HDF5 Subfiling VFD

September 30, 2022

The HDF Group

Jordan Henderson
Scot Breitenfeld
The HDF Group

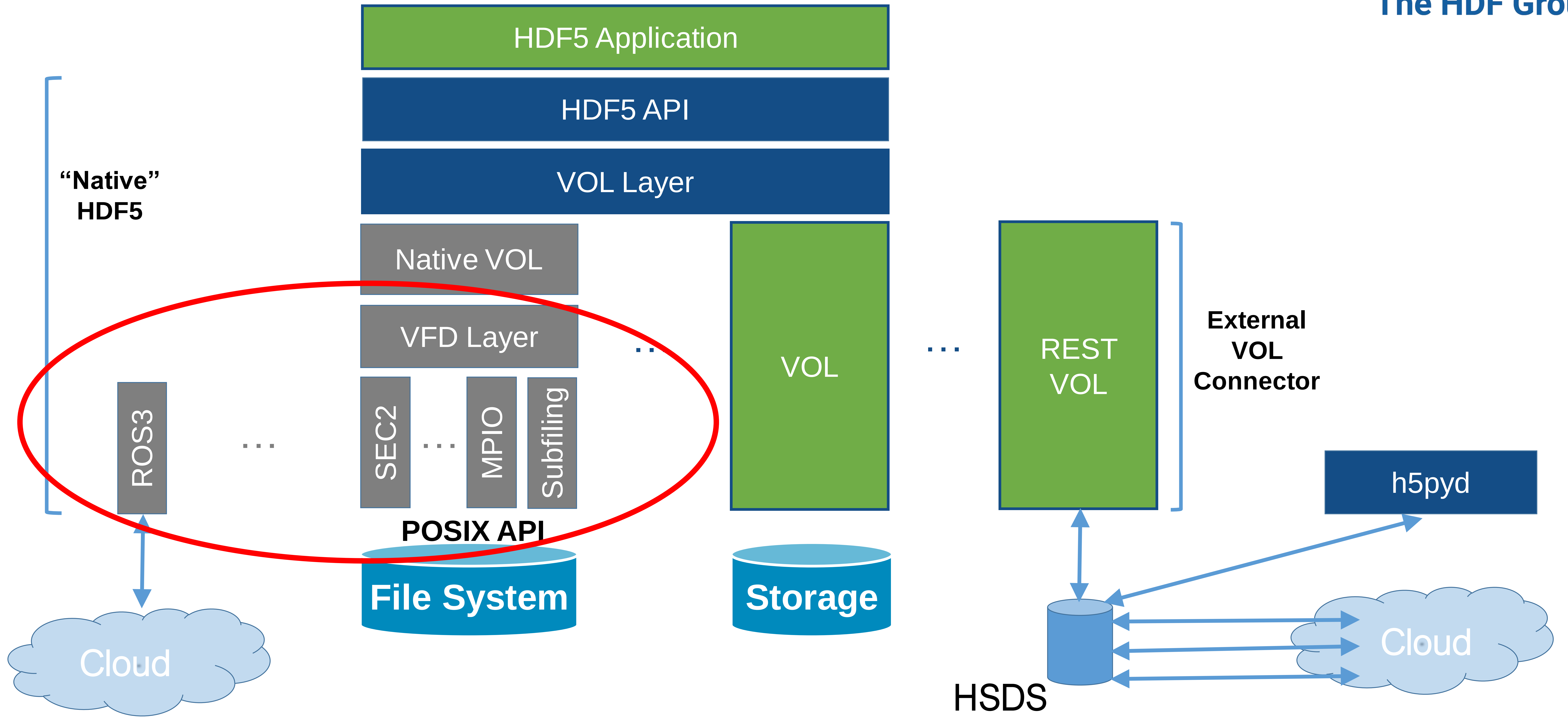
Quick Recap

- HDF5 Virtual File Drivers (VFDs) allow users to define mapping between HDF5 address space and underlying storage
  - Sec2 VFD – Uses POSIX I/O on a single file
  - Core VFD – I/O directly on memory
  - Multi/Family VFDs – Data/metadata written to separate files in a defined way
  - Etc.

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

HDF5 1.13 Library Architecture

The HDF Group



Subfiling VFD

Availability and Requirements

- Initial version released in HDF5 1.13.2 release
  - Further development work has been merged to develop branch for HDF5 1.13.3 and 1.14.0 releases
- HDF5 must be built with parallel support enabled
  - Must enable subfiling when building HDF5. It’s not enabled by default
- C11 capable compiler support is required
- Requires MPI_Init_thread to be called by HDF5 application and requires MPI_THREAD_MULTIPLE level of threading support by MPI implementation

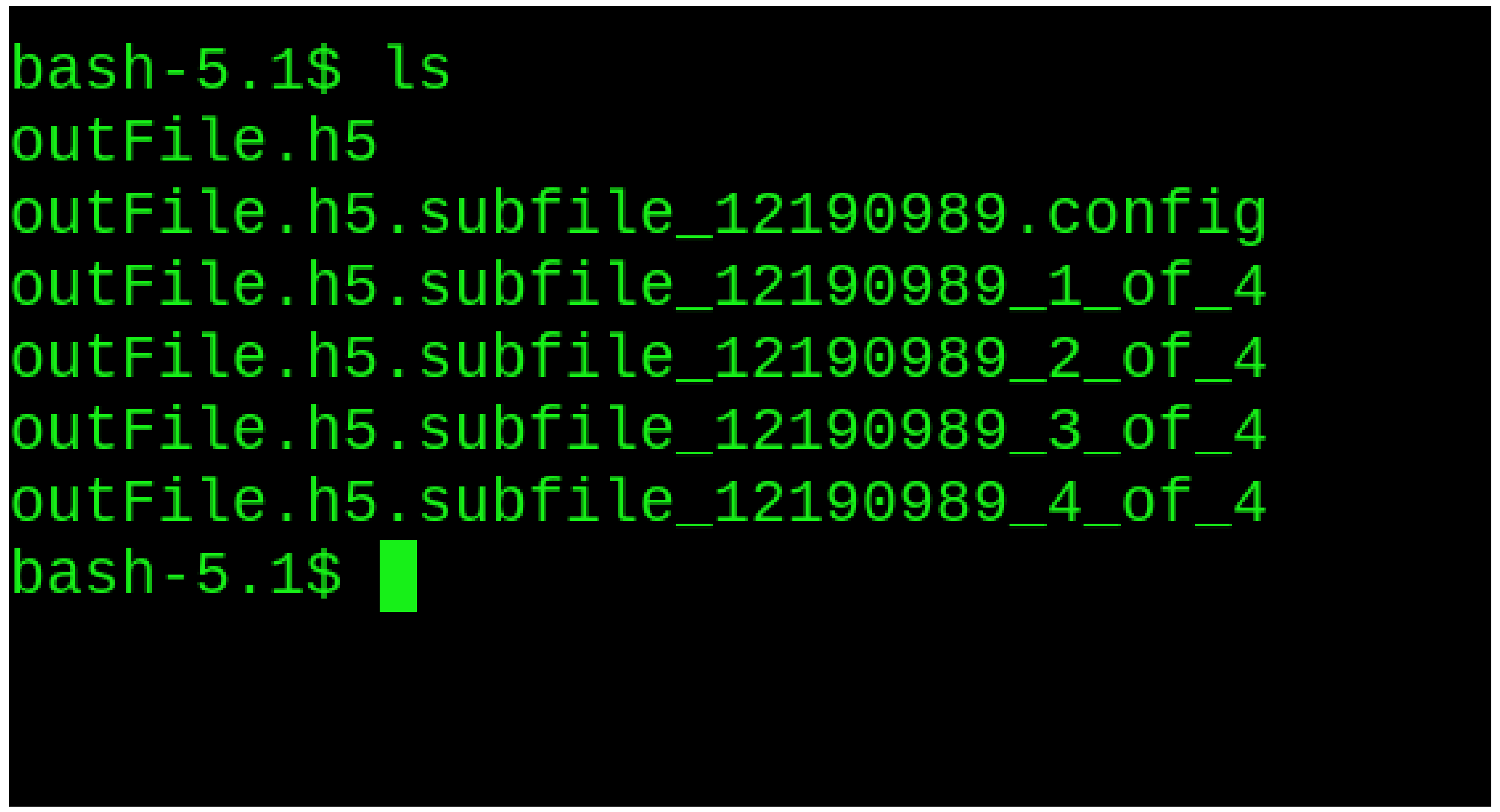
What is it?

- An MPI-based parallel file driver that allows an HDF5 application to distribute an HDF5 file across a collection of subfiles in equal-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 round-robin fashion
  - Defaults to 1 subfile per machine node with 32MiB data stripes
- Try to find a middle ground between single shared file and file-per-process approaches to parallel I/O
  - Minimize the locking issues of single shared file approach
  - Avoid some complexity and reduce total number of files compared to file-per-process approach
  - Designed to be flexible and configurable for different machines

What is it? (continued)

- Uses a system of "I/O concentrators" - subset of available MPI ranks which control subfiles and operate I/O worker thread pools
  - N-to-1 mapping from subfiles -> I/O concentrator ranks
  - Subfiles are assigned round-robin across the available I/O concentrator ranks, as determined by the chosen I/O concentrator selection method
  - 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

- Outputs several files per logical HDF5 file
  - HDF5 stub file
  - Subfiling VFD configuration file
  - Subfiles



```
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
bash-5.1$ 
```

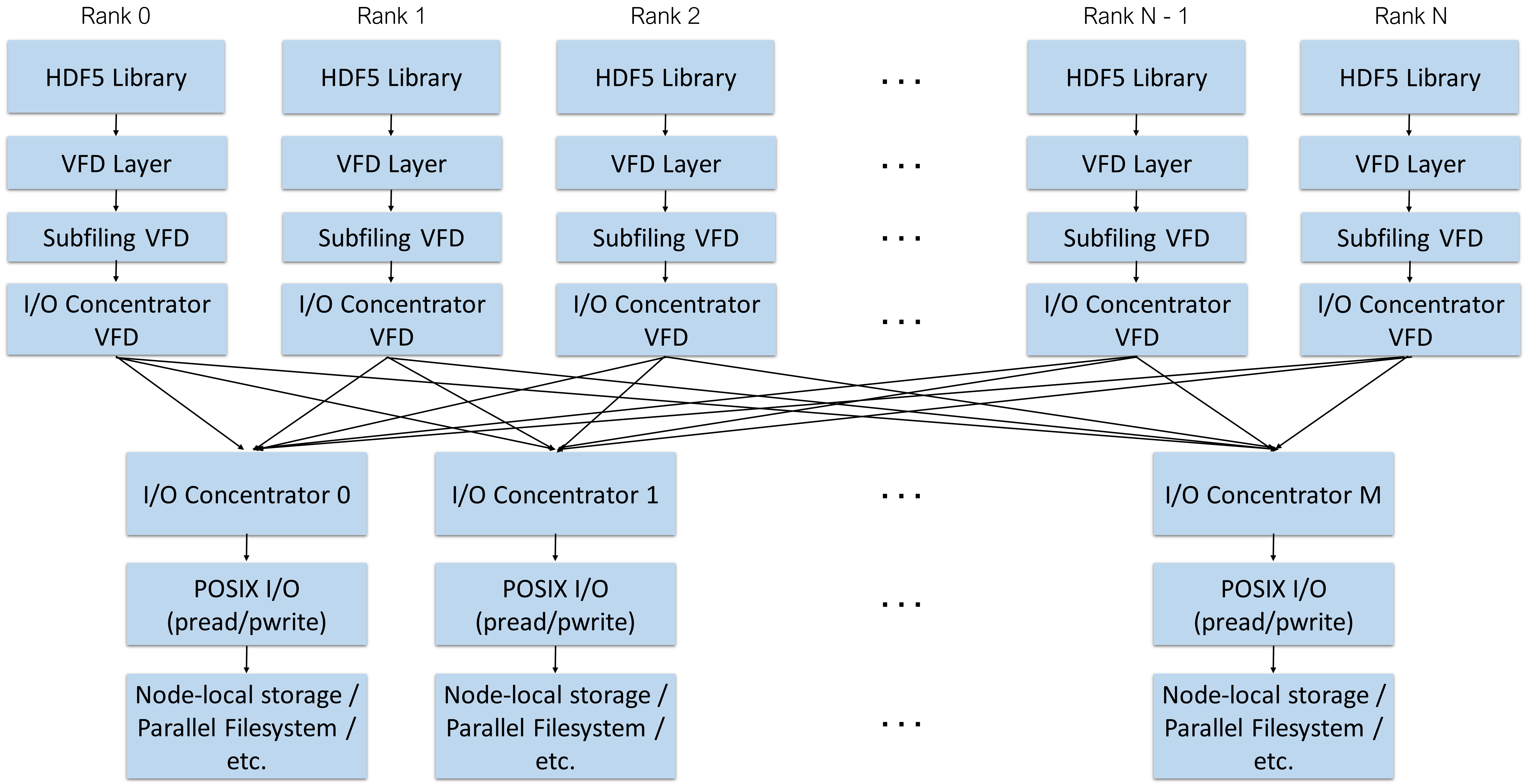
Current Architecture

The HDF Group

- Subfiling VFD stacked on top of I/O Concentrator VFD on each MPI rank
  - Subfiling VFD manages subfiling information (data stripe size, subfile count, HDF5 stub file, etc.) and breaks down I/O requests into offset/length vectors based on data stripe size and file offset
  - I/O Concentrator VFD receives I/O vectors and queues I/O calls to appropriate I/O concentrator

- Subset of MPI ranks selected as I/O concentrators
  - Each controls one or more subfiles
  - Receive I/O calls from I/O concentrator VFDs, translate to subfile-local file offsets and relay I/O call to appropriate subfile

Current Architecture



Subfiling HDF5 stub file

- Looks like 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 configuration file

- VFD creates and uses a simple configuration file to determine 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

```
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

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

- Currently co-located with HDF5 stub file and subfiling configuration file
  - Future development may allow placing of subfiles elsewhere

- Two ways to read data from subfiles with HDF5
  - Use Subfiling VFD
  - Stitch subfiles back together with external tooling to obtain an ordinary HDF5 file

New API Calls

```
herr_t
H5Pset_fapl_subfiling(hid_t fapl_id, const H5FD_subfiling_config_t *vfd_config);
```

Modifies the given File Access Property List to use the Subfiling VFD and configures the VFD according to the parameters set in the specified subfiling configuration structure. The subfiling configuration structure may be NULL, in which case default values are used.

```
herr_t
H5Pget_fapl_subfiling(hid_t fapl_id, H5FD_subfiling_config_t *config_out);
```

Returns the subfiling parameters that were set on the given File Access Property List, or default values if no subfiling parameters were set

Subfiling configuration structure

The HDF Group

https://github.com/HDFGroup/hdf5/blob/develop/src/H5FDsubfiling/H5FDsubfiling.h

```
typedef struct H5FD_subfiling_config_t {
    uint32_t magic;                      /* Must be set to H5FD_SUBFILING_FAPL_MAGIC                   */
    uint32_t version;                    /* Must be set to H5FD_SUBFILING_CURR_FAPL_VERSION            */
    hid_t    ioc_fapl_id;                /* The FAPL setup with the stacked VFD to use for I/O concentrators */
    hbool_t  require_ioc;                /* Whether to use the IOC VFD (currently must always be TRUE)      */
    H5FD_subfiling_params_t shared_cfg; /* Subfiling/IOC parameters (stripe size, stripe count, etc.)      */
} H5FD_subfiling_config_t;

typedef struct H5FD_subfiling_params_t {
    H5FD_subfiling_ioc_select_t ioc_selection; /* Method to select I/O concentrators          */
    int64_t                     stripe_size;   /* Size (in bytes) of data stripes in subfiles */
    int32_t                     stripe_count;  /* Target number of subfiles to use            */
} H5FD_subfiling_params_t;
```

Subfiling configuration structure (continued)

https://github.com/HDFGroup/hdf5/blob/develop/src/H5FDsubfiling/H5FDsubfiling.h

```
typedef enum {
    SELECT_IOC_ONE_PER_NODE = 0, /* Default                              */
    SELECT_IOC_EVERY_NTH_RANK,   /* Starting at rank 0, select-next += N */
    SELECT_IOC_WITH_CONFIG,      /* NOT IMPLEMENTED: Read-from-file      */
    SELECT_IOC_TOTAL,            /* Starting at rank 0, mpi_size / total */
    ioc_selection_options        /* Sentinel value                       */
} H5FD_subfiling_ioc_select_t;
```

Example

The HDF Group

https://github.com/HDFGroup/hdf5/blob/develop/examples/ph5_subfiling.c

```
H5FD_subfiling_config_t vfd_config;

hid_t plist_id = H5Pcreate(H5P_FILE_ACCESS);

H5Pset_mpi_params(plist_id, MPI_COMM_WORLD, MPI_INFO_NULL);

H5Pget_fapl_subfiling(plist_id, &vfd_config); /* Get a default subfiling configuration */

/* Set desired subfiling parameters */
...

H5Pset_fapl_subfiling(plist_id, &vfd_config);

hid_t file_id = H5Fcreate(H5FILE_NAME, H5F_ACC_TRUNC,
                          H5P_DEFAULT, plist_id);
```

Rank 0 pre-create example

- Rank 0 must know the target number of subfiles to create

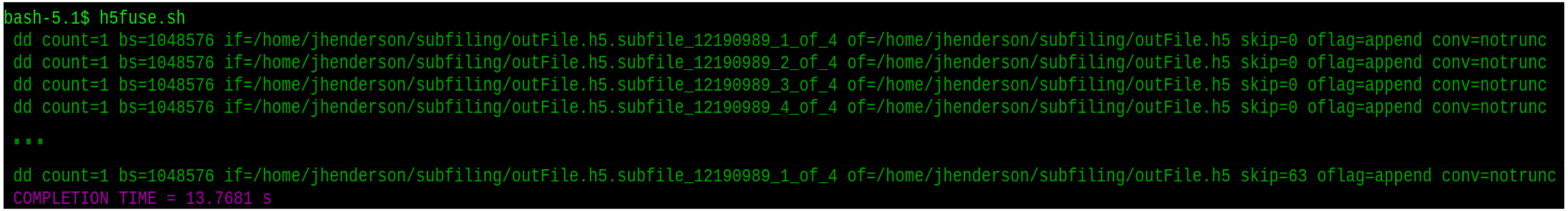
```
if (mpi_rank == 0) {
    H5FD_subfiling_config_t vfd_config;

    hid_t plist_id = H5Pcreate(H5P_FILE_ACCESS);

    H5Pset_mpi_params(plist_id, MPI_COMM_SELF, MPI_INFO_NULL);

    H5Pget_fapl_subfiling(plist_id, &vfd_config); /* Get a default subfiling configuration */

    vfd_config.shared_cfg.stripe_count = 20; /* Set target number of subfiles to 20 */

    H5Pset_fapl_subfiling(plist_id, &vfd_config);

    hid_t file_id = H5Fcreate(H5FILE_NAME, H5F_ACC_TRUNC,
                              H5P_DEFAULT, plist_id);
}

...
```

H5fuse script

- https://github.com/HDFGroup/hdf5/blob/develop/utils/subfiling_vfd/h5fuse.sh.in

- Reads a Subfiling VFD configuration file and fuses the file's subfiles back together into a single HDF5 file using dd

- Installed under 'bin' directory of HDF5 installation as 'h5fuse.sh'



```
bash-5.1$ h5fuse.sh
 dd count=1 bs=1048576 if=/home/jhenderson/subfiling/outFile.h5.subfile_12190989_1_of_4 of=/home/jhenderson/subfiling/outFile.h5 skip=0 oflag=append conv=notrunc
 dd count=1 bs=1048576 if=/home/jhenderson/subfiling/outFile.h5.subfile_12190989_2_of_4 of=/home/jhenderson/subfiling/outFile.h5 skip=0 oflag=append conv=notrunc
 dd count=1 bs=1048576 if=/home/jhenderson/subfiling/outFile.h5.subfile_12190989_3_of_4 of=/home/jhenderson/subfiling/outFile.h5 skip=0 oflag=append conv=notrunc
 dd count=1 bs=1048576 if=/home/jhenderson/subfiling/outFile.h5.subfile_12190989_4_of_4 of=/home/jhenderson/subfiling/outFile.h5 skip=0 oflag=append conv=notrunc
 ...
 dd count=1 bs=1048576 if=/home/jhenderson/subfiling/outFile.h5.subfile_12190989_1_of_4 of=/home/jhenderson/subfiling/outFile.h5 skip=63 oflag=append conv=notrunc
 COMPLETION TIME = 13.7681 s
```

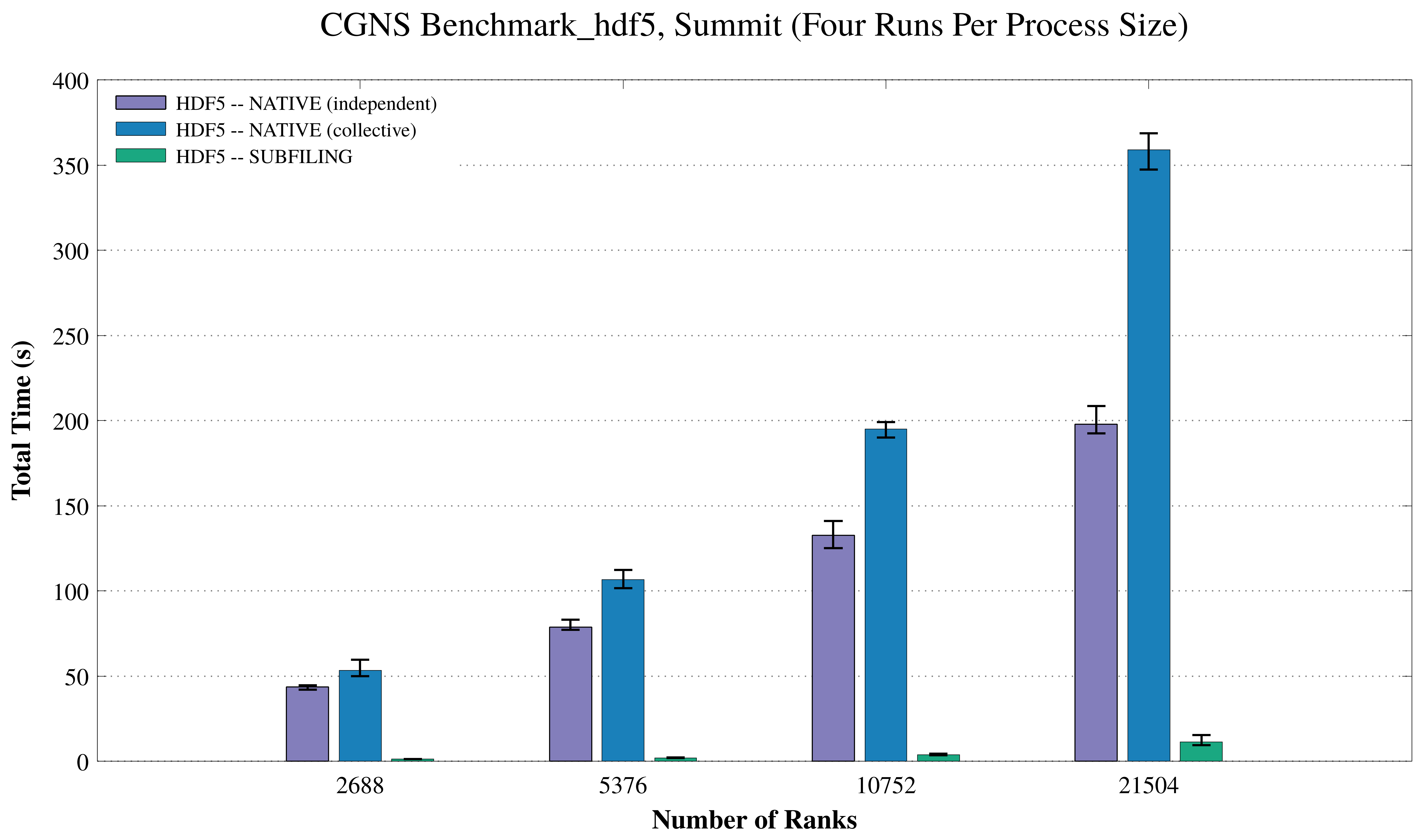
Performance Results

Performance Results

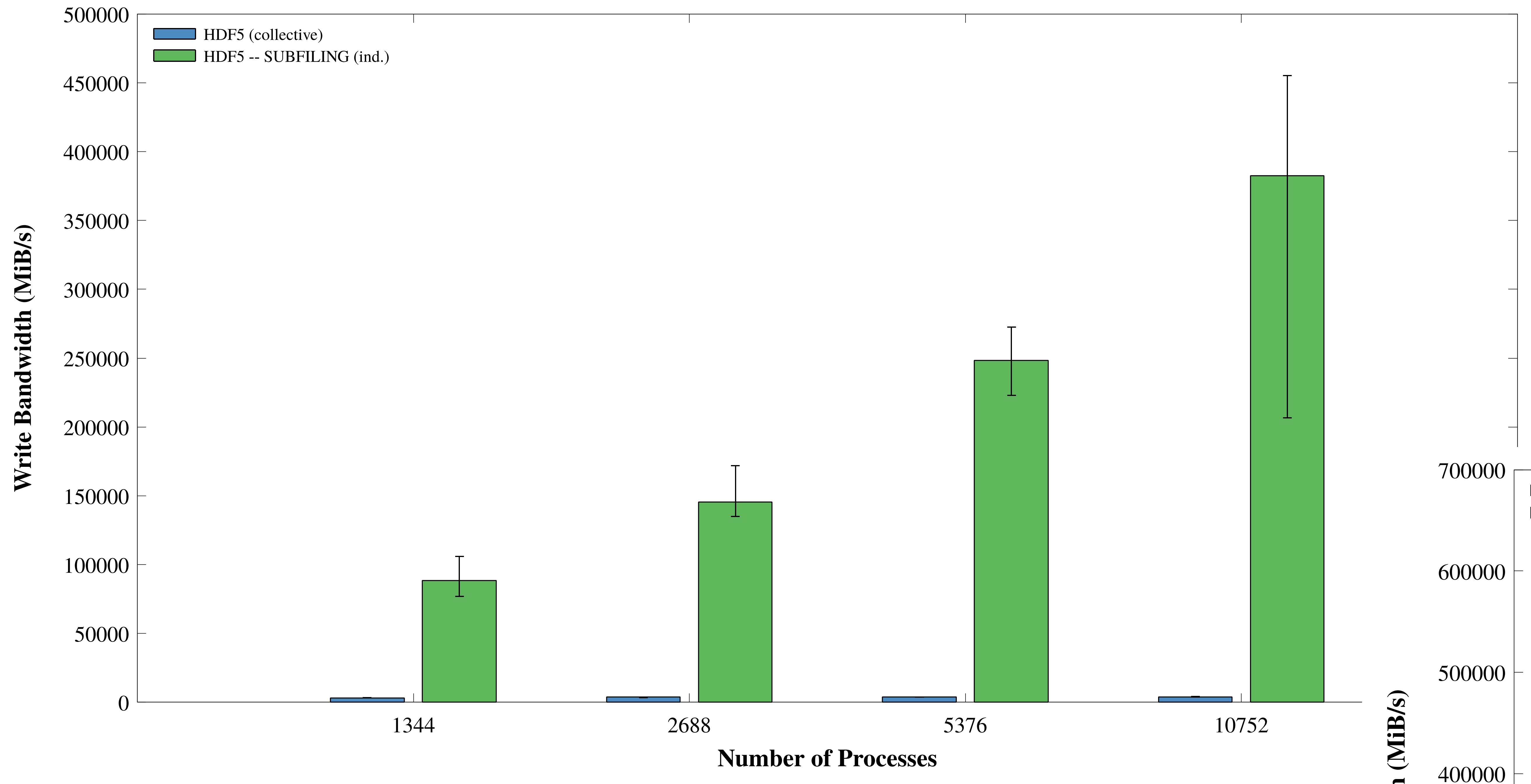
- CGNS = **C**omputational Fluid Dynamics (CFD) **G**eneral **N**otation **S**ystem
  - Standardize CFD I/O.
  - Subfiling version in the subfiling branch of the CGNS library.
  - *Benchmark_hdf5.c* writes and reads: mesh coordinates, element connectivity and solution data.
    - Summit (GPFS), using a mesh of 130 million (for 21k ranks), 6-node pentahedral elements.
      - The number of elements is halved as the ranks are decreased.

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

Performance Results

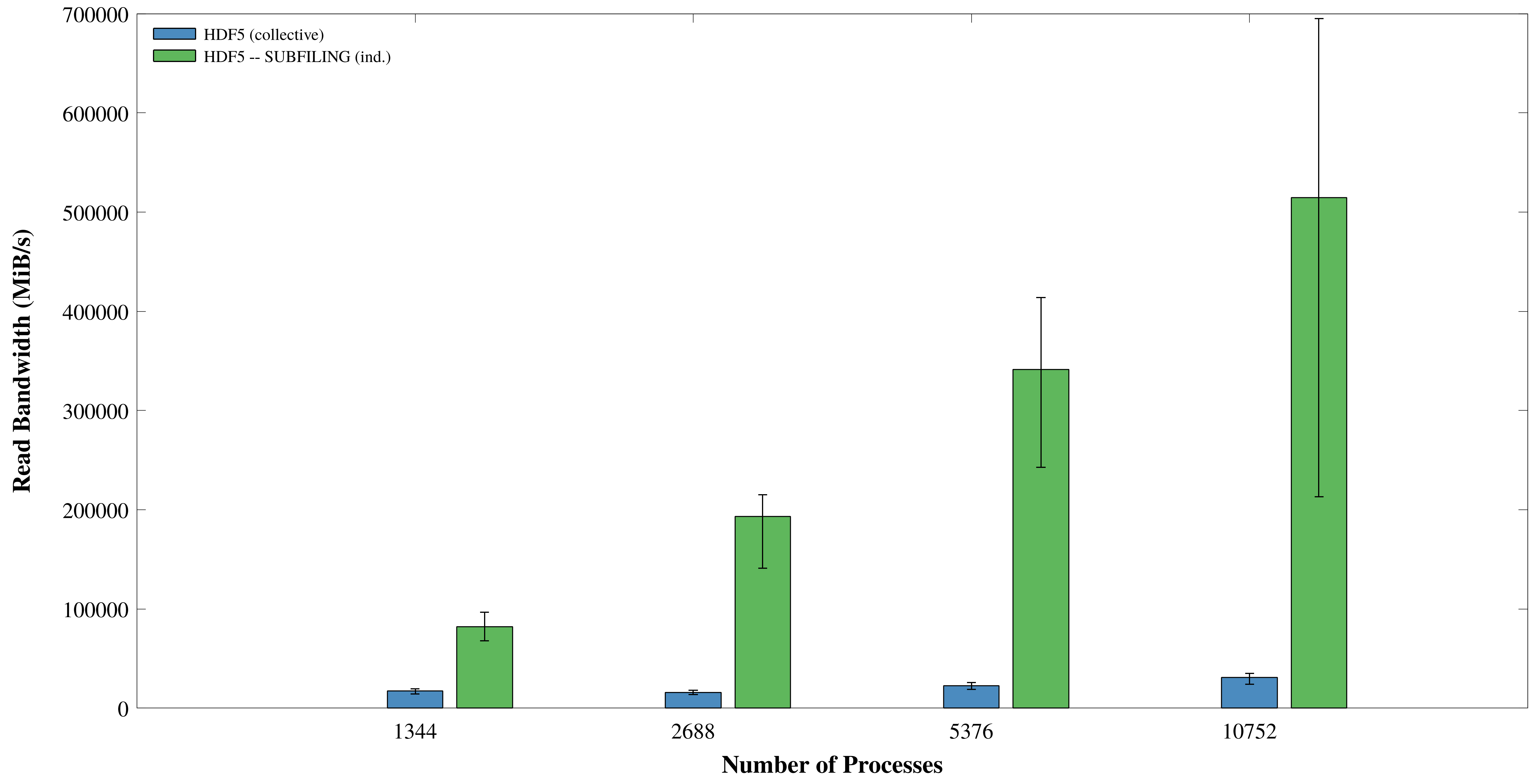


Performance Results -- Summit



- IOR, modified version for subfiling

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



More Information and Documentation

- HDF5 Subfiling VFD RFC
  - https://github.com/HDFGroup/hdf5doc/blob/master/RFCs/HDF5_Library/VFD_Subfiling/RFC_VFD_subfiling_200424.docx
- Subfiling VFD Documentation
  - https://docs.hdfgroup.org/hdf5/develop/_h5_f_dsubfiling_8h.html

Acknowledgments

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.

THANK YOU!

Questions & Comments?