The HDF Group
New VFDs and SWMR Re-Design and Re-Implementation

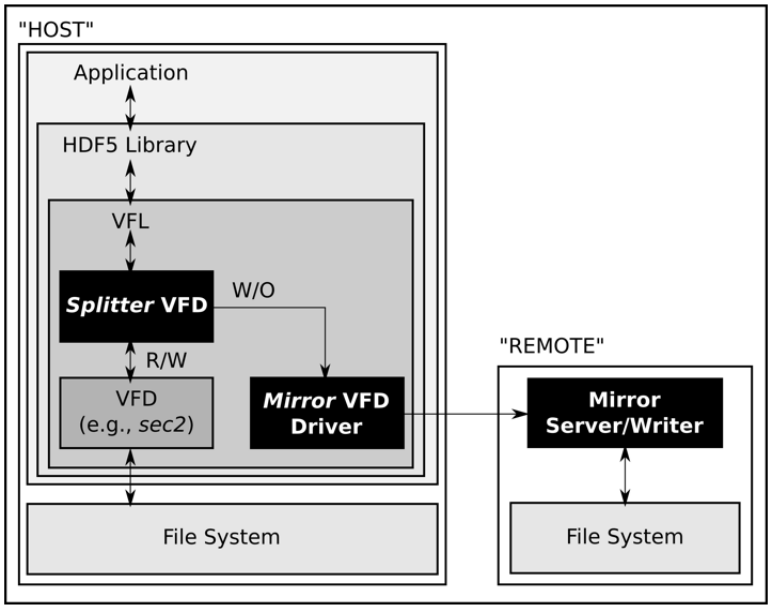
October 14, 2020

The HDF Group

John Mainzer
Principle Architect
The HDF Group

Mirror VFD

The HDF Group



Duplicates all writes to the local HDF5 file on the remote system. Mirror server must run on the remote server.

Splitter VFD relays all writes both to the local VFD and to the Mirror VFD Driver. The Mirror Driver relays writes over the net to the Mirror Writer, which writes to the remote copy of the file.

Sockets based. Not optimized for performance.

In 1.10.7, should be in 1.12.1.

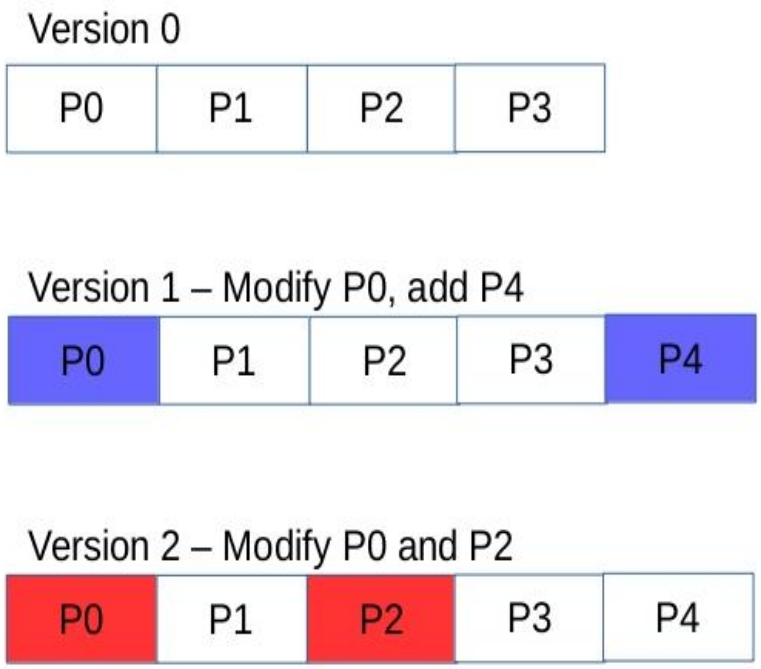
Onion VFD

The HDF Group

- Supports coarse version control and provenance management.
  - “Coarse”, because only versions as of file close are recoverable.
  - Each version is annotated with the date, ID of user, and an optional comment.
- Implemented at the VFD level -- largely transparent to the HDF5 library proper. The VFD:
  - Breaks the logical HDF5 file into pages.
  - The first time a page is modified after file open, the VFD does a copy on write, and applies further changes to the copy.
  - For each version, index maps logical page to the appropriate physical page.
- Minimal implementation in test and debug

Onion VFD – An Oversimplified Example

The HDF Group



Logical Page X	Maps to physical page Y in		
	Version 0	Version 1	Version 2
P0	0	4	6
P1	1	1	1
P2	2	2	7
P3	3	3	3
P4	N/A	5	5

SWMR Re-Design and Re-Implementation

The HDF Group

Objectives:

- Improve modularity and maintainability
- Full SWMR
- Support max time from write to visibility to readers
- Support non POSIX file systems – i.e. NFS
- Parallel SWMR

VFD SWMR – Conceptual Overview

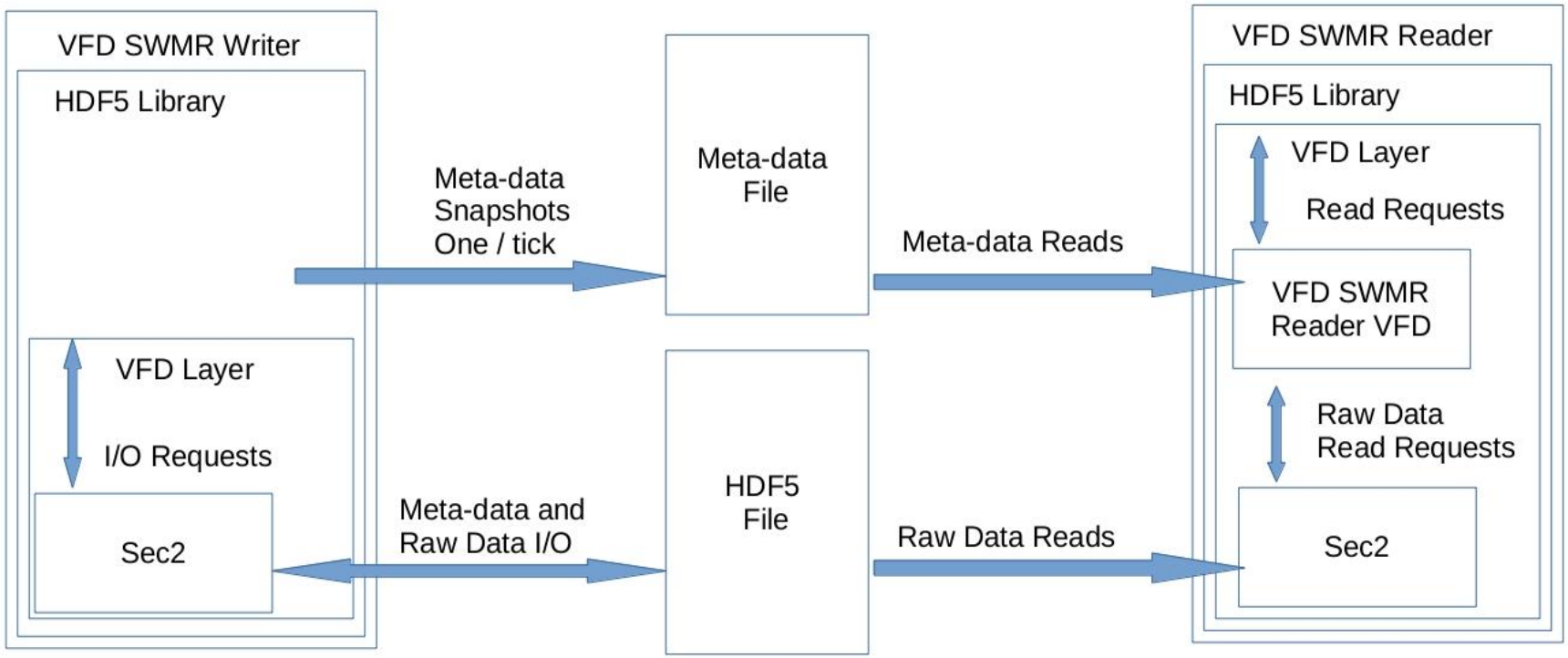
The HDF Group

An Oversimplified Cycle of Operation:

- Writer takes snapshots of HDF5 file metadata once per tick, and writes them to an auxiliary metadata file.
- Readers use a specialized VFD (Virtual File Driver) to intercept metadata read requests, and satisfy them from a snapshot in the metadata file. Check for a new snapshot once per tick.
- Snapshots expire after max_lag ticks.
- Assuming the file system can keep up, writes are visible to readers within three ticks.

VFD SWMR – An Oversimplified Diagram

The HDF Group



VFD SWMR – Design Implications

The HDF Group

Metadata snapshot design makes VFD SWMR transparent to most of the HDF5 library – only the metadata cache, page buffer, and VFD layer have to know about SWMR. Thus:

- Most elements of HDF5 just work – yielding almost "Full SWMR". Major exceptions are:
  - Variable Length Raw Data
  - Long running API calls can overrun max_lag ticks
- Upper levels of the library can ignore SWMR, reducing maintenance costs, and avoiding SWMR overhead for non-SWMR applications
- Snapshots don't require POSIX semantics, thus support for NFS is possible
- Snapshot design can also work in parallel – requires parallel page buffer.

VFD SWMR – Current Status

The HDF Group

- Friendly User release is available now
  - Only Unix / Linux for now – We are working on Windows
  - Writer flushes raw data at end of tick – We plan to make this configurable to maximize throughput
  - Only cursory benchmarks so far, with no attempt to optimize.
    - Writer overhead ranges from nil for a few large datasets, to a factor of 2 for many small datasets with two extensible dimensions.
    - No data yet on reader overhead
  - See Forum for further details
    - Announcement (https://forum.hdfgroup.org/t/announcement-from-the-hdf-group-first-alpha-release-of-vfd-swmr/7534)
    - Join the VFD SWMR mailing list (https://www.hdfgroup.org/swmr-vfd-mailing-list/)
- Aim for first production version by the end of May

THANK YOU!

Questions & Comments?

Acknowledgment:

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number DE-SC0018504.