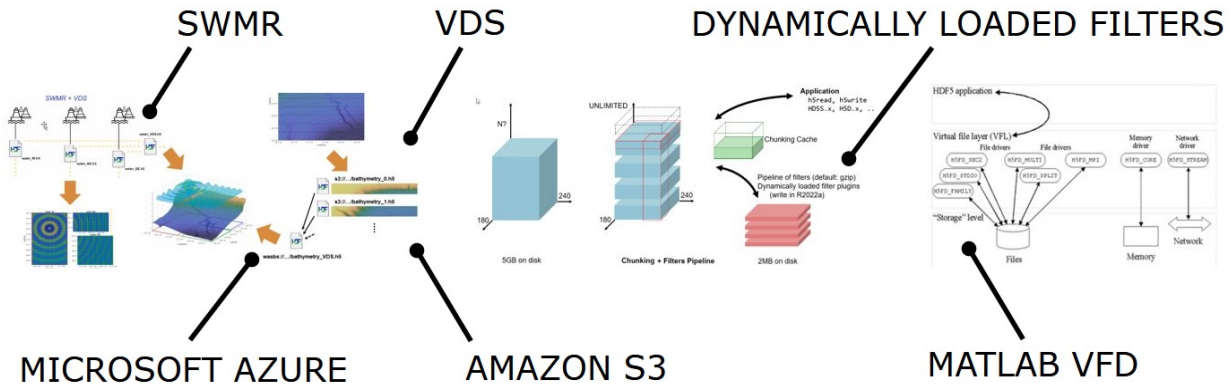
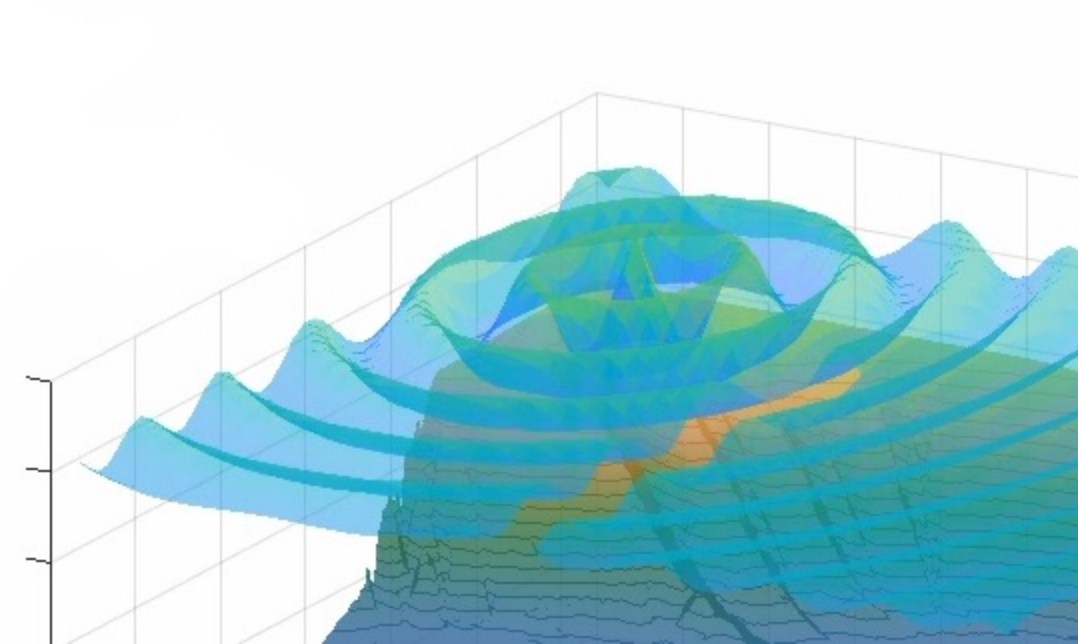
MATLAB and HDF5: Compression, Cloud, *and* Community

Ellen Johnson
Senior Software Engineer, MathWorks
European HDF5 User Group
May 31, 2022



Agenda

- Overview of HDF5 in MATLAB
- New in R2022a
- Demo
- Community Collaborations
- Future work
- Wrap-up and Q&A



Scientific Data in MATLAB

Scientific data formats

- HDF5, HDF4, HDF-EOS2
- NetCDF (with OPeNDAP)
- FITS, CDF, BIL, BIP, BSQ

Image file formats

- TIFF, JPEG, PNG, JPEG2000, HDR, *and more*

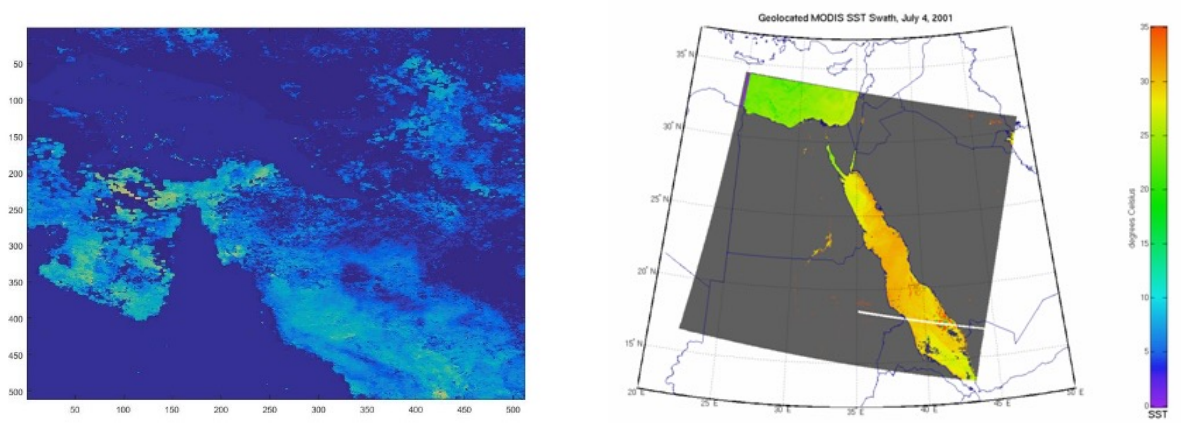
Vector data file formats

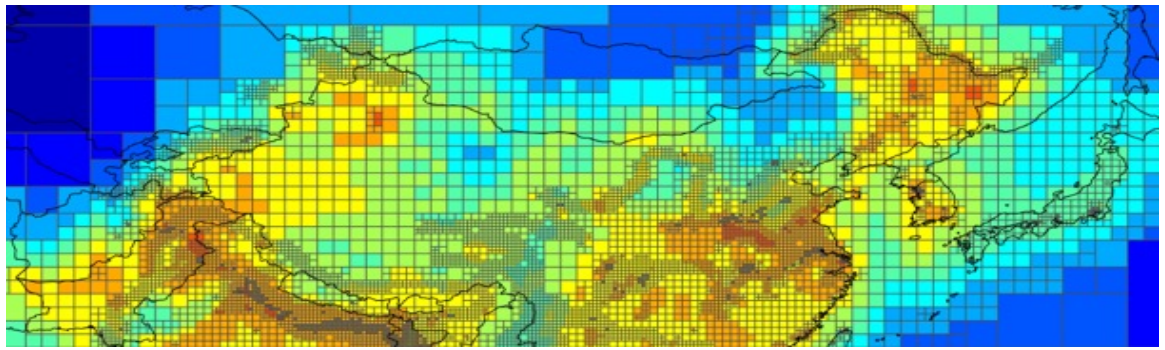
- ESRI Shapefiles, KML, GPS *and more*

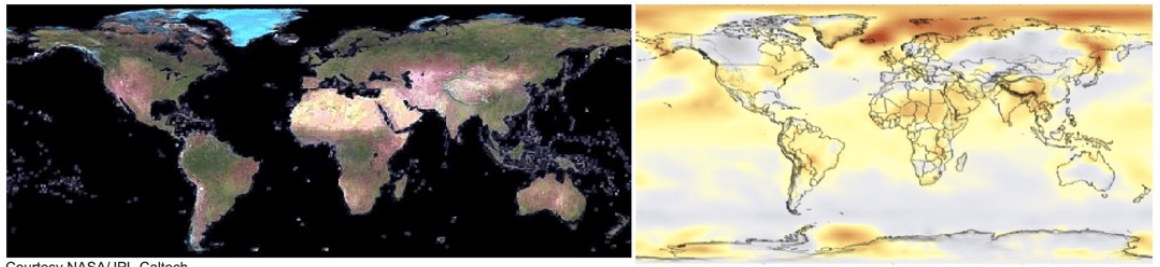
Raster data file formats

- GeoTIFF, NITF, USGS and SDTS DEM, NIMA DTED, *and more*

Web Map Service (WMS)







Courtesy NASA/JPL-Caltech

Courtesy NASA/Goddard Space Flight Center Scientific Visualization Studio

HDF5 in MATLAB

Two HDF5 interfaces

- High-level (HL) : Ease-of-use, less control
- Low-level (LL) : Wraps HDF5 C library, more control

Using the High-Level HDF5 interface:

```
h5disp("example.h5", "/g4/lat");
data = h5read("example.h5", "/g4/lat").'
```

Using the Low-Level HDF5 interface:

```
fid = H5F.open("example.h5");
dset_id = H5D.open(fid, "/g4/lat");
data = H5D.read(dset_id).'
H5D.close(dset_id);
H5F.close(fid);
```

```
HDF5 example.h5
Dataset 'lat'
    Size:  19
    MaxSize:  19
    Datatype:   H5T_IEEE_F64LE (double)
    ChunkSize:  []
    Filters:  none
    FillValue:  0.000000
    Attributes:
        'units':  'degrees_north'
        'CLASS':  'DIMENSION_SCALE'
        'NAME':  'lat'
data = 1x19
     -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10    20 ...



data = 1x19
     -90   -80   -70   -60   -50   -40   -30   -20   -10     0    10    20 ...
```

HDF5 in MATLAB

HDF5 Files

Hierarchical Data Format, Version 5

High-level access functions make it easy to read

High-Level Functions
Easily view, read, and write HDF5 files

Low-Level Functions
Interact directly with HDF5 library functions

High-Level Functions

Easily view, read, and write HDF5 files

Functions

h5create	Create HDF5 dataset
h5disp	Display contents of HDF5 file
h5info	Information about HDF5 file
h5read	Read data from HDF5 dataset
h5readatt	Read attribute from HDF5 file
h5write	Write to HDF5 dataset
h5writeatt	Write HDF5 attribute

7 high-level functions

~330 low-level functions

HDF5 Library Packages

Library (H5)	General-purpose functions for use with entire HDF5 library
Attribute (H5A)	Metadata associated with datasets or groups
Dataset (H5D)	Multidimensional arrays of data elements and supporting metadata
Dimension Scale (H5DS)	Dimension scale associated with dataset dimensions
Error (H5E)	Error handling
File (H5F)	HDF5 file access
Group (H5G)	Organization of objects in file
Identifier (H5I)	HDF5 object identifiers
Link (H5L)	Links in HDF5 file
MATLAB (H5ML)	MATLAB Utility functions not part of HDF5 C library
Object (H5O)	Objects in file
Property (H5P)	Object property lists
Reference (H5R)	HDF5 references
Dataspace (H5S)	Dimensionality of dataset
Datatype (H5T)	Datatype of elements in a dataset
Filters and Compression (H5Z)	Inline data filters, data compression

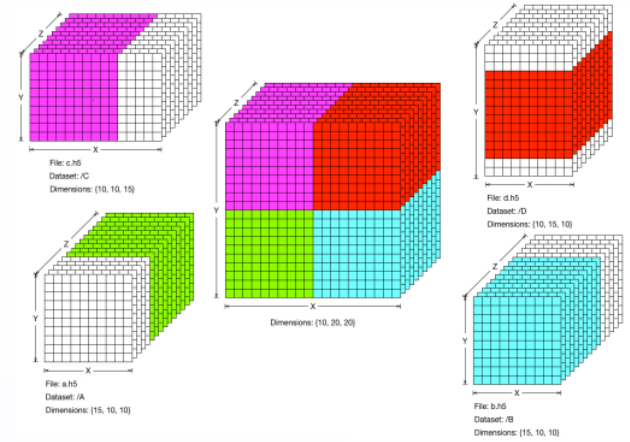
What We've Been Up To

R2020b

HDF5 Interface: Cloud-enabled
- S3 and Azure: Read/Write
- Hadoop: Read-only
- Enabled for all HL and LL functions

R2021a

MAT-file v7.3 save/load: Cloud-enabled

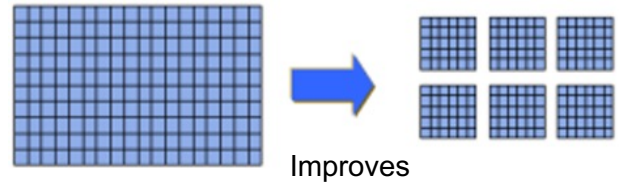


R2021b

Upgrade to HDF5 1.10.7

Support for Single-Writer/Multiple-Reader and Virtual Datasets

Chunked & Compressed



Improves storage efficiency, transmission speed

R2022a

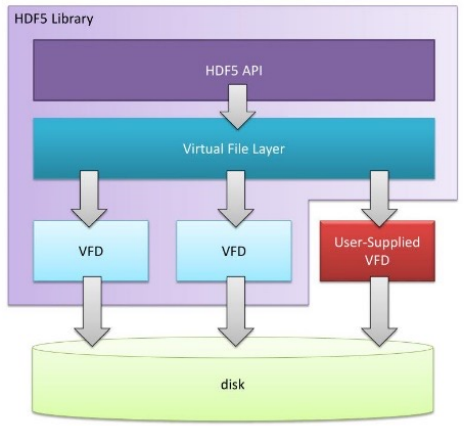
Upgrade to HDF5 1.10.8

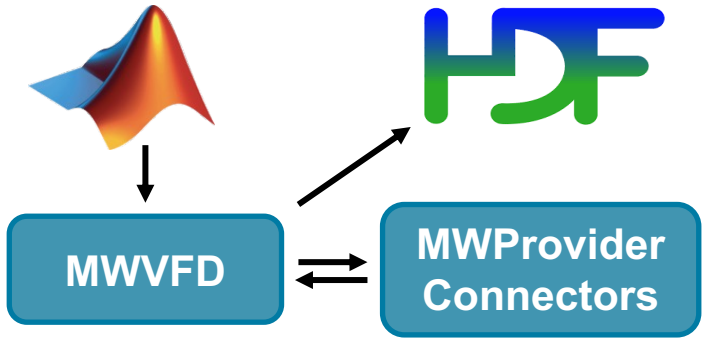
Writing datasets using Dynamically Loaded Filters

Refresher: Cloud Data Access

R2020b

- Wrote in-house HDF5 VFD
- Use in-house provider architecture
- Callbacks to HDF5 library





- S3 and Azure: Read/Write
- Hadoop: Read only
- Support in High and low-level interfaces
  including SWMR, VDS, Filter Plugins

amazon web services S3

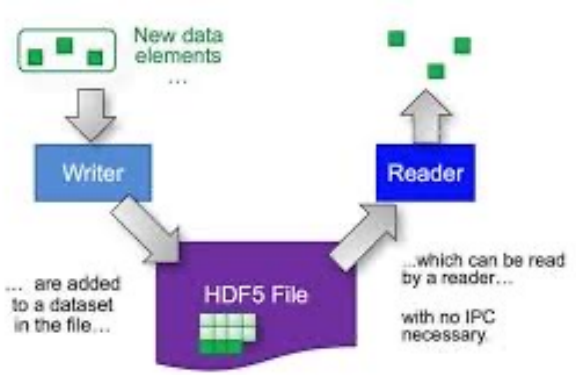
Azure hadoop

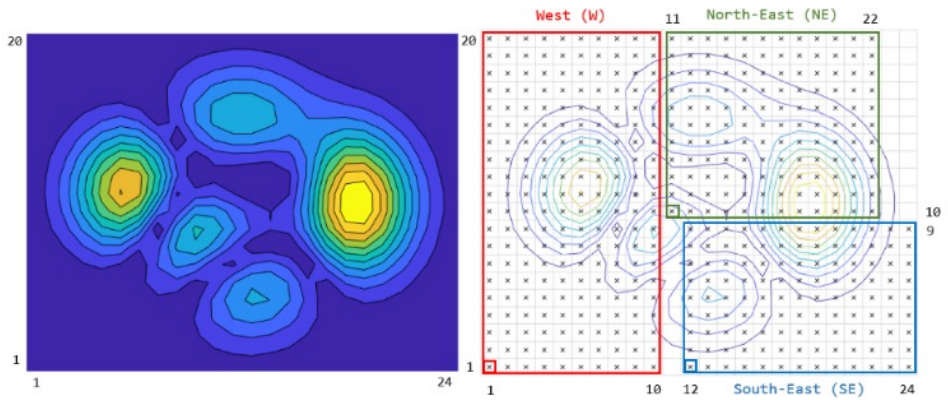
```
>> h5create("s3://h5test/myfile.h5","/ds1",[200 Inf],"ChunkSize",[20 20],"Deflate",9)
>> h5write("wasbs://h5test/myfile.h5","/ds1",rand(200,500),[1 1],[200 500])
>> h5read("hdfs://h5test/myfile.h5","/ds1")
```

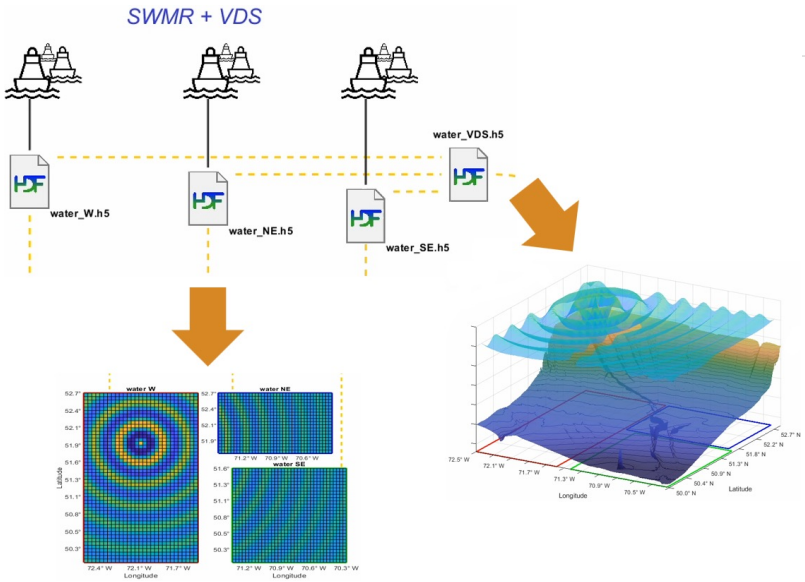
Refresher: SWMR and VDS

R2021b

- SWMR
- VDS
- Fine-tuning the MDC
- Partial Edge Chunk







Documentation | Examples | Functions | Apps | Videos | Answers | Trial Software | Product Updates

Read and Write Data Concurrently Using Single-Writer/Multiple-Reader (SWMR)

Overview

The Single-Writer/Multiple-Reader (SWMR) feature of the MATLAB® low-level HDF5 function interface allows you to append data to datasets or overwrite existing data while several reader processes concurrently read the new data from the file. The reader and writer processes can run on the same platform or different platforms, and no communication between the processes or file locking is needed.

To use SWMR, you must be familiar with the HDF5 SWMR programming model. For more information, see the HDF5 SWMR Documentation on The HDF Group website.

Work with HDF5 Virtual Datasets (VDS)

Overview

The HDF5 Virtual Dataset (VDS) feature allows you to access data from a collection of HDF5 files as a single, unified dataset, without modifying how the data is stored in the original files. Virtual datasets can have unlimited dimensions and map to source datasets with unlimited dimensions. This mapping allows a Virtual Dataset to grow over time, as its underlying source datasets change in size.

The VDS feature was introduced in the HDF5 library version 1.10. To use VDS, you must be familiar with the HDF5 VDS programming model. For more information, see the HDF5 Virtual Dataset documentation on The HDF Group website.

New: Writing Datasets with Dynamically Loaded Filters

R2022a

Now have full round-trip write/read for DLFs!

High-Level Interface

New *Name-Value* pairs in **`h5create`:**

`CustomFilterID` → Registered filter ID

`CustomFilterParameters` → Optional filter data

Example: BZIP2 filter

Filter ID = 307

Filter params = 6 (compression level)

```
h5create("myfile.h5","/dset1",[100 200],...
  "ChunkSize",[20 40],"CustomFilterID",307,...
  "CustomFilterParameters",6);

h5write("myfile.h5","/dset1",rand(100,200));
```

`CustomFilterID` — Filter identifier

[ ] (default) | positive integer

Filter identifier for the registered filter plugin assigned by The HDF Group, specified as a positive integer. For a list of registered filters, see the Filters page on The HDF Group website.

The default value for this argument means the data set does not use dynamically loaded filters for compression.

Data Types: `double`

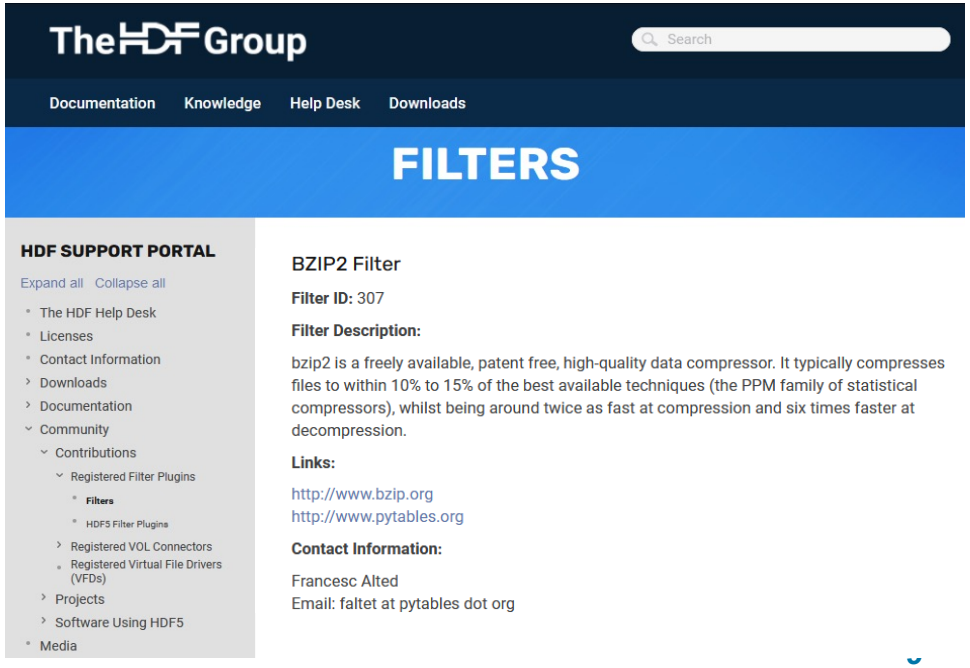
`CustomFilterParameters` — Filter parameters

[ ] (default) | numeric scalar | numeric row vector

Filter parameters for third-party filters, specified as a numeric scalar or row vector. If you specify the `CustomFilterID` without also specifying this argument, the `h5create` function passes an empty vector to the HDF5 library and the filter uses default parameters.

This name-value argument corresponds to the `cd_values` argument of the `H5Pset_filter` function in the HDF5 library.

Data Types: `double`



New: Writing Datasets with DLFs

R2022a

Low-level interface

```
H5P.set_filter(plistId,filterId,flags,cdValues)
```

Same example: BZIP2 filter

Filter ID = 307

cdValues = 6 (compression level)

```
% create the file and dataspace, set the chunking
fileId = H5F.create("myfile.h5","H5F_ACC_TRUNC","H5P_DEFAULT","H5P_DEFAULT");
spaceId = H5S.create_simple(2,[200 100],[]);
dcplId = H5P.create("H5P_DATASET_CREATE");
H5P.set_chunk(dcplId,[40 20]);

% set BZIP2 filter (registered ID = 307), verify filter is available
H5P.set_filter(dcplId,307,"H5Z_FLAG_OPTIONAL",6);
avail = H5Z.filter_avail(307)

% create the dataset and write the data
dsetId = H5D.create(fileId,"dset1","H5T_STD_I32LE",spaceId,"H5P_DEFAULT",dcplId,"H5P_DEFAULT");
H5D.write(dsetId,"H5ML_DEFAULT","H5S_ALL","H5S_ALL","H5P_DEFAULT",data);
```

H5P.set_filter
H5P.get_filter
H5P.modify_filter
H5P.remove_filter
H5P.get_filter_by_id
H5P.get_nfilters
H5P.all_filters_avail

H5Z.get_filter_info
H5Z.filter_avail

H5P.set_filter

Add filter to filter pipeline

For custom third-party filters, specify `filter` as the numeric filter identifier assigned by The HDF Group.

- `flags` — Constant that specifies the behavior when the filter fails. Valid values are:
  - `'H5Z_FLAG_OPTIONAL'` — Filter is excluded from the pipeline for the chunk in which the filter failed. The filter does not participate in the pipeline during a subsequent read of the chunk.
  - `'H5Z_FLAG_MANDATORY'` — The HDF5 library issues an error upon filter failure. The library writes all chunks processed by the filter before the failure occurred.
- `cd_values` — Array containing auxiliary data for the filter.

Reading Datasets with DLFs

Reading (since R2015a) just works!

(As long as filter plugin is installed and **HDF5_PLUGIN_PATH** is set)

```
% explore file contents
h5disp("myfile.h5")
```

```
HDF5 myfile.h5
Group '/'
    Dataset 'dset1'
        Size:  100x200
        MaxSize:  100x200
        Datatype:   H5T_IEEE_F64LE (double)
        ChunkSize:  20x40
        Filters:  unrecognized filter (HDF5 bzip2 filter; see http://www.hdfgroup.org/services/contributions.html)
        FillValue:  0.000000
```

```
% read a hyperslab
data = h5read("myfile.h5","/dset1",[1 1],[8 8],[10 20])
```

```
  data =
      0.4362    0.1035    0.6518    0.4199    0.9746    0.8252    0.6761    0.0730
      0.1604    0.9797    0.0477    0.9842    0.0910    0.3018    0.7048    0.4659
      0.4122    0.6901    0.4410    0.5075    0.4662    0.1697    0.1581    0.3335
      …
```

New DLF Topic Page

R2022a

Read and Write HDF5 Datasets Using Dynamically Loaded Filters

R2022a

The HDF5 library and file format enables using filters on data chunks before they are written to or after they are read from disk. Compression filters, for example, can substantially reduce the size of the data to be stored on disk and improve the overall performance of reading from and writing to an HDF5 dataset.

The HDF5 library includes a small set of internal filters, and MATLAB® supports most of them. While these filters work relatively well, they may not always provide an optimal performance improvement. For this reason, the HDF5 library and MATLAB support dynamically loaded filters, a mechanism that enables loading third-party filters at run time and adding them to the filter pipeline. To use dynamically-loaded filters, install filter plugins for reading datasets that were created using third-party filters, or for creating and writing datasets using third-party filters.

Install Filter Plugins

MATLAB supports three internal HDF5 filters: Deflate (GZIP), Shuffle, and Fletcher32. To read or write datasets using third-party filters, install and configure filter plugins.

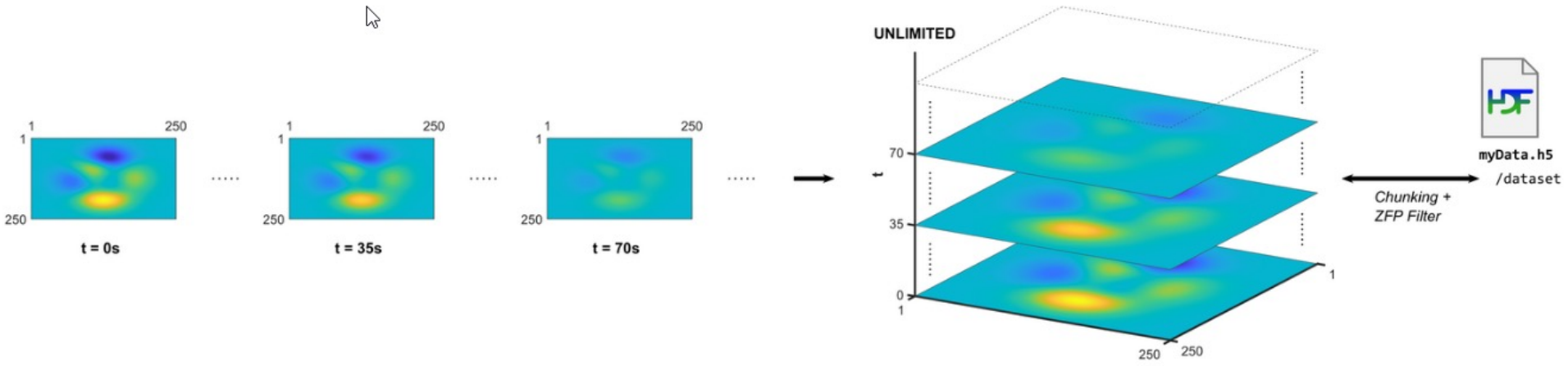
1. Install the relevant filter plugins:
   - On Windows® and Mac — Download and install the plugin binaries for your operating system from The HDF Group.

     Install the bundle of filter plugins for the version of HDF5 shipped with your MATLAB release. To query the version of HDF5 in your MATLAB release, use `H5.get_libversion`.
   - On Linux® — Get the filter plugin source code and build it against the version of HDF5 that is shipped with MATLAB. To obtain the filter plugin source code, see The HDF Group - Filters. Alternatively, you can build HDF5 from source using the instructions and the export map file from Build HDF5 Filter Plugins on MATLAB Answers™, and then build the filter plugin against your built HDF5 library.
2. Set the `HDF5_PLUGIN_PATH` environment variable to point to the local installation of the plugins:
   - On Windows — Set the environment variable using *System Properties > Advanced > Environment Variables*.
   - On Linux and Mac — Set the environment variable from the terminal before starting MATLAB.
3. Restart MATLAB.

Write Datasets Compressed with Third-Party Filters

You can create and write an HDF5 dataset using either the high-level interface (such as `h5create` and `h5write`) or low-level interface (such as `H5D.create` and `H5D.write`). To write a dataset with a third-party filter, first identify the filter ID and parameters from The HDF Group - Filters page.

For example, create an HDF5 dataset for a time series of 2-D arrays that grows with time. Use data chunking to enable an unlimited third dimension, and use a ZFP filter to compress the data chunks, as shown in this illustration.



DLF Workflow Requirements

R2022a

Install filter plugin

- Use prebuilt binaries (i.e., hdf5plugin binaries from THG)
- Build from source

Set **`HDF5_PLUGIN_PATH`** environment variable

- **Windows:** System Properties → Advanced → Environment Variables
- **Linux and Mac:** set from terminal, then start MATLAB from terminal

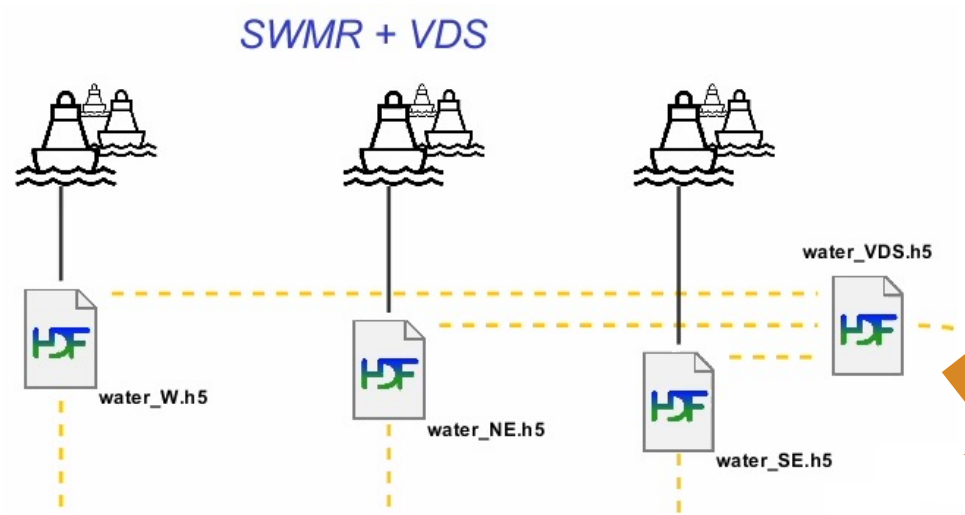
Linux example using `tcsh`:

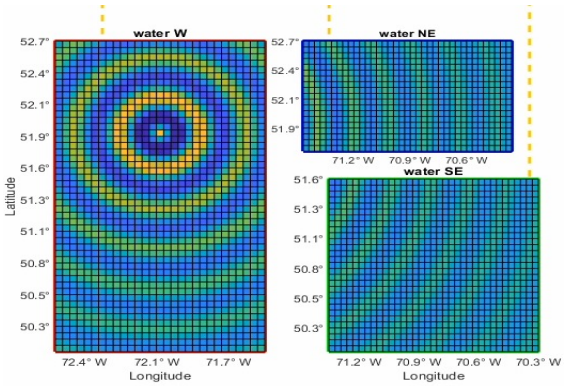
```
setenv HDF5_PLUGIN_PATH /ellenj/h5dlf/BZIP2-plugin/plugins/lib
```

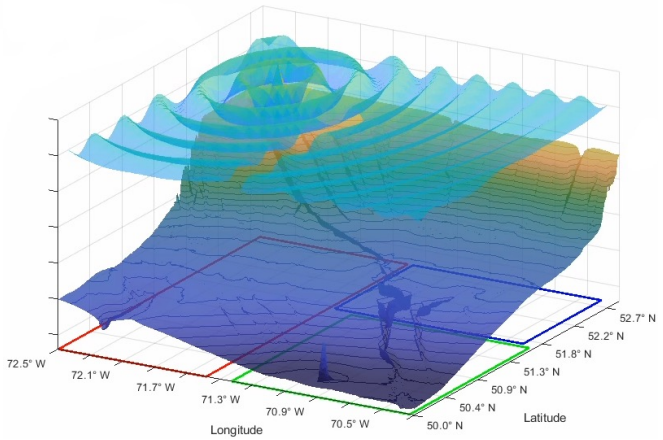
Start MATLAB, use **`getenv`** to verify path is set:

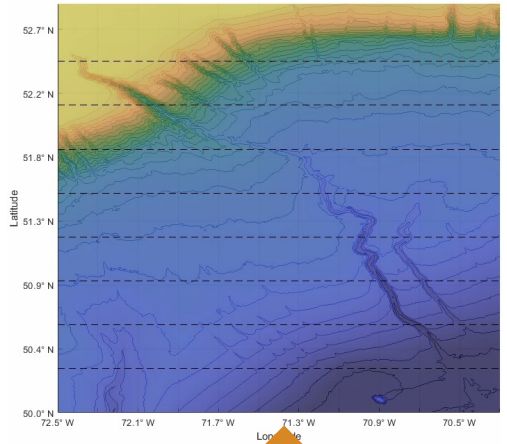
```
>> getenv("HDF5_PLUGIN_PATH")
   '/ellenj/h5dlf/BZIP2-plugin/plugins/lib'
```

Demo – MATLAB Meets HDF5 in the Cloud









BLOSC

s3://…/bathymetry_0.h5

BZIP2

s3://…/bathymetry_1.h5

wasbs://…/bathymetry_VDS.h5

GEBCO Gridded Bathymetry Data: https://www.gebco.net/data_and_products/gridded_bathymetry_data/
GEBCO Compilation Group (2020) GEBCO 2020 Grid (doi:10.5285/a29c5465-b138-234d-e053-6c86abc040b9)


MathWorks®

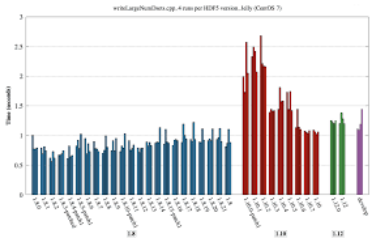
The HDF Group

MATLAB and Community Collaborations

Continuing existing and forging new connections!

The HDF Group

- Long-standing relationship
- Monthly meetings – technical and planning
- Security (CVEs), Performance

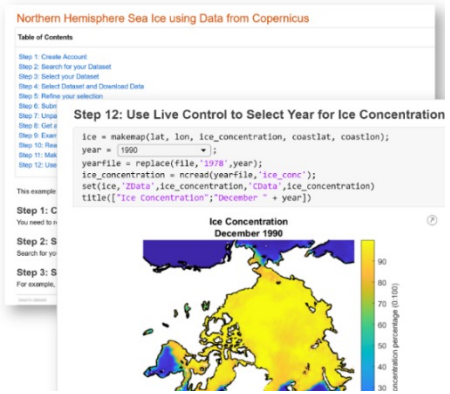


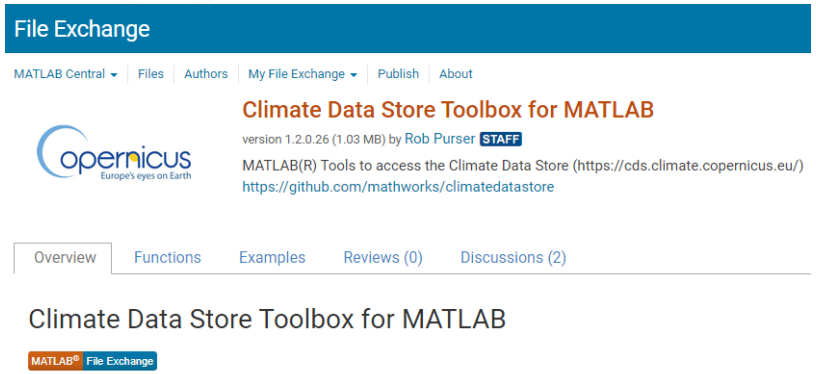
Neurodata Without Borders

- Newly formed MatNWB working group
- Consult and collaborate on projects, features

Earth and Climate Science

- *Under consideration:*
  - Climate data connectors
  - Contributions to open-source community toolboxes like Climate Data Toolbox





Future Work and Community Engagement

Highest priority

- Ship one HDF5 version
- Improved filter plugin experience (ship prebuilt plugin binaries)
- VDS and SWMR control in high-level interface
- Performance

Community Engagement

- Continue working with THG
- High-energy physics community – provide feedback on DLF, SWMR, VDS and wish-lists
- Earth/Climate Data Providers – please host more HDF5 data on cloud!

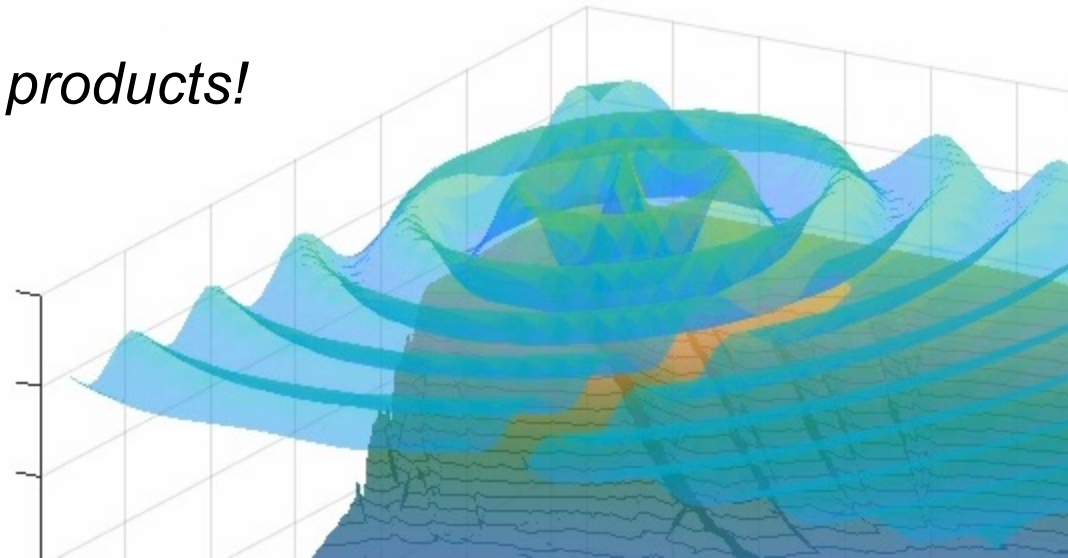
Wrap-up and Q&A

- MATLAB current with 1.10 branch – 22a ships with 1.10.8
- Full roundtrip support for Dynamically Loaded Filters
- Comprehensive doc pages for SWMR, VDS, DLFs
- Expanding community involvement

We love hearing feedback – it helps us improve our products!

Reach out with any questions or wish-lists!

- ellenj@mathworks.com



Acknowledgements

- GEBCO Gridded Bathymetry Data: https://www.gebco.net/data_and_products/gridded_bathymetry_data/
  GEBCO Compilation Group (2020) GEBCO 2020 Grid (doi:10.5285/a29c5465-b138-234d-e053-6c86abc040b9)
- The HDF Group: www.hdfgroup.com
- HDF5 VDS RFC: https://portal.hdfgroup.org/display/HDF5/RFC+HDF5+Virtual+Dataset

Thank You!