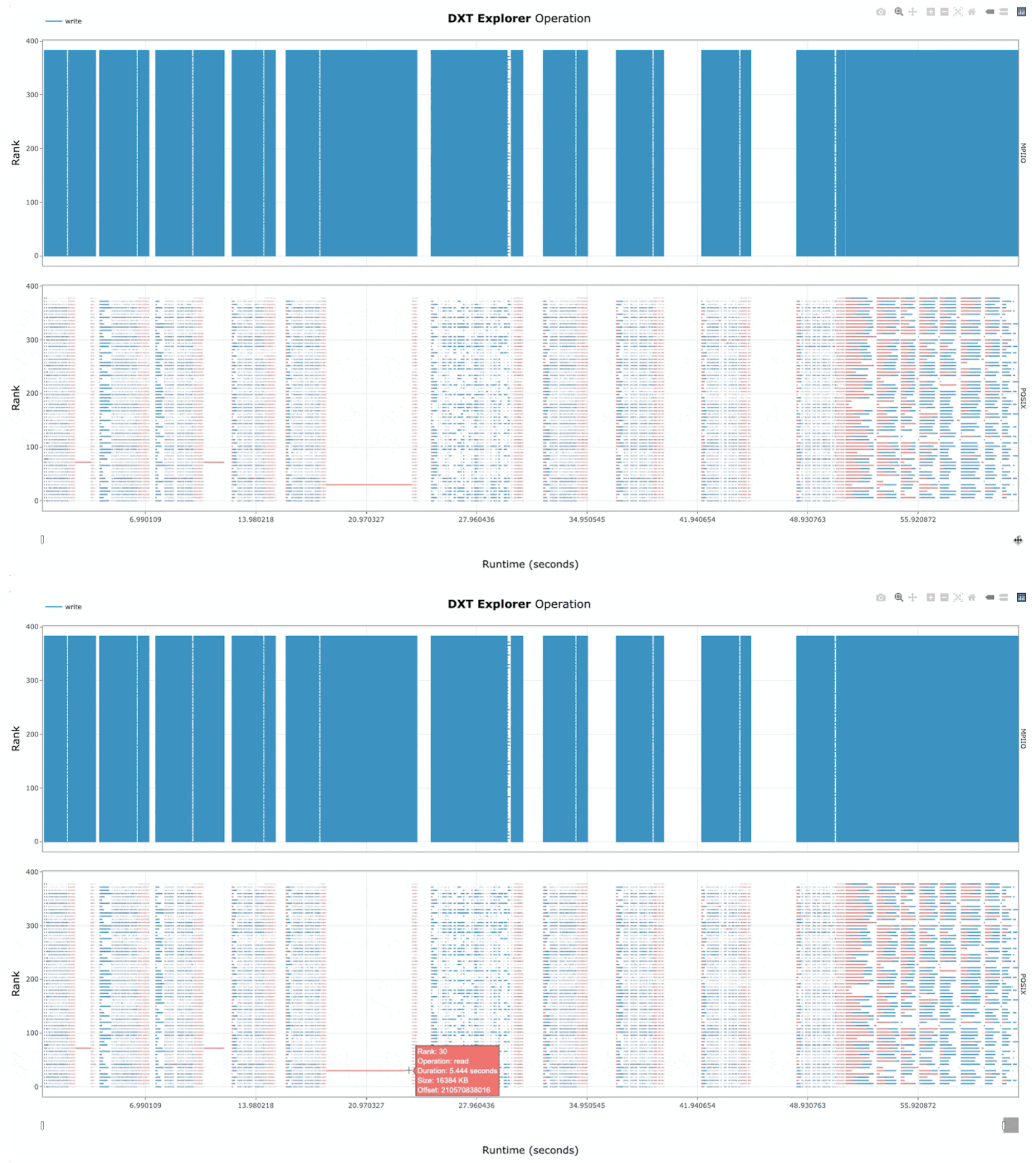


Tuning: Visualizing I/O performance

- To better identify I/O performance bottlenecks
  - Developed DXT Explorer to visualize Darshan Extended Traces
  - In collaboration with the DataLib Darshan team
  - PDSW 2021 paper (held in conjunction with SC21)
- Identified and tuned performance of
  - WarpX, FlashIO, 3D decomposition benchmarks
  - **2X to 19X performance improvements**



FlashIO benchmark on Summit - Baseline vs. Optimized



github.com/hpc-io/dxt-explorer

docker pull hpcio/dxt-explorer

Take home

- ECP ExaIO - HDF5 project
  - Supports numerous exascale applications to use HDF5 efficiently
  - Features
    - Async I/O, caching and prefetching using node-local storage, subfiling, multi-dataset I/O API, parallel compression tuning, GPU Direct Storage (GDS) VFD
  - Tools
    - h5bench parallel I/O benchmark suite
    - DXT Explorer for visualizing I/O performance

Thank you!

U.S. DEPARTMENT OF ENERGY | Office of Science

Argonne Leadership Computing Facility
an Office of Science user facility

- **Contacts**
  - Suren Byna (LBNL) SByna@lbl.gov
  - Scot Breitenfeld (The HDF Group) brtnfld@hdfgroup.org
  - Kathryn Mohror (LLNL) mohror1@llnl.gov
  - Sarp Oral (ORNL - OLCF) oralhs@ornl.gov
  - Venkat Vishwanath (ANL - ALCF) venkat@anl.gov

HDF5 User Support:
HDF Helpdesk: help@hdfgroup.org
HDF Forum: https://forum.hdfgroup.org/

UnifyFS:
Kathryn Mohror: mohror1@llnl.gov