Argonne Training Program on Extreme-Scale Computing

ATPESC 2021

August 1– August 13, 2021

M. Scot Breitenfeld

The HDF Group

Outline

- Foundations of HDF5
  - What is HDF5
  - HDF5 Fundamentals – A Simple Problem
- Intro to HDF5 performance fundamentals
  - Raw data writes with metadata impacts
  - Tuning metadata for parallel HDF5 applications

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

What is HDF5?

The HDF Group
BERKELEY LAB
Lawrence Berkeley National Laboratory
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

What is HDF5?

- Hierarchical Data Format version 5 (**HDF5**)
  - An extensible **data model**
    - Structures for data organization and specification
  - Open-source **software** (I/O library and tools)
    - Performs I/O on data organized according to the data model
    - Works with POSIX and other types of backing store: Object Stores (DAOS, AWS S3, AZURE, Ceph, etc.), memory hierarchies and other storage devices
  - Open **file format** (POSIX storage only)
  - This talk's focus is geared toward traditional file systems

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HDF5 is designed for…

- High volume and complex data
  - HDF5 files of GBs sizes are common
- Every size and type of system (portable)
  - Works on from embedded systems, desktops and laptops to exascale systems
- Flexible, efficient storage and I/O
  - See variety of backing store
- Enabling applications to evolve in their use of HDF5 and to accommodate new models
  - Data can be added, removed and reorganized in the file
- Supporting long-term data preservation
  - Petabytes of remote sensing data, including data for long term climate research is in NASA archives now

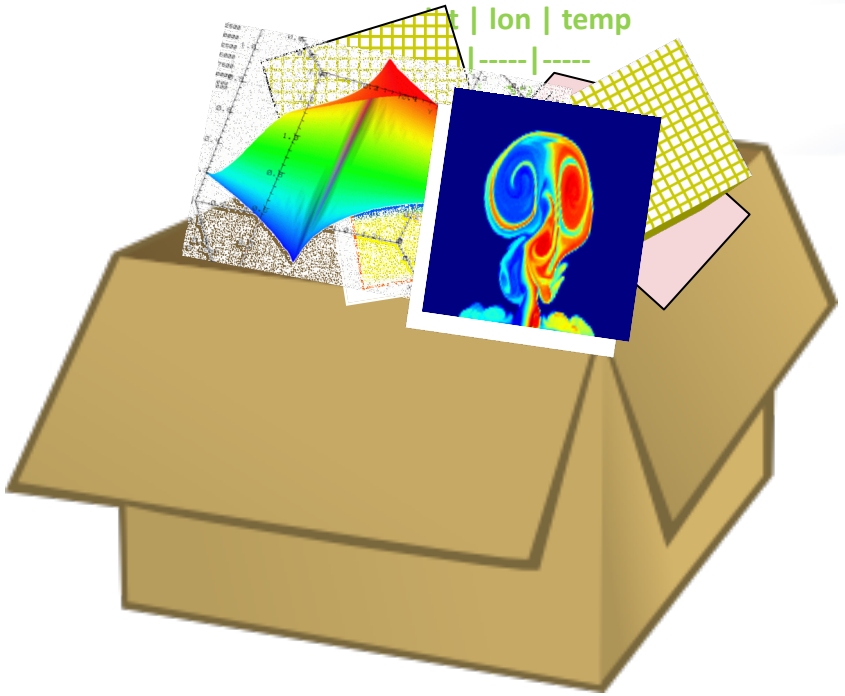
The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HDF5 Data model

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HDF5 File

An HDF5 file is a **structured container** that holds data objects.



The HDF Group

BERKELEY LAB
Lawrence Berkeley National Laboratory

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

HDF5 Data Model

Dataset –
Organize and contain data elements

Dataspace –
Describes logical layout of the data elements

HDF5 Objects

Attribute –
User-defined metadata

Datatype –
Describes individual data elements

File

Link –
Organize data objects

Group –
Organize data objects

HDF5 Software

HDF5 home page: http://hdfgroup.org/HDF5/

- Latest releases: HDF5 1.10.7 (1.12.1)

HDF5 source code:

- Written in C, and includes optional C++, Fortran, Java APIs, and High-Level APIs
- Contains command-line utilities (h5dump, h5repack, h5diff, ..) and compile scripts

HDF5 pre-built binaries:

- When possible, include C, C++, Fortran, Java, and High-Level libraries. Check ./lib/libhdf5.settings file.
- Built with the SZIP and ZLIB external libraries

3rd party software:

- h5py (Python)
- http://h5cpp.org/ (Contemporary C++ including support for MPI I/O)

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

External Help Resources

- Reference manuals, examples
  - https://portal.hdfgroup.org/display/HDF5/HDF5
- The HDF Group Webinars https://www.hdfgroup.org/category/webinar/
  - Past topics covered
    - Introduction to HDF5
    - HDF5 Advanced Features
    - HDF5 VOL connectors
- Direct help from the HDF Group
  - Help desk: help@hdfgroup.org
  - HDF5 office hours: a weekly mini-seminar/Q&A session, Tuesdays 1-1:30 p.m. Central
- Help from the HDF5 Community via the HDF forum

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HDF5 Fundamentals – A Simple Problem

- Writing multiple 2D array variables over time:

ACROSS P processes arranged in a **R x C** process grid
  FOREACH step 1 .. **S**
    FOREACH count 1 .. **A**
      CREATE a double **ARRAY** of size **[X,Y]** | **[R*X,C*Y]** (Strong | Weak)
      (**WRITE** | **READ**) the **ARRAY** (**to** | **from**) an HDF5 file

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Fundamentals – Missing Information

- How are the array variables represented in HDF5?
  - 2D, 3D, 4D datasets
  - Are the extents known a priori?
  - How are the dimensions ordered?
  - Groups?
- What order is the data written, and is the data read the same way?
- What's the storage layout?
  - How many physical files?
  - Contiguous or chunked, etc.
  - Is the data compressible?
- What's the file system or data store?
- Collective vs. independent MPI-IO

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

One Kind of Performance Hurdle

- HDF5 has a complex-looking interface
  - Complexity does not necessarily mean difficult to use
  - Users may require such complexity to achieve their goals
    - **Goal**: Self-describing share-friendly data layout
      - Tuning performance and efficiency with the constraint of using a standardized file format (netCDF, CGNS, etc.)
    - **Goal**: Fastest I/O possible
      - Tuning for check-points by minimizing metadata, large write blocks.
  - The complexity of the HDF5 workflow and underlying hardware may make the HDF5 tasks unavoidably complex.

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Other Sources of Performance Variability

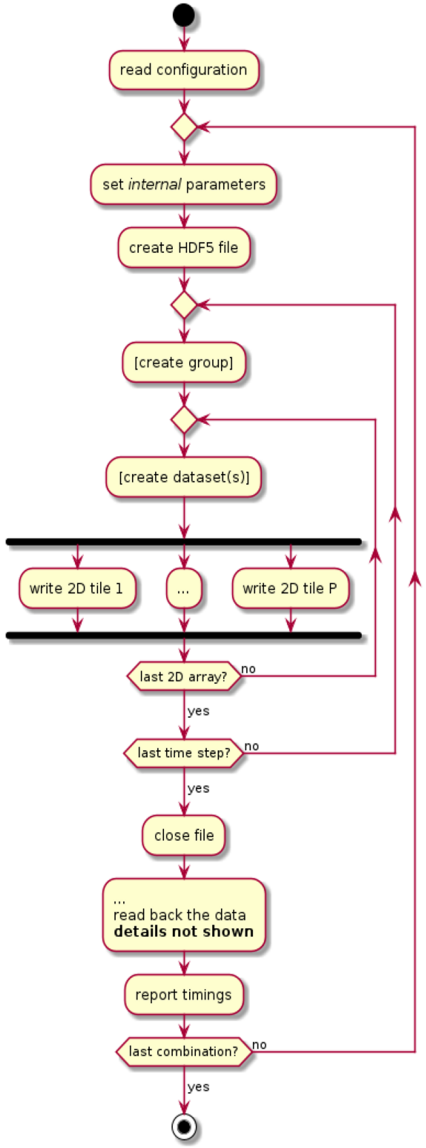
- Hardware
- System configuration and activity of other users
- **HDF5 property lists**
  - Nearly 180 APIs
  - Controls storage properties for HDF5 objects
  - Controls in-flight HDF5 behavior
  - About 100 *H5Pset_** functions
    - ≤ $p_1 * \ldots * p_{100}$ combinations!
    - How many are tested?
  - What does *H5P_DEFAULT* mean?
    - (No, you can't control that one)
  - What is the effect of using H5P_DEFAULT?

https://portal.hdfgroup.org/display/HDF5/Property+Lists

The HDF Group BERKELEY LAB Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

Back to earlier example – Application Model

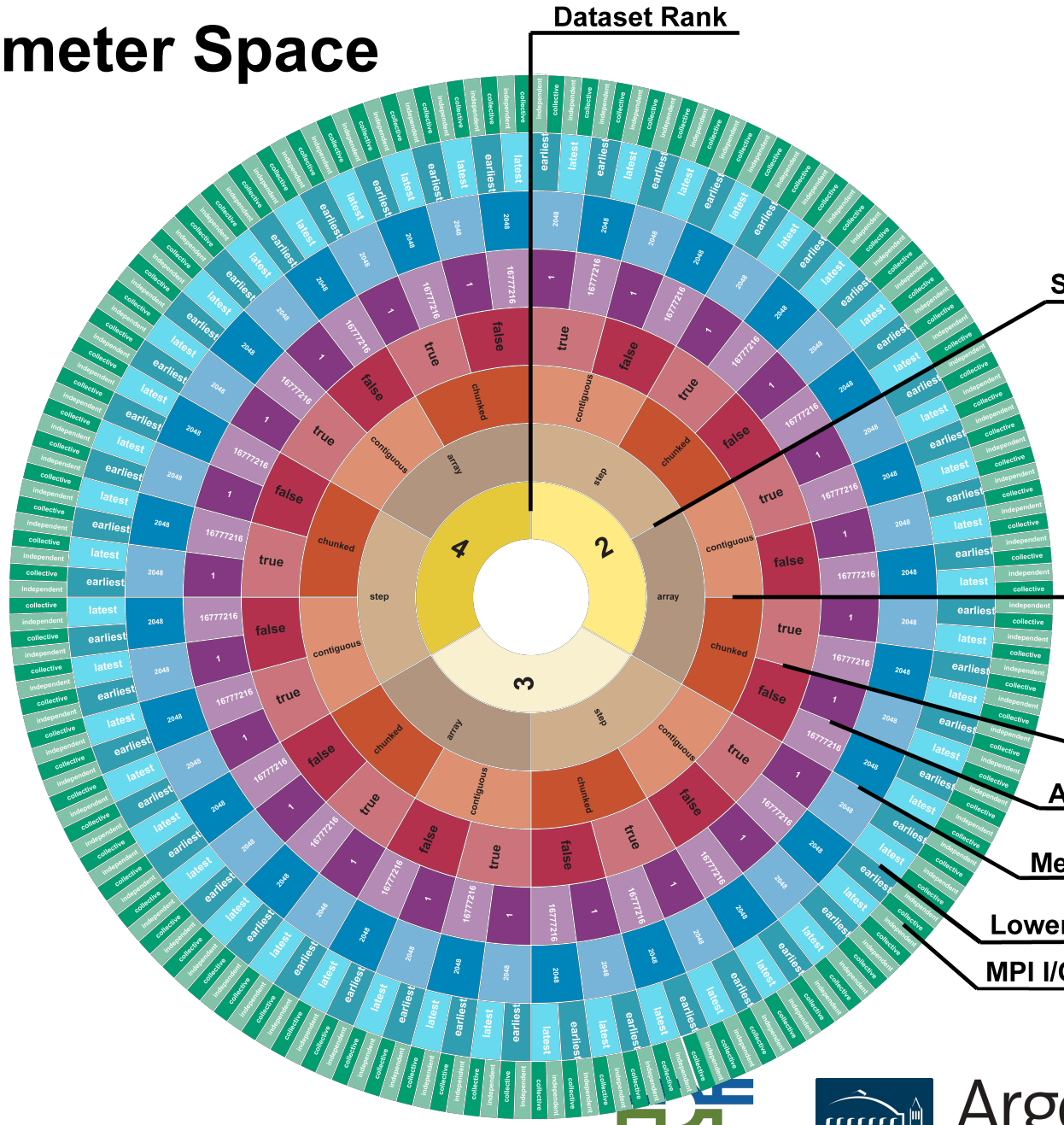
- Good or bad news:
  - There are *several* different ways to handle the data in HDF5, for example:
    - Many 2D datasets or attributes
    - A few 3D datasets
    - A 4D dataset
  - There are many ways to use HDF5 properties
    - Chunking
    - Data alignment
    - Metadata block size
    - Collective/Independent I/O
  - Ideally, performance would be more or less the same
  - **HDF5 I/O**[1] test explores the HDF5 parameter space



1 https://github.com/HDFGroup/hdf5-iotest

The HDF Group BERKELEY LAB Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

HDF5 Parameter Space



The HDF Group

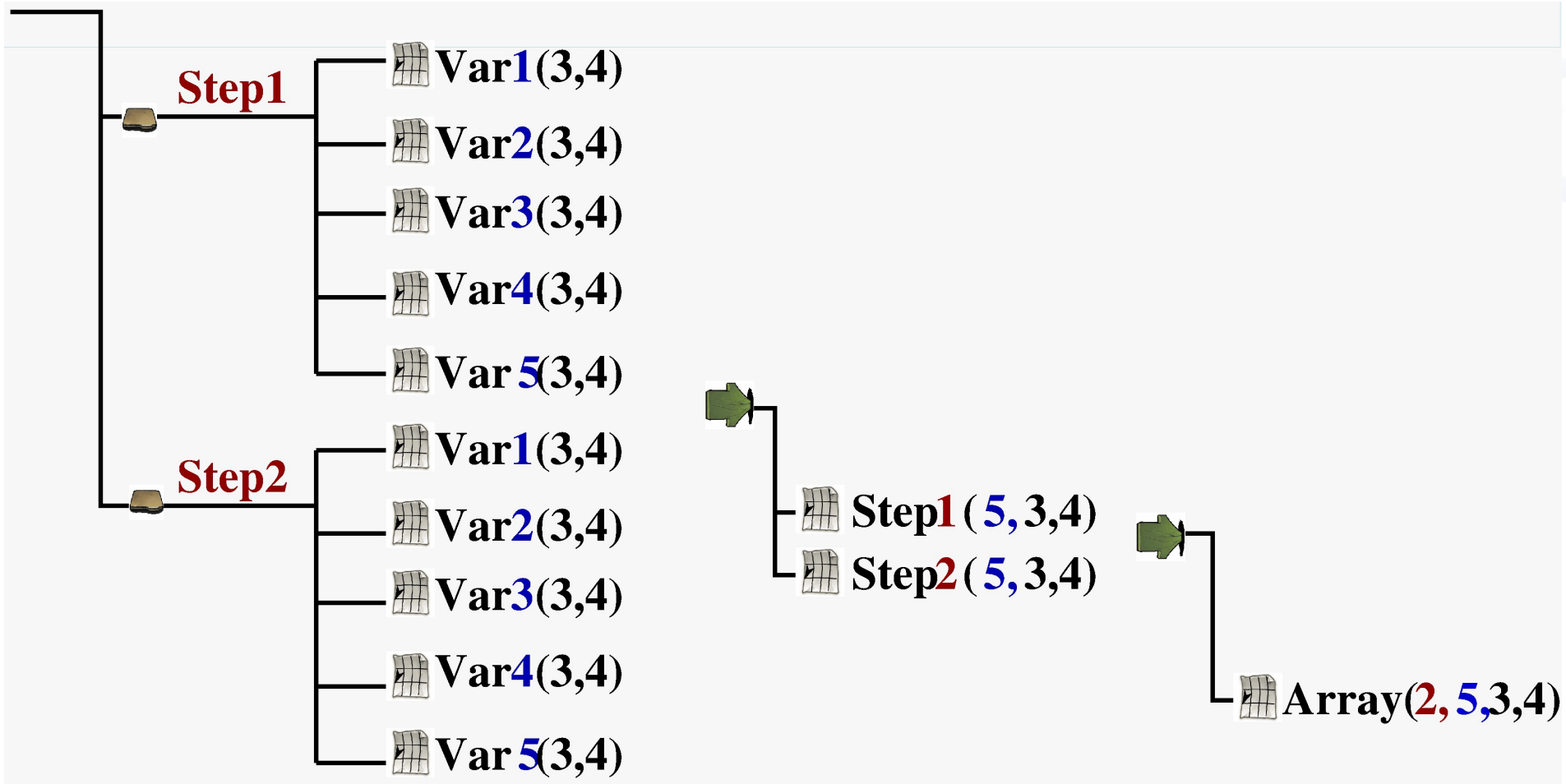
BERKELEY LAB
Lawrence Berkeley National Laboratory

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

IO Pattern Model

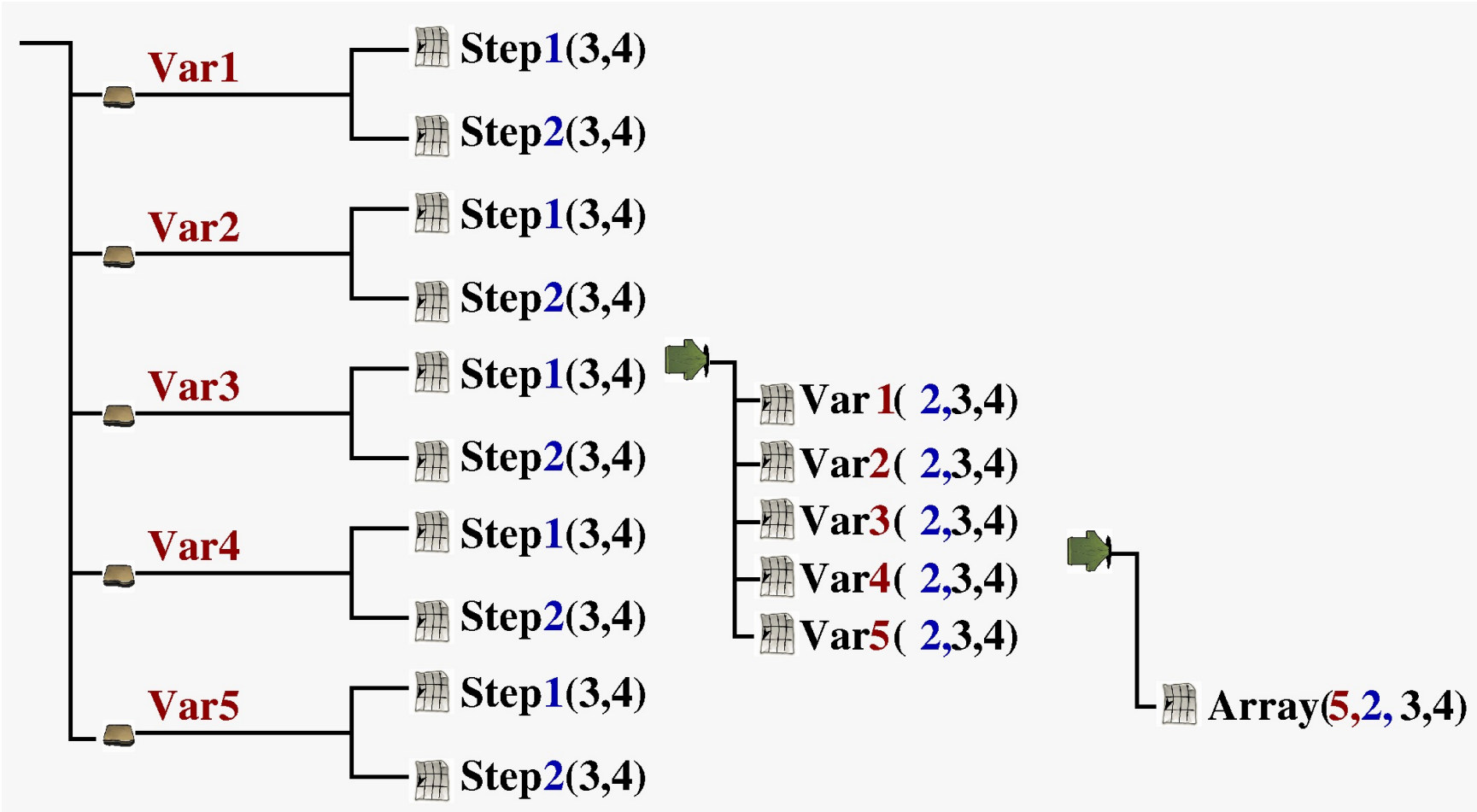
Step based IO Pattern



The HDF Group BERKELEY LAB Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

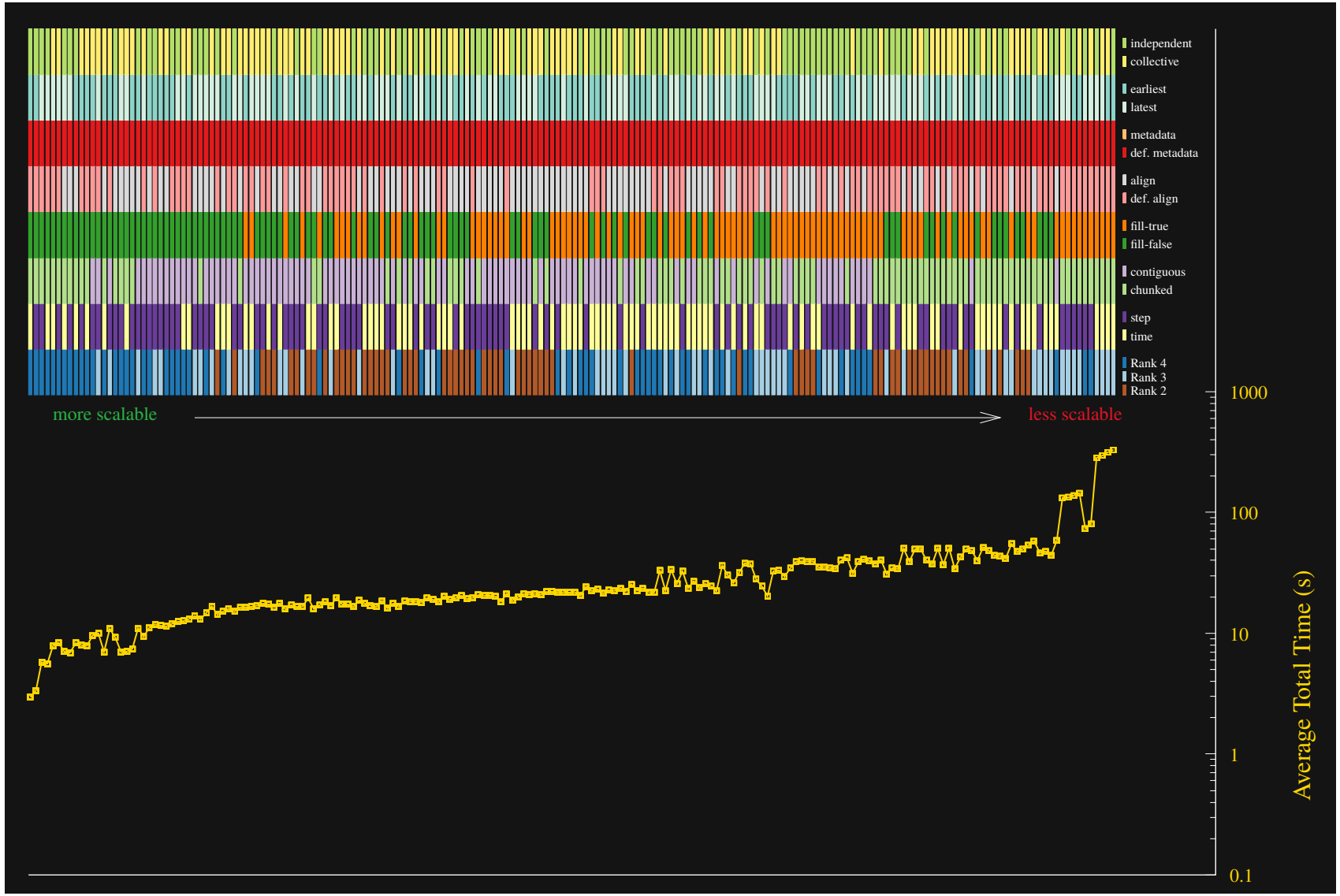
IO Pattern Model

Array based IO Pattern



The HDF Group BERKELEY LAB Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

Performance as a function of HDF5 parameter space



- Summit, weak scaling (42 to 2688)
- Best had:
  - four rank array (layout)
  - chunked
  - no fill values
  - default alignment
  - independent I/O

Use Case:
Raw data writes with metadata impacts

Tuning HACC (Hardware/Hybrid Accelerated Cosmology Code)

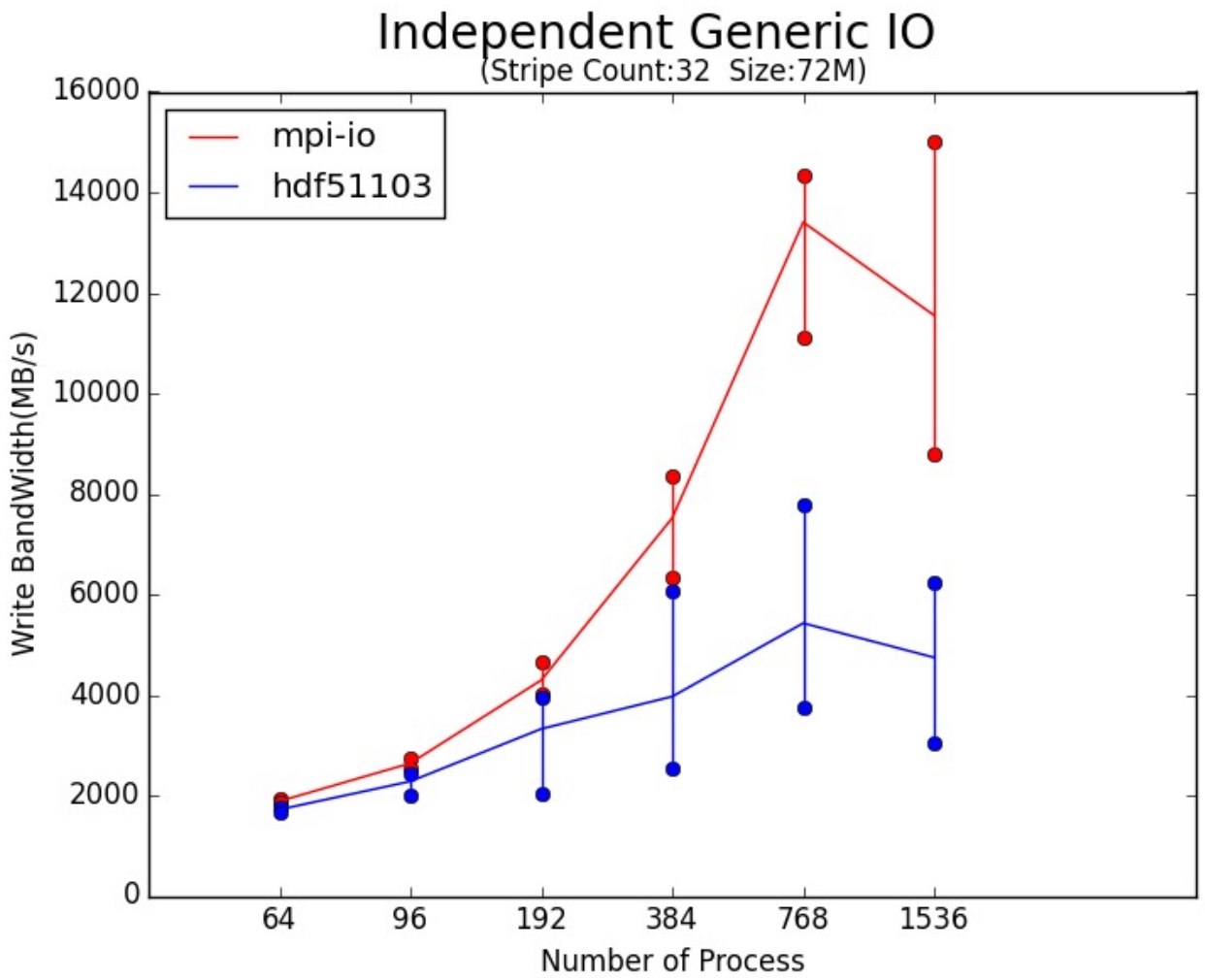
The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HDF5 Metadata

- Describes HDF5 objects in the file
  - Examples:
    - Object headers
      - Contain users' attributes
    - Local and global heaps
    - B-tree nodes
- Small (KBs) comparing to raw data
- Calls for HDF5 metadata can result in many small reads and writes.

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

HAAC Performance Results



Includes times for file open and close

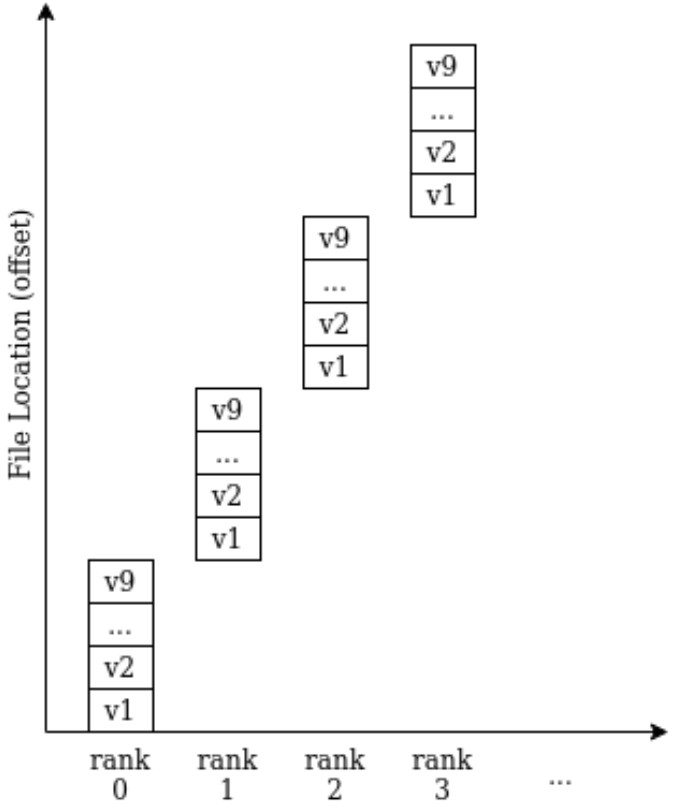
The HDF Group

BERKELEY LAB

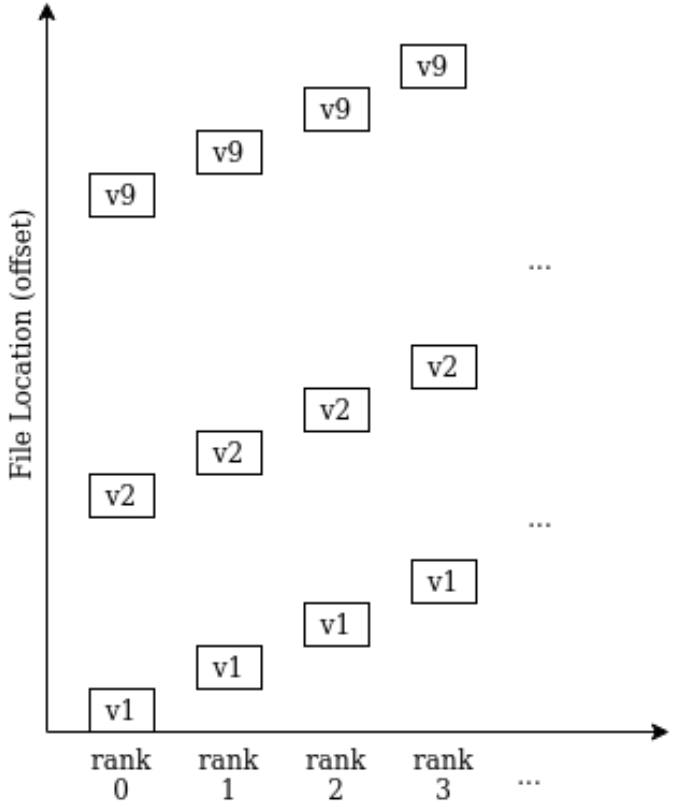
Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

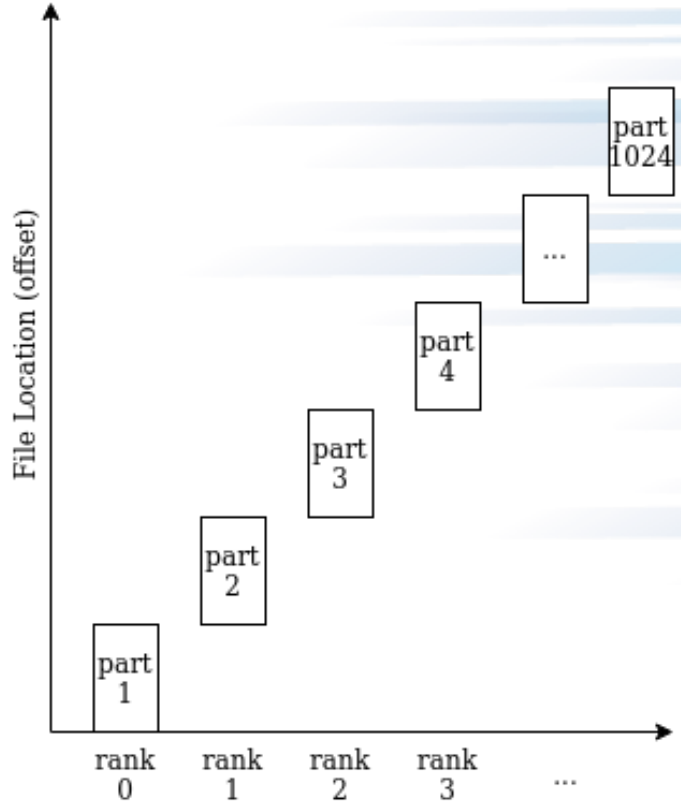
3 Access Patterns



MPI Interleaved = HDF5 Multi



MPI Contiguous = HDF5 Individual



HDF5 Compound

The HDF Group BERKELEY LAB Argonne NATIONAL LABORATORY ECP EXASCALE COMPUTING PROJECT

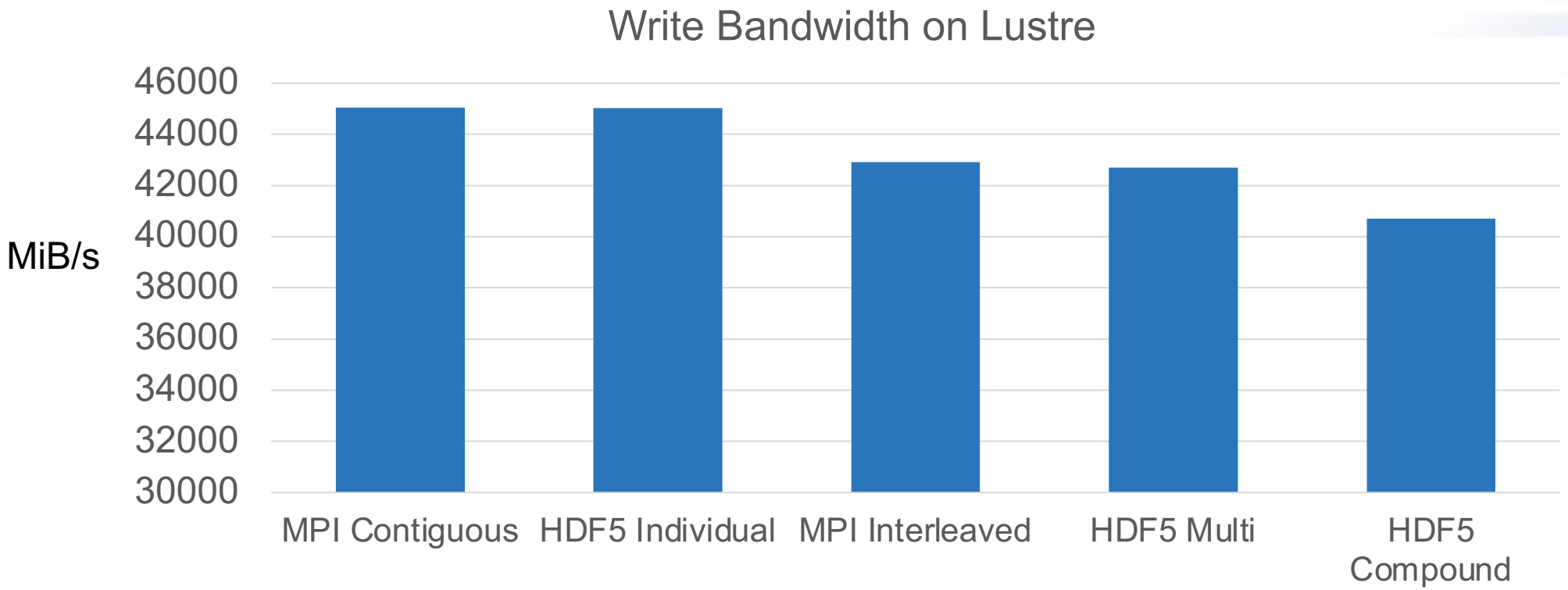
How to match the pure MPI performance

- First find the best stripe size and stripe count using a pure MPI implementation.
- Decide which data layout gives the best performance
  - On Lustre, MPI Contiguous (HDF5 Individual) is better
  - On GPFS, MPI Interleaved (HDF5 Multi) is better
- **Tune HDF5 to match the performance of MPI implementation**

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

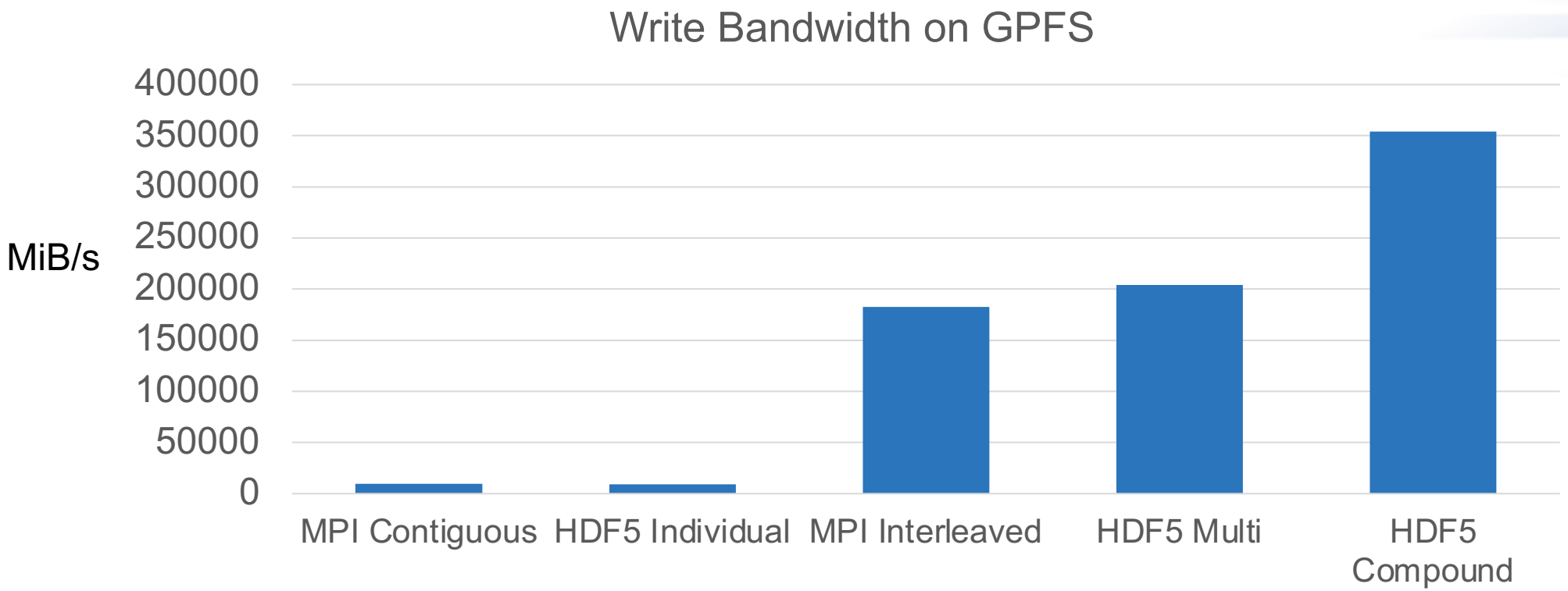
How to match the pure MPI performance

- **Tune HDF5 to match the performance of MPI implementation**
  - **H5Pset_meta_block_size(8MB) on Lustre**



How to match the pure MPI performance

- **Tune HDF5 to match the performance of MPI implementation**
  - **H5Pset_meta_block_size(32MB) on GPFS**



Conclusions

- Current version of HDF5 doesn't have a way to match the MPI-Interleaved access pattern. HDF5_Multi achieves the same pattern.
- Collective I/O does not help since the request size is already very big.
- MPI_Interleaved (HDF5_Multi) is better on GPFS.
  - Because the stripe size is small and users can not change it.
- MPI_Contiguous (HDF5_Individual) is better on Lustre.
  - Because we use a big stripe size (128M)
  - For smaller stripe sizes (e.g. < 36M), MPI_Interleaved is better.

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Tuning Metadata for parallel HDF5 applications

Features for Metadata tuning performance

The HDF Group

BERKELEY LAB
Lawrence Berkeley National Laboratory

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Write Metadata Collectively

- **Symptoms:** Many users reported that `H5Fclose()` is very slow and doesn't scale well on parallel file systems.
- **Diagnosis:** HDF5 metadata cache issues very small accesses (one write per entry). We know that parallel file systems don't do well with small I/O accesses.
- **Solution:** Gather up all the entries of an epoch, create an MPI derived datatype, and issue a single collective MPI write.

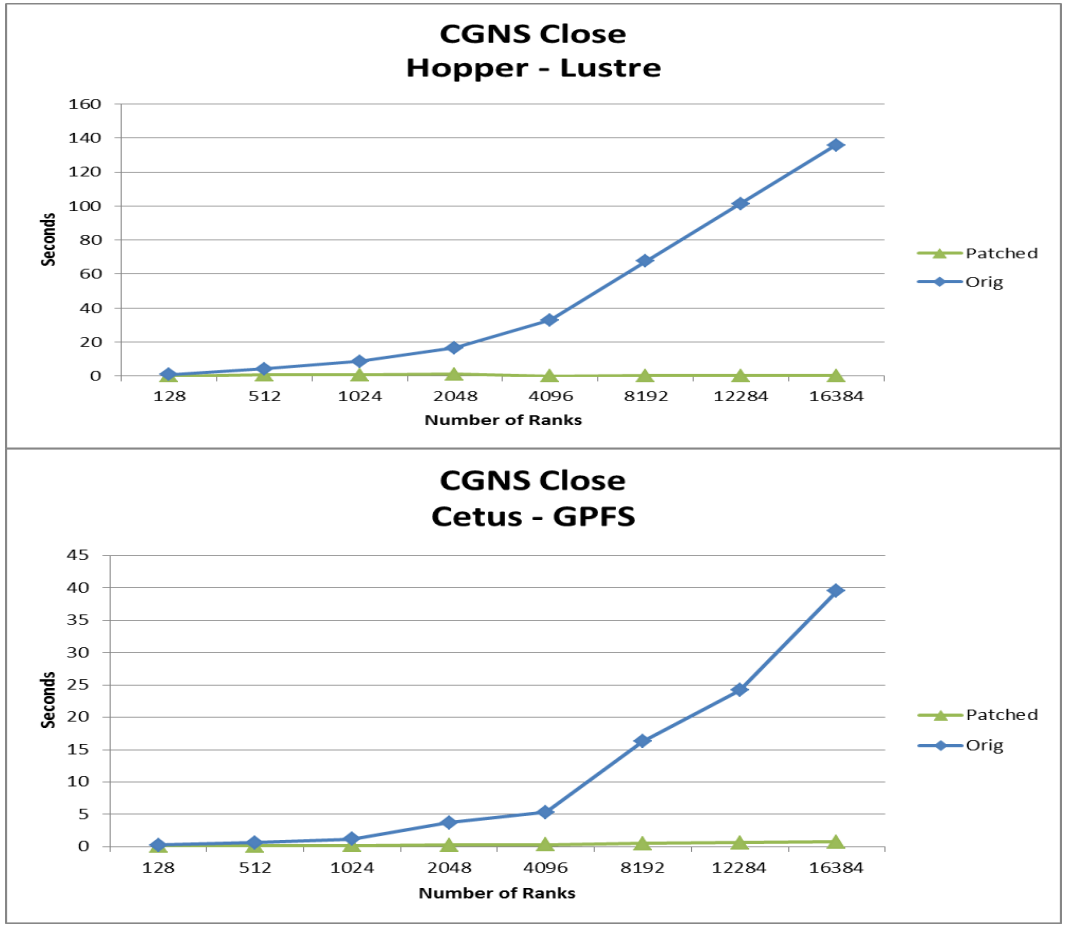
The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Collective Metadata I/O Functions

H5P_SET_COLL_METADATA_WRITE	Establishes I/O mode property setting, collective or independent, for metadata writes
H5P_GET_COLL_METADATA_WRITE	Retrieves I/O mode property setting for metadata writes
H5P_SET_ALL_COLL_METADATA_OPS	Establishes I/O mode, collective or independent, for metadata read operations
H5P_GET_ALL_COLL_METADATA_OPS	Retrieves I/O mode for metadata read operations

The HDF Group
BERKELEY LAB
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Closing a CGNS File …



The HDF Group

BERKELEY LAB
Lawrence Berkeley National Laboratory

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT


EXASCALE COMPUTING PROJECT


Argonne
NATIONAL LABORATORY

Thank you!


U.S. DEPARTMENT OF
ENERGY
Office of
Science


National Nuclear Security Administration