HDF5: State of the Union

Quincey Koziol, Ruth Aydt
{koziol,aydt}@hdfgroup.org
The HDF Group

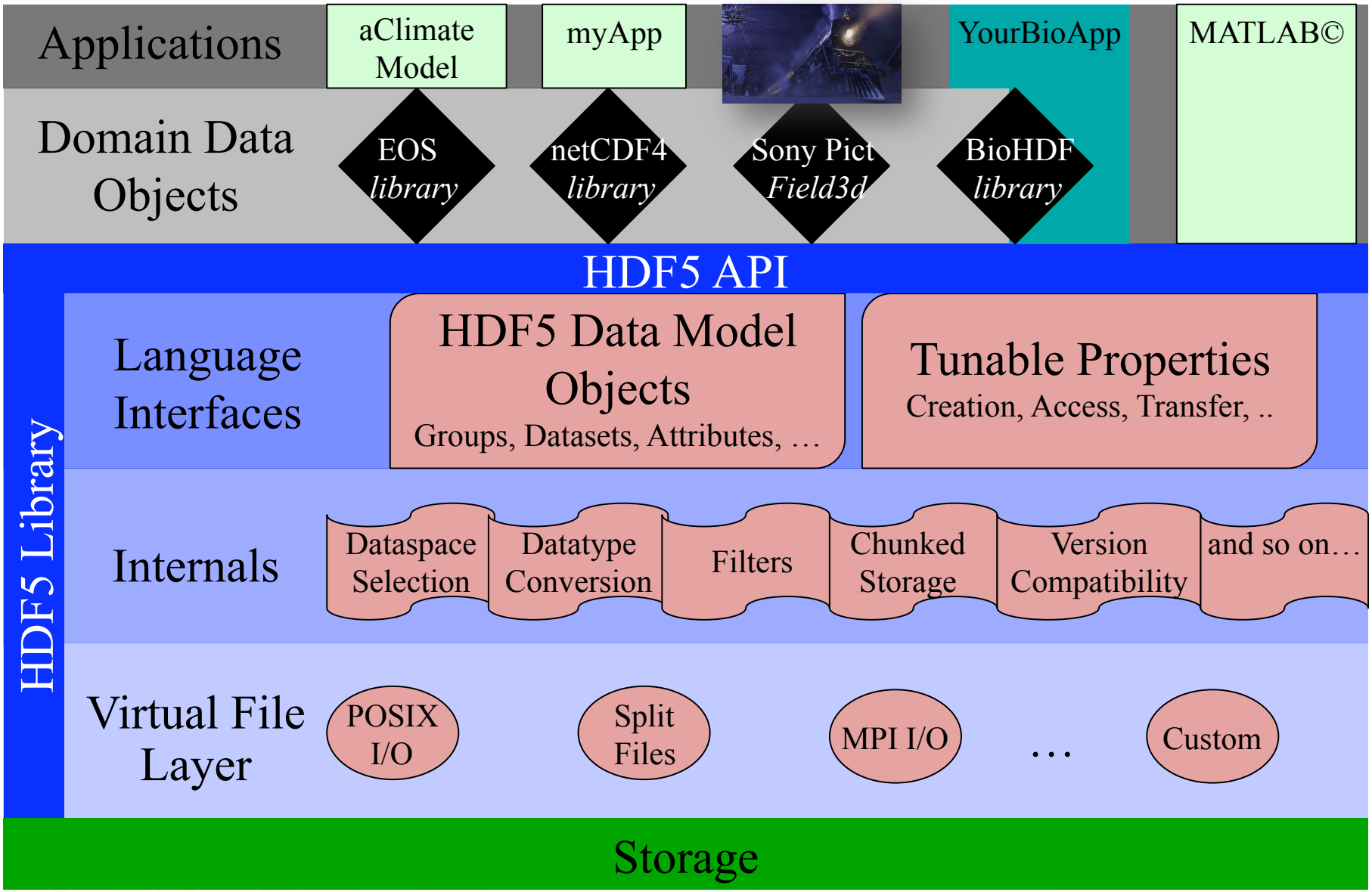
SC09 – HDF5 BOF

November 19, 2009

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 HDF

- Our 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
- Improvements
  - Concurrent access
  - Improving parallel I/O performance
  - Improving real-time write performance
  - Improving high level language support

practice

DOI:10.1145/1562764.1562781

Article development led by acmqueue
queue.acm.org

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.

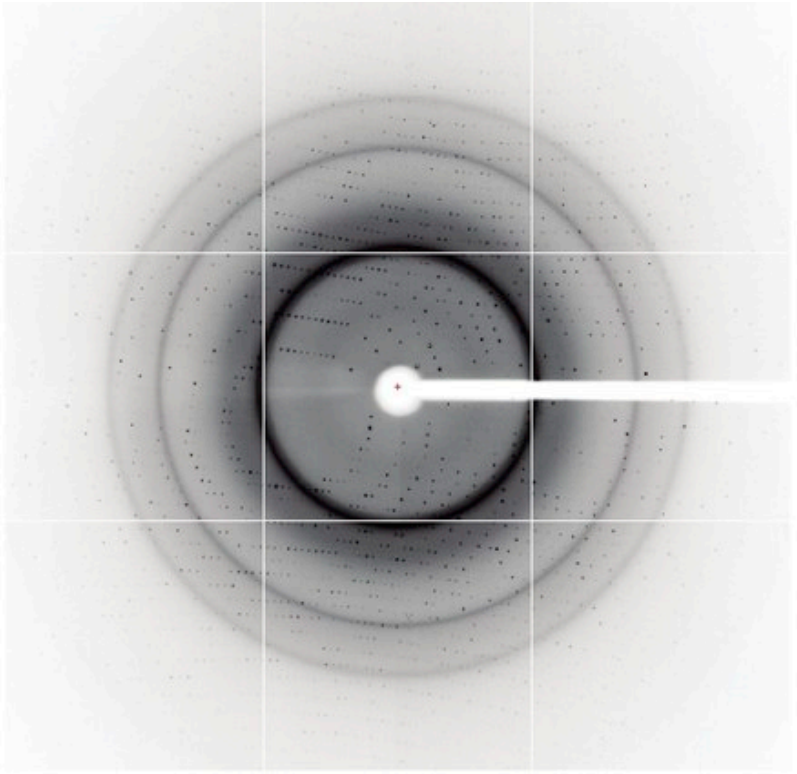
Several factors are forcing the obsolescence: rapid increases in the number of pixels per image; acceleration in the rate at which images are produced; changes in image designs to cope with new scientific instrumentation and concepts; collaborative requirements for interoperability of images collected 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 major 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.



An x-ray diffraction image taken by Michael Soltis of LSAC on SSRL BL9-2 using an ADSC Q315 detector (SN901).

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,[5] may be so interrelated that they represent different 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

COMMUNICATIONS OF THE ACM – October 2009

Cool recent application

-

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

The HDF Group

- Established in 1988
  - 18 years at University of Illinois National Center for Supercomputing Applications
  - 4 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 25 FTEs

The HDF Group Mission

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

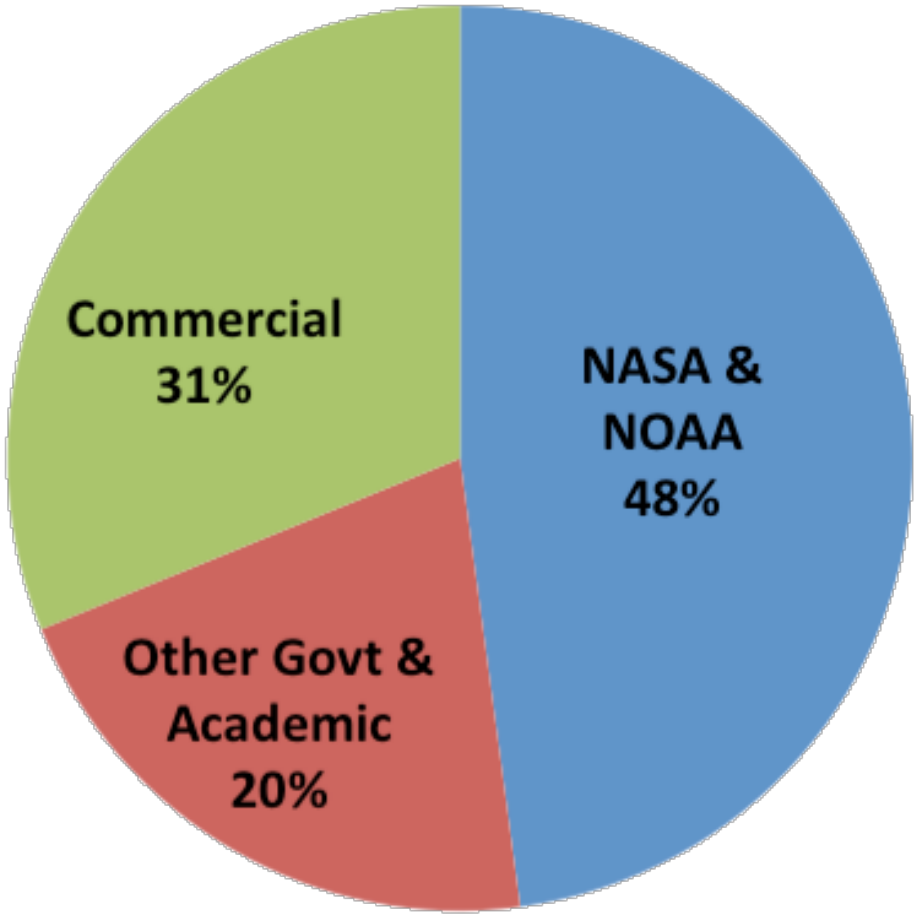
Goals of The HDF Group

- 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

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

HDF5 1.8.3 minor release (May 09)

- New functions
  - Improve flexibility when traversing external links
  - Validate object identifier
- Enabled data chunk cache properties to be set per dataset (per file in previous releases)
- Forward/backward compatibility issues
  - Modified library to be able to open files with corrupt root group symbol table messages
  - Also corrects corruption errors if found.

HDF5 1.8.4 minor release (Nov 09)

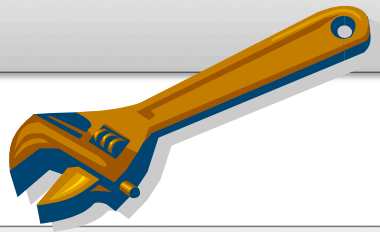
- Modified configure and make process to properly preserve user's CFLAGS and similar environment variables.
- Corrected a problem where library would re-write the superblock in a file opened for R/W access, even when no changes were made to the file.

HDF5 1.6 minor releases

- 1.6.9 May 09
  - Minor bug fixes
  - Same tools improvements as in 1.8.3
- 1.6.10 Nov 09
  - Minor bug fixes
  - Ability to embed library information in executable binaries
  - **This is a last release of 1.6 series**
    - announced in May 2009 – *no response*
    - ***This is your last chance!***

Topics

- 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
  - Ability to show data pointed to by dataset region references.
  - More options for dumping data into ASCII
    - Compatible with MS Excel
    - Compatible with h5import
- h5diff
  - Some new flags
    - Report non-comparable objects
    - Avoid NaN detection
    - Option to use system epsilon to compare floating-point numbers
  - Treats two INFINITY values as equal
  - Compares for strict equality first to improve performance
  - Fixed segmentation fault problem on variable length strings.

Major Improvements for Existing Tools

- h5stat
  - Fixed incorrect statistics on EOS big data files with corrupted headers.
- h5repack
  - Added ability to preserve group creation order
  - Fixed problem that 1.8 fails on a file created with 1.6.
  - When chunk size not specified, uses heuristics to set chunk size

Tool activities in the works

- New tool -- h5tail
  - 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

Conversion Tools

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 is on the way

- Includes all HDF java products
  - Java Wrapper API
  - Java Object API
  - HDFView
- Adds new features, such as dataset region reference
- Improves performance
- Release schedule
  - Beta 1: end of Nov. 09
  - Full release: end of Dec. 09

Full support of HDF5 1.8.x in hdf-java

- Full HDF5 1.8 support will be added to the release after version 2.6.
- We are looking for input
  - RFC: http://www.hdfgroup.uiuc.edu/RFC/HDF5/hdf-java/
- Java wrapper will be completed March 2010
- Object API and HDFView update to come later

Topics

What's up with The HDF Group?

Library Update

Tools update

HDF Java Products

Library development in the works

Other activities

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

Surviving a System Failure in HDF5

- 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 with Metadata, 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 released August 2009
- Adding support for asynchronous I/O of journal writes – Faster!
- Further development may support parallel I/O
- To be included in 1.10.0 release

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 “OpenPA” project - http://trac.mcs.anl.gov/projects/openpa

Other Library Features

- Saving space
  - New Chunk Indexing Methods
  - Store Partial Edge Chunks More Efficiently
  - Persistent File Free Space tracking/recovery
  - Allow a group's link info to be compressed
- Saving time
  - Aggregate neighboring metadata for faster metadata cache I/O

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)	Constant 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 *and appends*
2+ unlimited dimension	Improved B-tree*	Smaller index overhead	Faster

Parallel I/O Improvements

- Project with NERSC to improve HDF5 performance on parallel applications
- Up to 6x performance improvements on certain applications (so far)

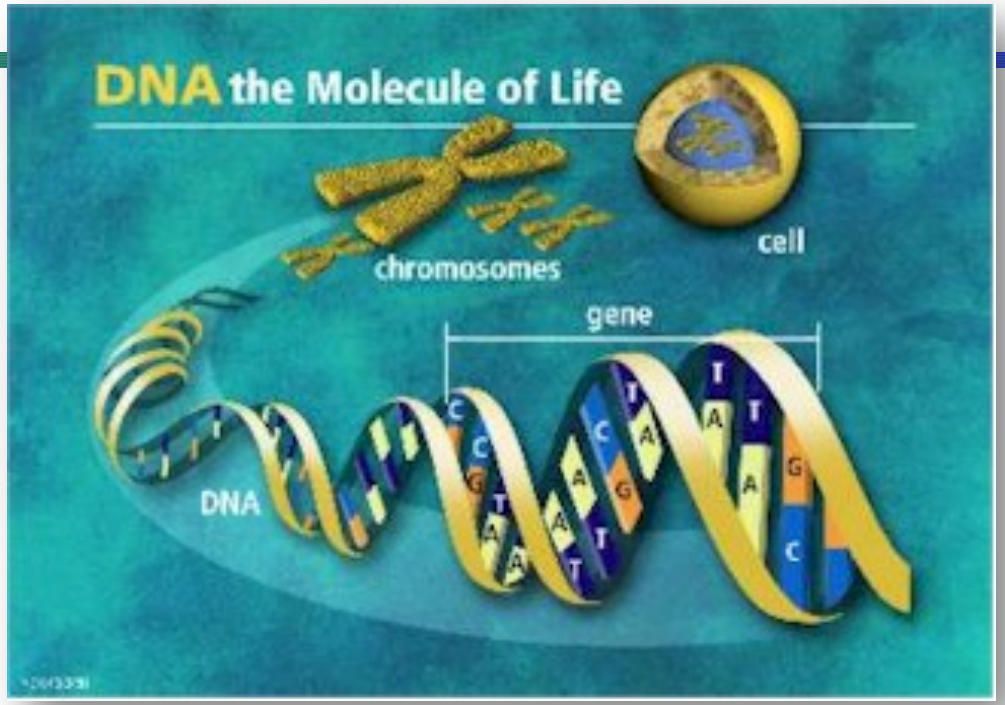
Parallel I/O Improvements

- Current work:
  - Restructured library code to number of file truncation operations
  - Restructured library code to detect same “shape” of selection in more cases, allowing optimized I/O path to be taken more often
- Future work:
  - Add high-level “tune for Lustre file system” API call(s)
  - Improvements to MPI-IO and MPI-POSIX VFDs and library algorithms for faster/better use of MPI

Topics

- What's up with The HDF Group?
- Library Update
- Tools update
- HDF Java Products
- Library development in the works

Other activities



NIH STTR with Geospiza, Seattle WA

BIOHDF : TOWARD SCALABLE BIOINFORMATICS INFRASTRUCTURES

Next Generation DNA Sequencing

"Transforms today's biology"

"Democratizing genomics"

NGS is Powerful

"Changing the landscape"

"Genome center in a mail room"

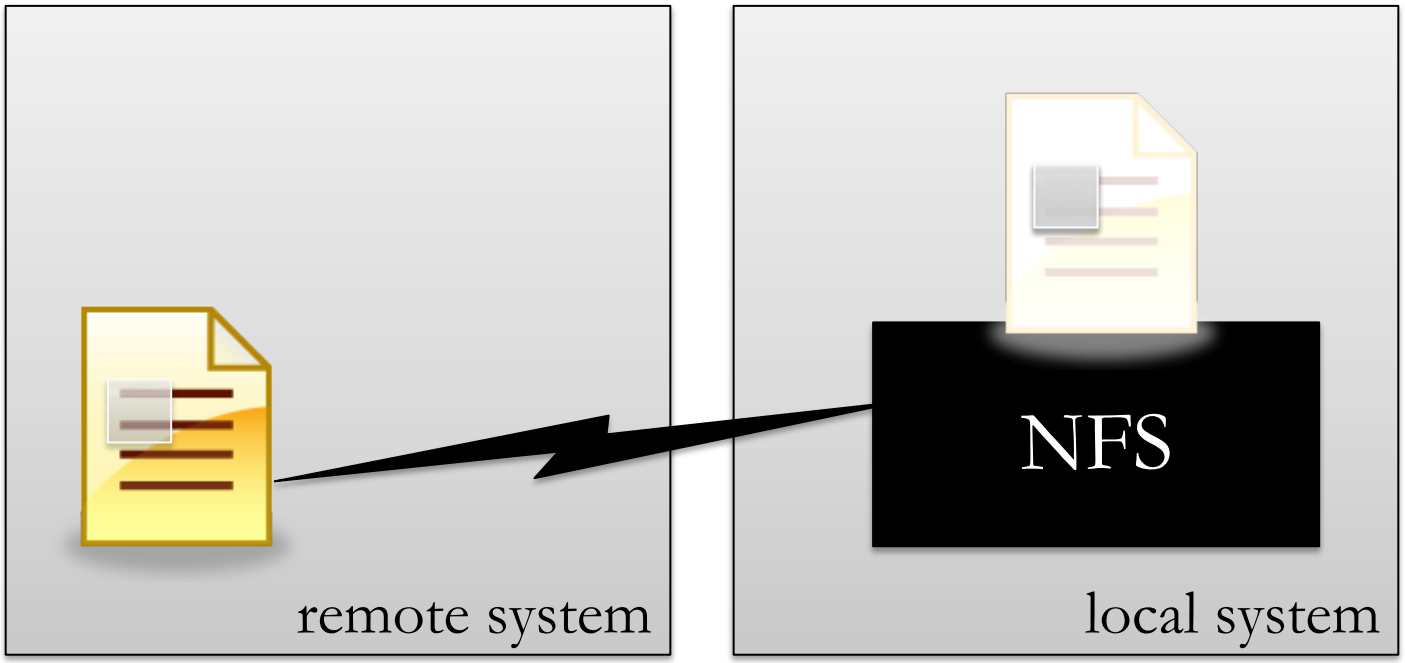
"The beginning of the end for microarrays"

BioHDF Project

- ***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**

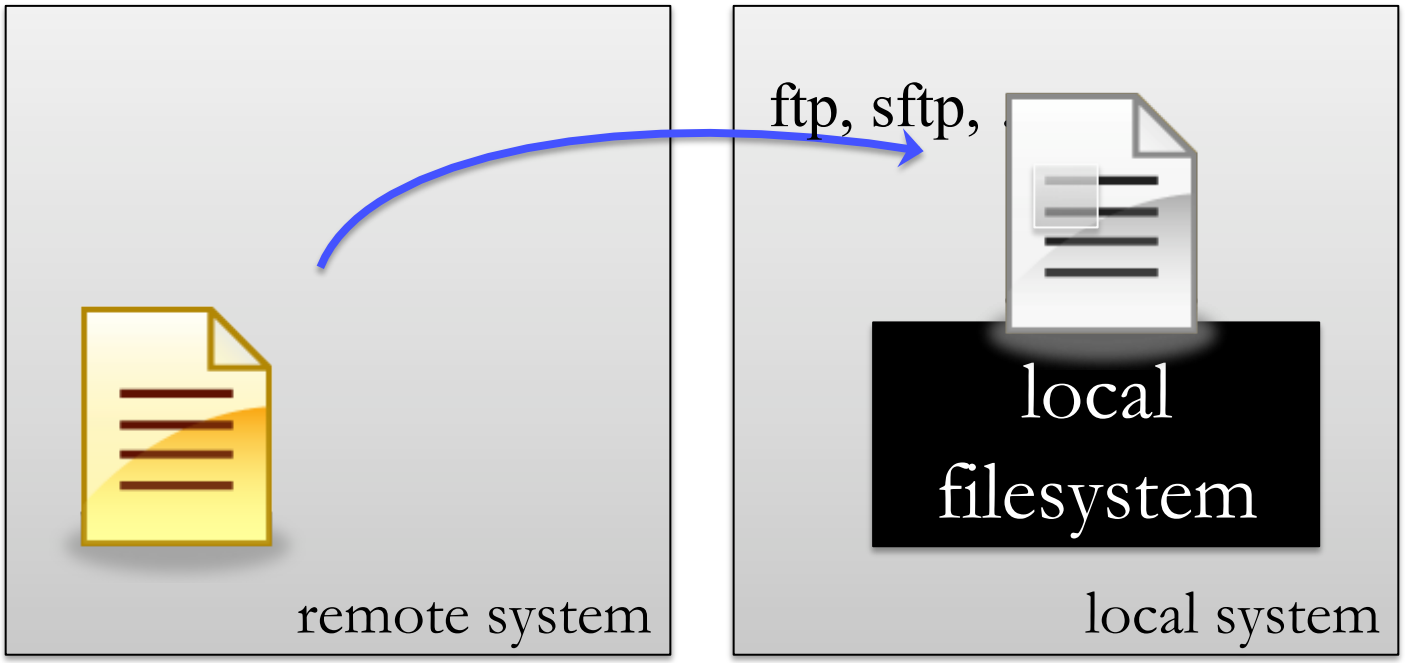
Performance evaluation of using SSHFS-FUSE to access HDF5 files

Using NFS



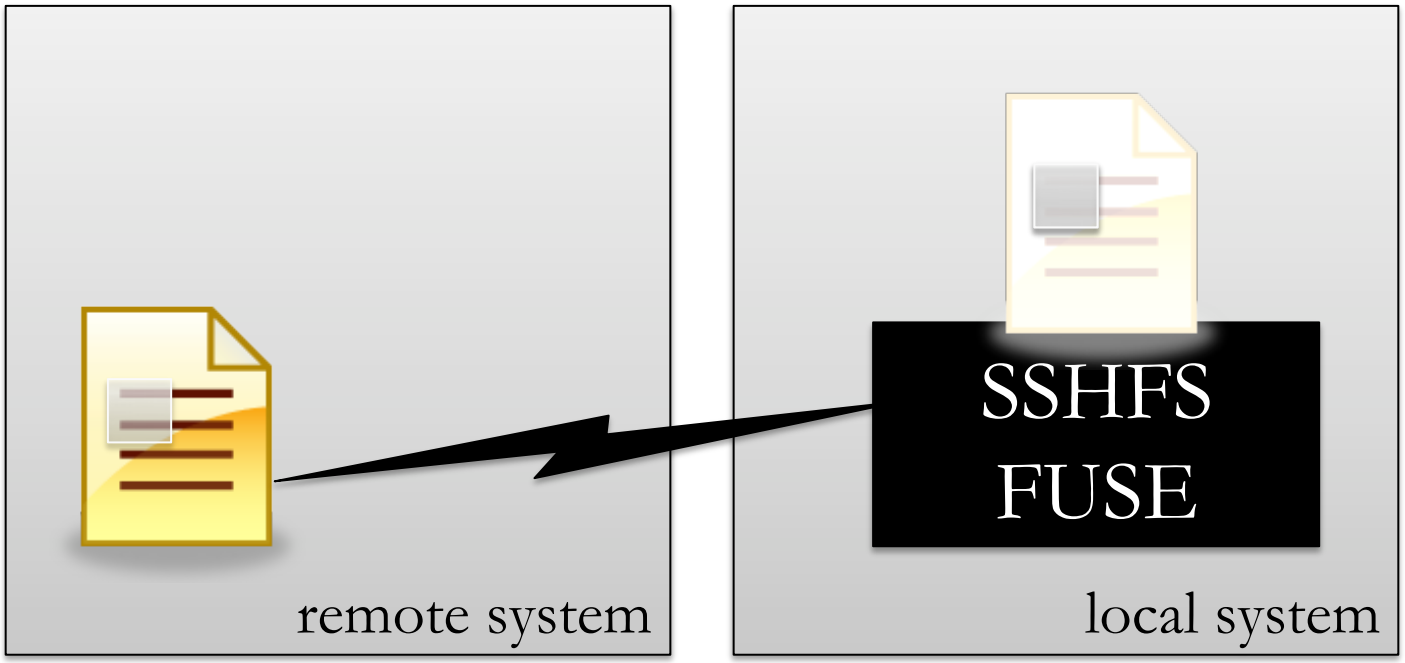
- However, NFS requires the system admin. to mount the remote file system.

1. Downloading a whole file



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

2. Using FUSE+SSHFS



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

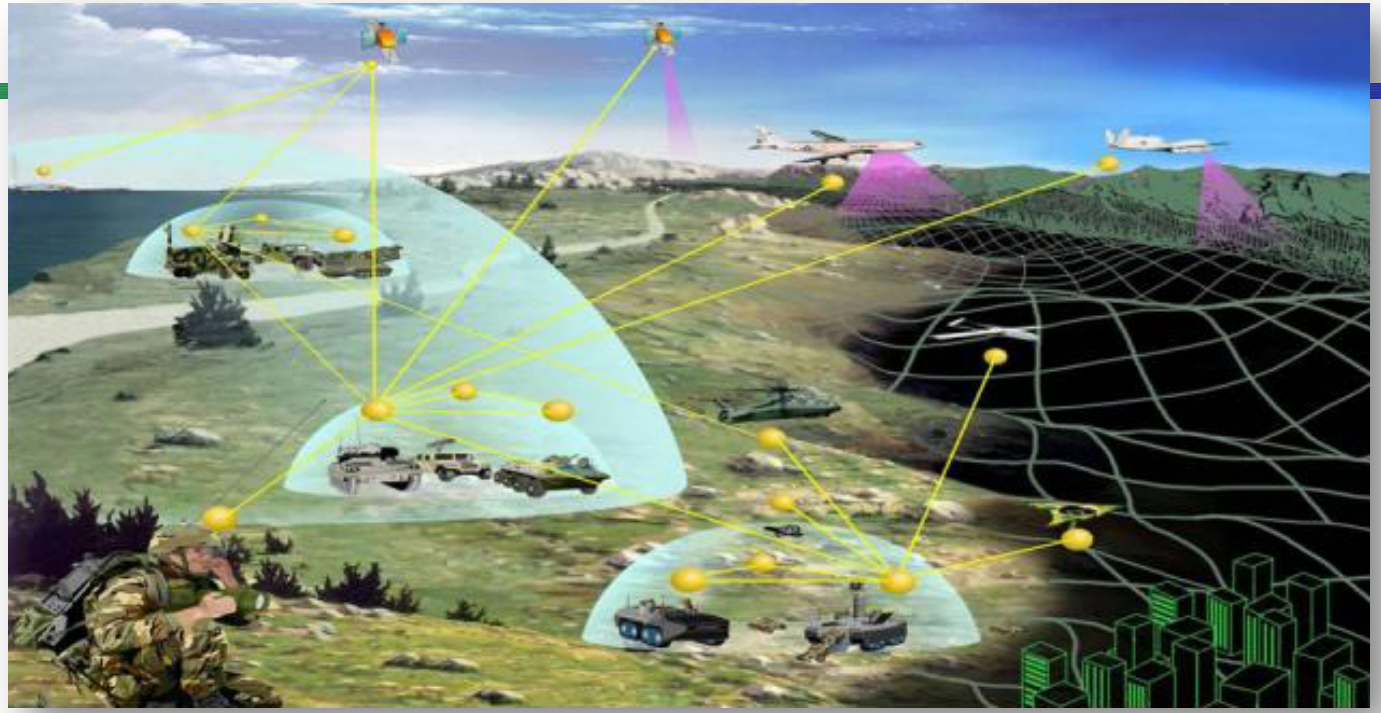
Results of experiment

- 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

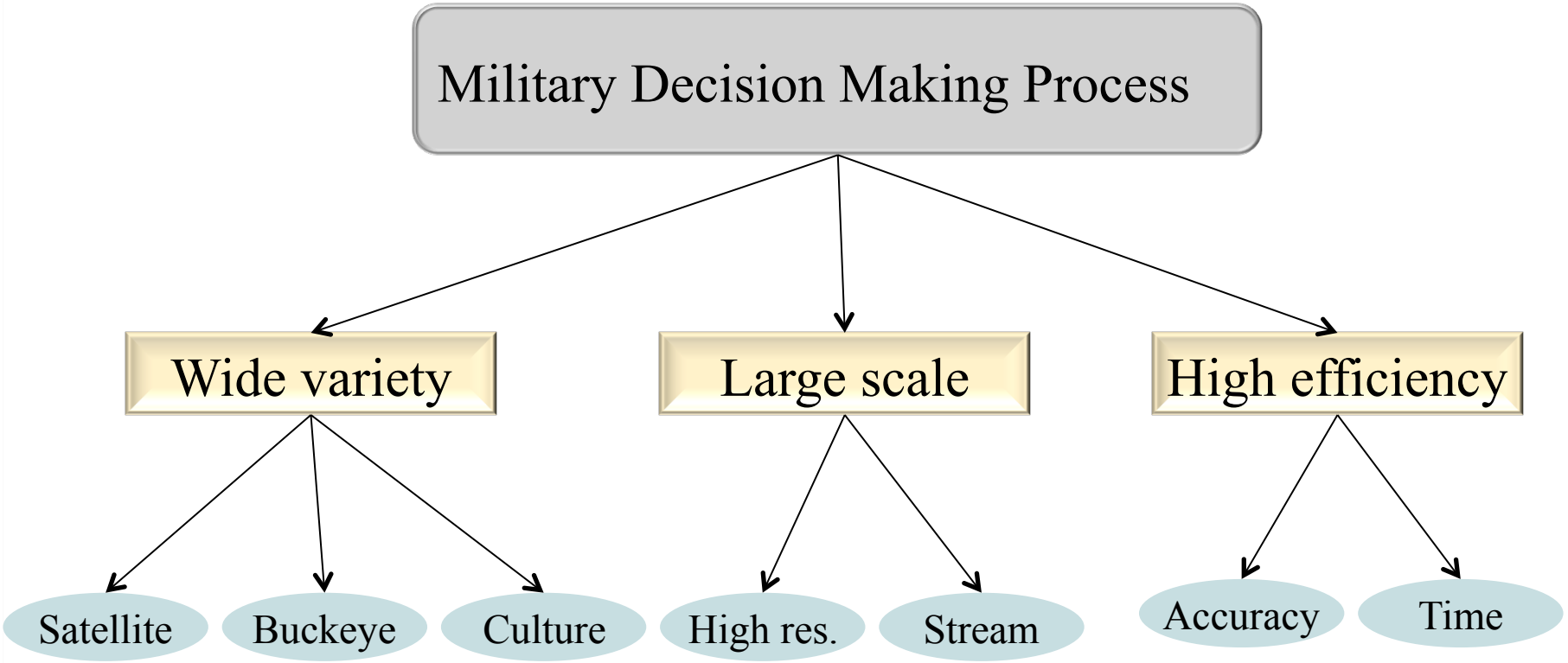
SSHFS consumed less time



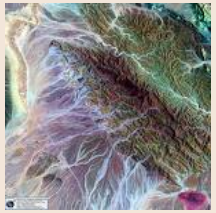
A Project with the Army Geospatial Center

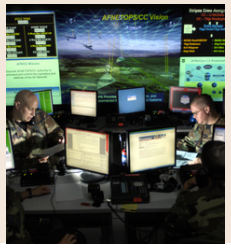
TRANSFORMING THE GEOCOMPUTATIONAL BATTLESPACE FRAMEWORK WITH HDF5

Data Challenges

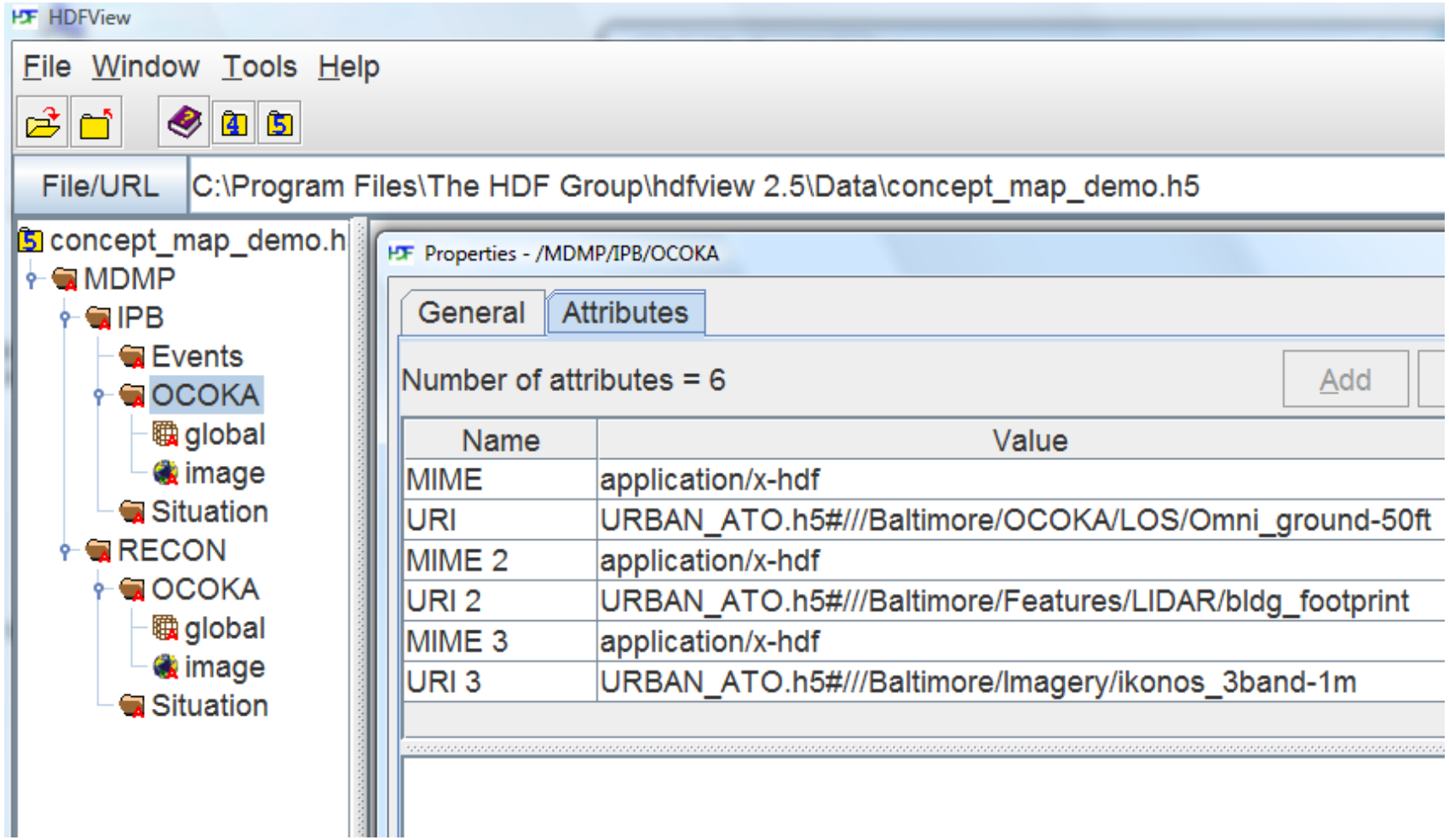




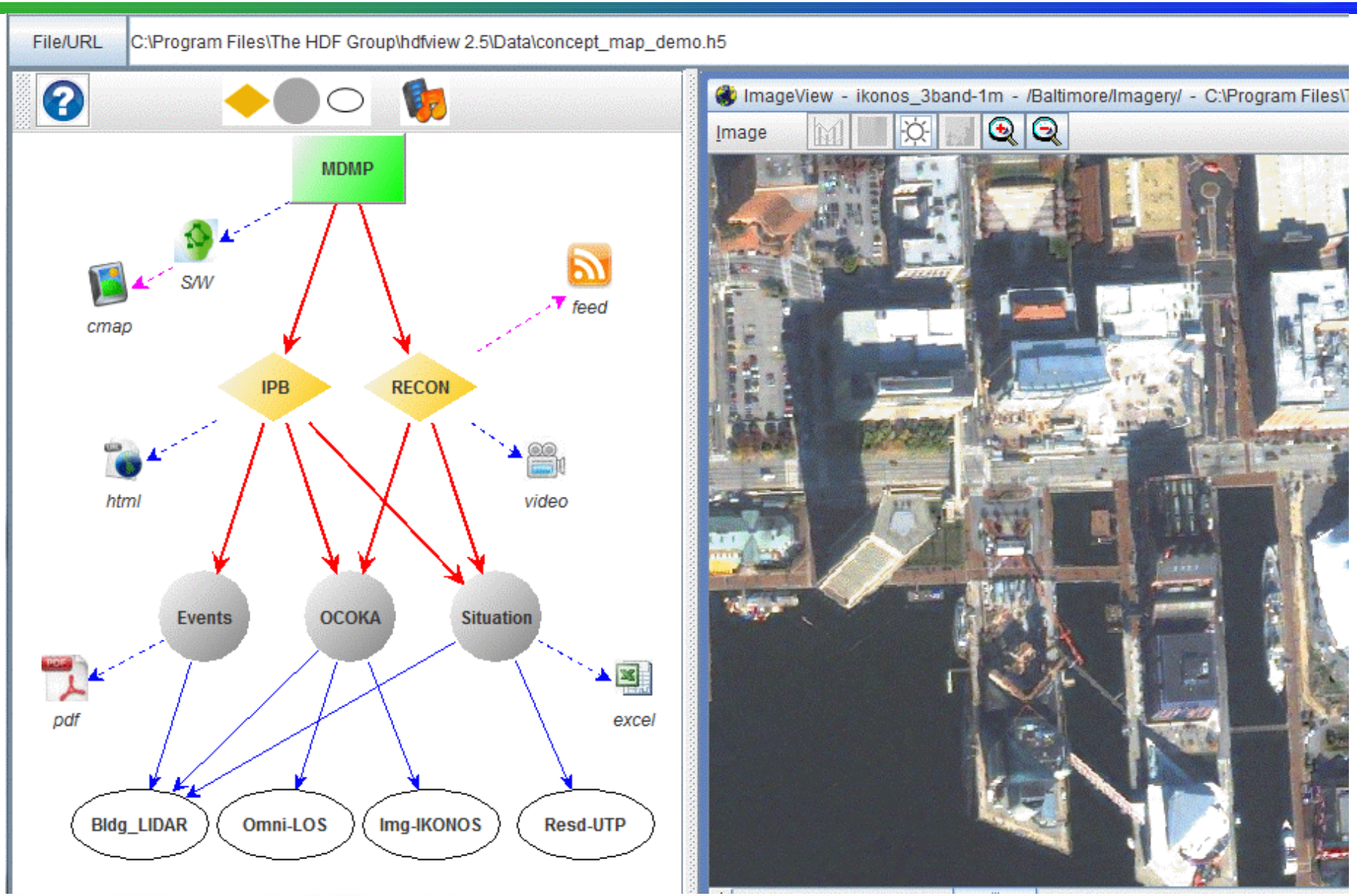





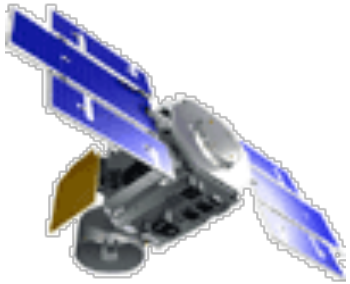

Concept Map : General HDFView



Plug-in: Concept Map



HDF-EOS library



HDF - EOS Tools and Information Center

EOS support

- 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

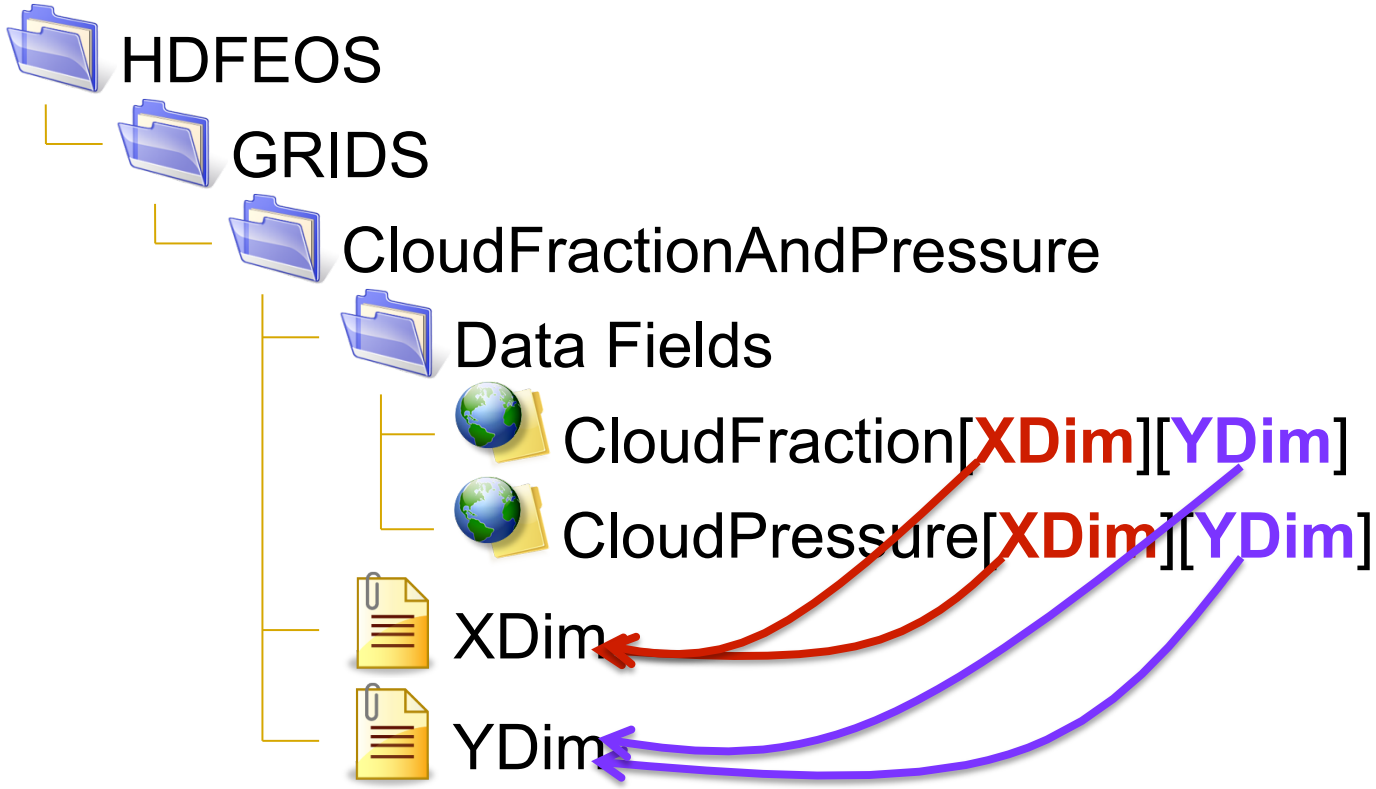
Accessing HDF-EOS5 files via netCDF-4 API

The Main Challenge

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

Our Solution – Augmentation

- 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/rfcs/

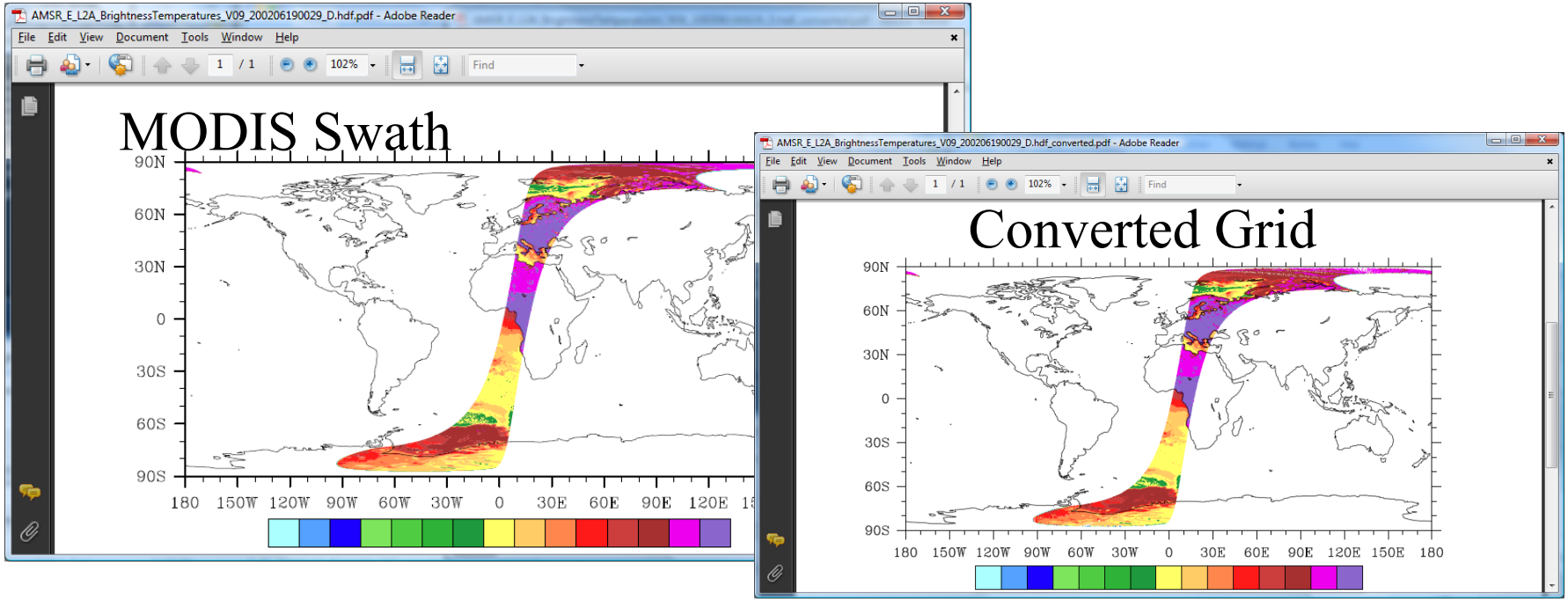

OPeNDAP

OPeNDAP

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

Swath to Grid conversion Tool

- 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

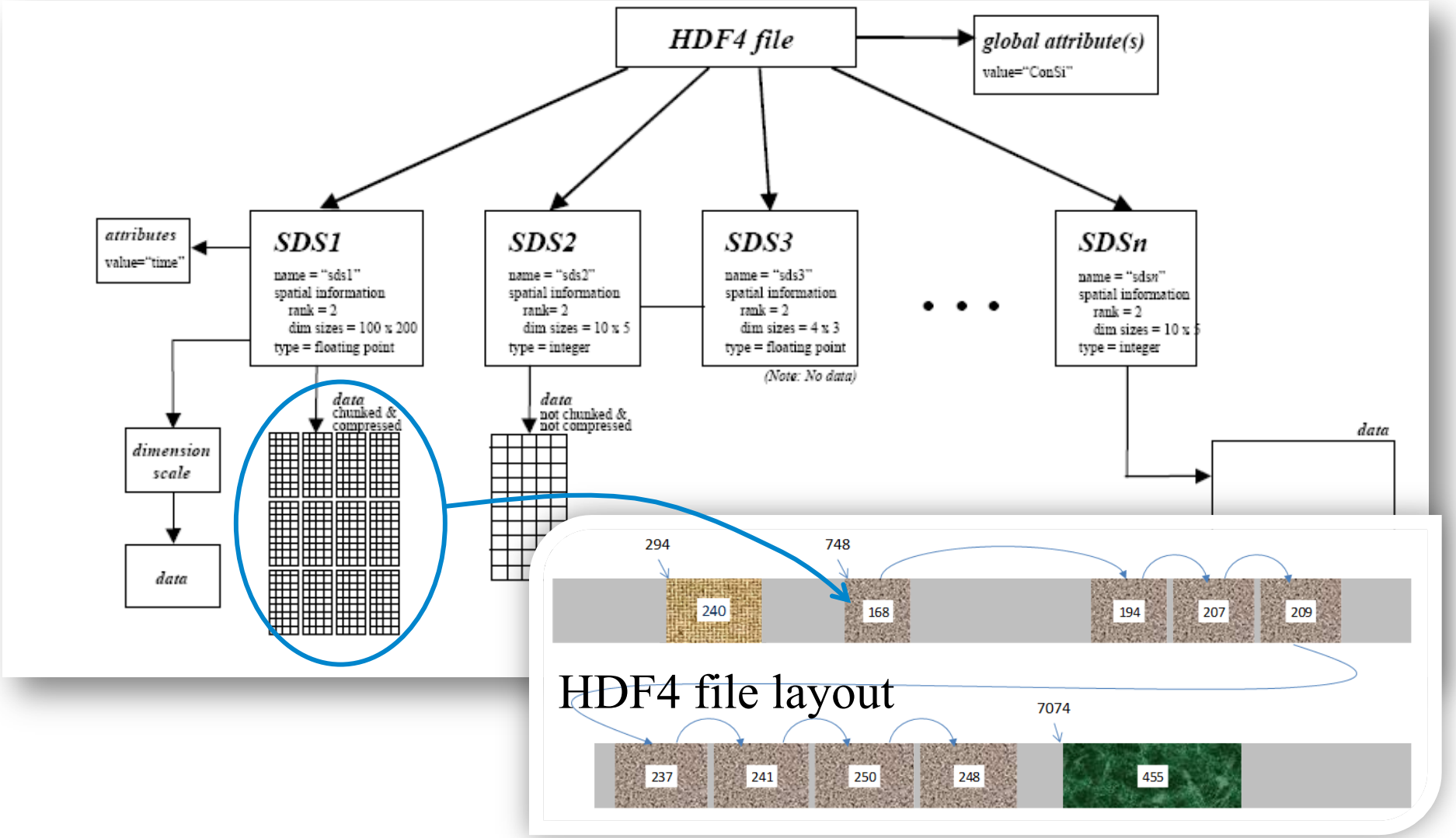
- Data accessibility and usability
  - Developed library of high level APIs to support NPP/NPOESS data management
  - Modified h5dump to display region references
  - Modified HDFView to view object and region references and quality flags
- System maintenance
- User support

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?