HDF5 Subfiling: A Scalable Approach to Exascale I/O

M. Scot Breitenfeld
Jordan Henderson
HDF5 User Group Meeting, 2024

Data Aggregation Challenges and Benefits

The HDF Group

- Benefits (typically for large node counts)
  - Better use of parallel I/O subsystems, such as node-local storage
  - By leveraging parallel I/O subsystems, we can effectively mitigate locking and contention issues, leading to substantial performance enhancements, especially at larger processor counts compared to a single-file approach
  - Reduces the complexity of *file-per-process*

- *Challenges*
  - It may still be burdensome working with many subfiles
    - Do the readers understand the data layout and organization
    - May need to combine the files into a valid format
      - can be expensive and negate any benefits from aggregation
    - Hiding data processing during computation to avoid with-out impacting compute performance
  - Unknown at what node count does aggregation start to benefit

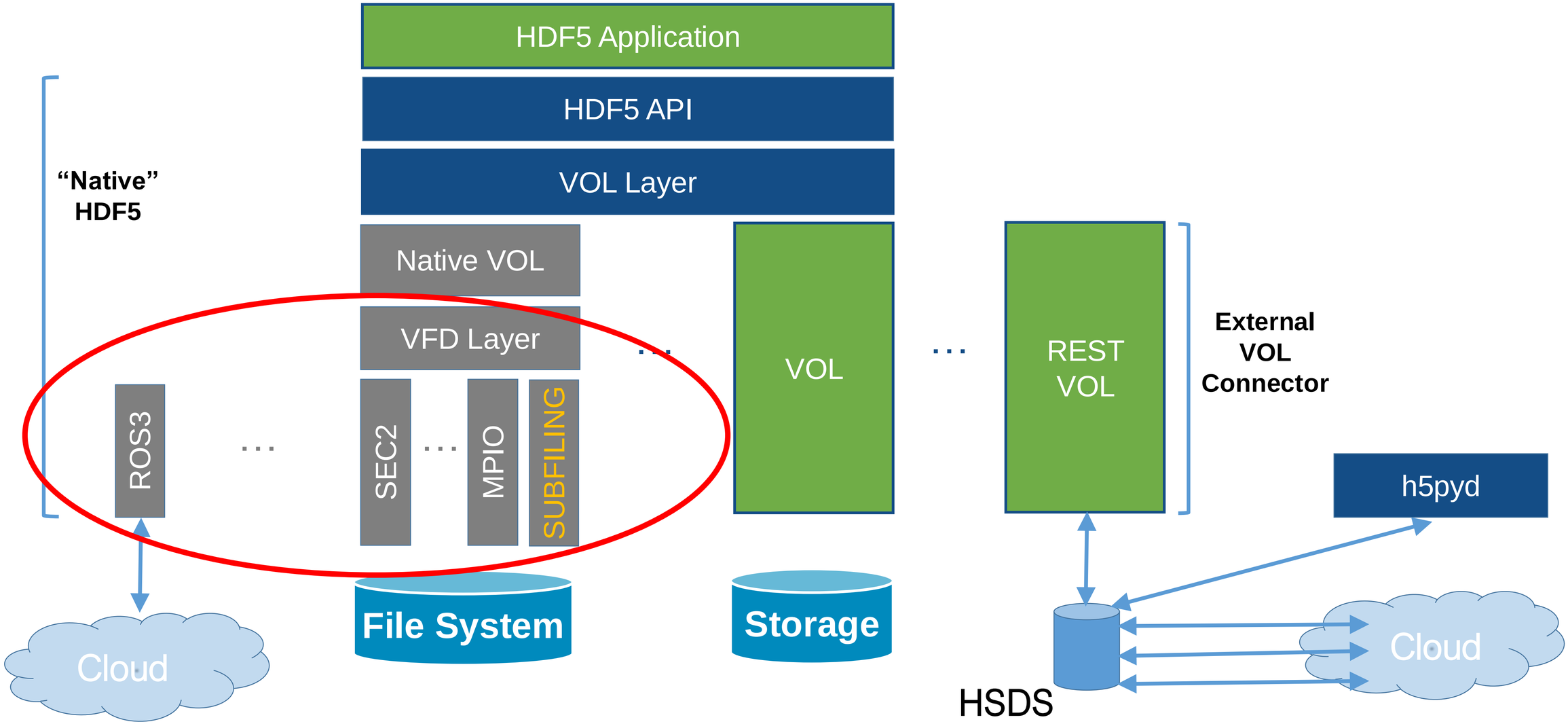
Quick Recap

The HDF Group

- HDF5 Virtual File Drivers (VFDs) allow users to define a mapping between HDF5 address space and underlying storage. Examples:
  - Sec2 VFD – Uses POSIX I/O on a single file
  - Core VFD – I/O directly on memory
  - MPI IO VFD – Use MPI IO for parallel I/O
  - For more, see File driver property list functions (H5P) in the Reference Manual

  - Set on an HDF5 File Access Property List by generic **`H5Pset_driver`** call, or by specialized driver-specific call

HDF5 1.14 Library Architecture



Subfiling VFD

Availability and Requirements

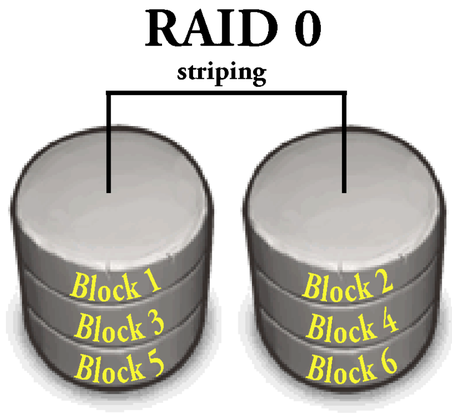
The HDF Group

- Introduced in HDF5 1.14.0
- HDF5 must be built with parallel support enabled
  - Plus, must enable subfiling when building HDF5. It's not enabled by default
- C11-capable compiler support is required

- **An application must use *MPI_Init_thread* and requires *MPI_THREAD_MULTIPLE* level of threading support by MPI implementation**

- A subfiling User's Guide is available

What is it?

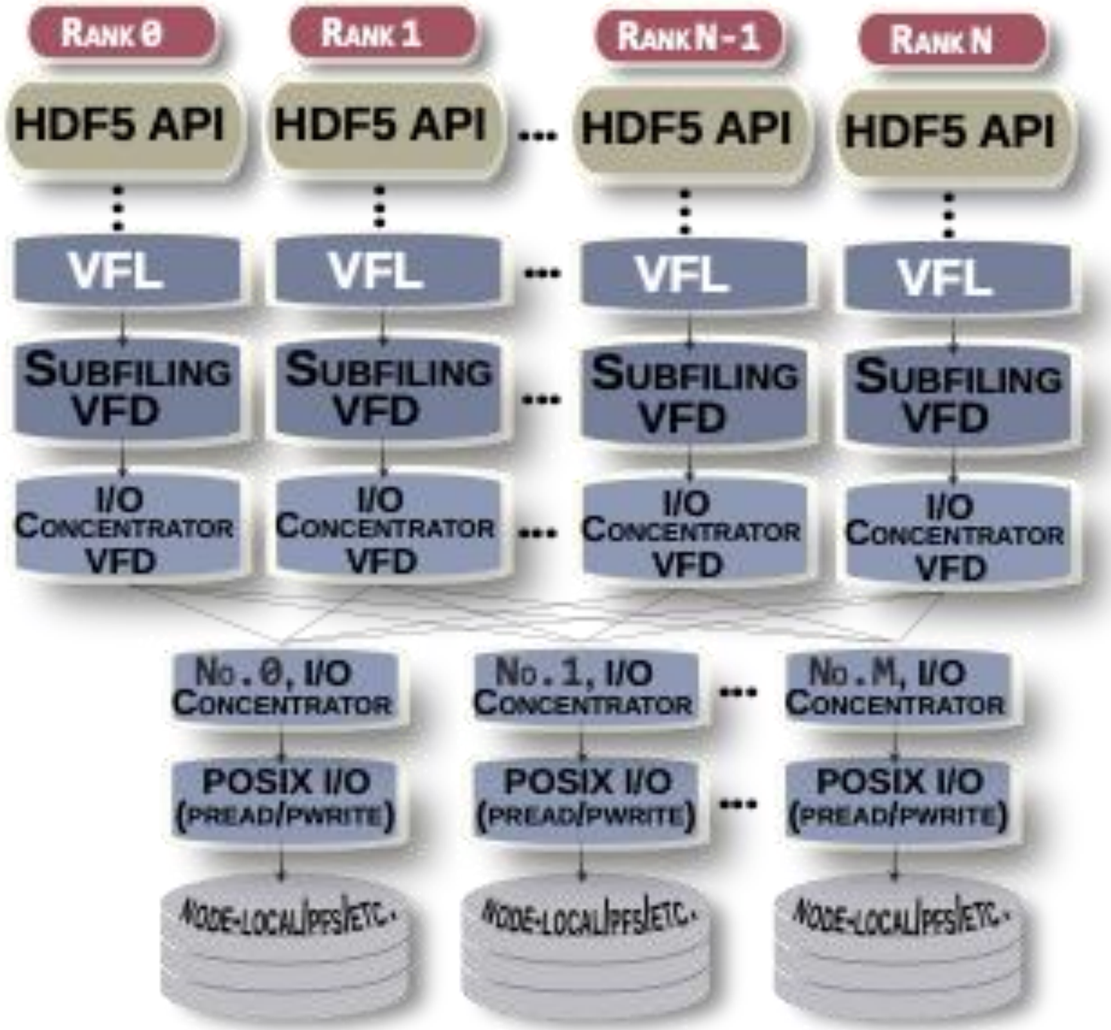


- An MPI-based parallel file driver used to split an HDF5 file across a collection of subfiles in equally sized data segment stripes.
  - Data stripe size is the amount of data (in bytes) that can be written to a subfile before data is placed in the next subfile in a round-robin (default) fashion
  - Defaults to 1 subfile per machine node with 32MiB data stripes
- Subfiling is a compromise between file-per-process (*fpp*) and a single shared file (*ssf*)
  - Minimize the locking issues of *ssf* approach
  - Avoid some complexity and reduce total number of files compared to *fpp* approach
  - Designed to be flexible and configurable for different machines

What is it? (continued)

The HDF Group

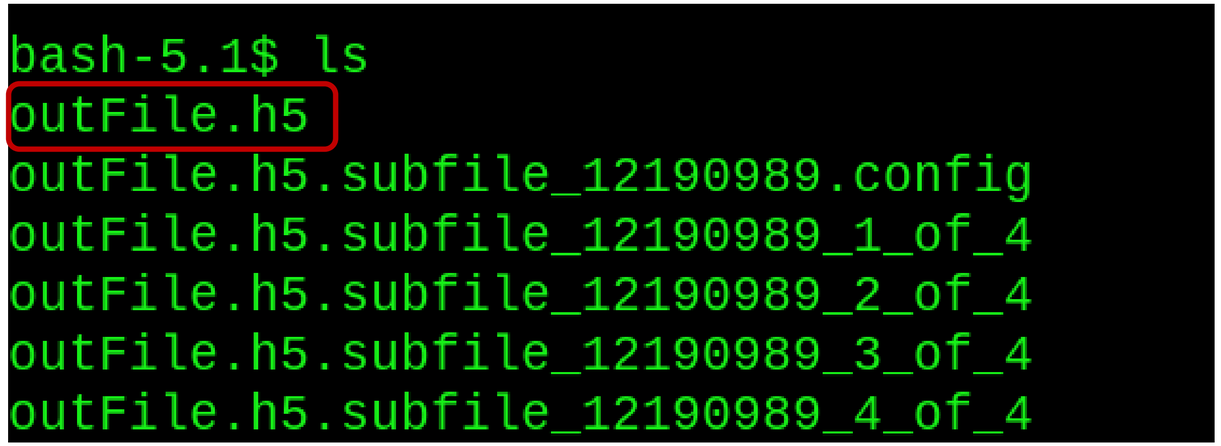
- Uses "I/O concentrators" - a subset of available MPI ranks that control subfiles and operate I/O worker thread pools.
  - N-to-1 mapping from subfiles -> I/O concentrator ranks
  - I/O from non-I/O-concentrator MPI ranks is forwarded to the appropriate I/O concentrator based on offset in the logical HDF5 file
  - Default: Subfiles are assigned round-robin across the available I/O concentrator ranks



Subfiling Output Files per Logical HDF5 File

The HDF Group

- HDF5 stub file
  - Appears as a normal HDF5 file; only contains HDF5 superblock information and subfiling parameter information
  - Useful for compatibility with HDF5 applications that read initial bytes of file, e.g., CGNS, NetCDF4
  - Inode value of stub file used to generate unique filenames for configuration file and subfiles



Subfiling Output Files per Logical HDF5 File

The HDF Group

Subfiling configuration text file

- A simple configuration file detailing the subfiling parameters for an existing file
- Validated against subfiling parameters stored in HDF5 stub file once logical HDF5 file has been opened
- Useful for external tooling to get subfiling parameter information

```
bash-5.1$ ls
outFile.h5
outFile.h5.subfile_12190989.config
outFile.h5.subfile_12190989_1_of_4
outFile.h5.subfile_12190989_2_of_4
outFile.h5.subfile_12190989_3_of_4
outFile.h5.subfile_12190989_4_of_4
```

```
stripe_size=1048576
aggregator_count=4
subfile_count=4
hdf5_file=/home/jhenderson/subfiling/outFile.h5
subfile_dir=/home/jhenderson/subfiling
outFile.h5.subfile_12190989_1_of_4
outFile.h5.subfile_12190989_2_of_4
outFile.h5.subfile_12190989_3_of_4
outFile.h5.subfile_12190989_4_of_4
```

Subfiles

Contains all the file data, including superblock information duplicated in HDF5 stub file

Subfiling

- Subfiling file driver is set on a File Access Property List

```
1.     plist_id = H5Pcreate(H5P_FILE_ACCESS);
2.     status = H5Pset_fapl_subfiling(plist_id, vfd_config);
3.     file_id = H5Fcreate(H5FILE_NAME, H5F_ACC_TRUNC, H5P_DEFAULT, plist_id);
4.     H5Pclose(plist_id);
```

Environment variables control options:

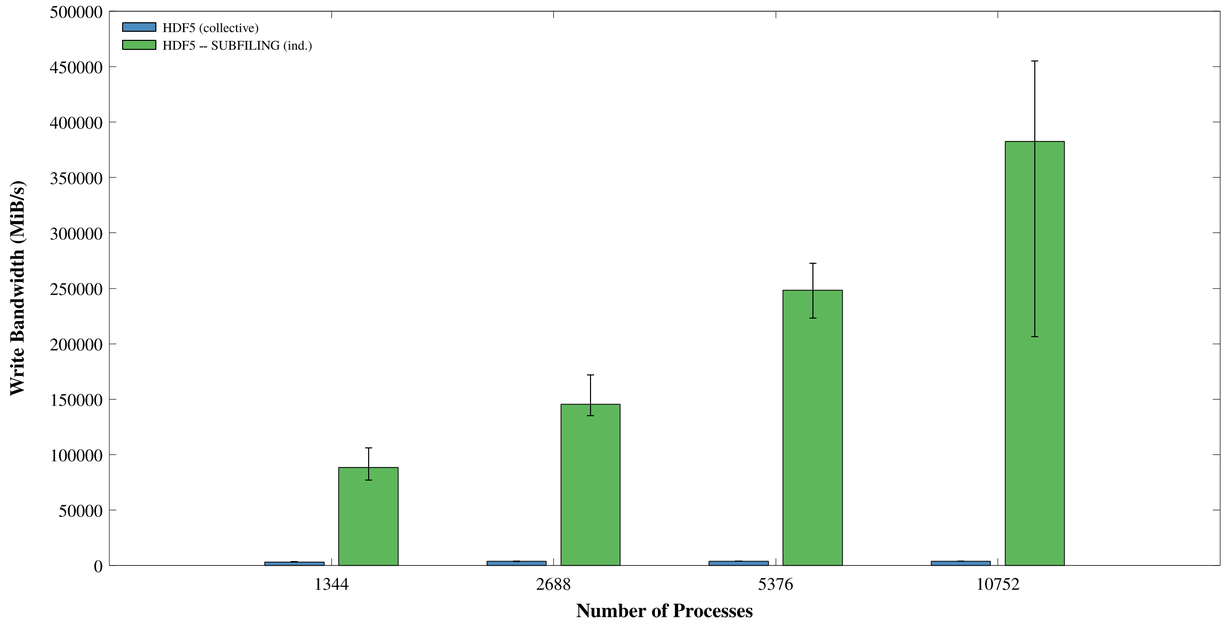
- **H5FD_SUBFILING_IOC_PER_NODE**– Number of I/O concentrators per node.
- **H5FD_SUBFILING_STRIPE_SIZE** – Maximum contiguous block of data that can be written to a single I/O Concentrator before moving on to the next IOC.
- **H5FD_IOC_THREAD_POOL_SIZE** – Sets the number of I/O Concentrator helper threads. **The default is four pool threads**.
- **H5FD_SUBFILING_CONFIG_FILE_PREFIX** — Sets the prefix of the configuration file. Useful when using node-local storage.
- **H5FD_SUBFILING_SUBFILE_PREFIX** – Sets the prefix for the subfiles. Useful when using node-local storage

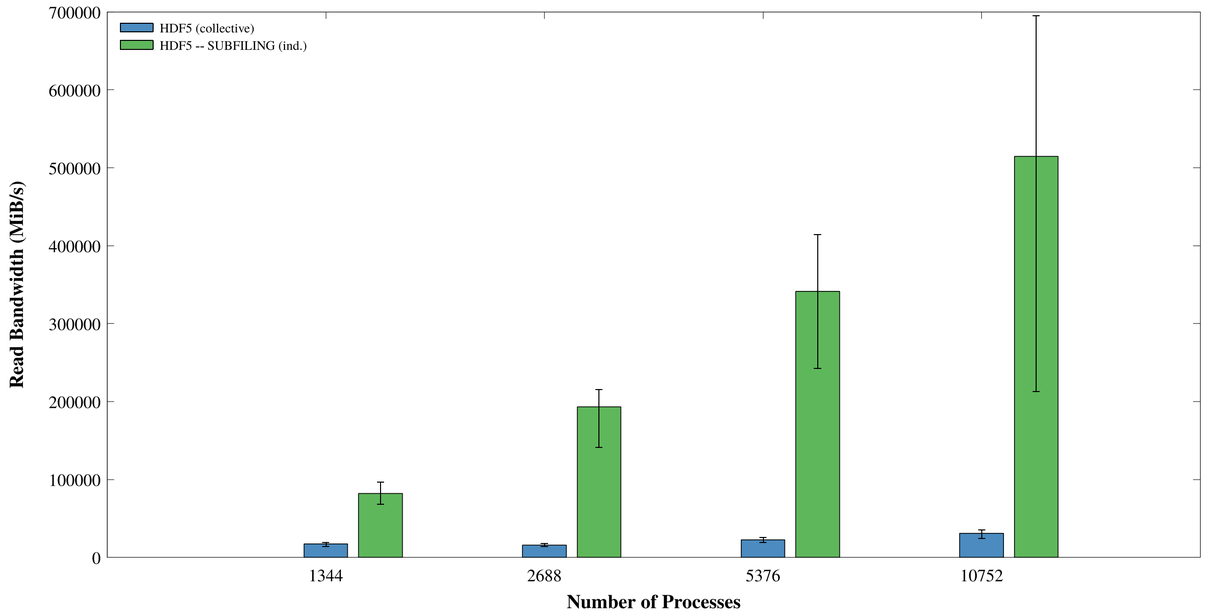
Performance Results

Subfiling – IOR on Summit (OLCF)

The HDF Group

Number of Ranks	File Size
1344	42GiB
2688	84 GiB
5376	168 GiB
10752	336 GiB

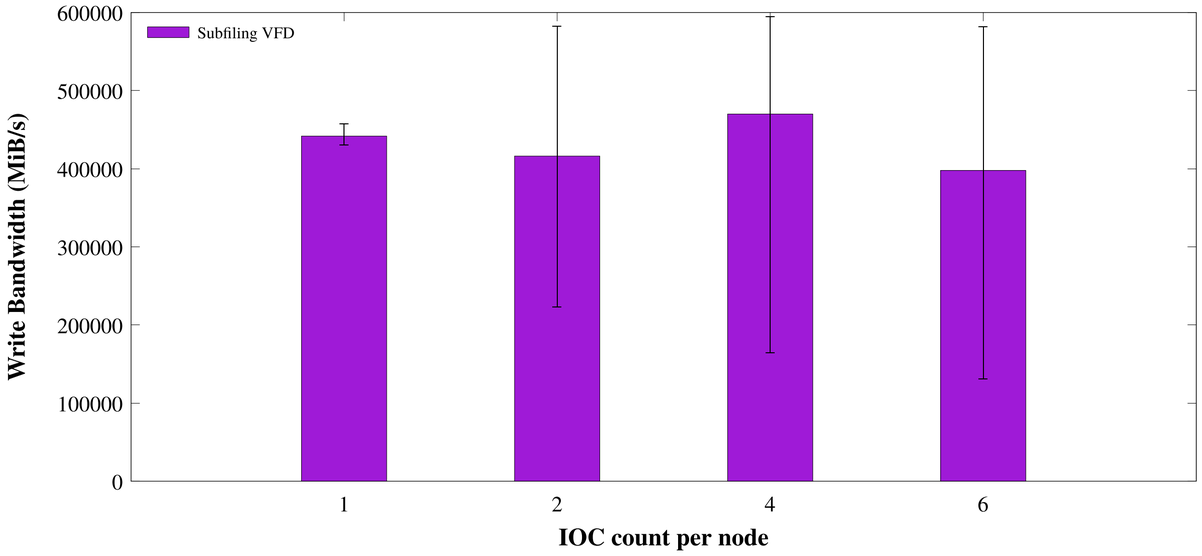


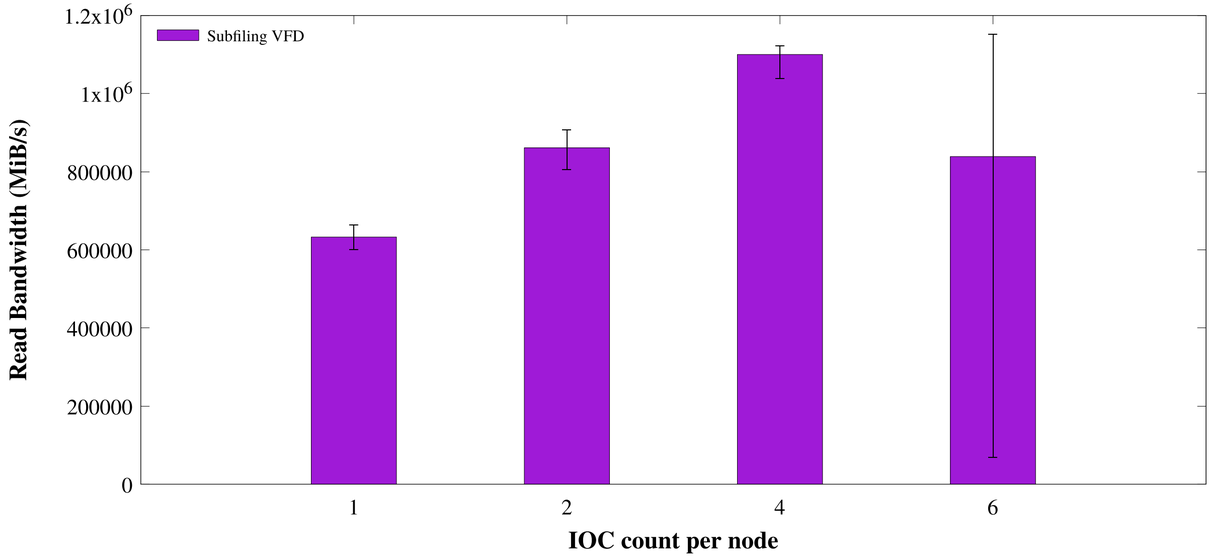


Subfiling – IOR on Summit (OLCF)

The HDF Group

Effects of the number of IOC, 10752 ranks

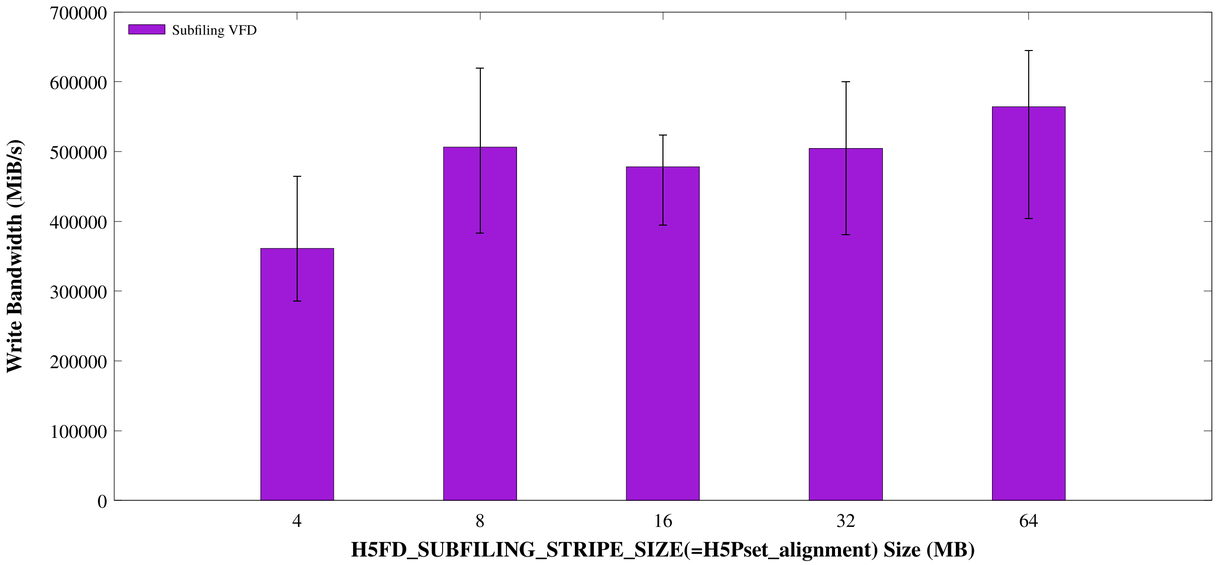


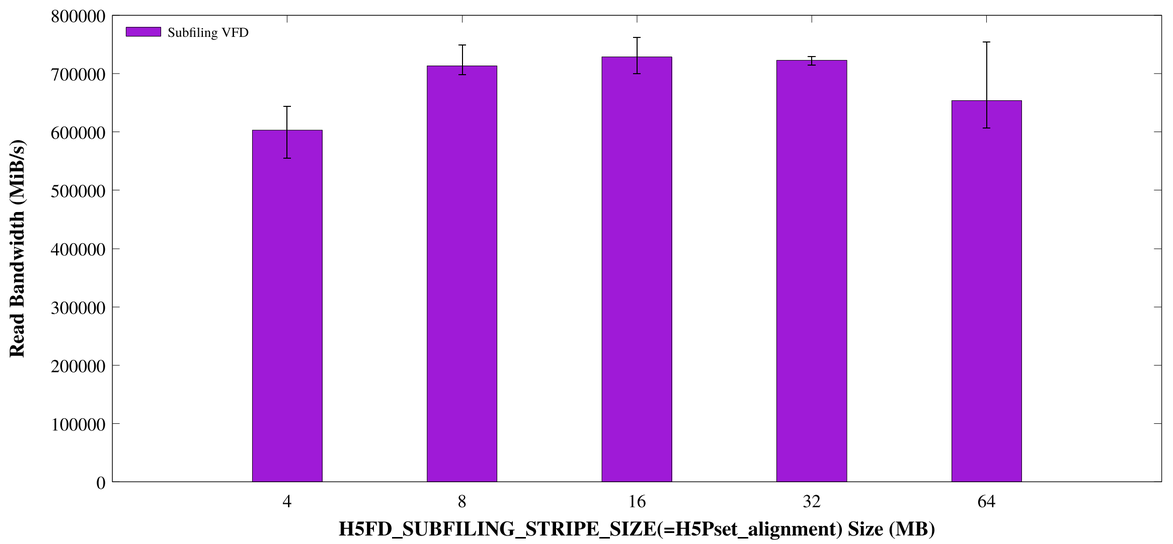


Subfiling – IOR on Summit (OLCF)

The HDF Group

Effects of stripe size, 10752 ranks

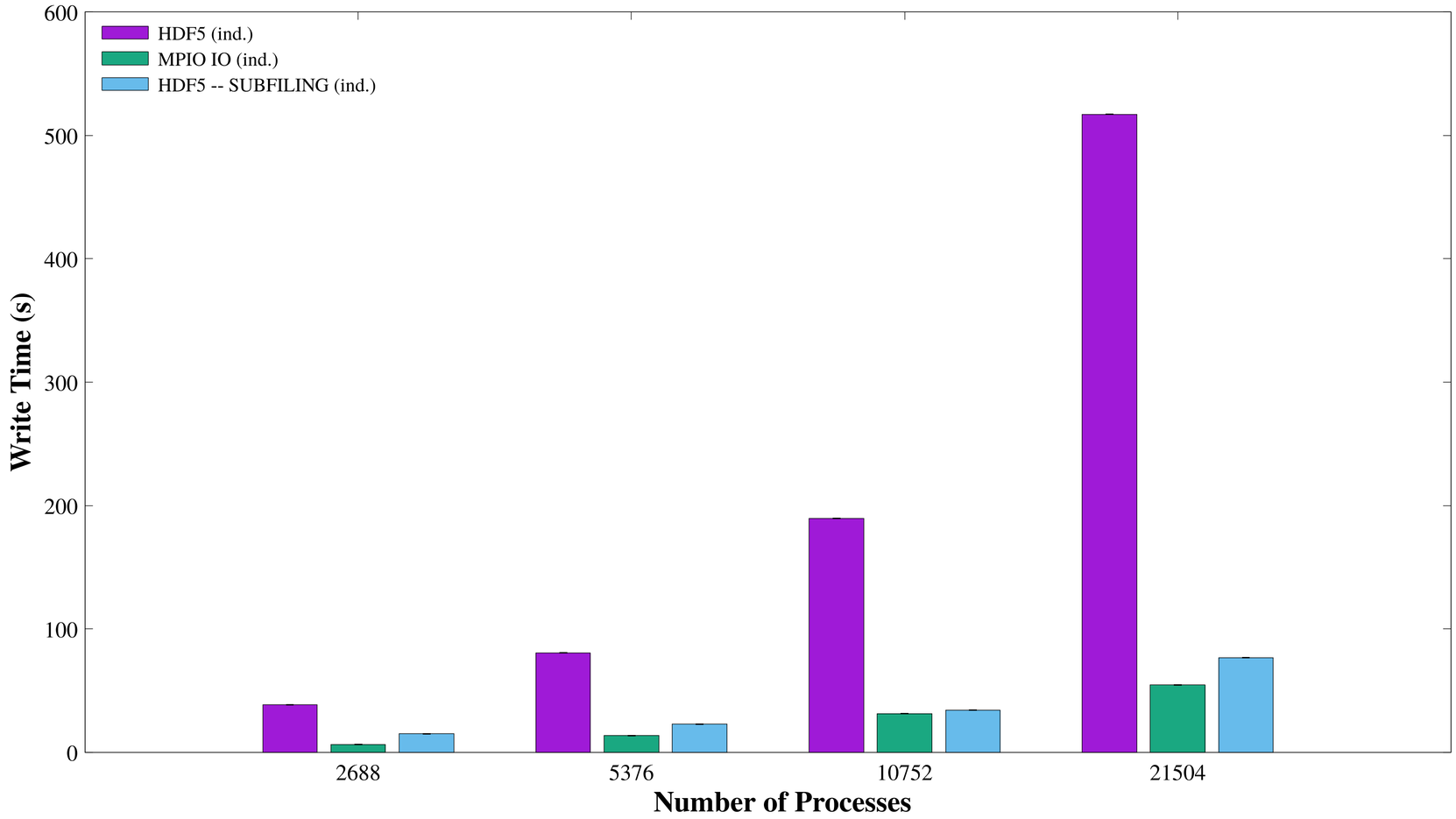




Subfiling – HACC on Summit (OLCF)

- HACC (Hardware Accelerated Cosmology Code) is an N-body code.
- Weak scaling, default (16 MiB stripe size)

Number of Ranks	File Size
1344	11 GiB
2688	21 GiB
5376	54 GiB
10752	106 GiB
21504	211 GiB
43008	1.7 TiB

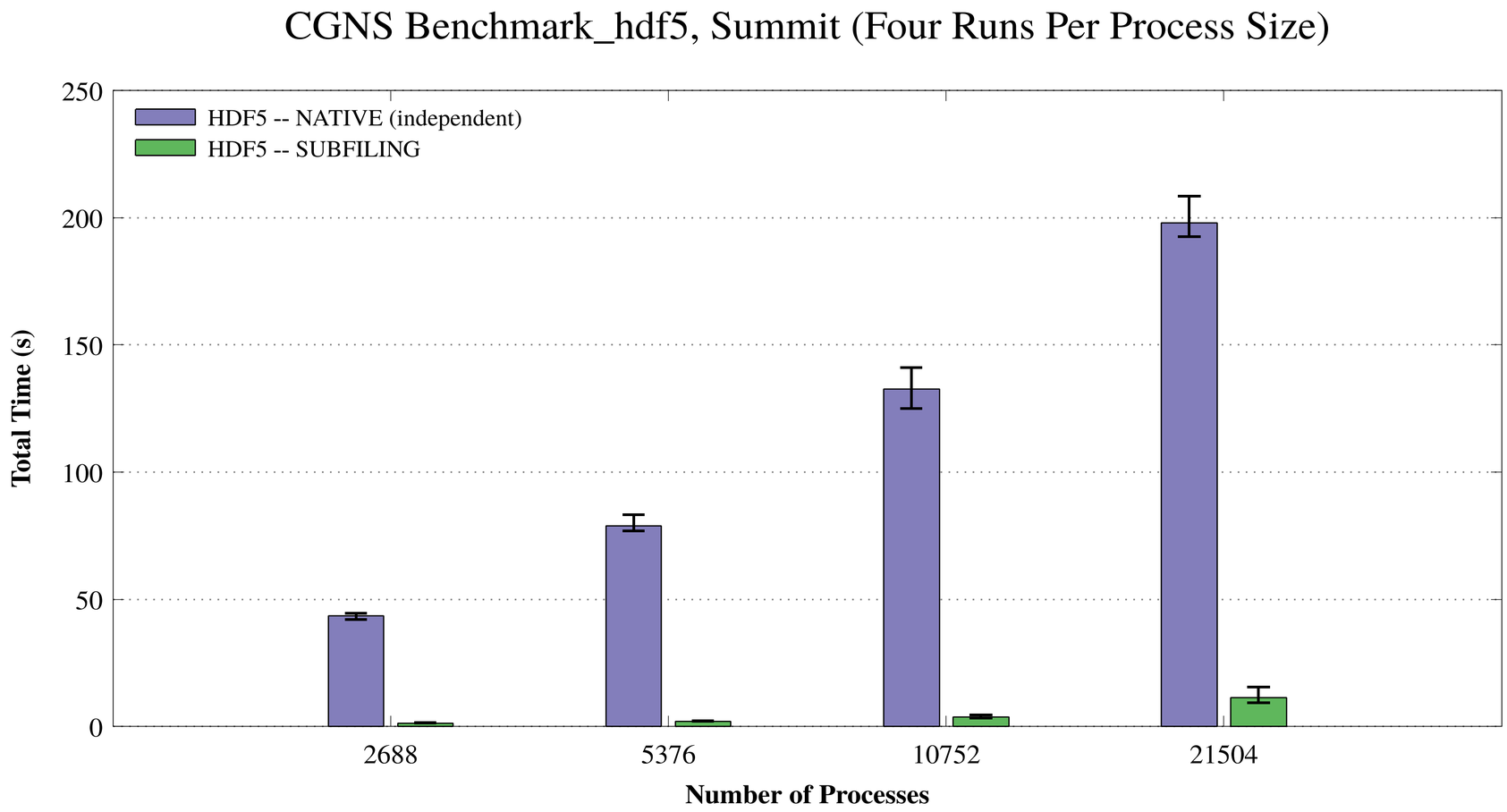


Subfiling – CGNS [1] on Summit (OLCF)

The HDF Group

- Default Subfiling settings.

Number of Ranks	HDF5 File Size
21504	53 GiB
10752	27 GiB
5376	14 GiB
2688	6.6 GiB



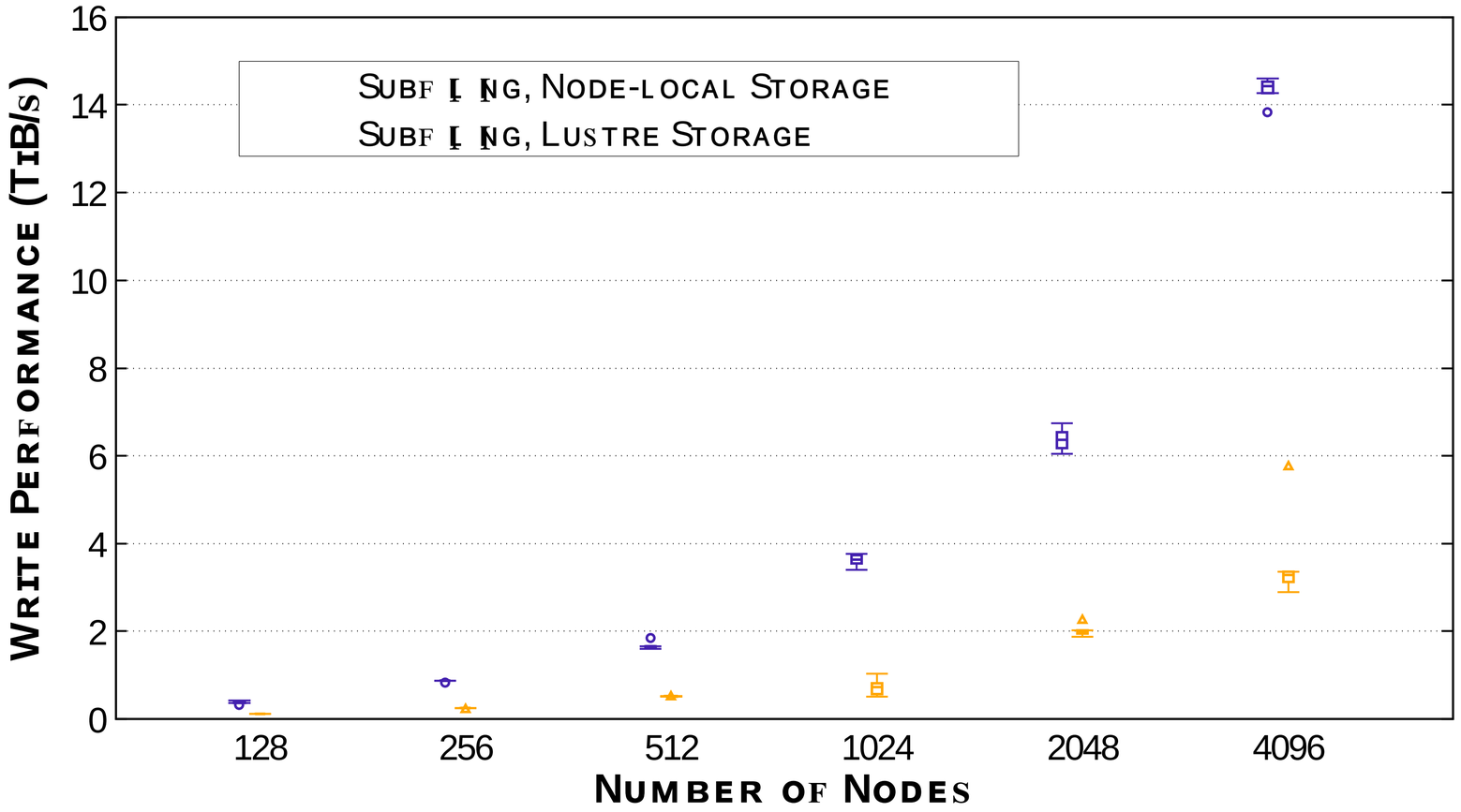
[1] CGNS = Computational Fluid Dynamics (CFD) General Notation System, cgns.org

Subfiling – ExaMPM [1] (Cabana [2]) on Frontier (OLCF)

The HDF Group

- GPU computation engine
  - Kokkos is used to transfer memory between GPU and CPUs
- Subfilings *pwrite* throughput for 4096 nodes

Number of Nodes	Size (GiB)	
	Per Output	Total
128	122	610
256	195	975
512	482	2410
1024	981	4905
2048	1950	9750
4096	2083	10415



[1] https://github.com/ECP-copa/ExaMPM , [2] https://github.com/ECP-copa/Cabana

H5fuse script

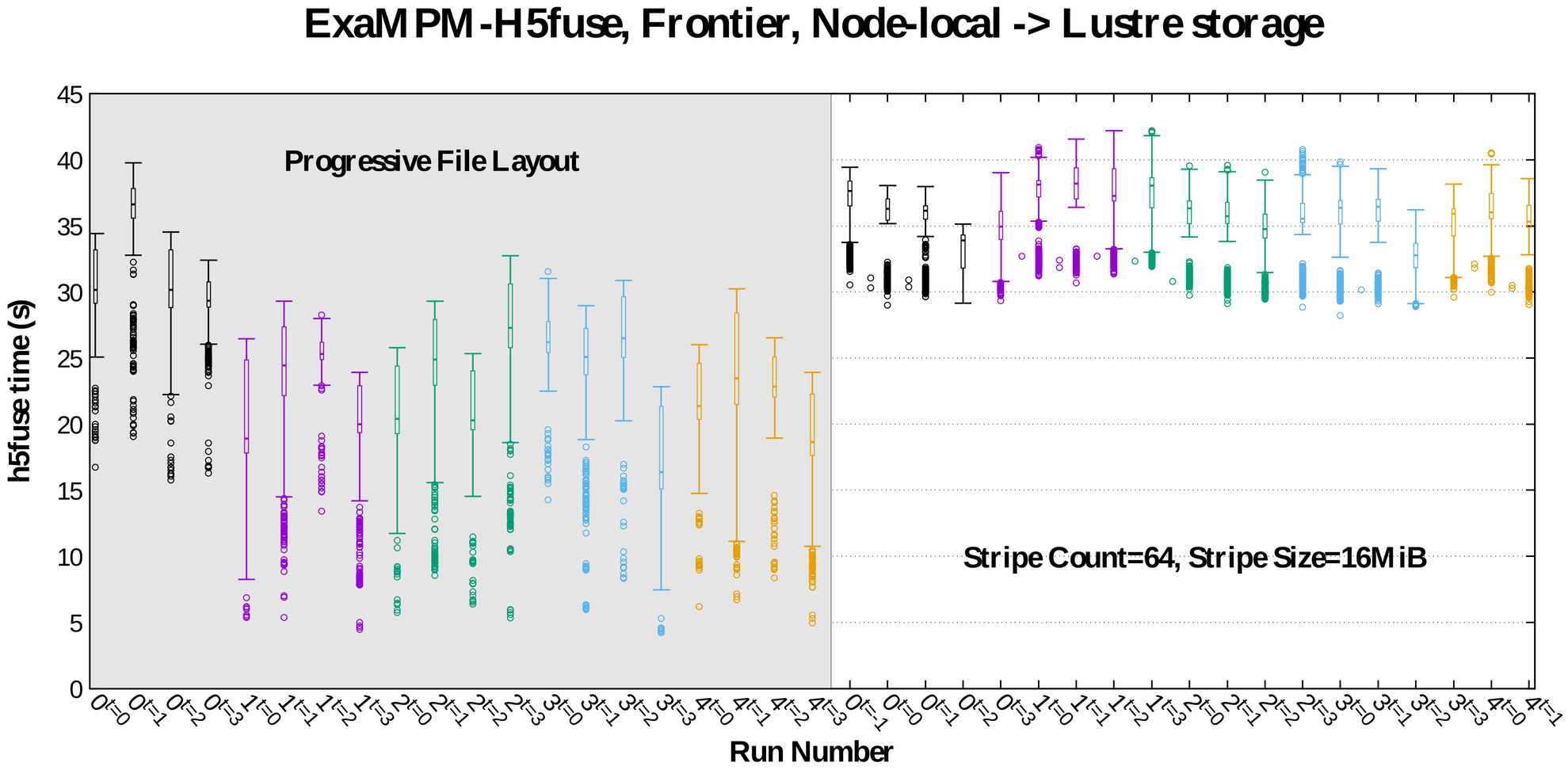
- https://github.com/HDFGroup/hdf5/blob/develop/utils/subfiling_vfd/h5fuse.in
- Reads a Subfiling VFD configuration file and fuses the subfiles back together into a single HDF5 file using dd
- Installed under '*bin*' directory of HDF5 installation as '*h5fuse*'

```
~/packages/cgns/src/ptests (subfiling) $ ./h5fuse -v
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_1_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=0 seek=0 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_1_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=1 seek=2 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_1_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=2 seek=4 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_1_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=3 seek=6 conv=notrunc
...
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_2_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=20 seek=41 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_2_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=21 seek=43 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_2_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=22 seek=45 conv=notrunc
 dd count=1 bs=8388608 if=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns.subfile_96536974_2_of_2 of=/home/brtnfld/packages/cgns/src/ptests/benchmark_000004.cgns skip=23 seek=47 conv=notrunc
COMPLETION TIME = 0.450 s
```

- Currently exploring calling *h5fuse* from within an application
  - Hide during the compute stage the single HDF5 file creation
  - Uses prototype API, which returns the the mapping of subfiles to mpi ranks.

H5fuse Performance, 1024 subfiles, ~900 GiB Total

- *H5fuse performance*
  - Combining node-local subfiles into the single HDF5 file on Lustre
  - Removing the subfiles after fused completed



20

THANK YOU!

Questions & Comments?

Acknowledgments

Some of this research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.