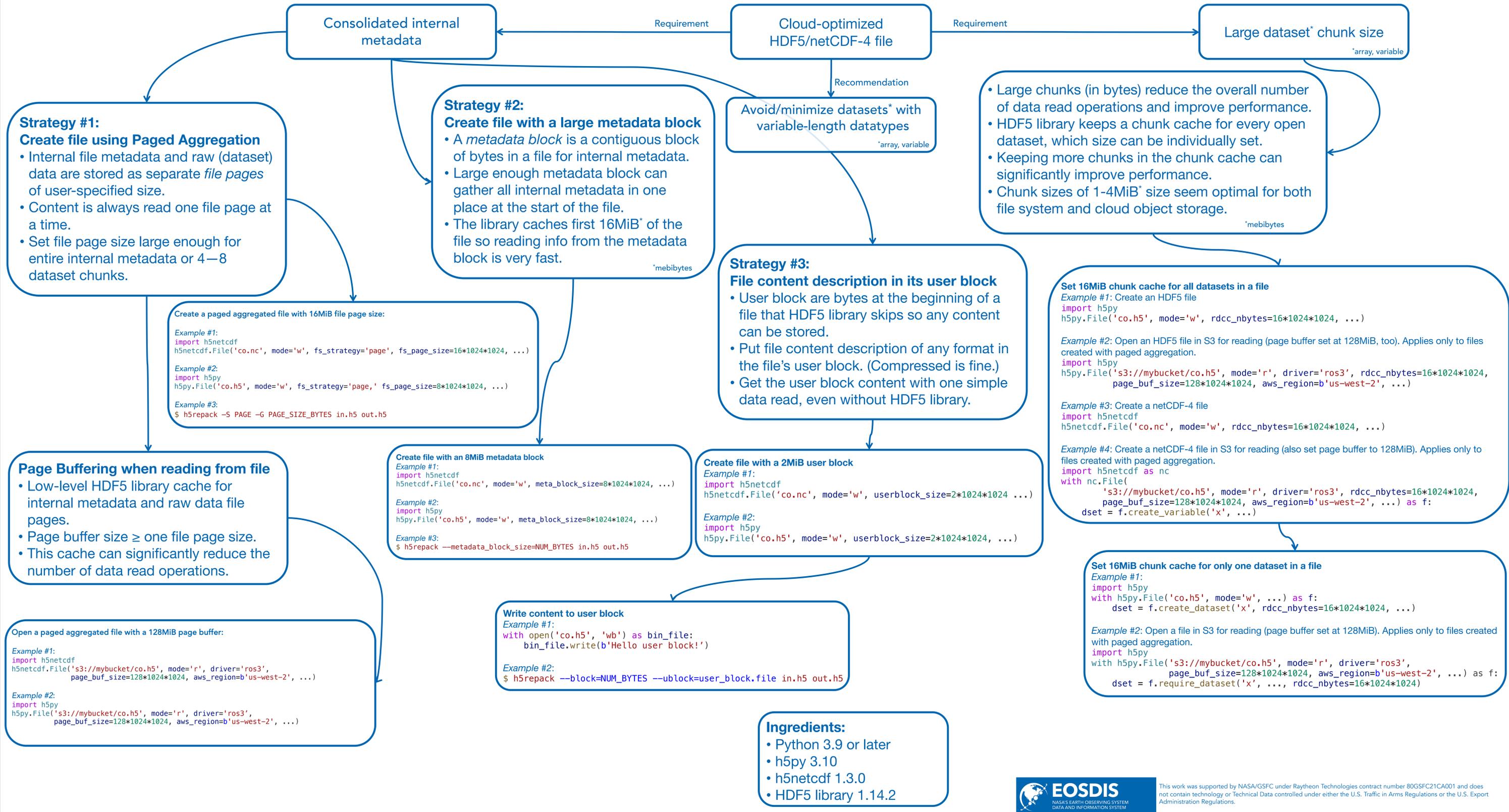


Strategies and Software for Optimizing HDF5/netCDF-4 Files in the Cloud

Aleksandar Jelenak, Dana Robinson (NASA EED-3/HDF Group)



Strategy #1: Create file using Paged Aggregation

- Internal file metadata and raw (dataset) data are stored as separate *file pages* of user-specified size.
- Content is always read one file page at a time.
- Set file page size large enough for entire internal metadata or 4–8 dataset chunks.

Create a paged aggregated file with 16MiB file page size:

```

Example #1:
import h5netcdf
h5netcdf.File('co.nc', mode='w', fs_strategy='page', fs_page_size=16*1024*1024, ...)

Example #2:
import h5py
h5py.File('co.h5', mode='w', fs_strategy='page', fs_page_size=8*1024*1024, ...)

Example #3:
$ h5repack -S PAGE -G PAGE_SIZE_BYTES in.h5 out.h5
  
```

Page Buffering when reading from file

- Low-level HDF5 library cache for internal metadata and raw data file pages.
- Page buffer size \geq one file page size.
- This cache can significantly reduce the number of data read operations.

Open a paged aggregated file with a 128MiB page buffer:

```

Example #1:
import h5netcdf
h5netcdf.File('s3://mybucket/co.h5', mode='r', driver='ros3',
page_buf_size=128*1024*1024, aws_region='us-west-2', ...)

Example #2:
import h5py
h5py.File('s3://mybucket/co.h5', mode='r', driver='ros3',
page_buf_size=128*1024*1024, aws_region='us-west-2', ...)
  
```

Strategy #2: Create file with a large metadata block

- A *metadata block* is a contiguous block of bytes in a file for internal metadata.
- Large enough metadata block can gather all internal metadata in one place at the start of the file.
- The library caches first 16MiB* of the file so reading info from the metadata block is very fast.

Create file with an 8MiB metadata block

```

Example #1:
import h5netcdf
h5netcdf.File('co.nc', mode='w', meta_block_size=8*1024*1024, ...)

Example #2:
import h5py
h5py.File('co.h5', mode='w', meta_block_size=8*1024*1024, ...)

Example #3:
$ h5repack --metadata_block_size=NUM_BYTES in.h5 out.h5
  
```

Create file with a 2MiB user block

```

Example #1:
import h5netcdf
h5netcdf.File('co.nc', mode='w', userblock_size=2*1024*1024 ...)

Example #2:
import h5py
h5py.File('co.h5', mode='w', userblock_size=2*1024*1024, ...)
  
```

Write content to user block

```

Example #1:
with open('co.h5', 'wb') as bin_file:
    bin_file.write(b'Hello user block!')

Example #2:
$ h5repack --block=NUM_BYTES --ublock=user_block.file in.h5 out.h5
  
```

- Ingredients:**
- Python 3.9 or later
 - h5py 3.10
 - h5netcdf 1.3.0
 - HDF5 library 1.14.2

Large chunks (in bytes) reduce the overall number of data read operations and improve performance. HDF5 library keeps a chunk cache for every open dataset, which size can be individually set. Keeping more chunks in the chunk cache can significantly improve performance. Chunk sizes of 1–4MiB* size seem optimal for both file system and cloud object storage.

Set 16MiB chunk cache for all datasets in a file

```

Example #1: Create an HDF5 file
import h5py
h5py.File('co.h5', mode='w', rdcc_nbytes=16*1024*1024, ...)

Example #2: Open an HDF5 file in S3 for reading (page buffer set at 128MiB, too). Applies only to files created with paged aggregation.
import h5py
h5py.File('s3://mybucket/co.h5', mode='r', driver='ros3', rdcc_nbytes=16*1024*1024,
page_buf_size=128*1024*1024, aws_region='us-west-2', ...)

Example #3: Create a netCDF-4 file
import h5netcdf
h5netcdf.File('co.nc', mode='w', rdcc_nbytes=16*1024*1024, ...)

Example #4: Create a netCDF-4 file in S3 for reading (also set page buffer to 128MiB). Applies only to files created with paged aggregation.
import h5netcdf as nc
with nc.File('s3://mybucket/co.h5', mode='r', driver='ros3', rdcc_nbytes=16*1024*1024,
page_buf_size=128*1024*1024, aws_region='us-west-2', ...) as f:
    dset = f.create_variable('x', ...)
  
```

Set 16MiB chunk cache for only one dataset in a file

```

Example #1:
import h5py
with h5py.File('co.h5', mode='w', ...) as f:
    dset = f.create_dataset('x', rdcc_nbytes=16*1024*1024, ...)

Example #2: Open a file in S3 for reading (page buffer set at 128MiB). Applies only to files created with paged aggregation.
import h5py
with h5py.File('s3://mybucket/co.h5', mode='r', driver='ros3',
page_buf_size=128*1024*1024, aws_region='us-west-2', ...) as f:
    dset = f.require_dataset('x', ..., rdcc_nbytes=16*1024*1024)
  
```