The Future of H5Coro

HDF Users Group (HUG) 2021

JP Swinski/NASA/GSFC

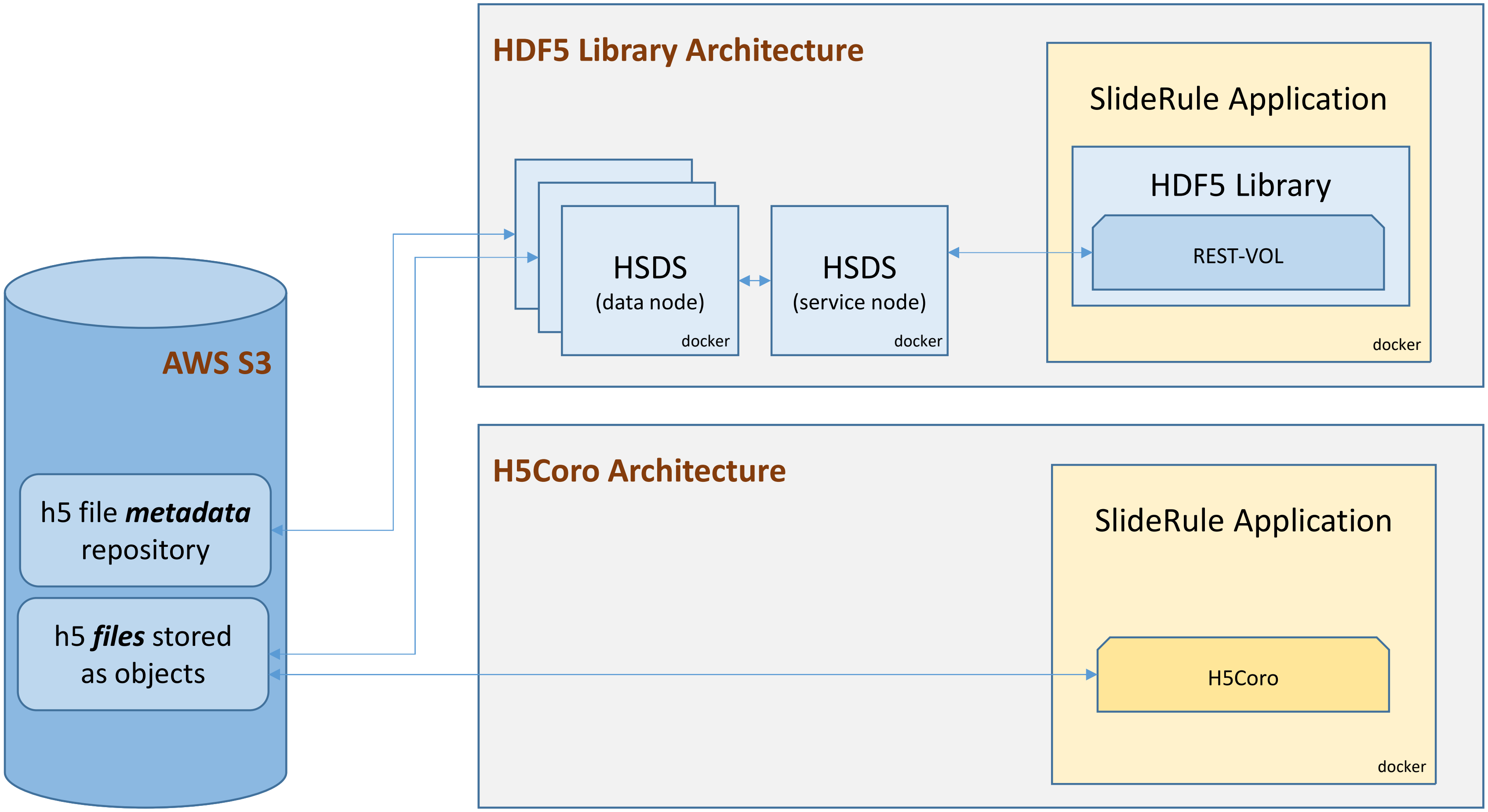
October 12, 2021

What is H5Coro

H5Coro is an independent implementation in **C++** of a subset of the HDF5 standard that is optimized for reading **static data from cloud-based storage systems**.

The H5Coro software is under active development by the University of Washington and NASA/Goddard Space Flight Center as a part of the SlideRule program for on-demand processing of **ICESat-2** data.

Where H5Coro Fits In



Key Features

- **All reads are concurrent**. Multiple threads within the same application can issue read requests through H5Coro and those reads will get executed in parallel.

- **Intelligent range gets** are used to read as many dataset chunks as possible in each read operation. This drastically reduces the number of HTTP requests to S3 and means there is no longer a need to re-chunk the data (it actually works better on smaller chunk sizes due to the granularity of the request).

- **The system is serverless.** H5Coro is linked into the running application and scales naturally as the application scales. This reduces overall system complexity.

- **No metadata repository is needed.** Instead of caching the contents of the datasets which are large and may or may not be read again, the library focuses on caching the structure of the file so that successive reads to other datasets in the same file will not have to re-read and re-build the directory structure of the file.

Performance Comparison

Library	File Storage	File Structure	Cached	Instance	Time (secs)
HDF5/REST-VOL	S3	Original	Yes	c5.2xlarge	9559 (~2 ½ hrs)
HDF5/REST-VOL	S3	Original	No	c5.2xlarge	9029
HDF5/REST-VOL	S3	Repacked	No	c5.2xlarge	3215
HDF5/REST-VOL	S3	Repacked	Yes	c5.2xlarge	3157
H5Coro	S3	Repacked	No	c5.xlarge	368
H5Coro	S3	Repacked	Yes	c5.xlarge	336
HDF5	Ext4	Original	No	desktop	154
H5Coro	S3	Original	No	c5.xlarge	116 (~2 mins)
H5Coro	S3	Original	Yes	c5.xlarge	72
H5Coro	Ext4/Buffered	Original	No	desktop	56

Current Status

- Actively deployed in ICESat-2’s SlideRule system.

- Added non-blocking read API that returns a *future*; supports greater parallelization.

- Python bindings continue to be improved with help from group of seismic researchers at UW.

Future Work

- API to return metadata associated with dataset (dimensions, data type)
- API to return directory listing of h5 file’s groups and datasets
- Support for reading *attribute* messages

What should the future hold?

Our goal is for H5Coro to become
a niche implementation optimized for reading
static data out of S3

What should the future hold?

Our goal is for H5Coro to become

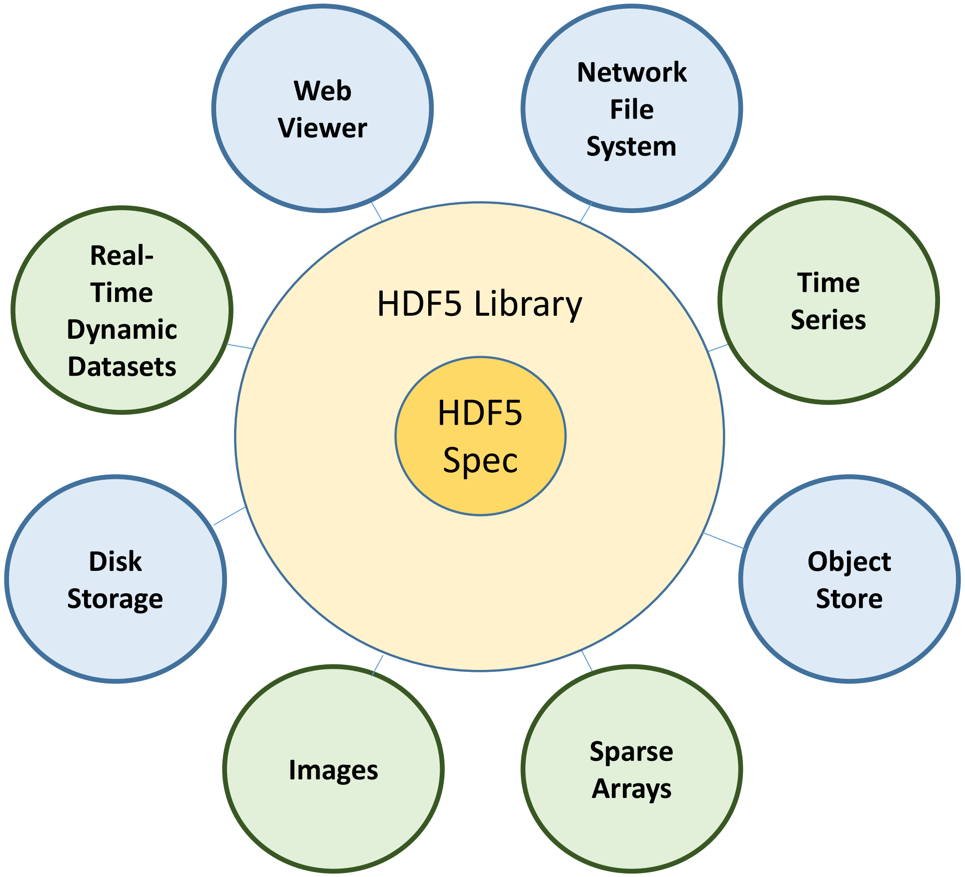
one of many

~~a~~ niche implementations optimized for reading

~~static data out of S3~~

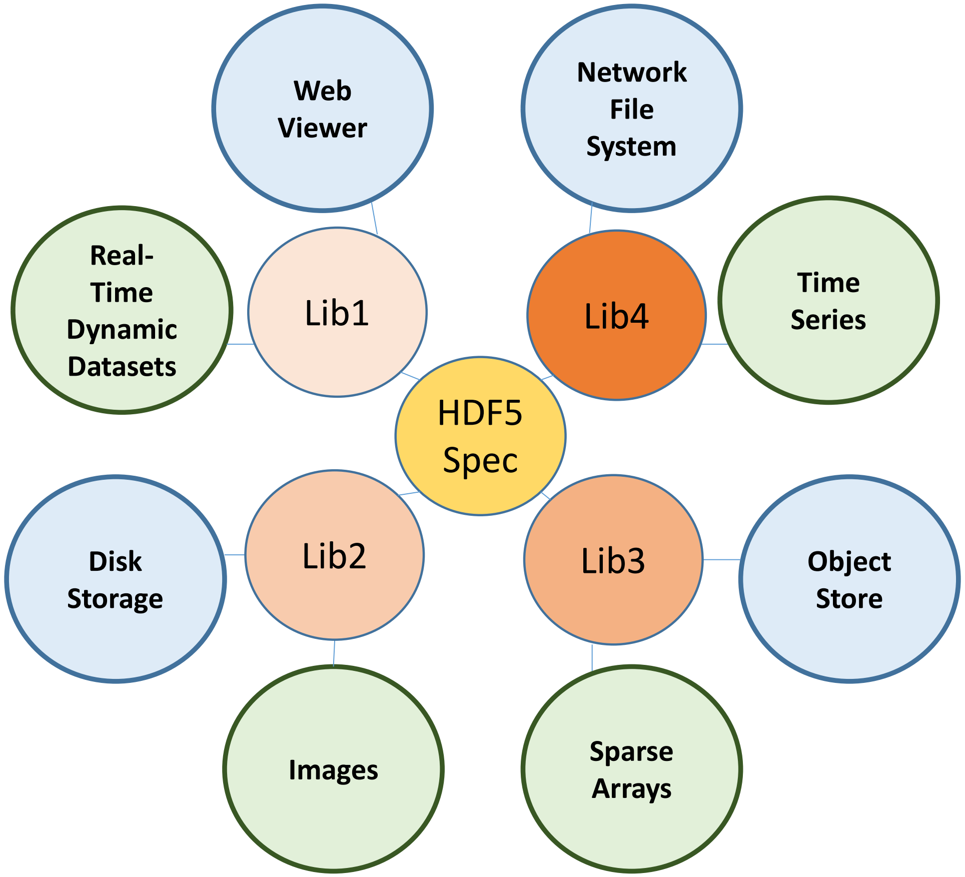
& writing application specific data

How to Handle Different Applications



Consolidated Library

How to Handle Different Applications



Special Purpose Libraries

Which is better?

Consolidated Library

?

Special Purpose Libraries

Which is better?

Consolidated Library

?

Special Purpose Libraries

We don’t need to choose… we can do both.

Which is better?

Consolidated Library

?

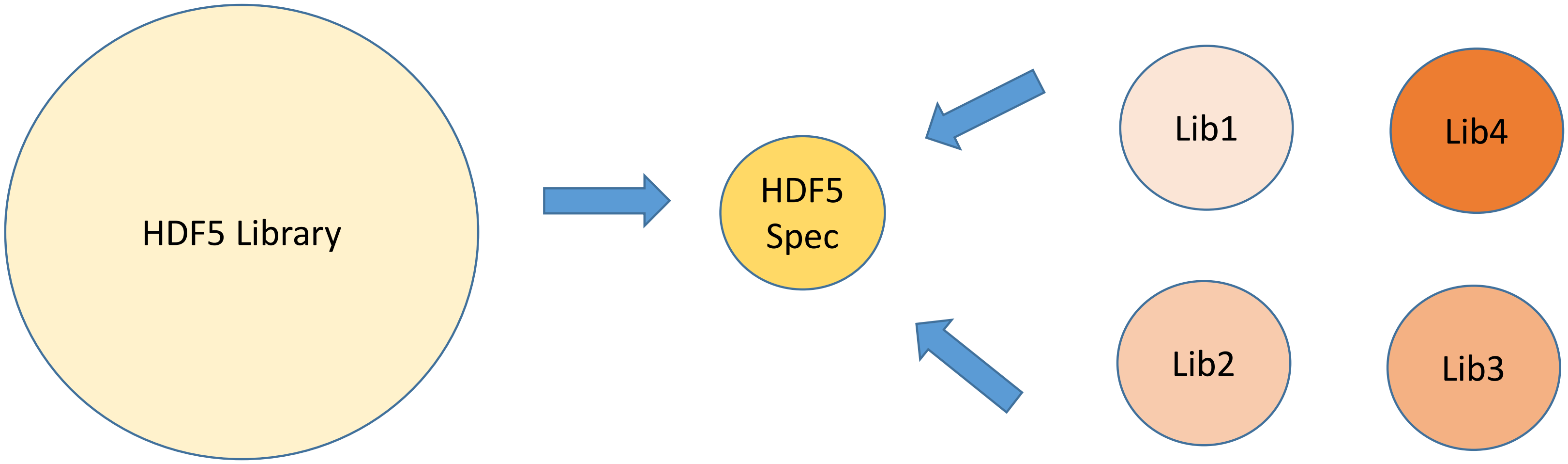
Special Purpose Libraries

We don’t need to choose… we can do both.
If you have a tight, simple, stable **specification**.

Best of both worlds

Consolidated Library

Special Purpose Libraries



Reference Implementation

Optimized Implementations

What steps can we take to promote the development of special purpose implementations of the HDF5 specification in different programming languages and for different applications?

Proposal

- Identify a small set of important data models and applications that cover many of the existing use cases for HDF5
  - Static write-once, read-many image data
  - High latency storage systems
  - …

- Define a subset of the HDF5 specification specifically suited to each data model and application identified above

- Build an option into the current HDF5 library to write (and check on read) H5 files that adhere to the subsetted specification.

H5Coro as an example

- Six weeks to get first working system

- A lot of the specification not implemented (and yet it works for many applications)

- A lot of the specification needed to be implemented just to get basic functionality

H5Coro Specifications ***NOT*** Supported

The following portions of the HDF5 format specification are intentionally not implemented:

- All write operations
- File free space management
- File driver information
- Virtual datasets

The following portions of the HDF5 format specification are intentionally constrained:

- Datasets with dimensions greater than 2 are flattened to 2 dimensions and left to the user to index.
- Only sequentially stored data can be read at one time, hyperslabs are not supported.
- Data type conversions are supported for fixed and floating point numbers only, but the intended use of the library is to return a raw memory block with the data values written sequentially into it, allowing the user to cast the memory to the correct array type.

H5Coro Support for File Structures

Format Element	Supported	Contains	Missing
Field Sizes	Yes	1, 2, 4, 8 bytes	
Superblock	Partial	Version 0	Version 1, 2, 3
B-Tree	Partial	Version 1	Version 2
Group Symbol Table	Yes	Version 1	
Local Heap	Yes	Version 0	
Global Heap	No		Version 1
Fractal Heap	Yes	Version 0	
Shared Object Header Message Table	No		Version 0
Data Object Headers	Yes	Version 1, 2	

H5Coro Support for Messages

Format Element	Supported	Contains	Missing
Shared Message	No		Version 1
NIL Message	Yes	Unversioned	
Dataspace Message	Yes	Version 1	
Link Info Message	Yes	Version 0	
Datatype Message	Partial	Version 1	Version 0, 2, 3
Fill Value (Old) Message	No		Unversioned
Fill Value Message	Partial	Version 2	Version 1, 3
Link Message	Yes	Version 1	
External Data Files Message	No		Version 1
Data Layout Message	Partial	Version 3	Version 1, 2
Bogus Message	No		Unversioned
Group Info Message	No		Version 0
Filter Pipeline Message	Yes	Version 1	
Attribute Message	No		Version 1
Object Comment Message	No		Unversioned
Object Modification Time (Old) Message	No		Unversioned
Shared Message Table Message	No		Version 0
Object Header Continuation Message	Yes	Version 1, 2	
Symbol Table Message	Yes	Unversioned	
Object Modification Time Message	No		Version 1
B-Tree ‘K’ Value Message	No		Version 0
Driver Info Message	No		Version 0
Attribute Info Message	No		Version 0
Object Reference Count Message	No		Version 0

H5Coro Support for Storage, Types, Filters

Format Element	Supported	Contains	Missing
Compact Storage	Yes		
Continuous Storage	Yes		
Chunked Storage	Yes		
Fixed Point Type	Yes		
Floating Point Type	Yes		
Time Type	No		
String Type	No		
Bit Field Type	No		
Opaque Type	No		
Compound Type	No		
Reference Type	No		
Enumerated Type	No		
Variable Length Type	No		
Array Type	No		
Deflate Filter	Yes		
Shuffle Filter	Yes		
Fletcher32 Filter	No		
Szip Filter	No		
Nbit Filter	No		
Scale Offset Filter	No		

Back of a Napkin H5 Cloud Standard

Field Size *(8 bytes)*

Superblock *(version 1)*

B-Tree *(version 1)*

Data Object Headers *(version 2)*

Messages: Dataspace, Datatype, Fill Value, Link, Data Layout, Filter Pipeline, Attribute, Object Comment, Header Continuation

Storage: compact, continue, chunked

Types: Fixed Point, Floating Point, Time, String

Filter: Deflate, Shuffle, Fletcher32, Szip, Nbit, Scale Offset

Acronyms

API	Application Program Interface
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
GSFC	Goddard Space Flight Center
HDF5	Hierarchical Data Format version 5
HTTP	Hypertext Transfer Protocol
ICESat-2	Ice, Cloud, and land Elevation Satellite, 2nd generation
IO	Input / Output
IP	Internet Protocol
NASA	National Aeronautics and Space Administration
NSIDC	National Snow and Ice Data Center
REST	Representational State Transfer
S3	Simple Cloud Storage Service
TCP	Transmission Control Protocol