CSE 5449: Intermediate Studies in Scientific Data Management

Lecture 4: I/O Software stack – I/O libraries, HDF5

Dr. Suren Byna

The Ohio State University

E-mail: byna.1@osu.edu

https://sbyna.github.io

01/19/2023

Summary of the last class

- Class projects
  - Homework:
    - Go through the projects and discuss if there are any questions / concerns
    - Select one project and let me know which one you would like to work on – Jan 26th
    - Provide an initial plan of execution – list tasks and timelines – Jan 26th
- What is parallel computing?
- High-level concept of parallel I/O

Today’s class

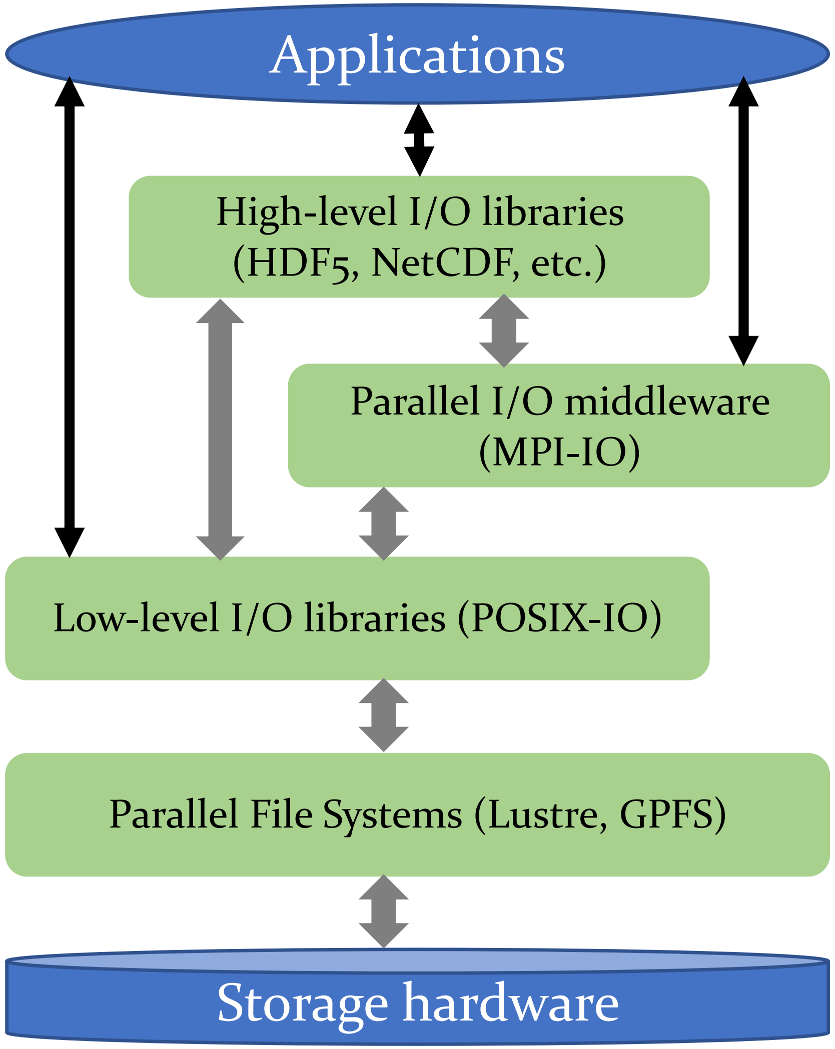
- Parallel I/O software stack
- An intro to HDF5

Class projects

5. Performance comparison of sub-filing in HDF5 and PnetCDF

- Background: Sub-filing is an approach to split a very large file into smaller files. However, there are pros / cons with the approach on how the data is organized.
- Question
  - Which of the HDF5 and PnetCDF sub-filing approaches are best?
  - What better strategies for sub-filing are there?
- Deliverable: A short paper describing
- Resources
  - Tuning HDF5 subfiling performance on parallel file systems https://escholarship.org/content/qt6fs7s3jb/qt6fs7s3jb.pdf
  - Using Subfiling to Improve Programming Flexibility and Performance of Parallel Shared-file I/O https://ieeexplore.ieee.org/document/5362452
  - Scalable Parallel I/O on a Blue Gene/Q Supercomputer Using Compression, Topology-Aware Data Aggregation, and Subfiling https://ieeexplore.ieee.org/document/6787260

Data storage and access – Software layers in HPC systems



Applications

High-level I/O libraries (HDF5, NetCDF, etc.)

Parallel I/O middleware (MPI-IO)

Low-level I/O libraries (POSIX-IO)

Parallel File Systems (Lustre, GPFS)

Storage hardware

High-level I/O libraries

- High-level I/O libraries for hiding the complexity of the I/O stack
- Easy to map memory-level data structures to file / storage data structures
- Often have rich application programming interfaces
- Examples:
  - netCDF, HDF5, PnetCDF, ADIOS, ROOT (sequential), FITS (sequential)
  - Higher-level interfaces for simplicity: h5py, netcdf4-python, h5cpp, h5part, NeXus, etc.

File systems

- A file system is a software that manages a collection of files on storage hardware
- In a sequential system (laptop, a workstation, a server, etc.), file system is part of the OS
  - ext3, ext4, JFS, XFS, BtrFS, APFS, …

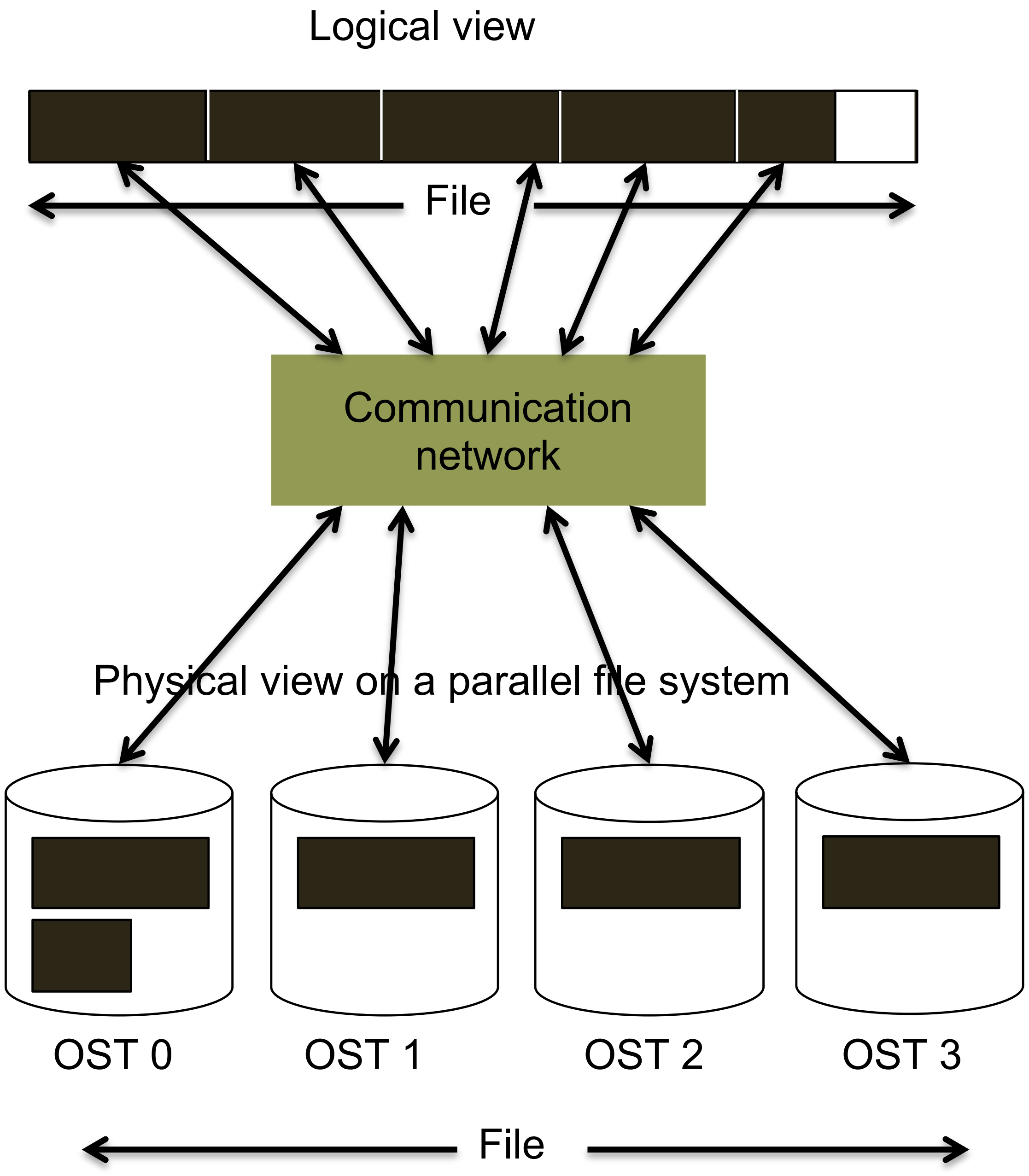
- File system functions
  - Specifying paths
  - Partitioning storage, mounting
  - Managing directories and drives
  - File extensions
  - Advanced: compression, data integrity, fault-tolerance, encryption, etc.

Parallel file systems

- Parallel file system is for managing multiple storage devices
- Gives the view of a single image
- Popular parallel file systems
  - Lustre, Spectrum Scale (GPFS), BeeGFS, GlusterFS, Ceph, Hadoop Distributed FS

Parallel I/O from file system view

- Typical building blocks of parallel file systems
    - Storage hardware – HDD or SSD RAID
    - Storage servers (in Lustre, Object Storage Servers [OSS], and object storage targets [OST]
    - Metadata servers
    - Client-side processes and interfaces
- Management
    - Stripe files for parallelism
    - Tolerate failures



Why I/O libraries?

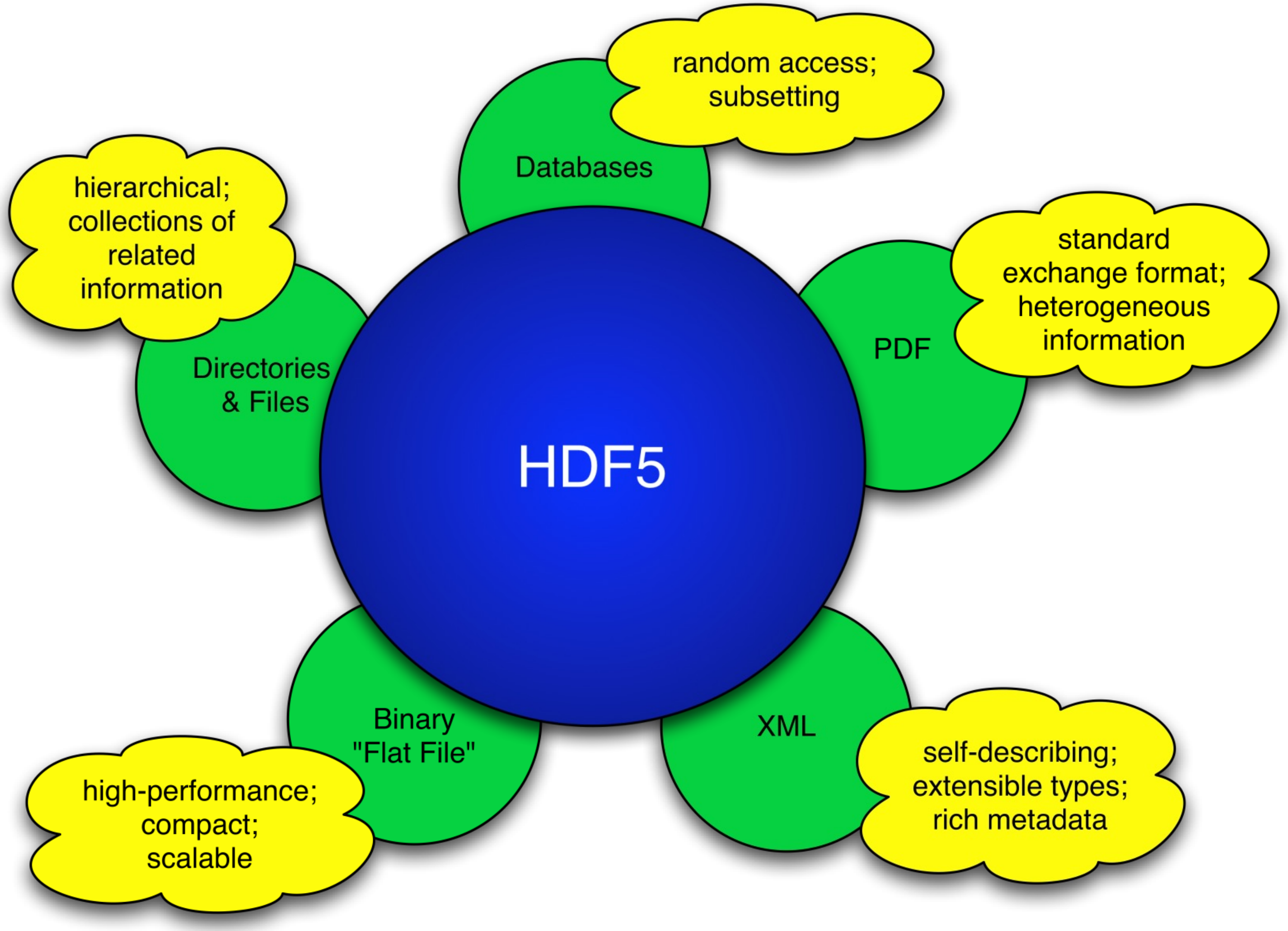
- Have you ever asked yourself:
  - How will I deal with file-per-processor I/O in the exascale era?
  - Do I need to be an “MPI / Lustre / DataWarp / … expert” to save my data?
  - Where *is* my checkpoint file?

- I/O libraries, such as HDF5 hide I/O complexity so you can concentrate on science
  - Optimized I/O to single shared file
  - “Sub-file” I/O from many processes to “n” files (“M➔N I/O”)

What is HDF5?

The following HDF5 slides are from Quincey Koziol, the HDF Group, and ExaHDF5

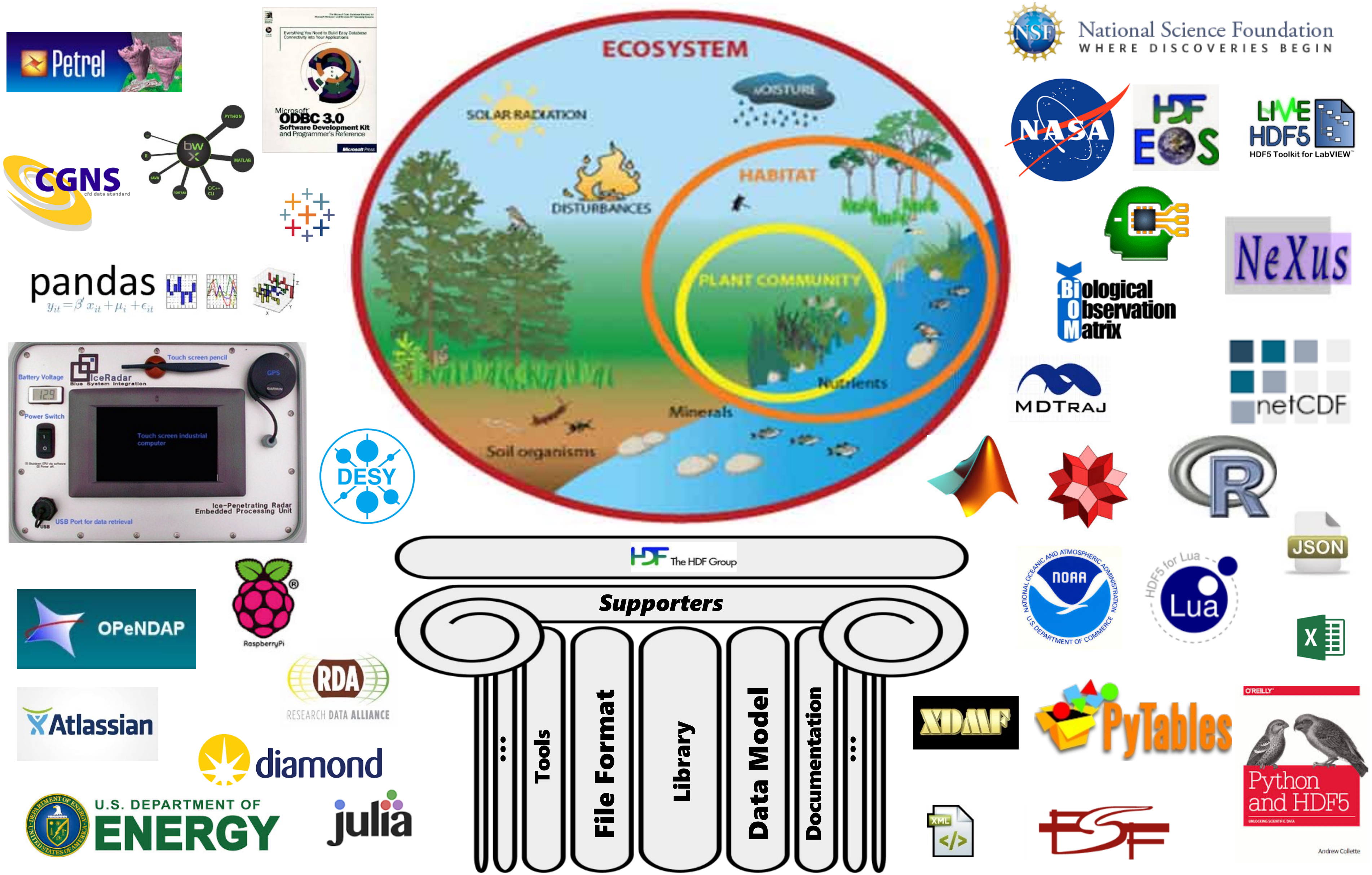
HDF5 is like …


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

HDF5 is designed …

- for high volume and / or complex data
- for every size and type of system – from cell phones to supercomputers
- for flexible, efficient storage and I/O
- to enable applications to evolve in their use of HDF5 and to accommodate new models
- to support long-term data preservation

HDF5 Ecosystem



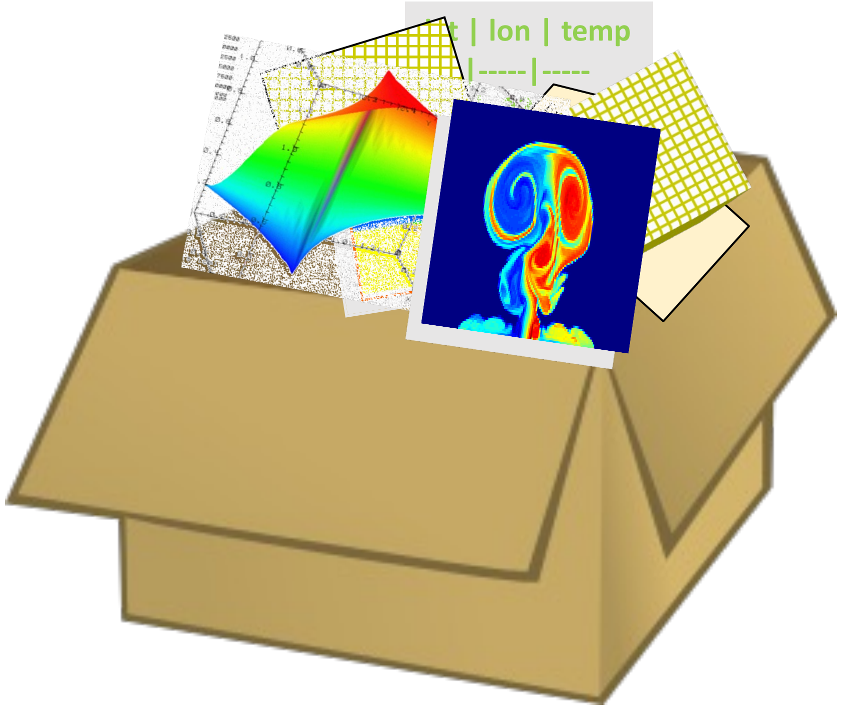
What is HDF5?

- HDF5 ➔ Hierarchical Data Format, v5
- Open **file format**
  - Designed for high volume and complex data
- Open-source **software**
  - Works with data in the format
- An extensible **data model**
  - Structures for data organization and specification

HDF5 Data model

HDF5 File

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



HDF5 Data Model

HDF5

Objects

Dataset

Group

Attribute

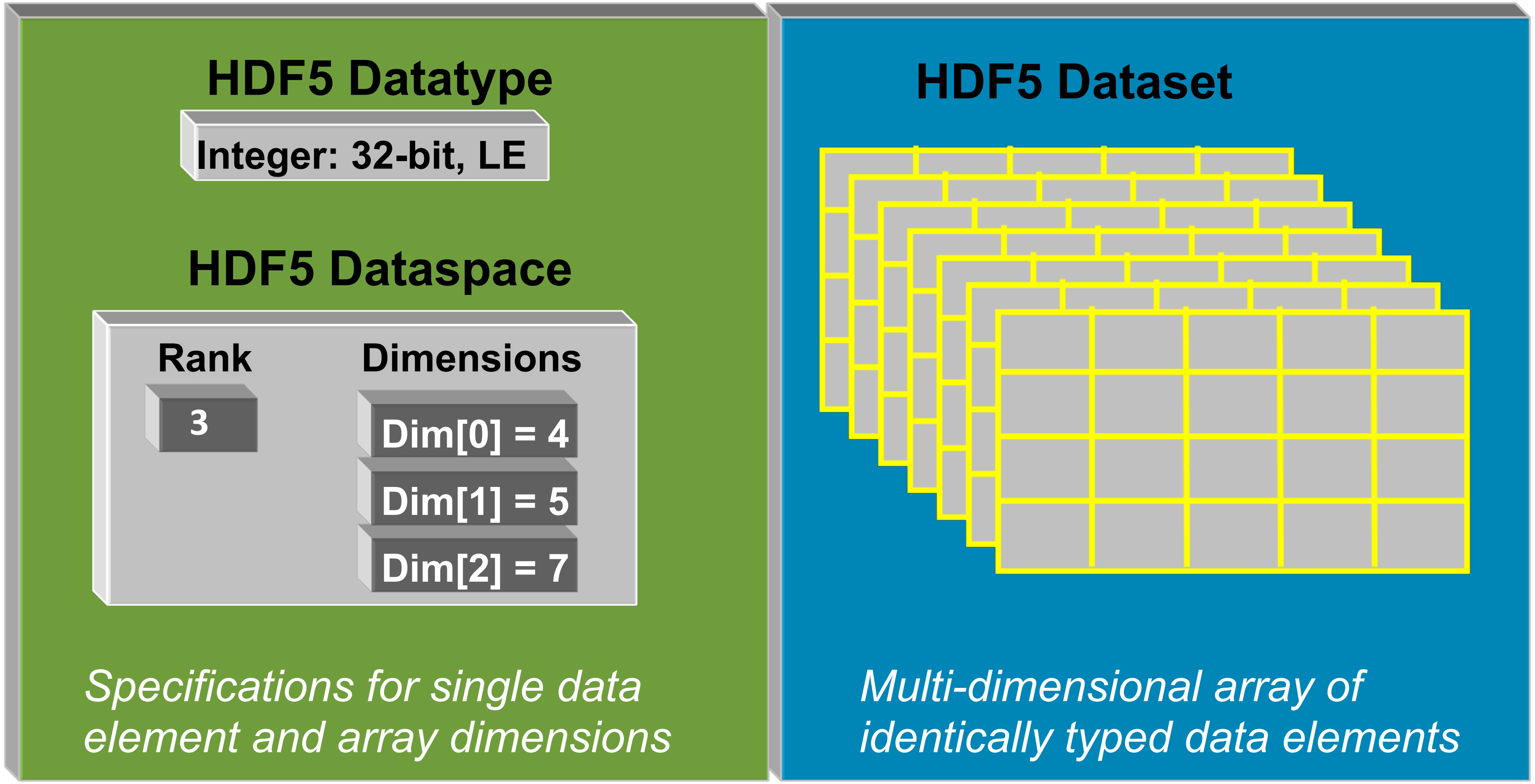
File

Link

Datatype

Dataspace

HDF5 Dataset


HDF5 Datatype
Integer: 32-bit, LE
HDF5 Dataspace
Rank
3
Dimensions
Dim[0] = 4
Dim[1] = 5
Dim[2] = 7
Specifications for single data element and array dimensions
HDF5 Dataset
Multi-dimensional array of identically typed data elements

- HDF5 datasets **organize and contain** data elements.
  - HDF5 datatype describes individual data elements.
  - HDF5 dataspace describes the logical layout of the data elements.

HDF5 Dataspace

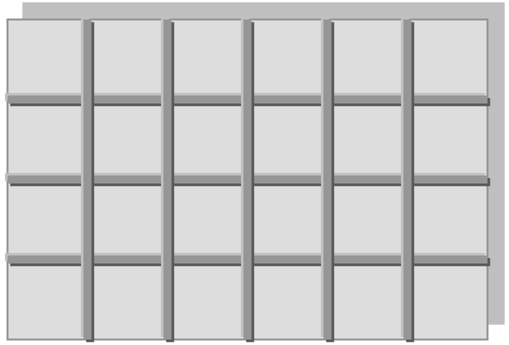
- Describes the logical layout of the elements in an HDF5 dataset
  - NULL
    - no elements
  - Scalar
    - single element
  - Simple array (*most common*)
    - multiple elements organized in a rectangular array
      - rank = number of dimensions
      - dimension sizes = number of elements in each dimension
      - maximum number of elements in each dimension
        - may be fixed or unlimited

HDF5 Dataspace

Two roles:

Dataspace contains spatial information

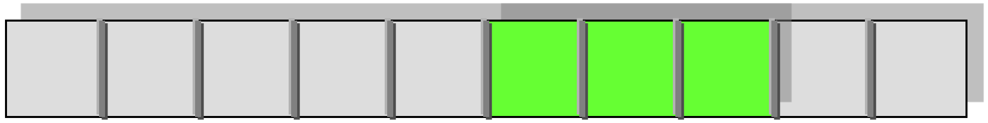
- Rank and dimensions
- Permanent part of dataset definition



Rank = 2

Dimensions = 4x6

Partial I/0: Dataspace describes application's data buffer and data elements participating in I/O



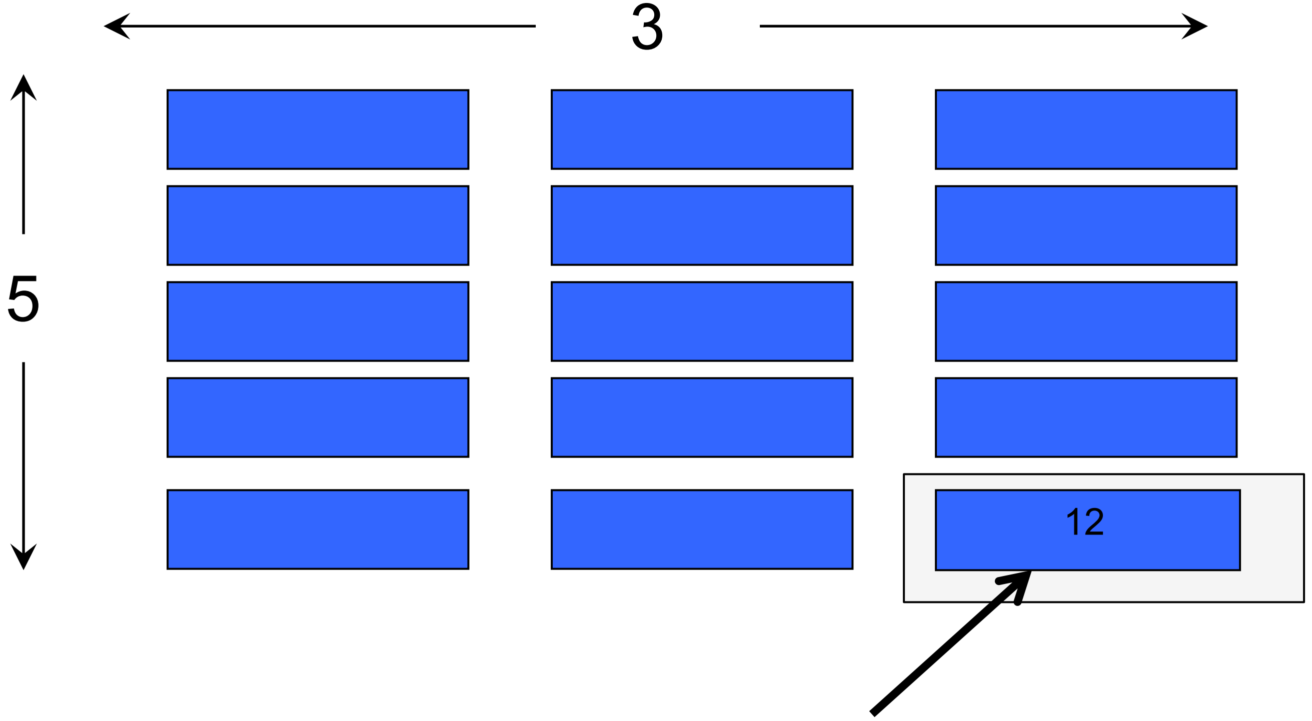
Rank = 1

Dimension = 10

HDF5 Datatypes

- Describe individual data elements in an HDF5 dataset
- Wide range of datatypes supported
  - Integer
  - Float
  - Enum
  - Array
  - User-defined (e.g., 13-bit integer)
  - Variable-length types (e.g., strings, vectors)
  - Compound (similar to C structs)
  - More …

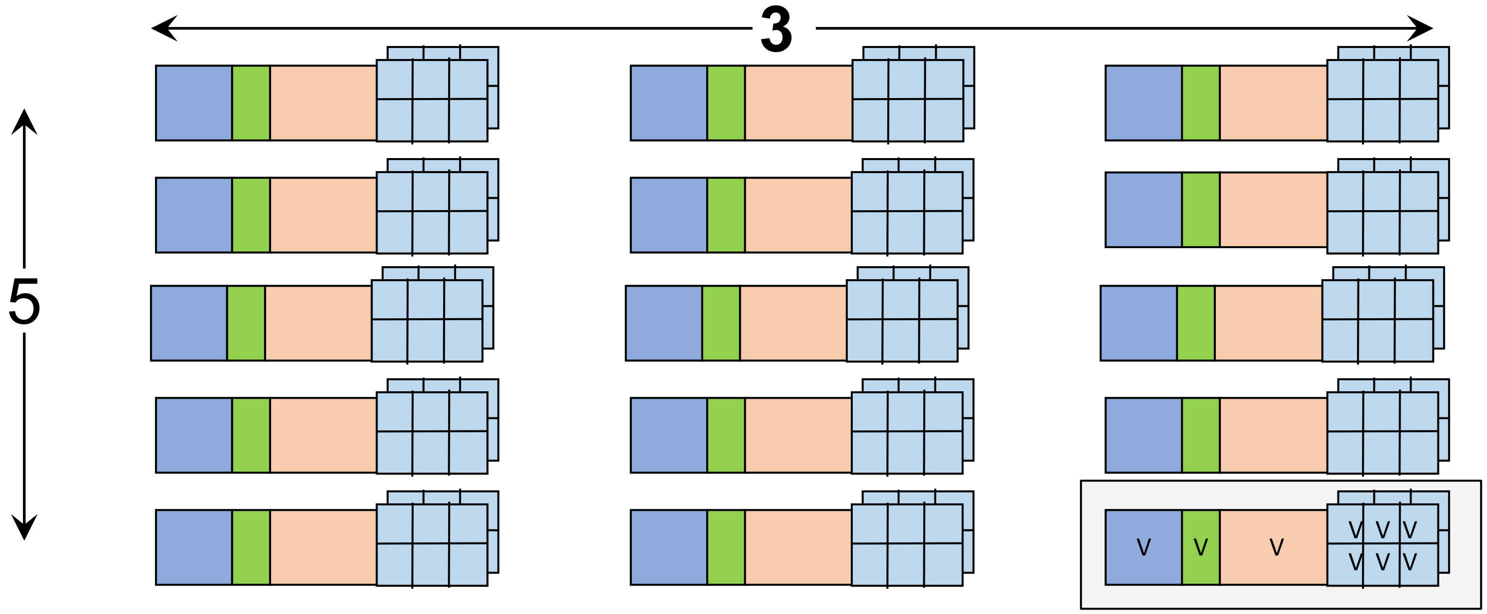
HDF5 Dataset



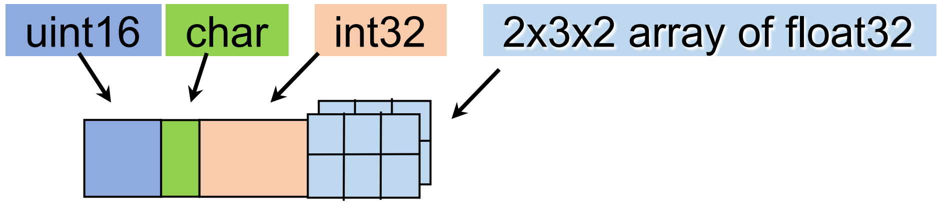
Datatype: 32-bit Integer

Dataspace: Rank = 2
Dimensions = 5 x 3

HDF5 Dataset with Compound Datatype



Compound Datatype:

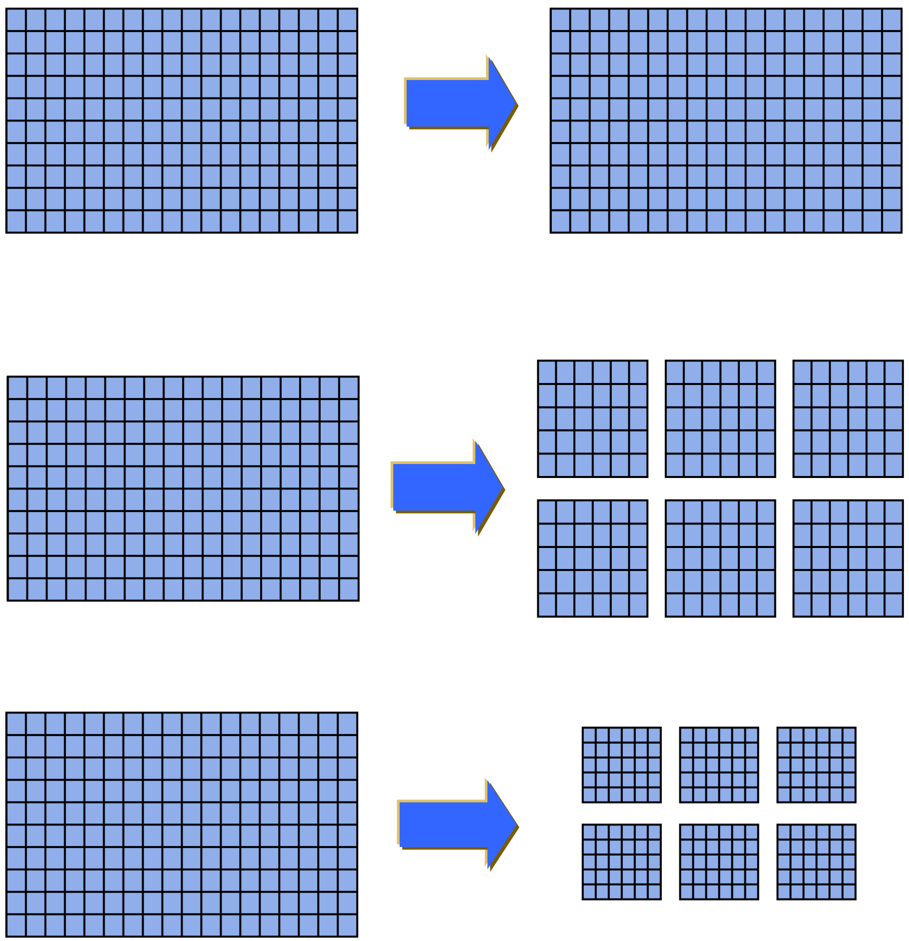


Dataspace: Rank = 2
Dimensions = 5 x 3

How are data elements stored?

Buffer in memory

Data in the file



Contiguous (default)

Data elements stored physically adjacent to each other

Chunked

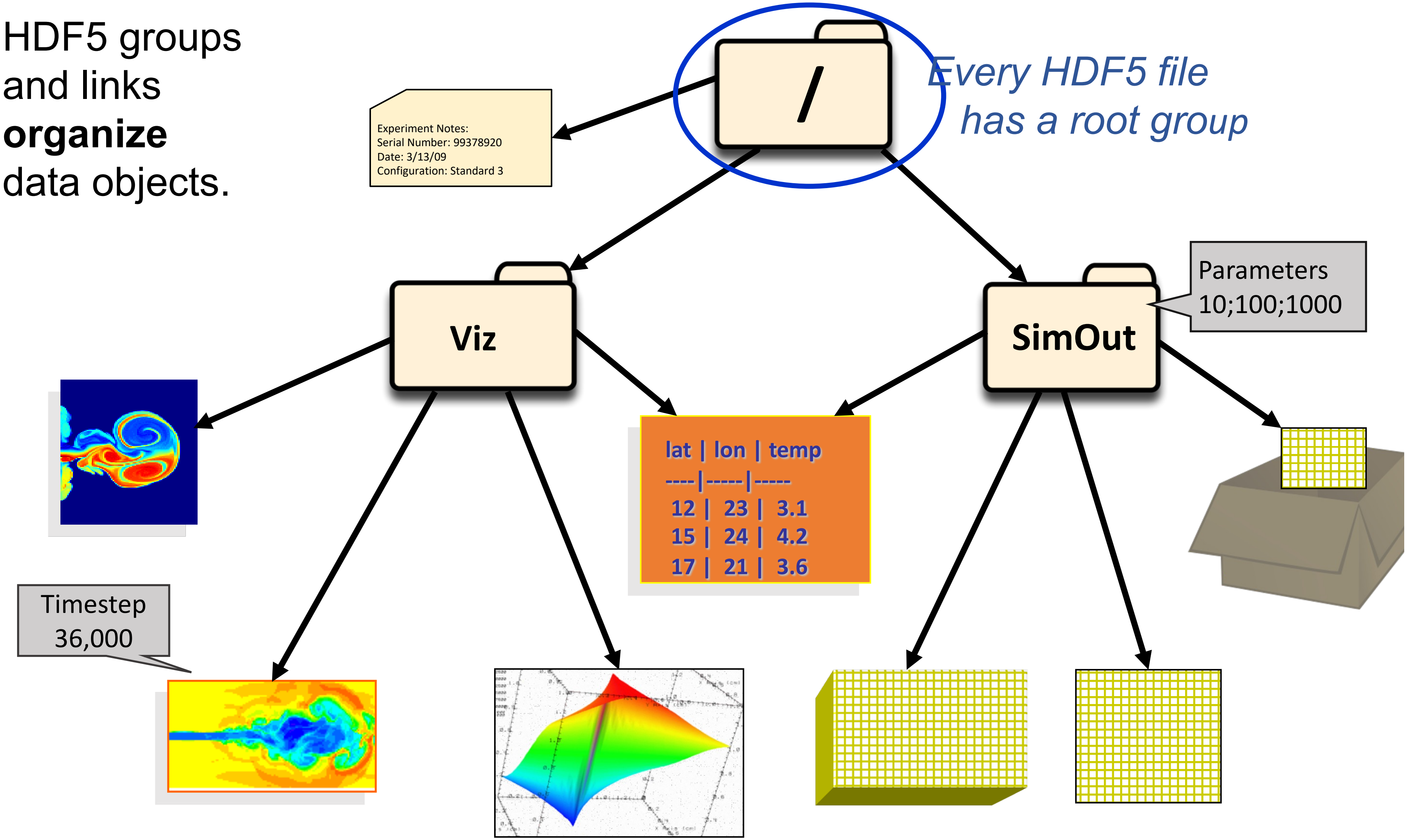
Better access time for subsets; extendible

Chunked & Compressed

Improves storage efficiency, transmission speed

HDF5 Groups and Links

HDF5 groups and links **organize** data objects.



Every HDF5 file has a root group

Experiment Notes:
Serial Number: 99378920
Date: 3/13/09
Configuration: Standard 3

Parameters
10;100;1000

lat	lon	temp
12	23	3.1
15	24	4.2
17	21	3.6

Timestep
36,000

HDF5 Attributes

- Typically contain user metadata
- Have a name and a value
- Attributes “decorate” HDF5 objects
- Value is described by a datatype and a dataspace
- Analogous to a dataset, but do not support partial I/O operations
  - Nor can they be compressed or extended

HDF5 software

HDF5 Home Page

HDF5 home page: http://www.hdfgroup.org/solutions/hdf5/

- Latest release: HDF5 1.14.0

HDF5 source code:

- Written in C, and includes optional C++, Fortran, and Java APIs
  - Along with “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 and require the SZIP and ZLIB external libraries

Useful Tools For New Users

`h5dump`:

Tool to “dump” or display contents of HDF5 files

`h5cc, h5c++, h5fc`:

Scripts to compile applications (like mpicc, …)

HDFView: Java browser to view HDF5 files
http://support.hdfgroup.org/products/java/hdfview/

HDF5 Examples (C, Fortran, Java, Python, Matlab, …)
http://support.hdfgroup.org/HDF5/examples/

Homework

- Install HDF5 on your laptop or on OSC
- Go to https://docs.hdfgroup.org/hdf5/develop/_h_d_f5_examples.html
  - Run the Examples from Learning the Basics page
  - Report the observations in the next class

Summary of today’s class

- Parallel I/O software stack
- I/O libraries, HDF5
  - Homework: Install HDF5 and run examples

After the class, slides are uploaded to: https://osu.instructure.com/courses/141406/files

Also available at: https://sbyna.github.io/teaching/5449-sdm.html

Next class

- Discussion of class projects
- More HDF5 I/O API
- Parallel HDF5 concepts