



HDF5 Update

Quincey Koziol

koziol@hdfgroup.org

The HDF Group

HEC-FSIO Workshop

August 3, 2010



HDF5 Technology Platform

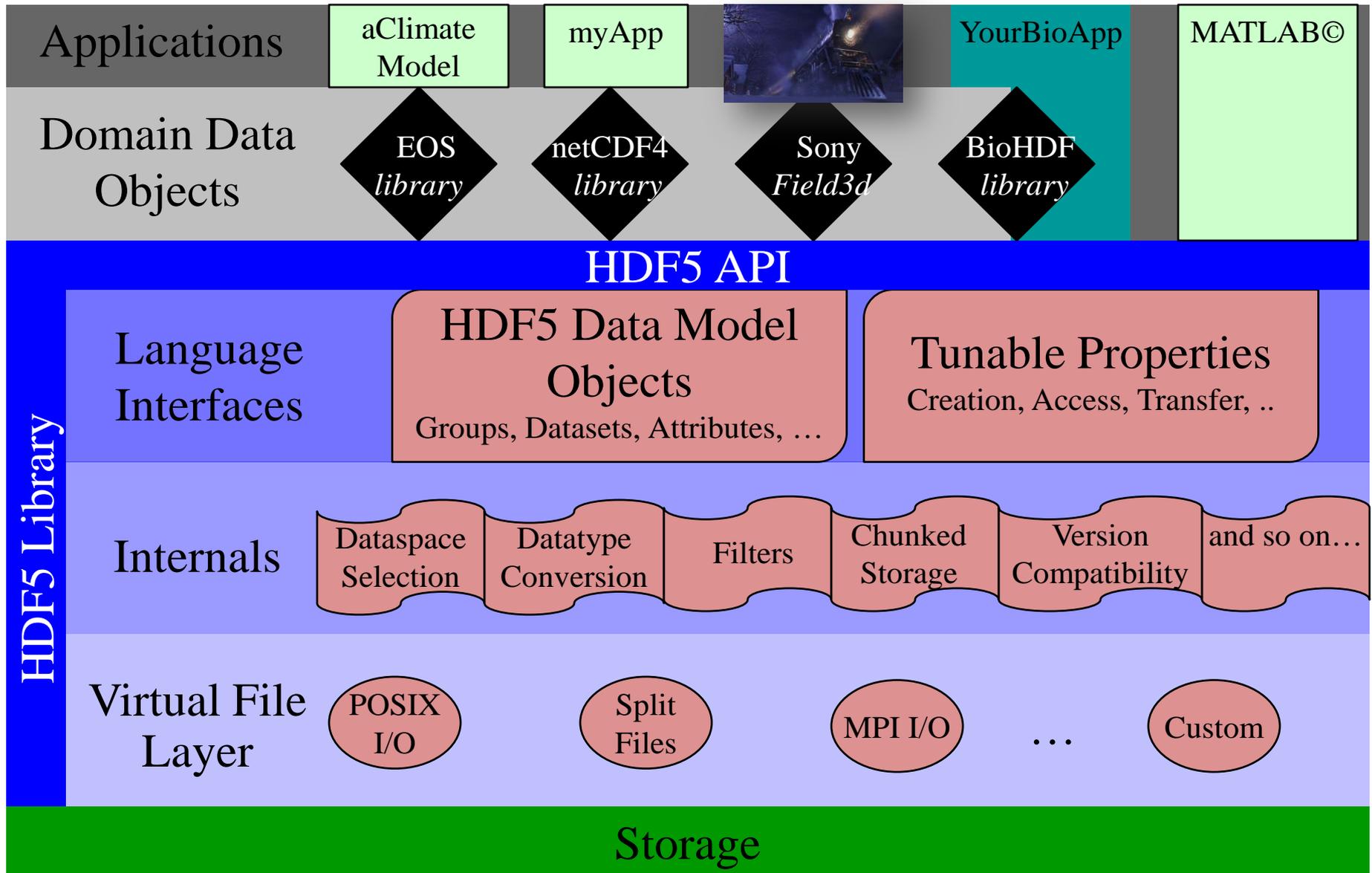
- **HDF5 Abstract Data Model**
 - Defines the “building blocks” for data organization and specification
 - Files, Groups, Links, Datasets, Attributes, Datatypes, Dataspaces

- **HDF5 Software**
 - Tools
 - Language Interfaces
 - HDF5 Library

- **HDF5 Binary File Format**
 - Bit-level organization of HDF5 file
 - Defined by HDF5 File Format Specification



HDF5 API and Applications





Data challenges addressed by HDF5

- Ability to organize complex collections of data
- Efficient and scalable data storage and access
- A growing need to integrate a wide variety of types of data
- The evolution of data technologies
- Long term preservation of data



Areas of increased recent interest

- Life sciences
 - Gene sequencing
 - Biomedical imaging
- High performance computing (HPC)
- Microsoft products (HPC, .NET, others)
- Database integration (and replacement)
- Improvements
 - Concurrent access
 - Improving parallel I/O performance
 - Improving real-time write performance
 - Improving high level language support

The biosciences need an image format capable of high performance and long-term maintenance. Is HDF5 the answer?

BY MATTHEW T. DOUGHERTY, MICHAEL J. FOLK, EREZ ZADOK, HERBERT J. BERNSTEIN, FRANCES C. BERNSTEIN, KEVIN W. ELICEIRI, WERNER BENGER, CHRISTOPH BEST

Unifying Biological Image Formats with HDF5

THE BIOLOGICAL SCIENCES need a generic image format suitable for long-term storage and capable of handling very large images. Images convey profound ideas in biology, bridging across disciplines. Digital imagery began 50 years ago as an obscure technical phenomenon. Now it is an indispensable computational tool. It has produced a variety of incompatible image file formats, most of which are already obsolete.

Several factors are forcing the obsolescence: rapid increases in the number of pixels per image;

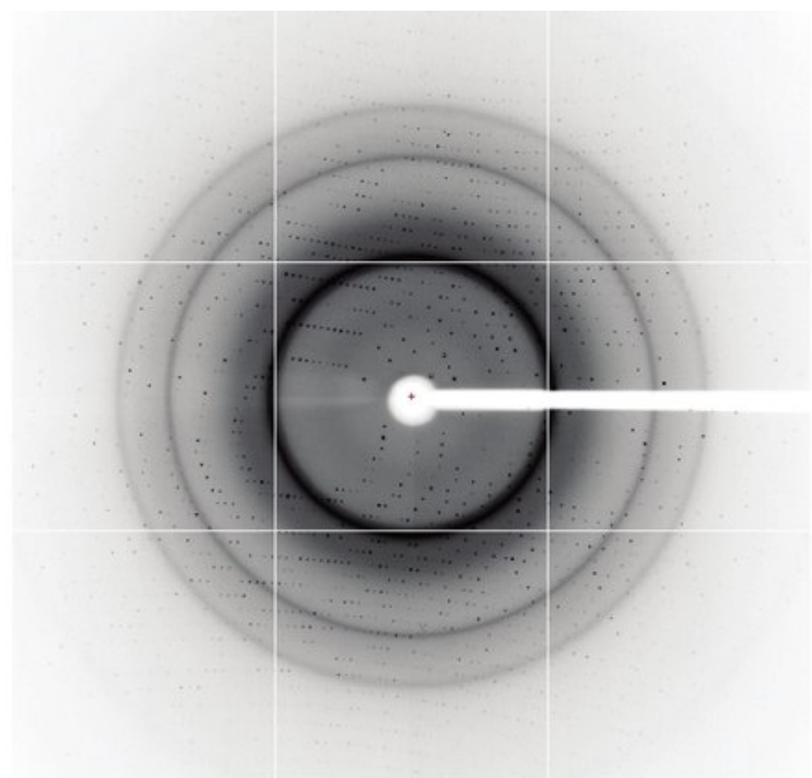
acceleration in the rate at which images are produced; changes in image designs to cope with new scientific instrumentation and concepts; collaborative requirements for interoperability of images collected in different labs on different instruments; and research metadata dictionaries that must support frequent and rapid extensions. These problems are not unique to the biosciences. Lack of image standardization is a source of delay, confusion, and errors for many scientific disciplines.

There is a need to bridge biological and scientific disciplines with an image framework capable of high computational performance and interoperability. Suitable for archiving, such a framework must be able to maintain images far into the future. Some frameworks represent partial solutions: a few, such as XML, are primarily suited for interchanging metadata; others, such as CIF (Crystallographic Information Framework),² are primarily suited for the database structures needed for crystallographic data mining; still others, such as DICOM (Digital Imaging and Communications in Medicine),³ are primarily suited for the domain of clinical medical imaging.

What is needed is a common image framework able to interoperate with all of these disciplines, while providing high computational performance. HDF (Hierarchical Data Format)⁴ is such a framework, presenting a historic opportunity to establish a coin of the realm by coordinating the imagery of many biological communities. Overcoming the digital confusion of incoherent bio-imaging formats will result in better science and wider accessibility to knowledge.

Semantics: Formats, Frameworks, and Images

Digital imagery and computer technology serve a number of diverse biological communities with terminology differences that can result in very different perspectives. Consider the word *format*. To the data-storage community the hard-drive format will play a ma-



An x-ray diffraction image taken by Michael Soltis of LSAC on SSRL BL9-2 using an ADSC Q315 detector (5N901).

ajor role in the computer performance of a community's image format, and to some extent, they are inseparable. A format can describe a standard, a framework, or a software tool; and formats can exist within other formats.

Image is also a term with several uses. It may refer to transient electrical signals in a CCD (charge-coupled device), a passive dataset on a storage device, a location in RAM, or a data structure written in source code. Another example is *framework*. An image framework might implement an image standard, resulting in image files created by a software-imaging tool. The framework, the standard, the files, and the tool, as in the case of HDF,⁴ may be so interrelated that they represent dif-

ferent facets of the same specification. Because these terms are so ubiquitous and varied due to perspective, we shall use them interchangeably, with the emphasis on the storage and management of pixels throughout their lifetime, from acquisition through archiving.

Hierarchical Data Format Version 5

HDF5 is a generic scientific data format with supporting software. Introduced in 1998, it is the successor to the 1988 version, HDF4. NCSA (National Center for Supercomputing Applications) developed both formats for high-performance management of large heterogeneous scientific data. Designed to move data efficiently between secondary storage and memory,

HDF5 translates across a variety of computing architectures. Through support from NASA (National Aeronautics and Space Administration), NSF (National Science Foundation), DOE (Department of Energy), and others, HDF5 continues to support international research. The HDF Group, a nonprofit spin-off from the University of Illinois, manages HDF5, reinforcing the long-term business commitment to maintain the format for purposes of archiving and performance.

Because an HDF5 file can contain almost any collection of data entities in a single file, it has become the format of choice for organizing heterogeneous collections consisting of very large and complex datasets. HDF5 is

Cool recent application: Sony Imageworks' Field3D



Spiderman 3



The Polar Express



Topics

What's up with The HDF Group?

Library update

Tools update

HDF Java Products

Library development in the works

Other activities



What's up with The HDF Group?



The HDF Group Mission

To ensure long-term accessibility of HDF data through sustainable development and support of HDF technologies.



Goals of The HDF Group

- Maintain and evolve HDF for sponsors and communities that depend on it
- Provide support to the HDF communities through consulting, training, tuning, development, research
- Sustain the company for the long term to assure data access over time



The HDF Group Services

- Helpdesk and Mailing Lists
 - Available to all users as a first level of support
- Priority Support
 - Rapid issue resolution and advice
- Consulting
 - Needs assessment, troubleshooting, design reviews, etc.
- Training
 - Tutorials and hands-on practical experience
- Enterprise Support
 - Coordinating HDF activities across departments
- Special Projects
 - Adapting customer applications to HDF
 - New features and tools
 - Research and Development



The HDF Group

- Established in 1988
 - 18 years at University of Illinois - National Center for Supercomputing Applications
 - 4.5 years as independent *non-profit* company: “The HDF Group”
- The HDF Group owns HDF4 and HDF5
 - Basic HDF4 and HDF5 formats, libraries, and tools are open and free
- Currently employ ~30 FTEs

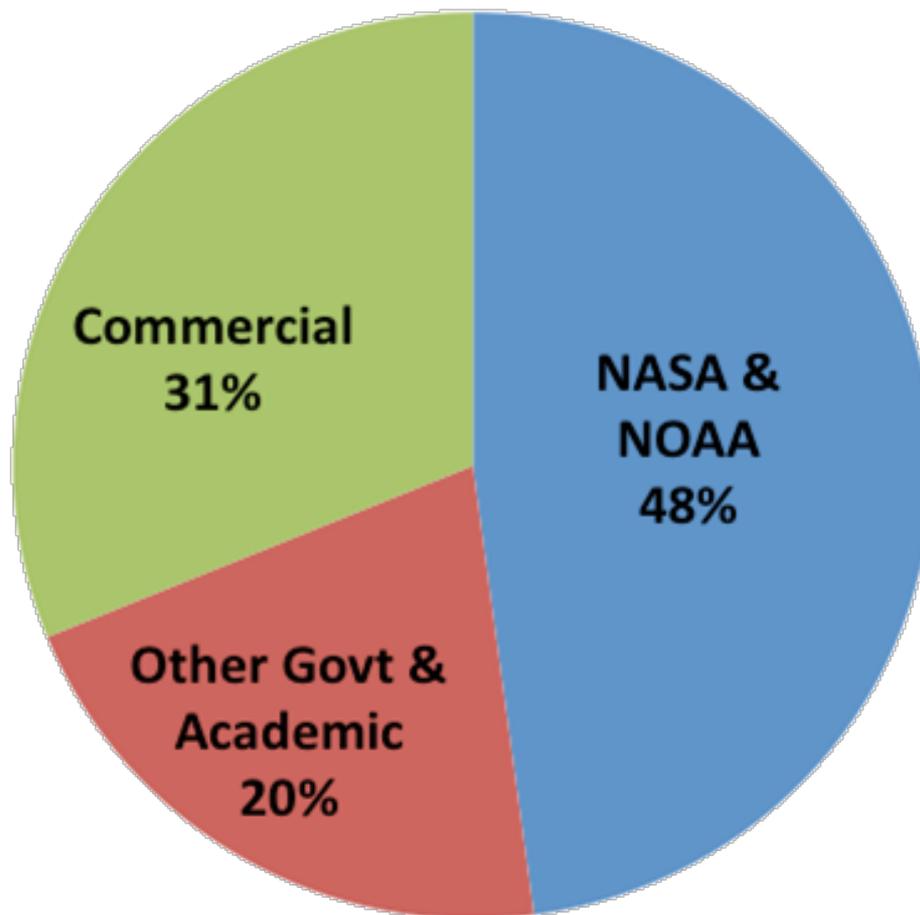


Members of the HDF support community

- NASA – Earth Observing System
- NOAA/NASA/Riverside Tech – NPOESS
- Army Geospatial Center
- A leading U.S. aerospace company
- A large financial firm
- An international semiconductor manufacturing company
- NIH/Geospiza (bio software company)
- University of Illinois/NCSA
- Sandia National Laboratory
- Lawrence Berkeley National Laboratory
- Lawrence Livermore National Laboratory
- DOE, via FOA grant proposals
- Projects for petroleum industry, vehicle testing, weapons research, others
- “In kind” support



Income Profile – past 12 months



Total income approximately \$3.4 million



Topics

What's up with The HDF Group?

Library update

Tools update

HDF Java Products

Library development in the works

Other activities



Basic Library Releases





HDF5 1.8.5 release (June '10)

- Initial CMake build support
- Fixed various “strict aliasing” issues, allowing compilation with “-O3” optimization
- Performance is substantially improved when extending a dataset with early allocation (i.e. when using parallel I/O).
- Updated FORTRAN and C++ wrappers
- Minor options added to h5dump and h5diff utilities
- Various bugs and performance issues

What's up with The HDF Group?

Library update

Tools update



HDF Java Products

Library development in the works

Other activities



Major Improvements for Existing Tools

- h5dump additions
 - Added the new packed bits feature which prints packed bits stored in an integer dataset
- H5diff
 - Added new flag --no-dangling-links
 - Added new flag --follow-symlinks



Tool activities in the works

- New tool – h5watch
 - Display new records appended to a dataset
- Improved code quality and testing
- Tools library: general purpose APIs for tools
 - Tools library currently only for our developers
 - Want to make it public so that people can use it in their products

Please send us your comments and requests regarding HDF5 conversion tools, such as:

- HDF4 to HDF5
- HDF5 to jpeg
- HDF5 to XML
- HDF5 to other formats?





Topics

What's up with The HDF Group?

Library update

Tools update

HDF Java Products

Library development in the works

Other activities



HDF-Java 2.6 released

- Includes all HDF java products
 - Java Wrapper API
 - Java Object API
 - HDFView
- Adds new features, such as dataset region reference
- Improves performance



Full support of HDF5 1.8.x in hdf-java

- Full HDF5 1.8 support will be added soon
- We are looking for input
 - RFC:
<http://www.hdfgroup.uiuc.edu/RFC/HDF5/hdf-java/>
- Java wrapper will be completed May 2010
- Object API and HDFView update to come later

What's up with The HDF Group?

Library update

Tools update

HDF Java Products

Library development in the works

Other activities





Surviving a System Failure

- Problem:
 - In the event of an application or system crash, data in HDF5 files are susceptible to corruption
 - Corruption can occur if structural metadata is being written when the crash occurs
- Initial Objective:
 - Guarantee an HDF5 file with consistent metadata can be reconstructed in the event of a crash



Crash Survivability in HDF5

- Approach: Metadata Journaling
 - When an HDF5 file is opened, a companion journal file is created
 - When an HDF5 function modifies metadata, this modification is recorded in the journal file
 - If the application crashes, a recovery program can replay the journal by applying all metadata writes, ensuring that all metadata in the file is correct



Metadata Journaling: Progress

- Feature complete (but only works w/serial I/O)
- Beta release in August 2009
- Added support for asynchronous I/O of journal writes – Faster!
- Adding support for asynchronous metadata entry writes





Single-Writer/Multiple-Reader Access

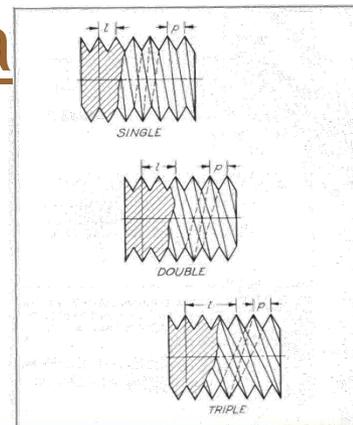
- Situation: A long-running process is modifying an HDF5 file and simultaneously other processes want to inspect data in the file.
- Solution: Single-Writer/Multiple-Reader (SWMR) File Access.
 - Allows simultaneous reading of HDF5 file while the file is being modified by another process
 - No inter-process coordination necessary
 - Also provides method of crash protection





Improved Multi-Threaded Concurrency

- Converting from “big lock” on code (entire library) to locks on internal library data structures
- Will improve ability to have multiple threads performing HDF5 operations simultaneously
- Working with Argonne MPICH team on “Open Portable Atomics” project - <http://trac.mcs.anl.gov/projects/openpa>





Other Library Features

- Saving time
 - New chunk indexing methods
 - Store partial edge chunks more efficiently
 - Aggregate neighboring metadata for faster metadata cache I/O
- Saving space
 - Persistent file free space tracking/recovery
 - Allow a group's link info to be compressed



New chunk indexing methods

Dataset type	Index type	Space improvements	Speed improvements
no unlimited dimensions, no I/O filters, no missing chunks	“implicit” no actual chunk index	Same storage space as contiguous dataset storage (no index)	Zero time lookups, Faster parallel I/O
no unlimited dimensions	“fixed sized” smaller chunk index	Smaller index overhead	Constant time lookups
1 unlimited dimension	“extensible array”	Smaller index overhead	Constant time lookups <i>and appends</i>
2+ unlimited dimension	Improved B-tree	Smaller index overhead	Faster than current B-trees



Future Library Improvements

- Work proposed to recent NSF SDCI CFP:
 - Expand built-in datatypes:
 - Boolean, complex, C99 types, etc.
 - Allow shared dataspace in file
 - More ways to add attributes to HDF5 objects:
 - Attributes on compound datatype fields
 - Attributes on regions within dataspace
 - Store compound datatypes in non-interleaved form
- Work proposed to recent NSF SI2 CFP:
 - Create “virtual object layer” within HDF5 library
 - Allows leveraging PLFS work under HDF5, along with efficient remote access to HDF5 data



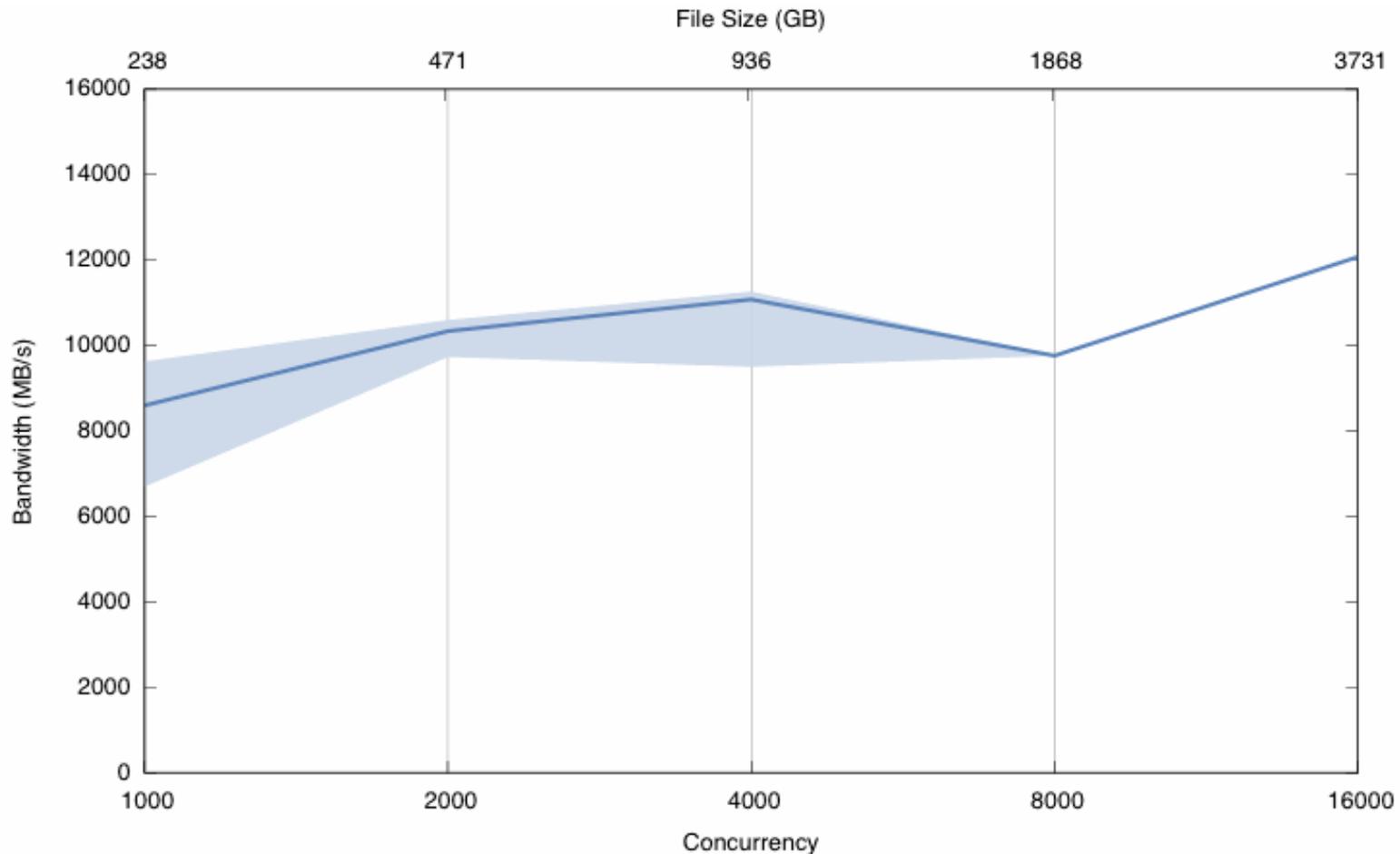
Parallel I/O in HDF5

- Goal is to be invisible: get same performance with HDF5 as with MPI I/O
- Project with NERSC to improve HDF5 performance on parallel applications:
 - 6-12x performance improvements on various applications (so far)



Parallel I/O In HDF5

- Up to 12GB/s to shared file (out of 15GB/s) on NERSC's franklin system:





Parallel I/O Improvements

- Current work: (will be in 1.8.6 release)
 - Reduce number of file truncation operations
 - Distribute metadata I/O over all processes
 - Detect same “shape” of selection in more cases, allowing optimized I/O path to be taken more often
- Upcoming work:
 - Add high-level “HPC” API interface
 - Improvements to MPI-IO and MPI-POSIX VFDs and library algorithms for faster/better use of MPI



- Pass along MPI Info hints to file open: *H5Pset_fapl_mpio*
- Use MPI-POSIX file driver to access file: *H5Pset_fapl_mpio*
- Align objects in HDF5 file: *H5Pset_alignment*
- Use collective mode when performing I/O on datasets: *H5Pset_dxpl_mpio* before *H5Dwrite/H5Dread*
- Avoid datatype conversions: make memory and file datatypes the same
- Advanced: explicitly manage metadata flush operations with *H5Fset_mdc_config*



Future Parallel I/O Improvements

- DOE Exascale FOA w/LBNL & PNNL Funded to:
 - Remove collective metadata modification restriction
 - Append-only mode, targeting restart files
 - Overlapping compute & I/O, with async I/O
 - Auto-tuning to underlying parallel file system
 - Improve resiliency of changes to HDF5 files
 - Bring FastBit indexing of HDF5 files into mainstream use for queries during data analysis and visualization



Future Parallel I/O Improvements

- Contract w/LLNL to do:
 - Scalable I/O performance tracking, testing and tuning
 - Virtual file driver enhancements
 - HPC Specific fast-tracking
 - Parallel interface enhancements
 - Exploratory design development
 - User support and routine maintenance

What's up with The HDF Group?

Library update

Tools update

HDF Java Products

Library development in the works

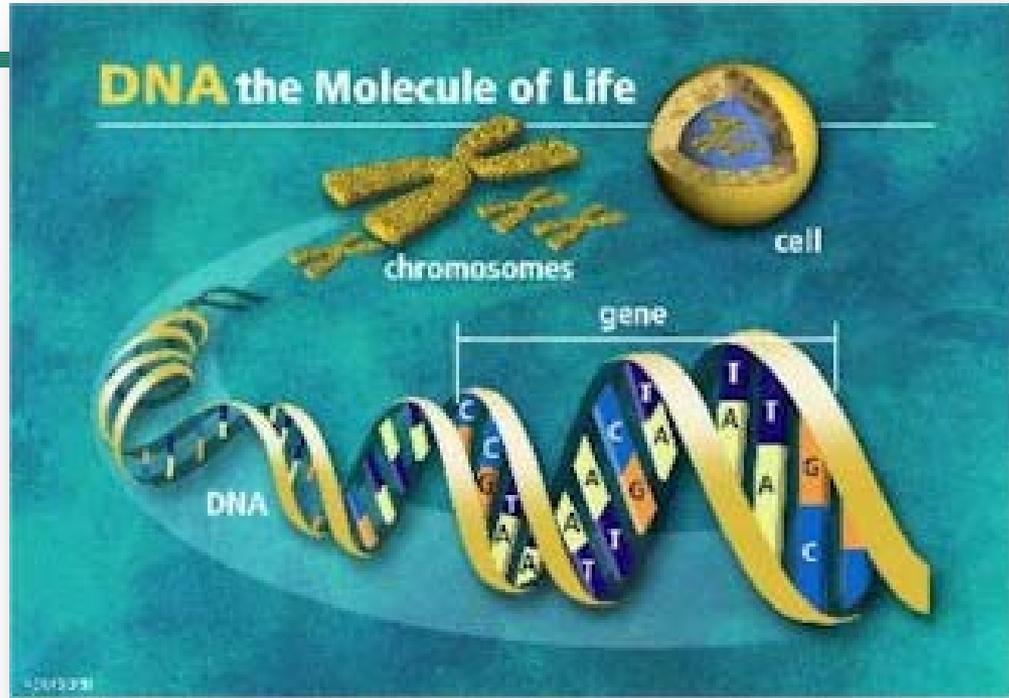


geospiza



Other activities





NIH STTR with Geospiza, Seattle WA

BIOHDF : TOWARD SCALABLE BIOINFORMATICS INFRASTRUCTURES



Next Generation DNA Sequencing

“Transforms today’s biology”

“Democratizing genomics”

“Changing the landscape”

“Genome center in a mail room”

“The beginning of the end for microarrays”

NGS is Powerful



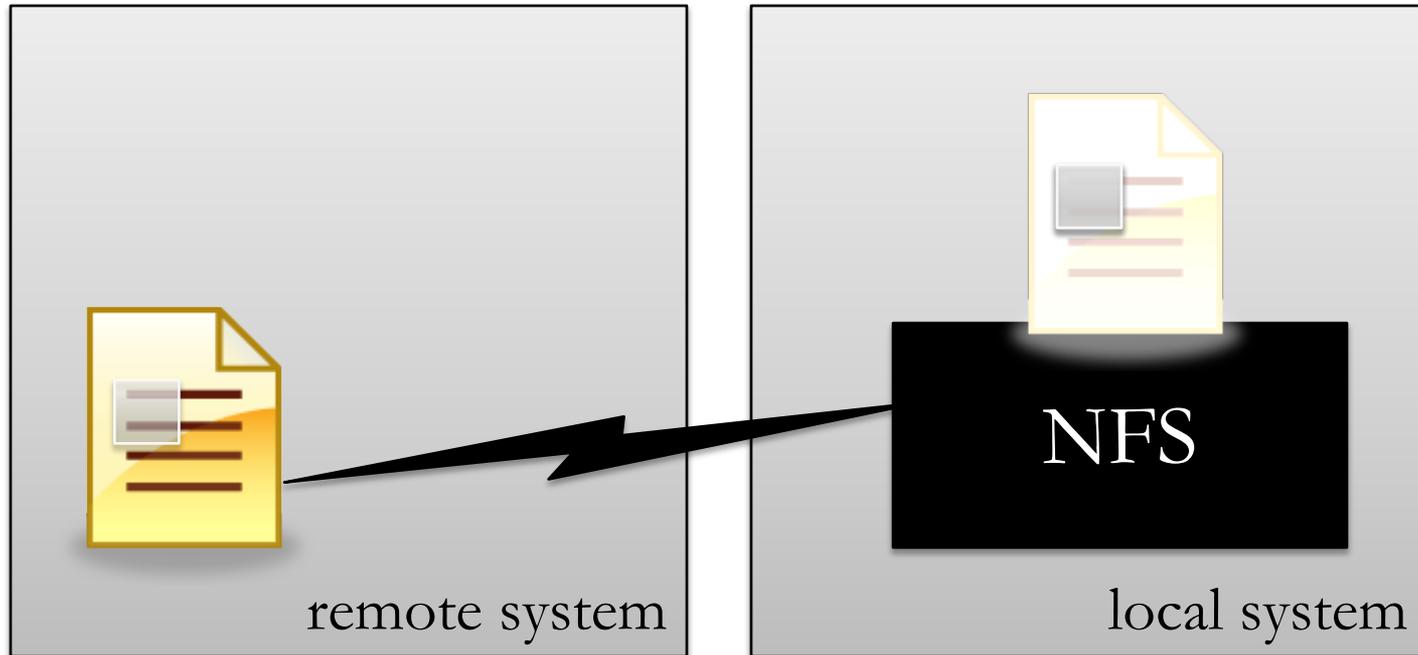


- ***Goal: Move bioinformatics problems from organizing and structuring data to asking questions and visualizing data***
 - Develop data models and tools to work with NGS data in HDF5
 - Create HDF5 domain-specific extensions and library modules to support the unique aspects of NGS data → BioHDF
 - Integrate BioHDF technologies into Geospiza products
- **Deliver core BioHDF technologies to the community as open-source software**



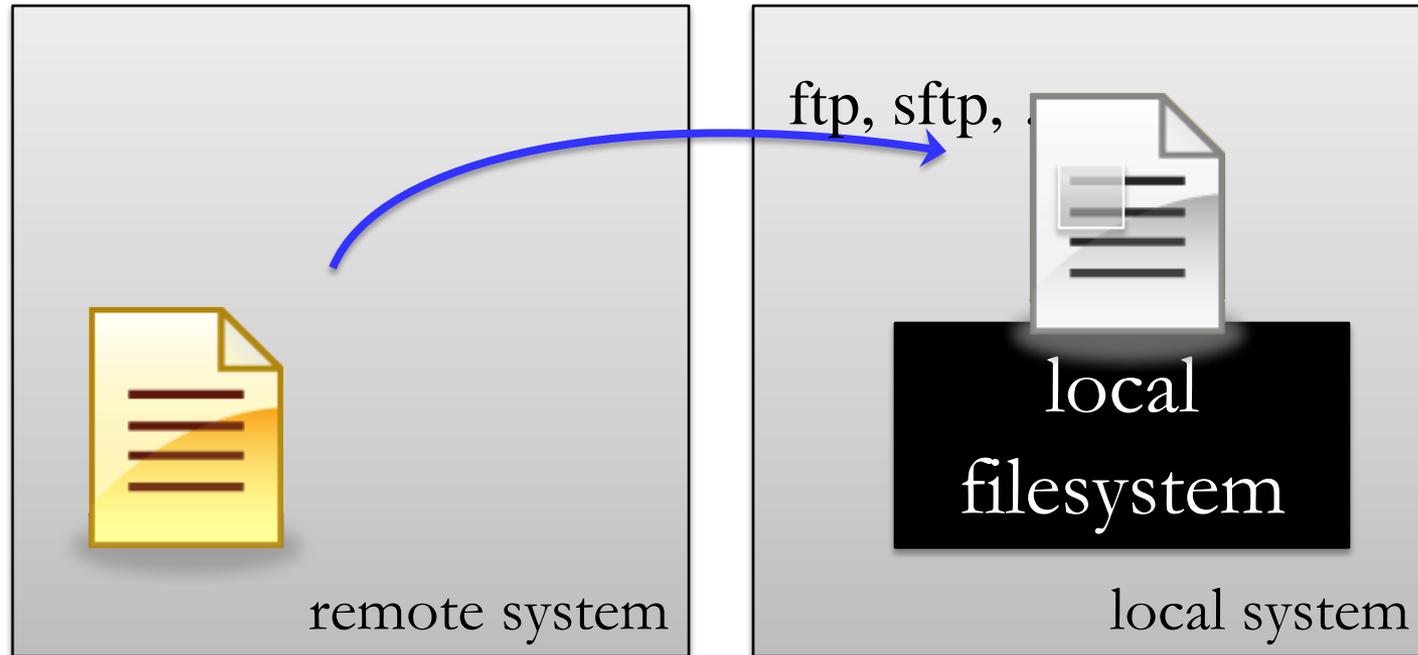
Performance evaluation of using SSHFS-FUSE to access HDF5 files

The word 'FUSE' is written in large, 3D, metallic-looking letters. The letters are gold-colored with a dark shadow underneath, giving them a three-dimensional appearance. The background is a light, hazy gradient.

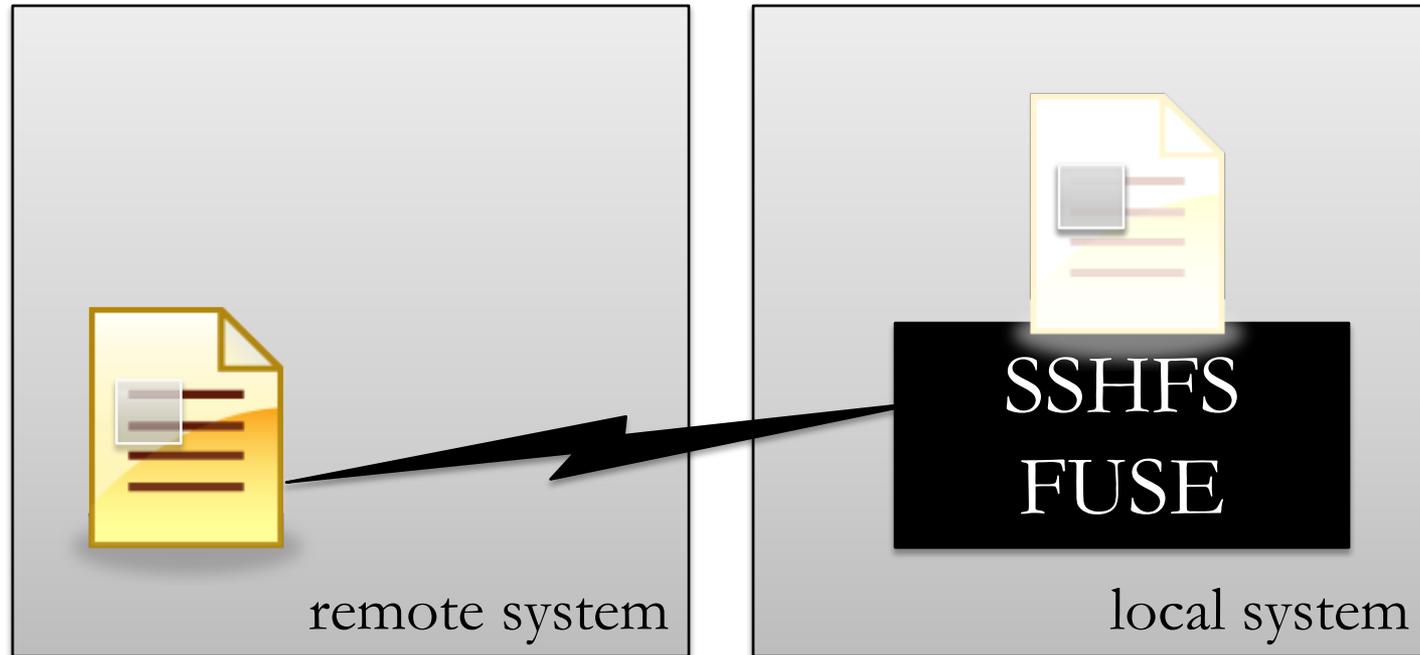


- However, NFS requires the system admin. to mount the remote file system.

1. Downloading a whole file



- What if only a small part is necessary from a huge file?



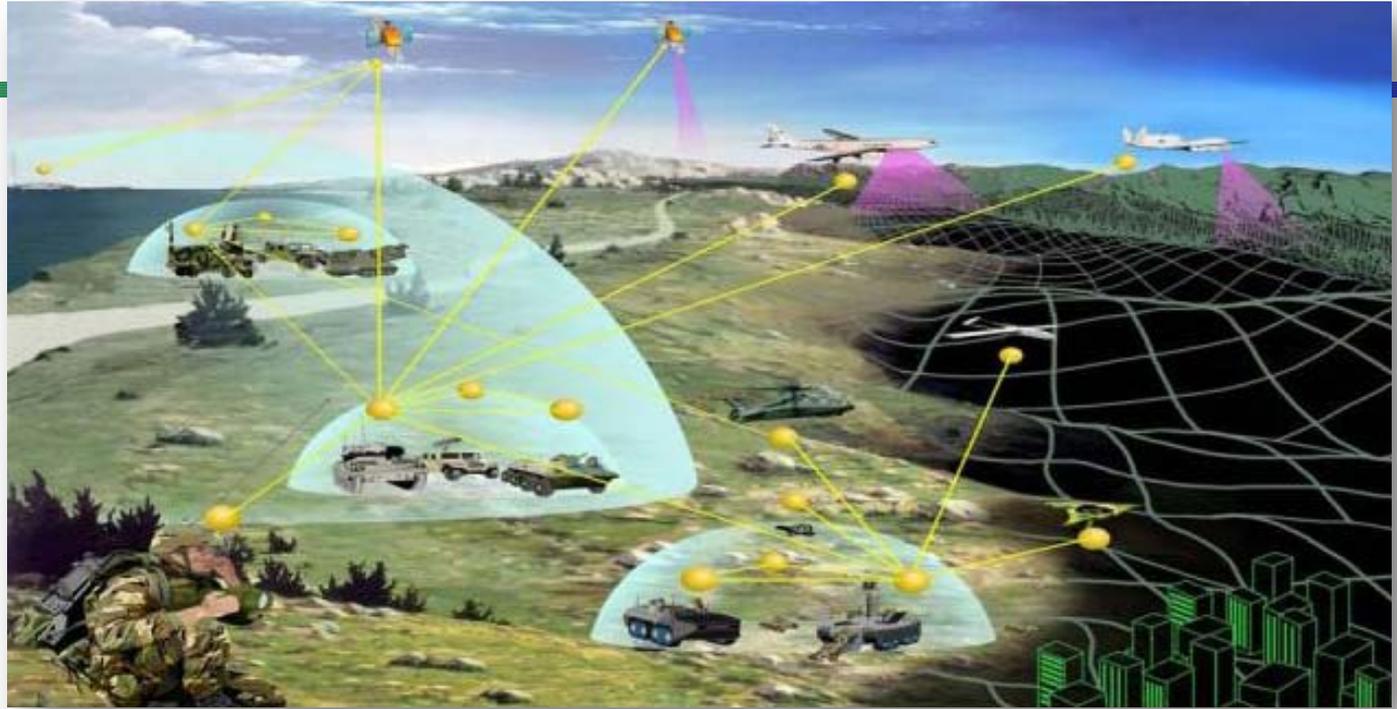
- If permission is granted to access FUSE, general users can mount remote filesystems.

- Elapsed time ratio
 - - SSHFS / downloading a whole file and subsetting

sshfs/download&local	File 1
Whole file	3.42
One dataset	0.23
One hyperslab	0.10

SSHFS
consumed
more time

SSHFS
consumed
less time



A Project with the Army Geospatial Center

TRANSFORMING THE GEOCOMPUTATIONAL BATTLESPACE FRAMEWORK WITH HDF5

Military Decision Making Process

Wide variety

Large scale

High efficiency

Satellite

Buckeye

Culture

High res.

Stream

Accuracy

Time





Concept Map : General HDFView

The screenshot shows the HDFView application window. The title bar reads "HDFView". The menu bar includes "File", "Window", "Tools", and "Help". The address bar shows the file path: "C:\Program Files\The HDF Group\hdfview 2.5\Data\concept_map_demo.h5".

The left pane displays a tree view of the concept map structure:

- concept_map_demo.h5
 - MDMP
 - IPB
 - Events
 - OCOKA
 - global
 - image
 - Situation
 - RECON
 - OCOKA
 - global
 - image
 - Situation

The right pane shows the "Properties" window for the selected "OCOKA" node. The "Attributes" tab is active, displaying "Number of attributes = 6" and an "Add" button. Below is a table of attributes:

Name	Value
MIME	application/x-hdf
URI	URBAN_ATO.h5#///Baltimore/OCOKA/LOS/Omni_ground-50ft
MIME 2	application/x-hdf
URI 2	URBAN_ATO.h5#///Baltimore/Features/LIDAR/bldg_footprint
MIME 3	application/x-hdf
URI 3	URBAN_ATO.h5#///Baltimore/Imagery/ikonos_3band-1m

File/URL C:\Program Files\The HDF Group\hdfview 2.5\Data\concept_map_demo.h5

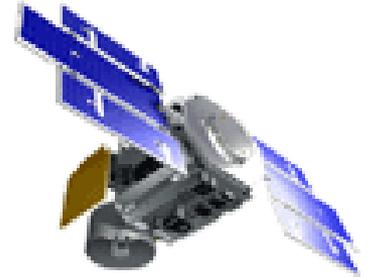
ImageView - ikonos_3band-1m - /Baltimore/Imagery/ - C:\Program Files\

```

graph TD
    MDMP[MDMP] --> IPB{IPB}
    MDMP --> RECON{RECON}
    IPB --> Events((Events))
    IPB --> OCOKA((OCOKA))
    RECON --> OCOKA
    RECON --> Situation((Situation))
    Events --> Bldg_LIDAR([Bldg_LIDAR])
    Events --> Omni-LOS([Omni-LOS])
    OCOKA --> Bldg_LIDAR
    OCOKA --> Omni-LOS
    OCOKA --> Img-IKONOS([Img-IKONOS])
    Situation --> Resd-UTP([Resd-UTP])
    Situation --> Img-IKONOS
  
```



HDF-EOS library



HDF - EOS Tools and Information Center



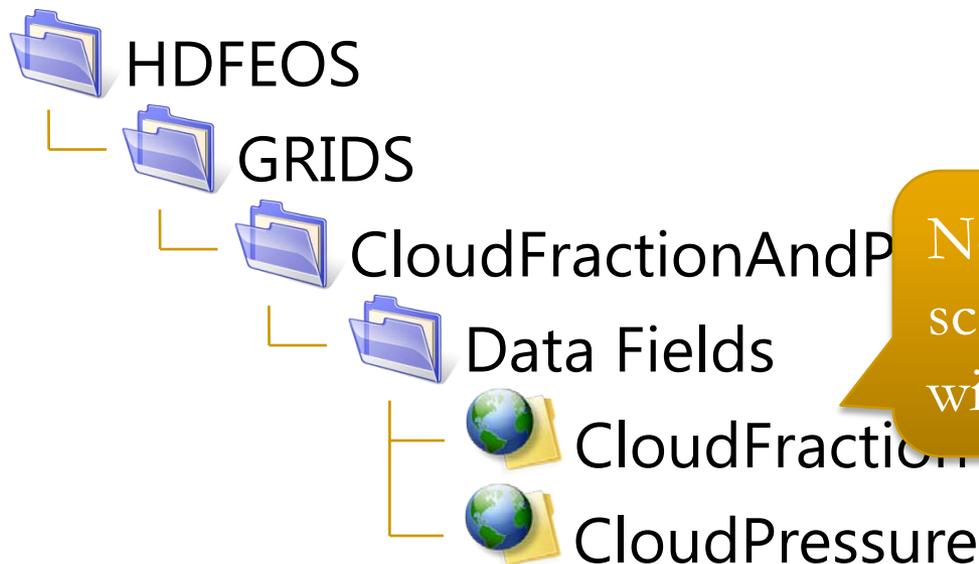
- HDF-EOS2 and HDF-EOS5
 - Automatic configuration with szip enabled/disabled
 - Now tested daily with HDF4 and HDF5 development code
- Updated the HDF-EOS website



HDF-EOS5/netCDF-4 Augmentation Tool

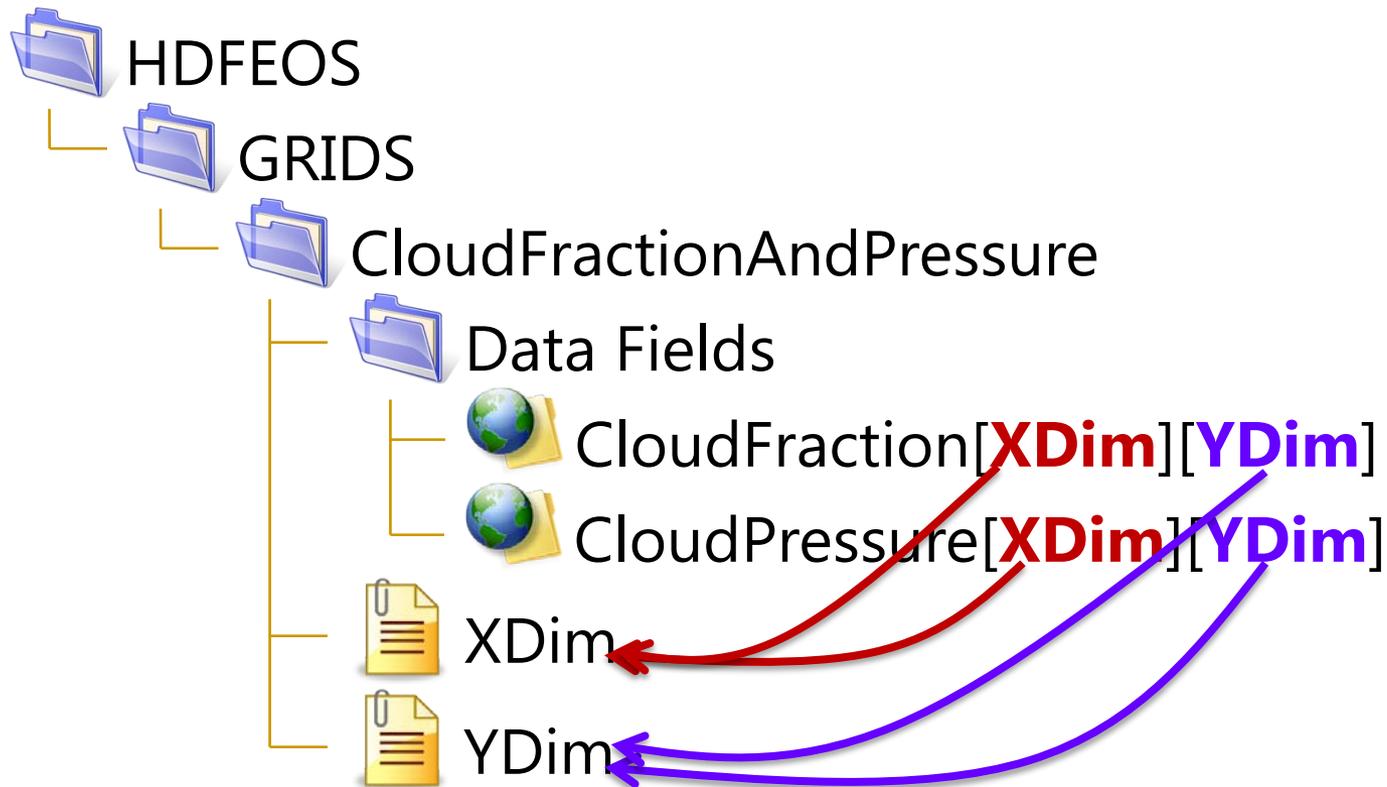
Accessing HDF-EOS5 files via netCDF-4 API

- NetCDF-4 model follows the HDF5 dimension scale model but HDF-EOS5 does not.



No HDF5 dimension scales are associated with this variable

- Provide dimensions required by netCDF-4





Special values in HDF5

- There are cases where a user may wish to specify more than one “special” value to describe non-standard data.
- We provide several examples (C, Fortran, IDL) on how to store special values:
 - <http://www.hdfgroup.org/pubs/rfc/>

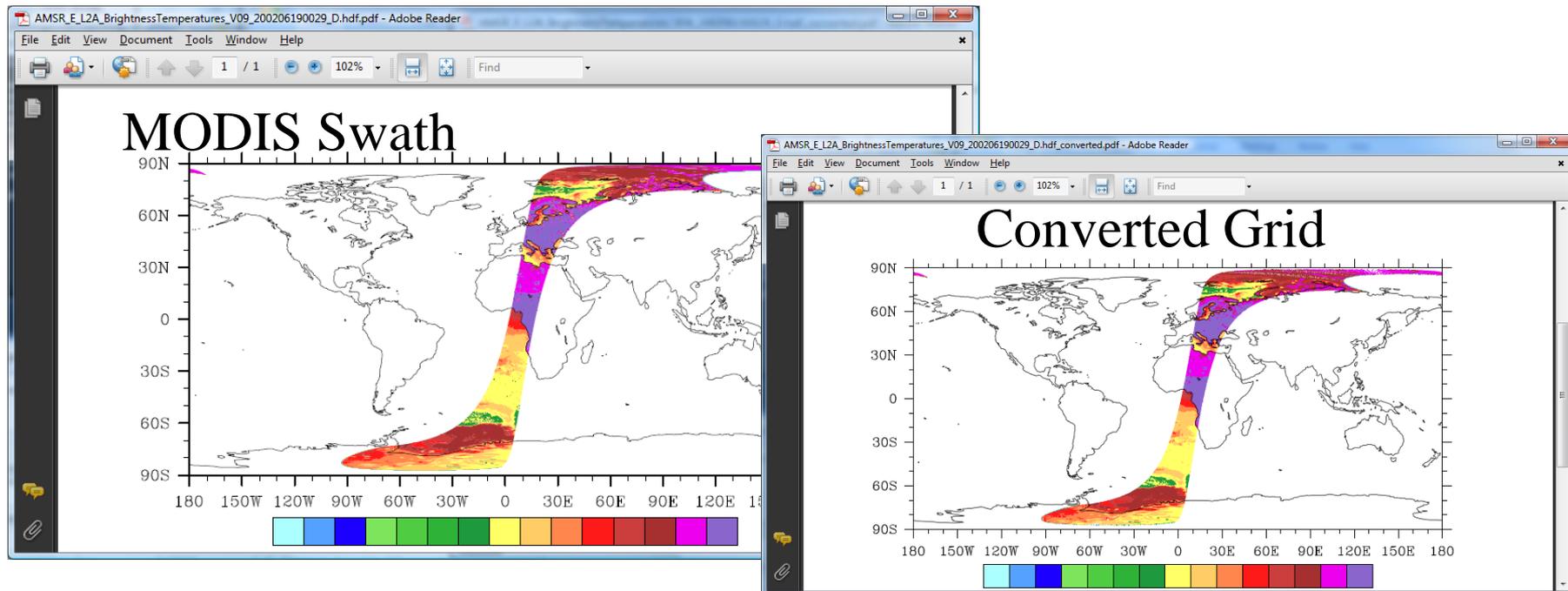


The logo for OPeNDAP, featuring the text 'OPeNDAP' in a blue, serif font. A red, curved line with a blue star at its end arches over the text.



- HDF5-OPeNDAP handler
 - Served OMI Swath data
- HDF4-OPeNDAP handler
 - Tested with some AIRS data and some MODIS data

- Request from NASA GES DISC
- Convert Swath to Grid
- Support both HDF-EOS2 and TRMM data
- Still in the development





Support for NPP/NPOESS by The HDF Group





Priorities for 2008-2009

- Data accessibility and usability
 - Developed library of high level APIs to support NPP/NPOESS data management
 - Modified h5dump to display region references
 - Modified HDFView to view object and region references and quality flags
- System maintenance
- User support



Data Pointed by Object References

File/URL G:\Projects\NPOESS\Data\SVI01-GIMFG_NPP_d2003125_t101038_e10116_b9_c2005829153351_dev.h5

SVI01-GIMFG_NPP_d2003125_t101

- All_Data
 - VIIRS-I1-SDR_All
 - VIIRS-IMG-FGEO_All
- Data_Products
 - VIIRS-I1-SDR
 - VIIRS-I1-SDR_Aggr
 - VIIRS-I1-SDR_Gran_0
 - VIIRS-IMG-FGEO
 - VIIRS-IMG-FGEO_Aggr
 - VIIRS-IMG-FGEO_Gran_0

TableView - VIIRS-I1-SDR_Aggr - /Data_Products/VIIRS-I1-SDR/ - G:\Pro

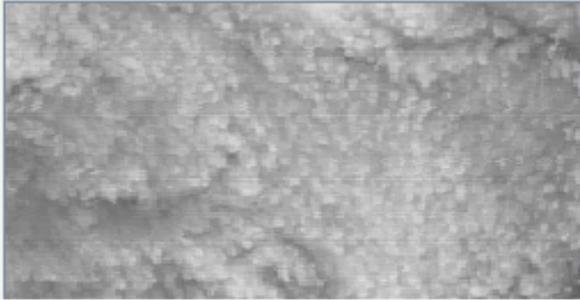
Table

0, 0 = /All_Data/VIIRS-I1-SDR_All/radiance_Array

	0
0	2928
1	3528
2	3800
3	4072
4	4344
5	4616
6	4888
7	5160
8	5432
9	6032
10	6304

radiance_Array - /All_Data/VIIRS-I1-SDR_

Image



radiance_Array - /All_Data/VIIRS-I1-SDR_All/ - G:\Projects\NPOESS\Dat

Table 0 1

26, 2364 = 884

	2356	2357	2358	2359	2360
0	889	898	889	889	880
1	871	861	861	871	871
2	830	821	811	821	821
3	898	898	898	898	898
4	834	834	834	825	825



Displaying quality flags in HDFView

- Can display data in decimal, binary, or hex.

The screenshot shows the HDFView interface with a file tree on the left and three data tables on the right. The file tree shows the path: SVI01-GIMFG_NPP_d2003125... > All_Data > VIIRS-I1-SDR_All > QF_VIIRS1SDR_Arra... (selected).

The three tables are:

- Table 1 (Decimal):** Shows data in decimal format. The columns are indexed 0-8 and rows 0-5.
- Table 2 (Binary):** Shows data in binary format (0s and 1s).
- Table 3 (Hexadecimal):** Shows data in hexadecimal format.

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	40	87	43	147	47	109	47	230	47
2	0	0	0	0	0	0	0	0	0
3	38	50	41	184	48	48	50	150	50
4	0	0	0	0	0	0	0	0	0
5	35	32	39	59	41	236	47	179	47

	0	1	2	3	4	5	6	7	8
0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
1	00101000	01010111	00101011	10010011	00101111	01101101	00101111	11100110	00101000
2	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
3	00100110	00110010	00101001	10111000	00110000	00110000	00110010	10010110	00110010
4	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
5	00100011	00100000	00100111	00111011	00101001	11101100	00101111	10110011	00101000

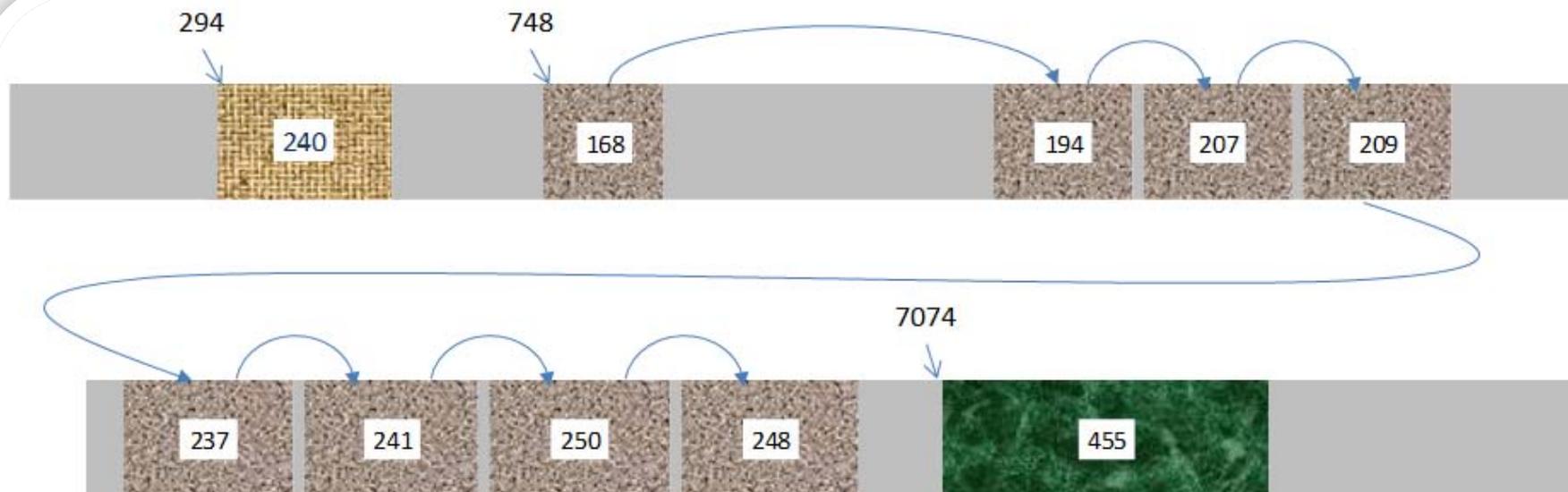
	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	28	57	2b	93	2f	6d	2f	e6	2f
2	0	0	0	0	0	0	0	0	0
3	26	32	29	b8	30	30	32	96	32
4	0	0	0	0	0	0	0	0	0
5	23	20	27	3b	29	ec	2f	b3	2f



NPOESS Project Information

- Project Web site
 - <http://www.hdfgroup.org/projects/npoess/>

HDF4 LAYOUT MAPS

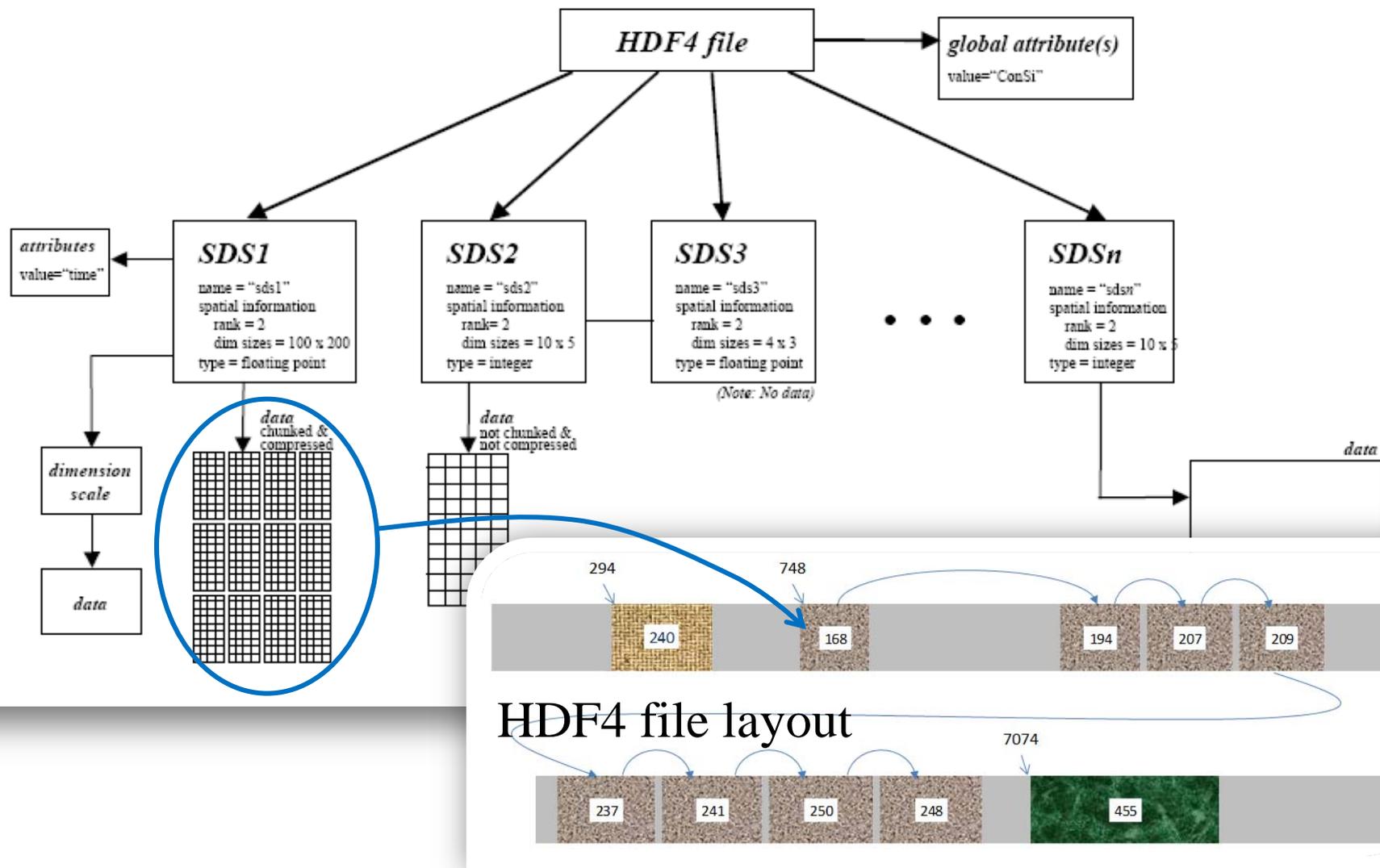




HDF4 Layout Map Project

- Problem
 - Long-term readability of HDF data depends on long-term availability of software
- Proposed solution
 - Create a map of the layout of data objects in an HDF file, allowing a simple reader to be written to access the data

Mapping a chunked SDS





Thank You!

Questions & Comments?