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

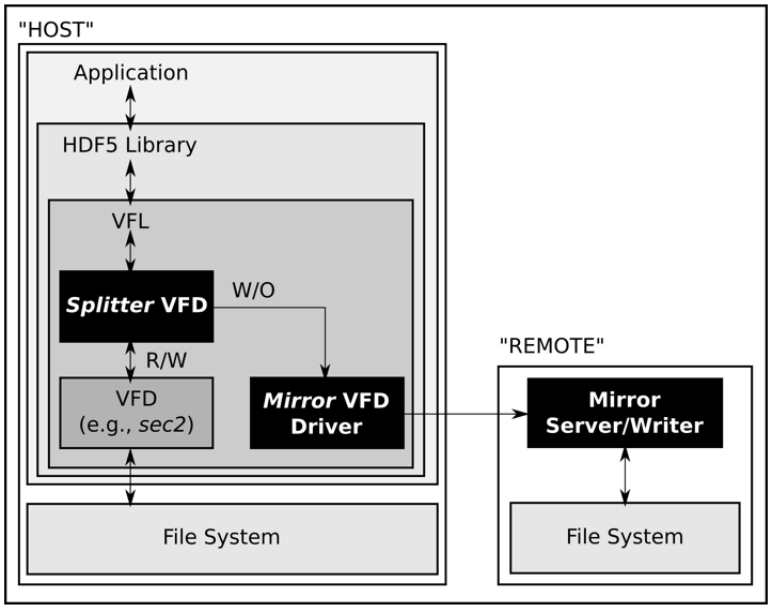
July 8, 2021

The HDF Group

John Mainzer
Principle Architect
The HDF Group

Splitter & Mirror VFDs

The HDF Group



Objective: Duplicate all writes to the local HDF5 file on a remote system.

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.

Mirror server must be running on the remote system.

Sockets based. Not optimized for performance. Factor of 6 slowdown is typical.

In 1.10.7 and 1.12.1.

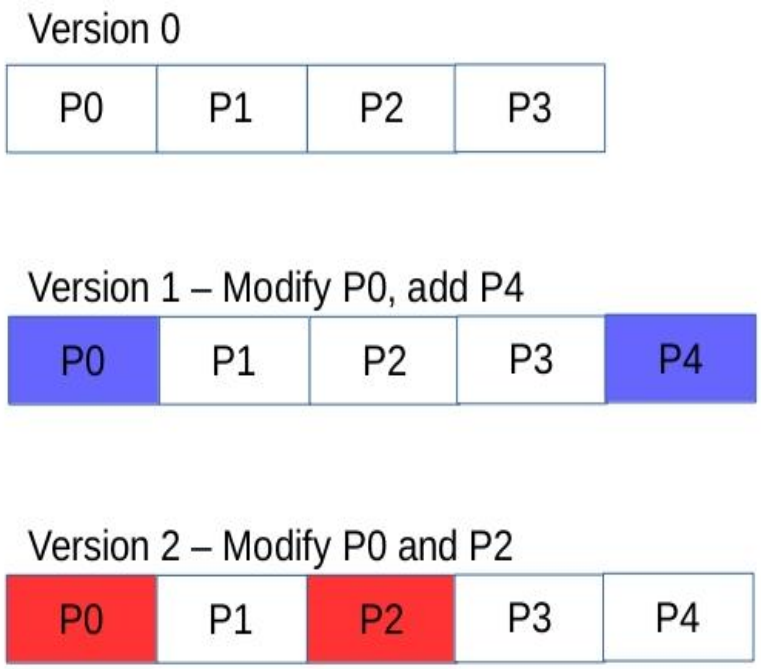
Onion VFD

The HDF Group

- Objective: support 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 of the file, index maps logical page to the appropriate physical page.
- Minimal implementation is complete. Release TBD.

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

SWMR (Single Writer Multiple Readers) allows one or more reader processes to read an HDF5 file as it is being written.

Objectives of re-design and re-implementation:

- 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

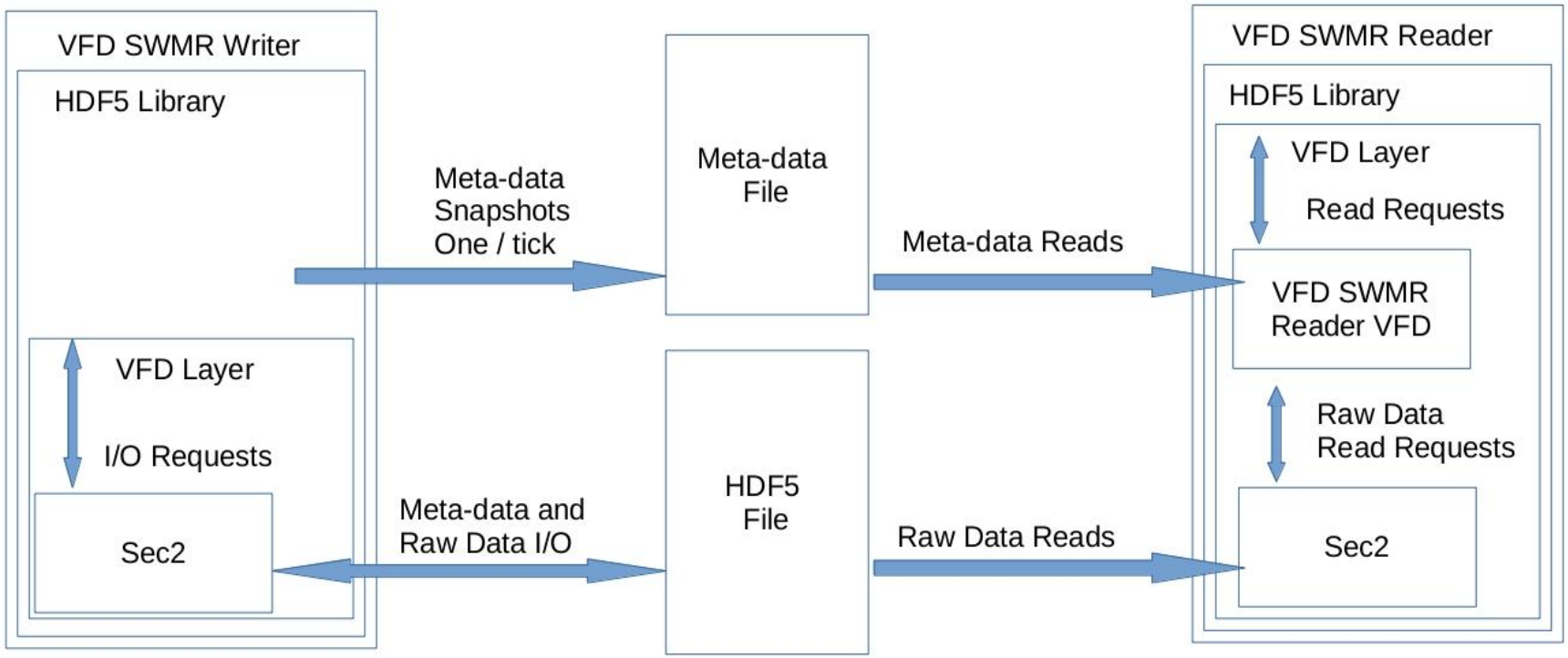
The HDF Group

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
  - Older versions of the file format can create arbitrarily large pieces of metadata, causing very large metadata files – thus only latest file format should be used with VFD SWMR.
- Upper levels of the library can ignore SWMR, reducing maintenance costs, and avoiding SWMR overhead for non-SWMR applications.

VFD SWMR – Design Implications (continued)

The HDF Group

Other implications:

- Snapshots don't require POSIX semantics, thus VFD SWMR for NFS is possible (but not implemented yet). Ditto for object stores once we have R/W VFDs that support them.
- Snapshot design can work in parallel – requires parallel page buffer.
- VFD SWMR on remote systems (via splitter & mirror VFDs) is possible (but not implemented yet).
- With minor tweaks, the snapshot design can support coarse metadata journaling for crash recovery.
- New HDF5 features should work with VFD SWMR with little or no additional development effort.

VFD SWMR – Current Status

The HDF Group

- Alpha release available now – new alpha release coming soon.
  - Only Unix / Linux / MacOS for now – We are working on Windows
  - Only serial / POSIX file systems for now. We hope to add NFS support by the end of the year.
  - Writer flushes raw data at the end of each tick – We are in the process of making this configurable to maximize throughput.
- Performance testing is in progress. No attempts at optimization so far. Some initial results shown in the following slides, but in a nut-shell:
  - Best performance with small numbers of data sets with at most one extensible dimension.
  - Group creation overhead is minimal.
  - File close can be slow.
- Aim for first production version by the end of 2021

Performance: Large Dataset Writes (initial results)

The HDF Group

Write n x 1000 x 1000 datasets (1 X 1000 x 1000 chunk size). On each cycle, extend each dataset and write one 1000 x 1000 frame. 4 KiB page size. 4 MiB page buffer. Raw data flushed on end of tick.

Number of Datasets	Regular HDF5 500 Cycles Total Dataset Write Time in Seconds	HDF5 with VFD SWMR 500 Cycles Total Dataset Write Time in Seconds	Percent Slowdown With VFD SWMR
10	32.3	29.4	-9%
100	284	318	+9%

Performance: Small Dataset Writes (initial results)

The HDF Group

WARNING: This benchmark is intentionally inefficient.
Write n x 16 x 16 datasets (1 x 16 x 16 chunks). On each cycle, extend each dataset and write one 16 x 16 frame. 4 KiB page size. 4 MiB page buffer. Raw data flushed on end of tick.

Number of Datasets	Regular HDF5 500 Cycles Total Dataset Write Time in Seconds	HDF5 with VFD SWMR 500 Cycles Total Dataset Write Time in Seconds	Percent Slowdown With VFD SWMR
100	0.87	0.93	9%
1000	11.5	12.8	9%
2000	23.0	35.2	53%
5000	70.7	149.0	111%
10000	160.9	359.1	123%

Performance: Group Creation (initial results)

The HDF Group

Create n groups in the root group. 4 KiB page size. 4 MiB page buffer.

Number of Groups	Regular HDF5 Total Group Creation Time in Seconds	HDF5 with VFD SWMR Total Group Creation Time in Seconds	Percent Slowdown With VFD SWMR
1000	0.064	0.069	7%
10000	0.516	0.500	-3%
100000	5.62	6.05	8%
1000000	83.1	97.8	18%

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.