Seamless Integration of Blosc2 and HDF5 for High-Performance Data Compression

Francesc Alted / @FrancescAlted@masto.social

The Blosc Development Team / @Blosc2@fosstodon.org

CEO ironArray / francesc@ironarray.io

2025 European HUG Meeting – DESY, Hamburg, Germany

May 26th 2025

Agenda

Intro

Support for **Blosc2 NDim in HDF5**

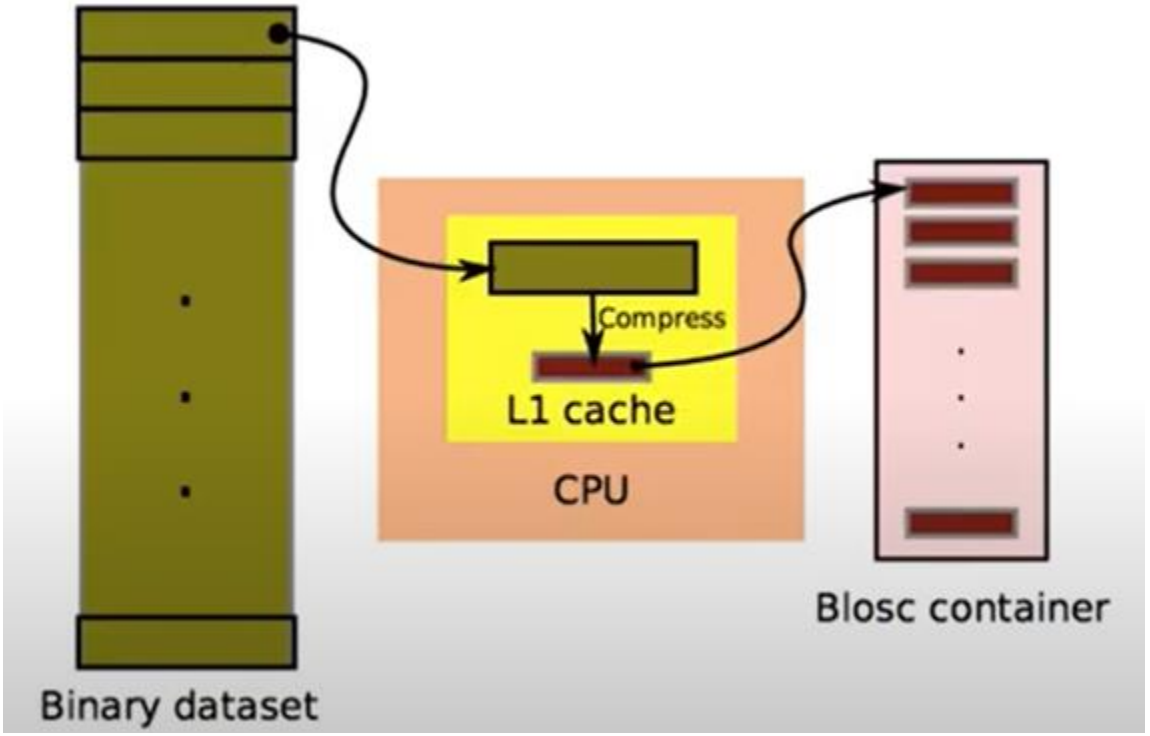
Plugins for **JPEG2000**

Caterva2: On-demand access to local/remote Blosc2/HDF5 data repositories

Intro

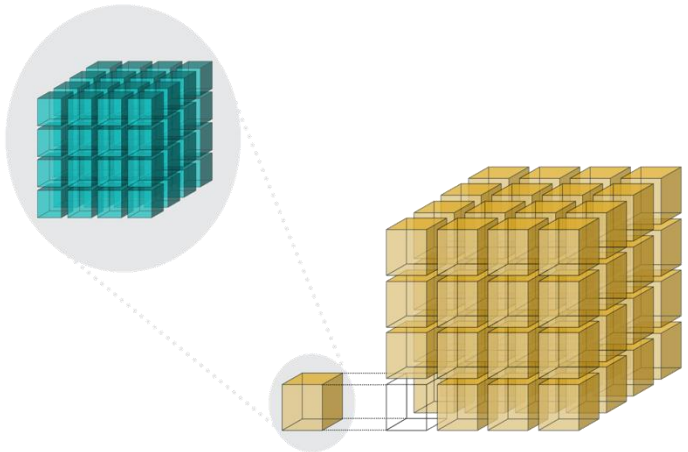
What is Blosc?

- A collection of codecs and filters for compressing binary data
- Goal: sending data from memory to CPU (and back) faster than *memcpy()*.
- Combining chunking and blocking: divide and conquer.



What is Blosc2?

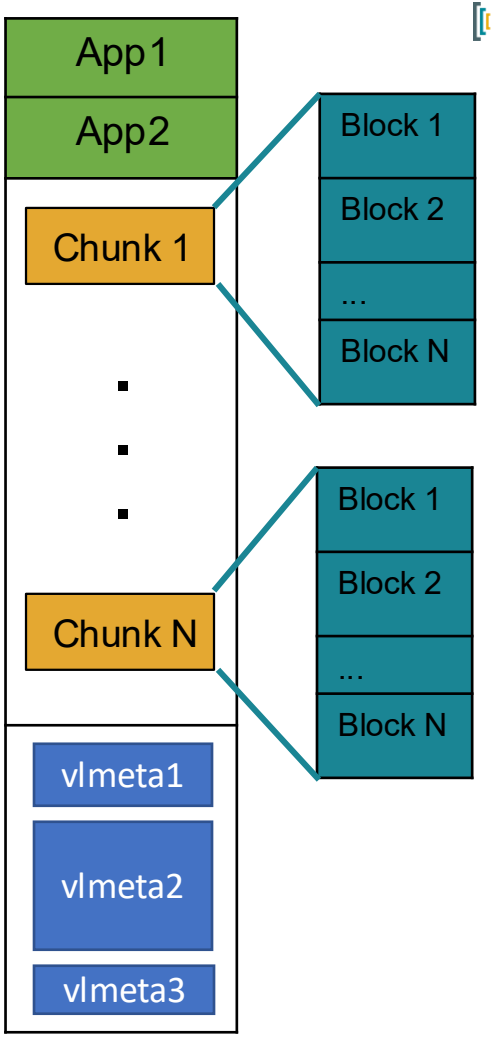
- ✓ Adds 63-bit containers
- ✓ Metalayers for adding info for apps and users
- ✓ Blosc2 NDim: Multi-dim blocks and chunks



Header:
Fixed Length
Metalayers

Data:
Super-Chunk

Trailer:
Var Length
Metalayers
(up to 2 GB)



Who is ironArray SLU?

- We are the developers of PyTables, numexpr and Blosc ecosystems
- Team of experts empowering you to harness the full potential of compression for big data: we are here to help!

ironArray

https://ironarray.io

Support for Blosc2 NDim in HDF5/h5py

Work done for the LEAPS INNOV Program

New version of Blosc2 plugin for HDF5

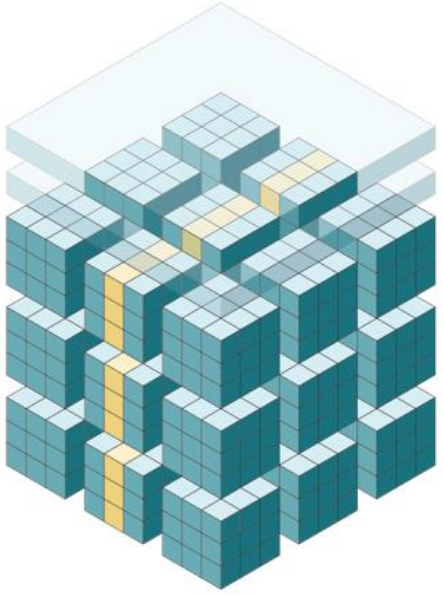
- Compression/decompression for HDF5 standard filter pipeline. Integrated in hdf5plugin.

- Optimization for Blosc2 NDim: useful for selective reading of blocks inside chunks; see b2h5py later.

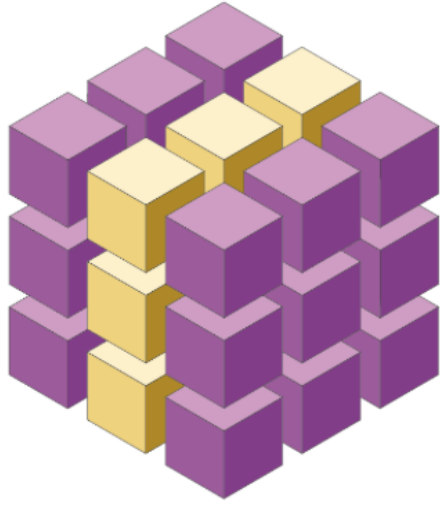
Thanks to Ivan Vilata (ironArray) and Thomas Vincent (ESRF)

Leveraging the second partition in Blosc2 NDim

Much more selective and hence, faster queries!



Blosc2 NDim

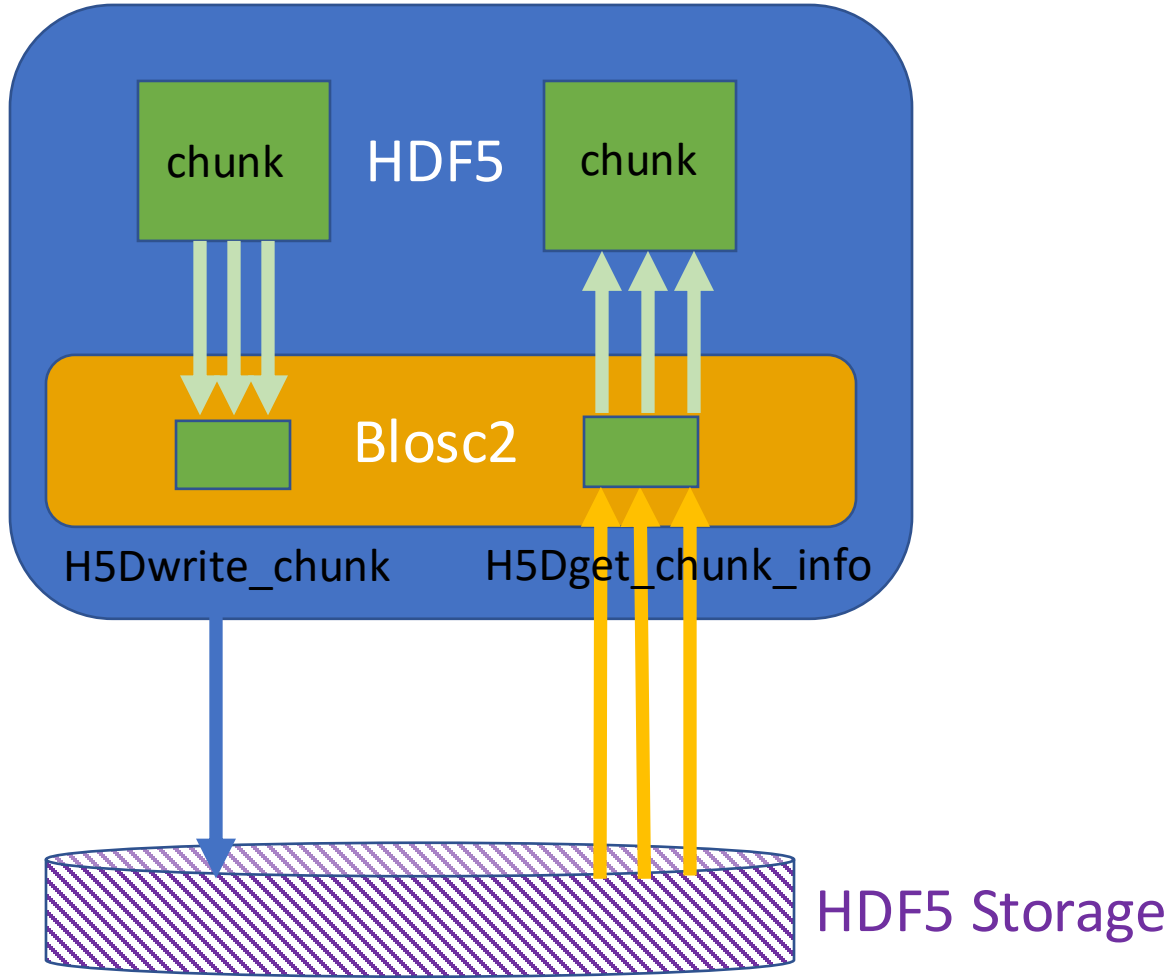


HDF5 / Zarr / others

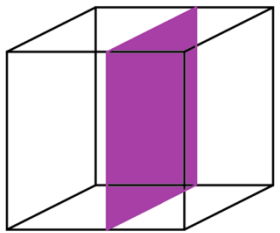
b2h5py: Use Blosc2 Inside Direct Chunking

- Both compression and decompression executed in parallel via Blosc2!
- Allows parallel I/O for reads
- Blosc2 NDim support: Chunk reads with enhanced selectivity from disk

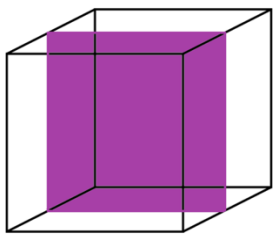
https://github.com/Blosc/b2h5py



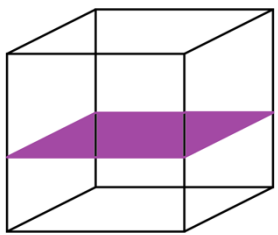
HDF5 pipeline vs direct chunking: Orthogonal slices in b2h5py/PyTables



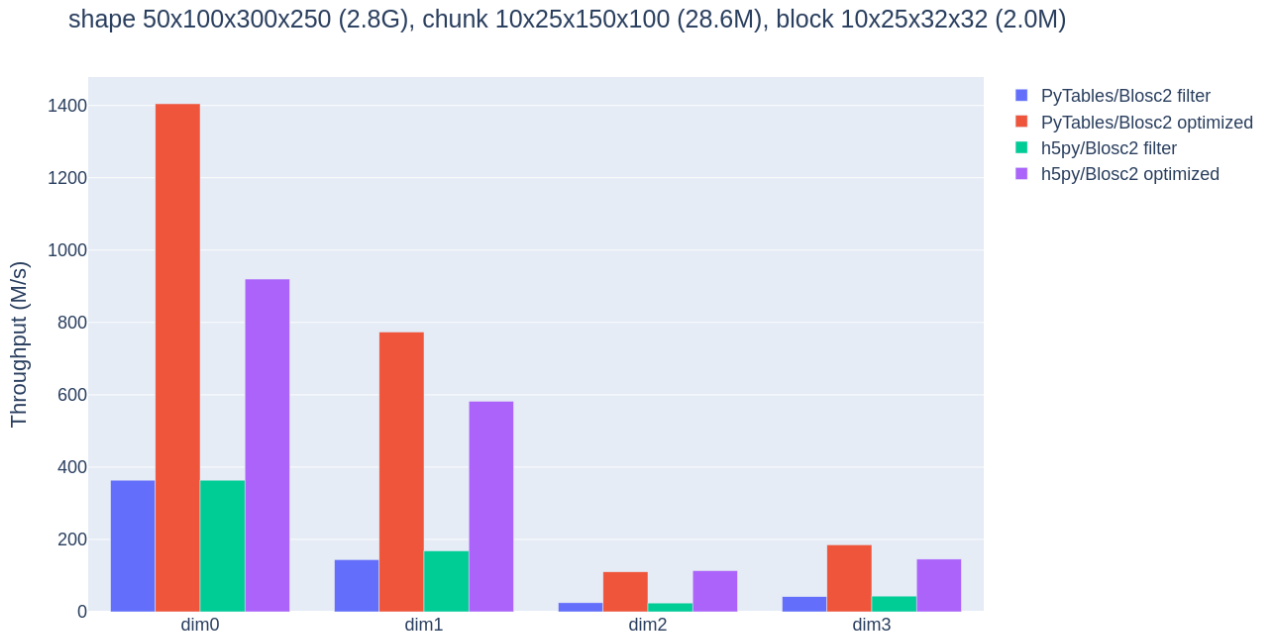
constant dim0



constant dim1



constant dim2



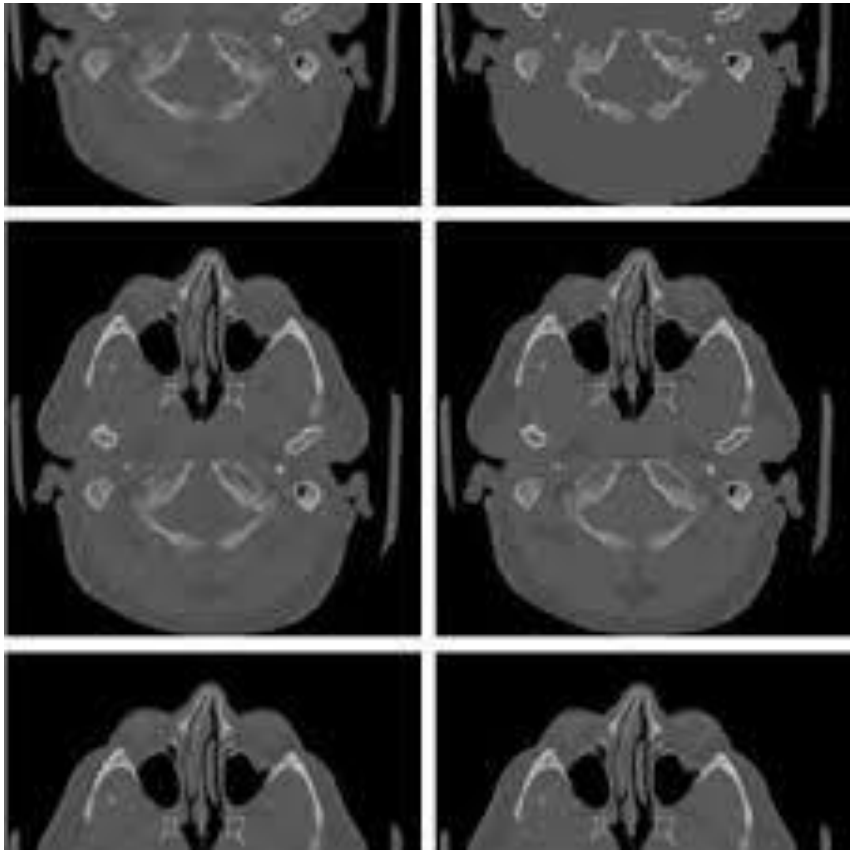
https://www.blosc.org/posts/pytables-b2nd-slicing/

JPEG2000 from Blosc2/HDF5/h5py

Enable compression/decompression with JPEG2000 codec
LEAPS INNOV program

Lossy compression with JPEG2000

- Can achieve great image quality in relatively high compression ratios (up to 10x).
- Already used in medical applications with great results.



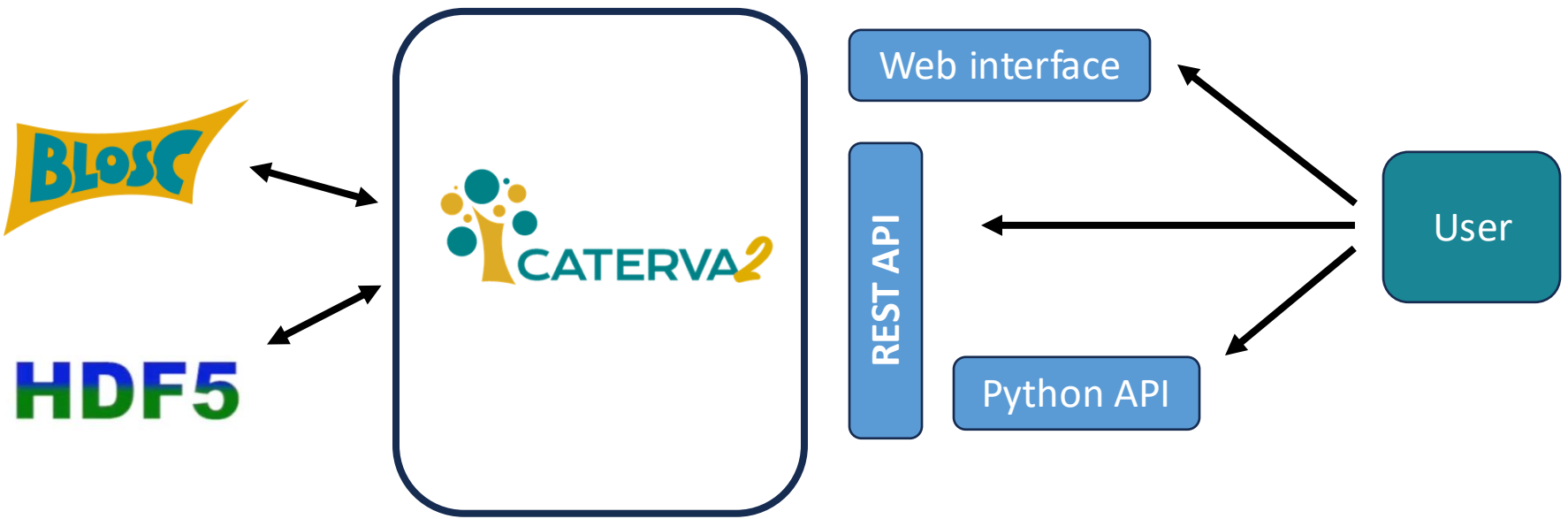
JPEG2000 grok and OpenHTJ2K as dynamic plugins for Blosc2

- OpenHTJ2K, an open source HTJ2K implementation.
- Grok, another implementation for HTJ2K; supports 16-bit gray images
- Packed and distributed as Python wheels:
  - $ pip install blosc2-openhtj2k
  - $ pip install blosc2-grok

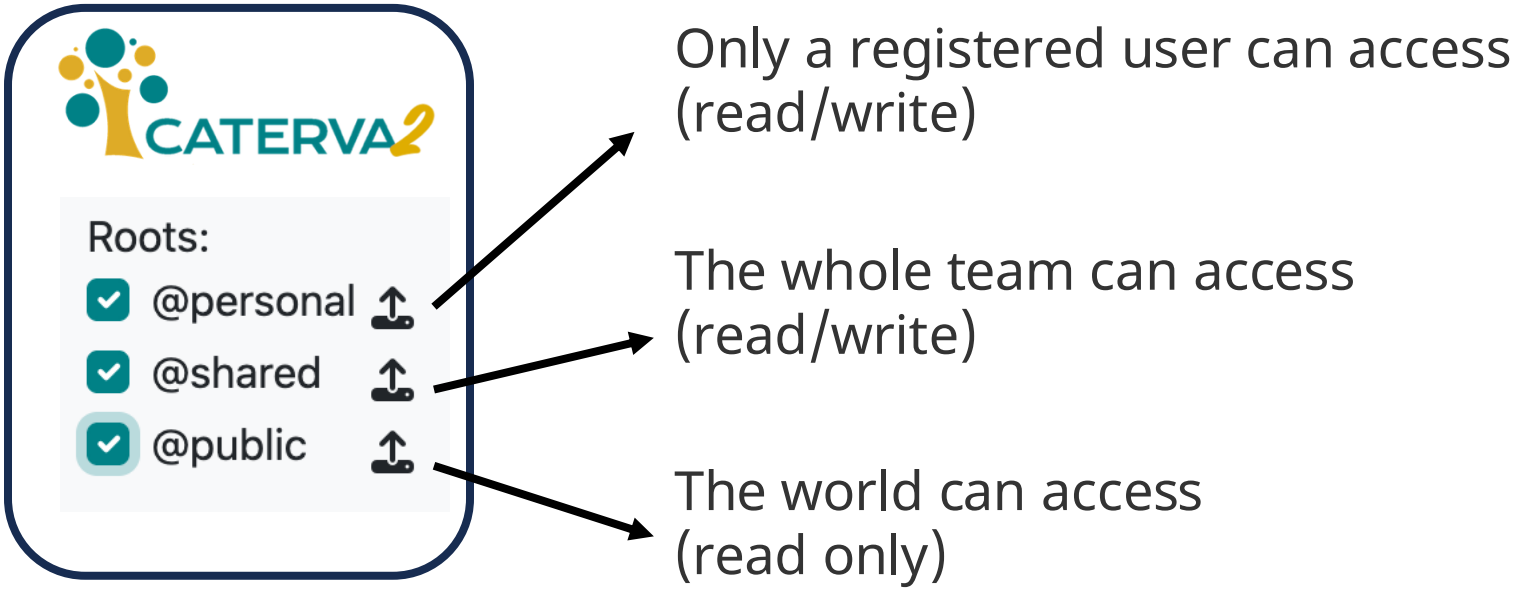
Thanks to Marta Iborra for her outstanding job!

Caterva2: On-demand access to local/remote Blosc2/HDF5 datasets

Caterva2
Serving data through Internet



Sharing data with your team, and the world!



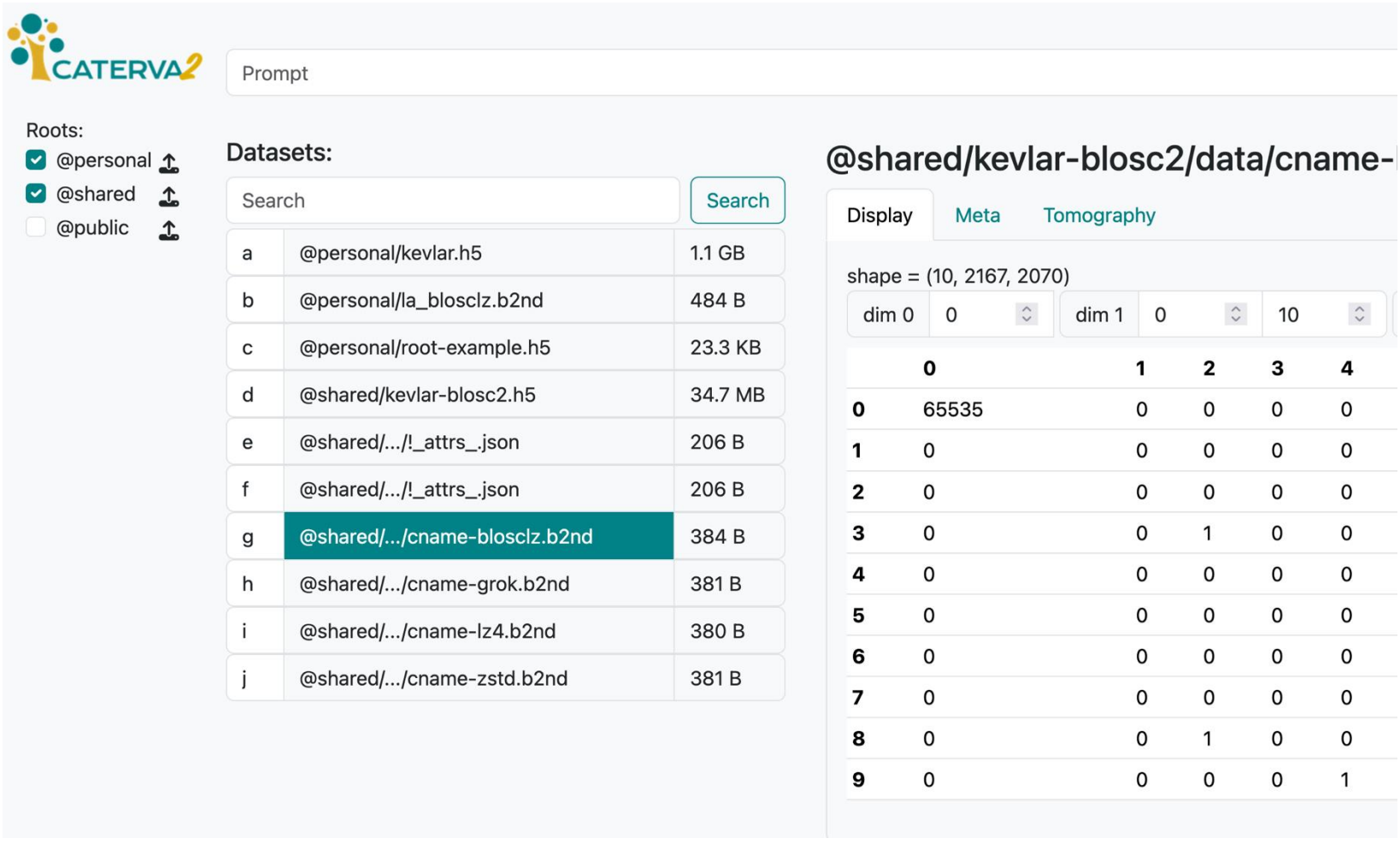
Demo time

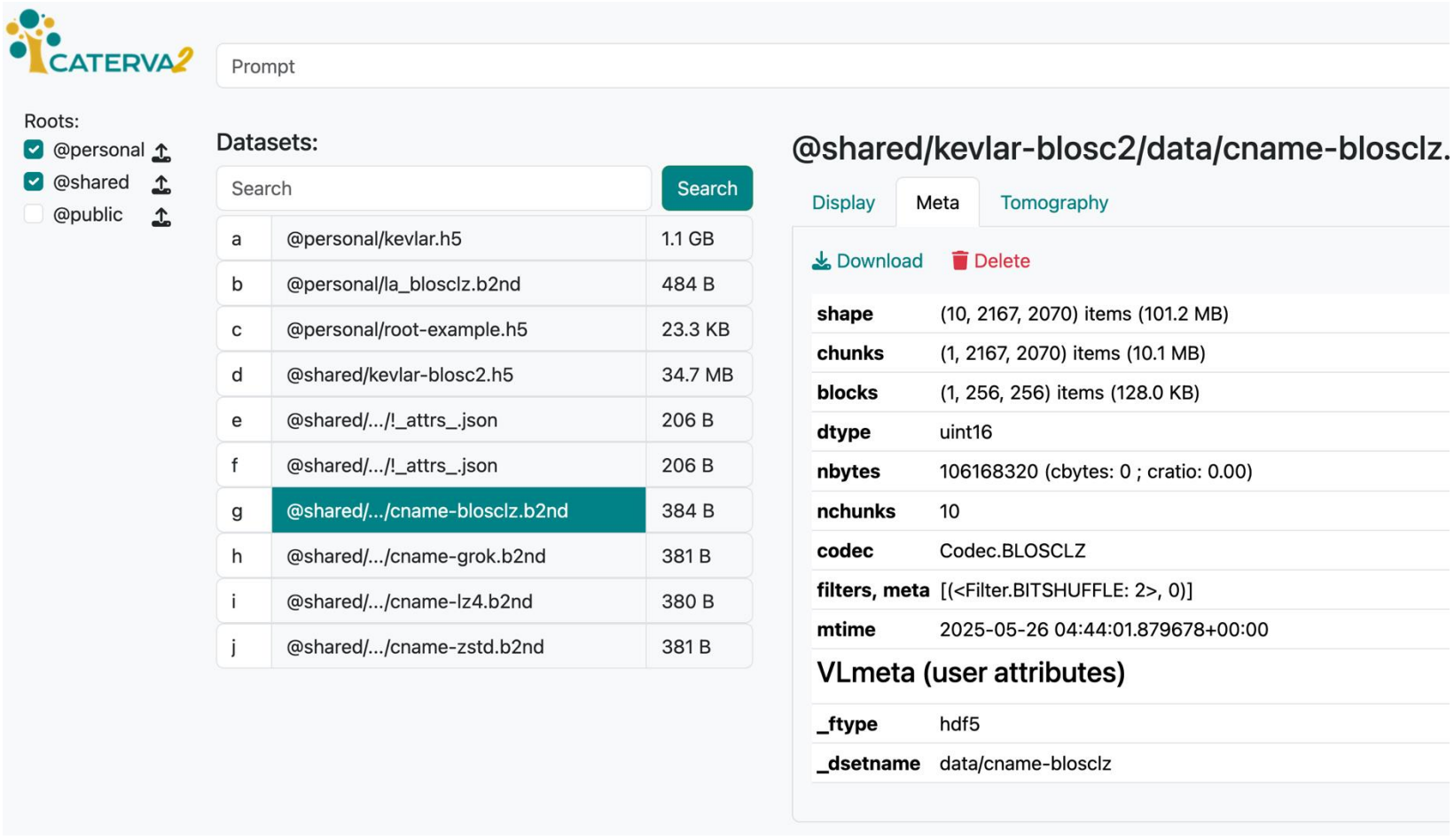
- Go to cat2.cloud/demo and try the interface by following me
- You can register to have write access; if not, you can still visualize data in cat2.cloud/demo/@public

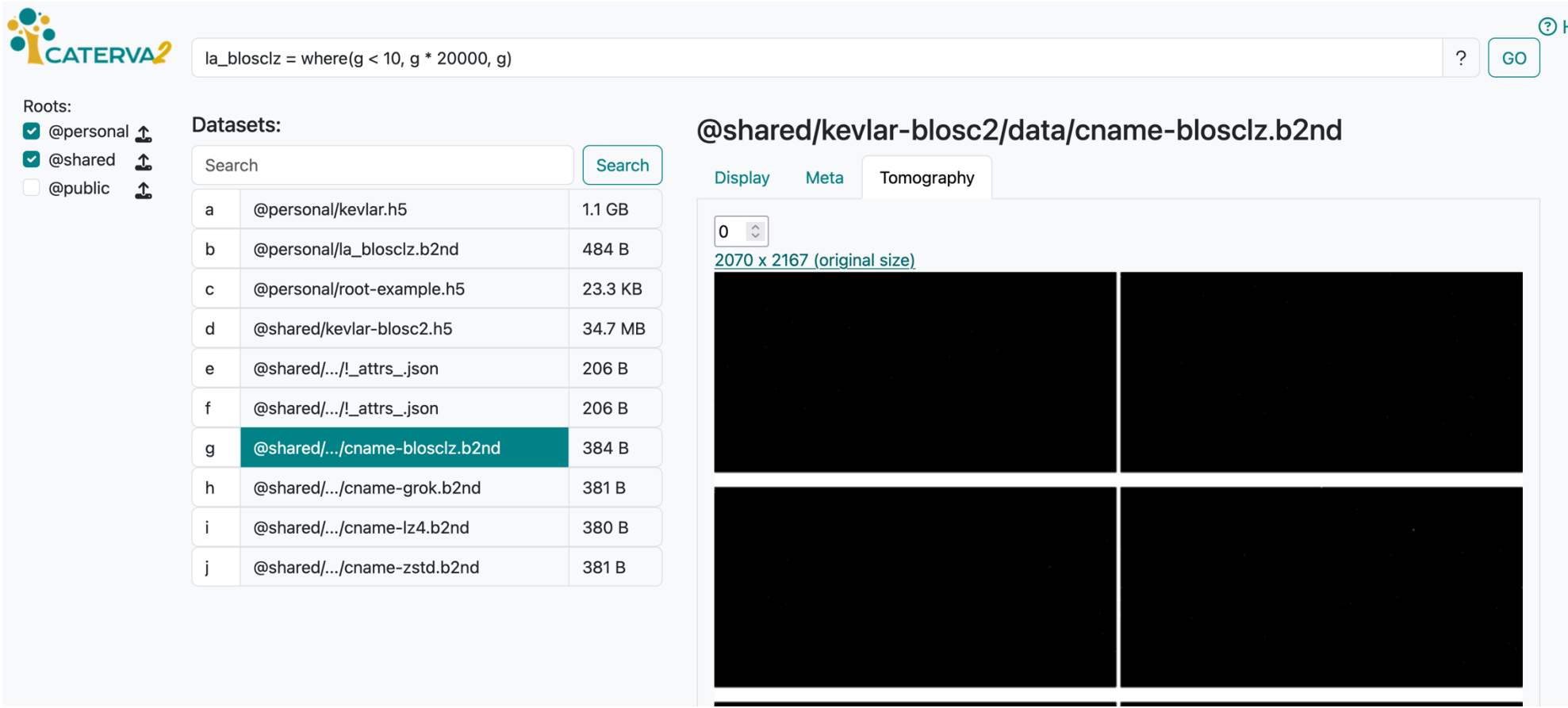
Helpers:

- Script for creating HDF5 files in demo
- Jupyter notebook for reproducing the demo

It is a demo service: do not use for sensible data!


CATERVA2
Prompt
Roots:
@personal
@shared
@public
Datasets:
Search
Search
a @personal/kevlar.h5 1.1 GB
b @personal/la_blosclz.b2nd 484 B
c @personal/root-example.h5 23.3 KB
d @shared/kevlar-blosc2.h5 34.7 MB
e @shared/.../!_attrs_.json 206 B
f @shared/.../!_attrs_.json 206 B
g @shared/.../cname-blosclz.b2nd 384 B
h @shared/.../cname-grok.b2nd 381 B
i @shared/.../cname-lz4.b2nd 380 B
j @shared/.../cname-zstd.b2nd 381 B
@shared/kevlar-blosc2/data/cname-
Display
Meta
Tomography
shape = (10, 2167, 2070)
dim 0 0
dim 1 0
10
0 1 2 3 4
0 65535 0 0 0 0
1 0 0 0 0 0
2 0 0 0 0 0
3 0 0 1 0 0
4 0 0 0 0 0
5 0 0 0 0 0
6 0 0 0 0 0
7 0 0 0 0 0
8 0 0 1 0 0
9 0 0 0 0 1


CATERVA2
Prompt
Roots:
@personal
@shared
@public
Datasets:
Search
Search
a @personal/kevlar.h5 1.1 GB
b @personal/la_blosclz.b2nd 484 B
c @personal/root-example.h5 23.3 KB
d @shared/kevlar-blosc2.h5 34.7 MB
e @shared/.../!_attrs_.json 206 B
f @shared/.../!_attrs_.json 206 B
g @shared/.../cname-blosclz.b2nd 384 B
h @shared/.../cname-grok.b2nd 381 B
i @shared/.../cname-lz4.b2nd 380 B
j @shared/.../cname-zstd.b2nd 381 B
@shared/kevlar-blosc2/data/cname-blosclz.
Display
Meta
Tomography
Download
Delete
shape (10, 2167, 2070) items (101.2 MB)
chunks (1, 2167, 2070) items (10.1 MB)
blocks (1, 256, 256) items (128.0 KB)
dtype uint16
nbytes 106168320 (cbytes: 0 ; cratio: 0.00)
nchunks 10
codec Codec.BLOSCLZ
filters, meta [(<Filter.BITSHUFFLE: 2>, 0)]
mtime 2025-05-26 04:44:01.879678+00:00
VLmeta (user attributes)
_ftype hdf5
_dsetname data/cname-blosclz


CATERVA2
la_blosclz = where(g < 10, g * 20000, g)
?
GO
Roots:
@personal
@shared
@public
Datasets:
Search
Search
a @personal/kevlar.h5 1.1 GB
b @personal/la_blosclz.b2nd 484 B
c @personal/root-example.h5 23.3 KB
d @shared/kevlar-blosc2.h5 34.7 MB
e @shared/.../!_attrs_.json 206 B
f @shared/.../!_attrs_.json 206 B
g @shared/.../cname-blosclz.b2nd 384 B
h @shared/.../cname-grok.b2nd 381 B
i @shared/.../cname-lz4.b2nd 380 B
j @shared/.../cname-zstd.b2nd 381 B
@shared/kevlar-blosc2/data/cname-blosclz.b2nd
Display
Meta
Tomography
0
2070 x 2167 (original size)

la_blosclz = where(g < 10, g * 20000, g)

Roots:

- [x] @personal
- [x] @shared
- [ ] @public

Datasets:

Search | Search

a	@personal/kevlar.h5	1.1 GB
b	@personal/la_blosclz.b2nd	484 B
c	@personal/root-example.h5	23.3 KB
d	@shared/kevlar-blosc2.h5	34.7 MB
e	@shared/.../!_attrs_.json	206 B
f	@shared/.../!_attrs_.json	206 B
g	@shared/.../cname-blosclz.b2nd	384 B

Roots:

- [x] @personal
- [x] @shared
- [ ] @public

Datasets:

Search | Search

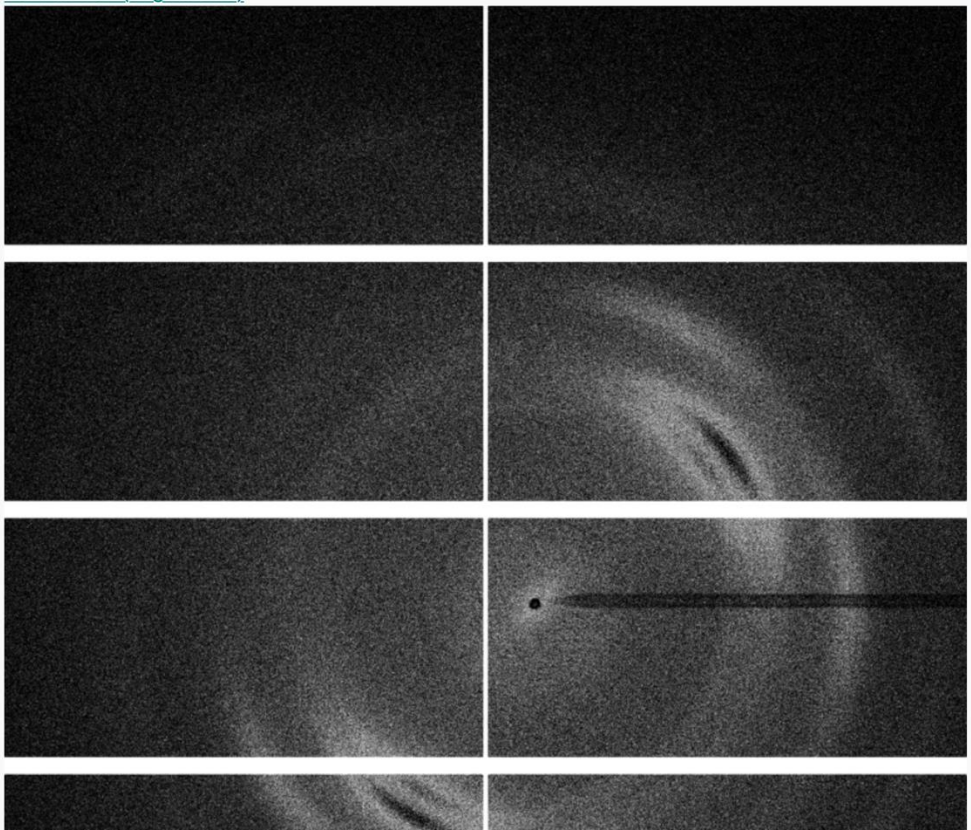
a	@personal/kevlar.h5	1.1 GB
b	@personal/la_blosclz.b2nd	484 B
c	@personal/root-example.h5	23.3 KB
d	@shared/kevlar-blosc2.h5	34.7 MB
e	@shared/.../!_attrs_.json	206 B
f	@shared/.../!_attrs_.json	206 B
g	@shared/.../cname-blosclz.b2nd	384 B
h	@shared/.../cname-grok.b2nd	381 B
i	@shared/.../cname-lz4.b2nd	380 B
j	@shared/.../cname-zstd.b2nd	381 B

@personal/la_blosclz.b2nd

Display | Meta | Tomography

2

2070 x 2167 (original size)



Caterva2 is Free Software

- It comes under the provisions of the Affero GPL
- Everyone can modify/contribute to it (whenever the license conditions are met)

Code available at
https://github.com/ironArray/Caterva2

Cat2Cloud: join the beta program!

- ironArray's Caterva2 in the cloud
- Have access to the developers
- Usability hints
- Suggest plugins
- It is free!

https://ironarray.io/cat2cloud

Conclusion

Progress made in integrating Blosc2 with HDF5

The Blosc2/ironArray teams have been working hard at:

- Implemented **native support for Blosc2 in HDF5**
- Plugins for **High Throughput JPEG 2000**
- **Caterva2**, making Blosc2/HDF5 data generally available with easy and efficiency.

Blosc2: a highly efficient and flexible tool for
compressing your data, your way

The future: better integration of Blosc2 and HDF5?

Take advantage of the structural machinery in HDF5 (Groups) to better integrate hierarchical storage for Blosc2/Caterva2

Interested? Get in touch!

Thanks to donors & contracts!

HUAWEI

ironArray

NUMFOCUS
OPEN CODE = BETTER SCIENCE

python SOFTWARE FOUNDATION

LEAPS INNOVATION

ESRF
The European Synchrotron

Google

Jeff Hammerbacher

Without them, we could not have possibly put Blosc2 into production status: Blosc2 2.0.0 came out in June 2021; now at 2.11.3.

Thanks! Questions?










contact@ironarray.io

Compress Better, Compute Bigger, Share Faster