CSE 5449: Intermediate Studies in Scientific Data Management

Lecture 16: Virtual Object Layer (VOL) and Intel DAOS

Dr. Suren Byna

The Ohio State University

E-mail: byna.1@osu.edu

https://sbyna.github.io

03/21/2023

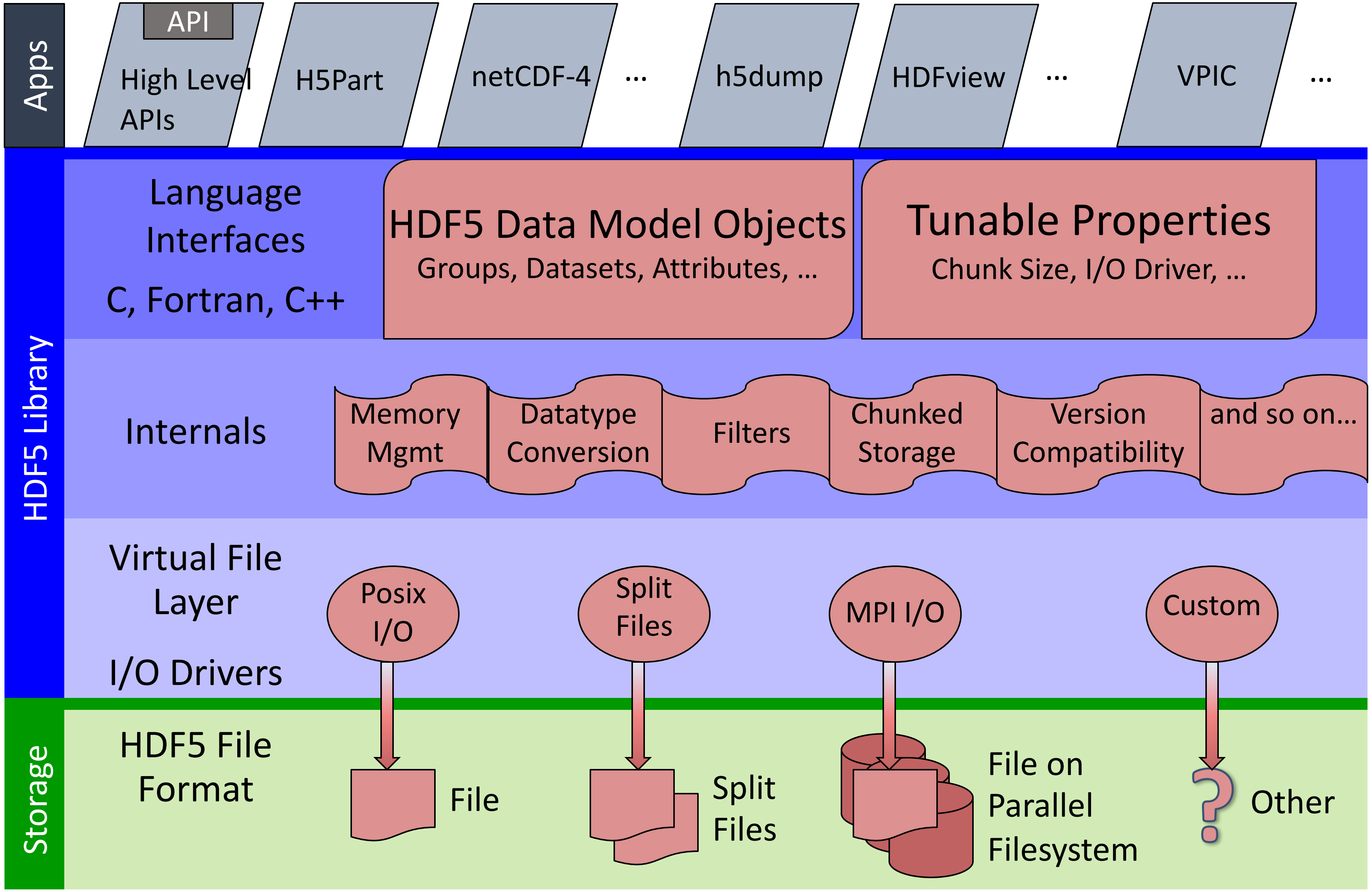
Today's class

- Any questions?

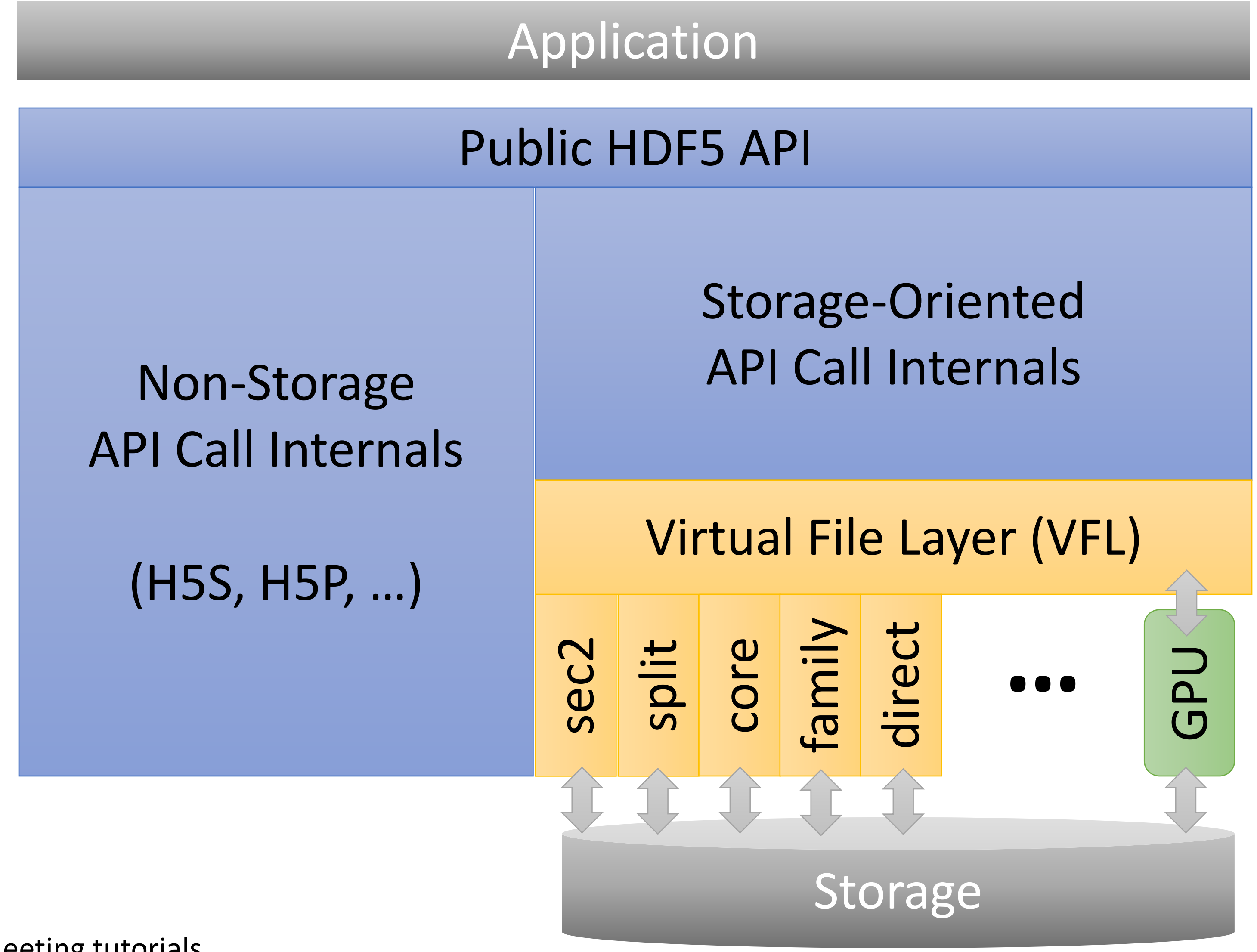
- Class presentation topic

- Today's class –
  - HDF5 optimizations – VOL and Async I/O

HDF5 Software Layers & Storage

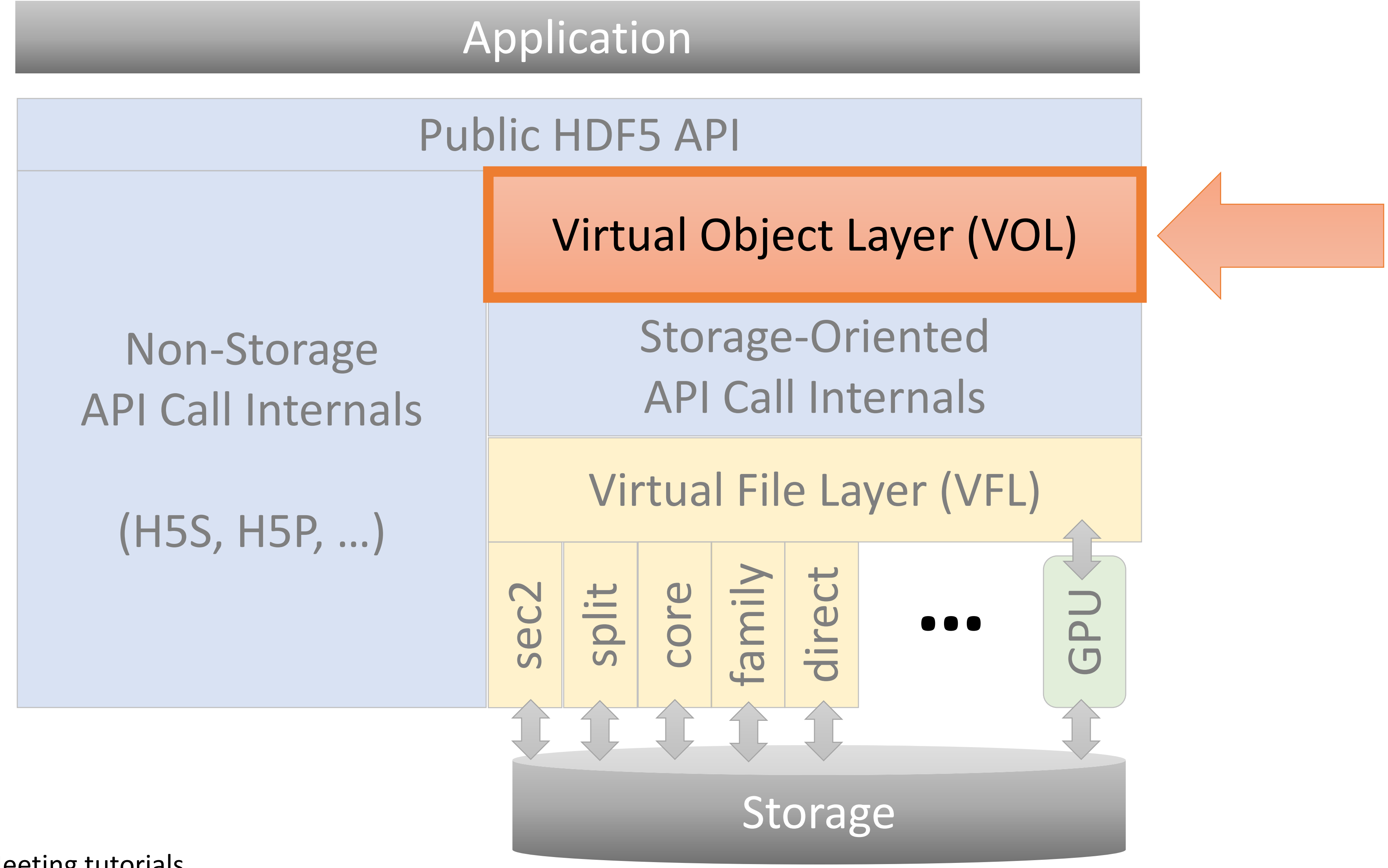


Original HDF5 Architecture (pre-1.12.0)



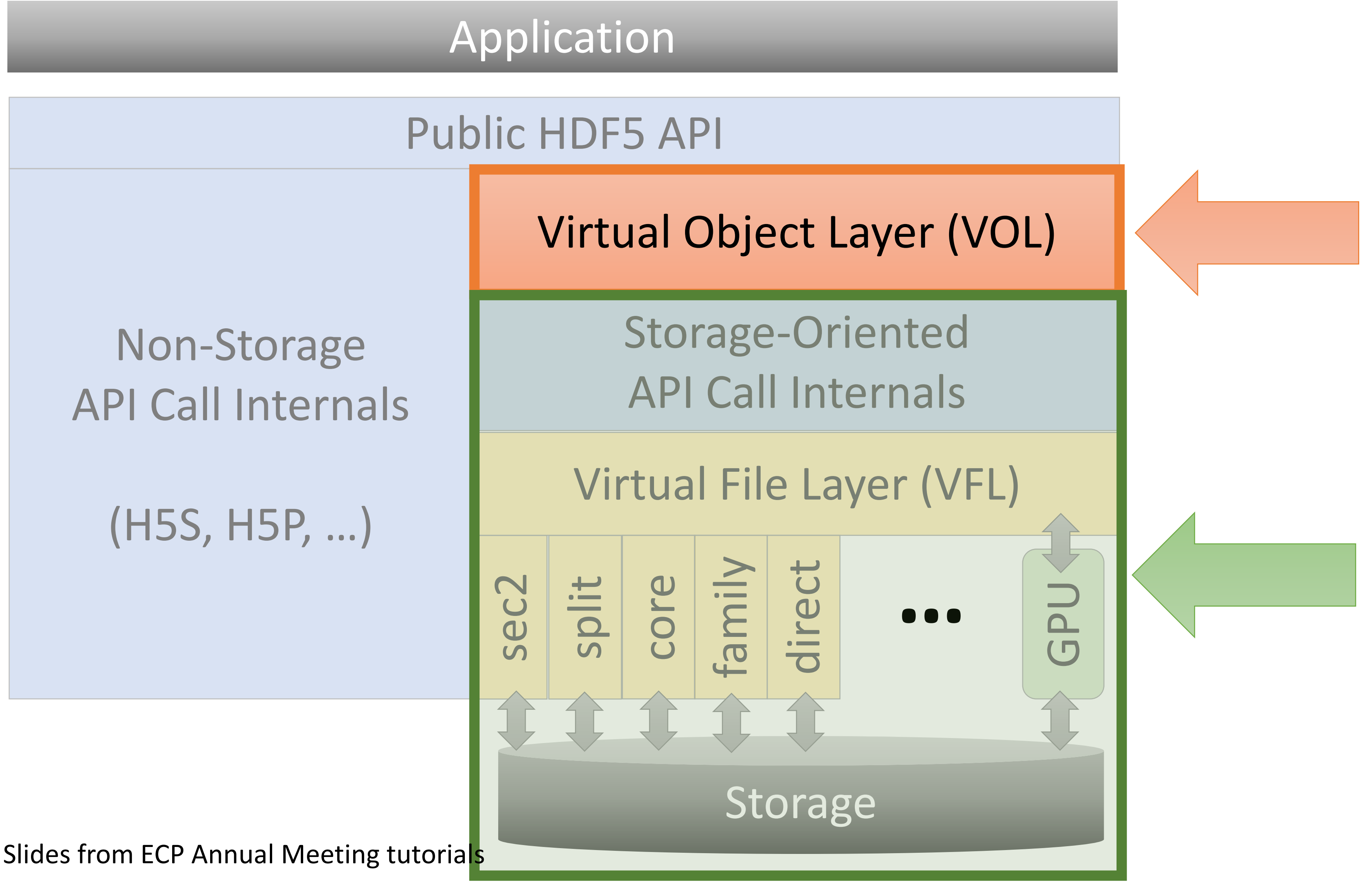
Slides from ECP Annual Meeting tutorials

HDF5 Architecture (1.12.0+)



Slides from ECP Annual Meeting tutorials

HDF5 Architecture (1.12.0+)

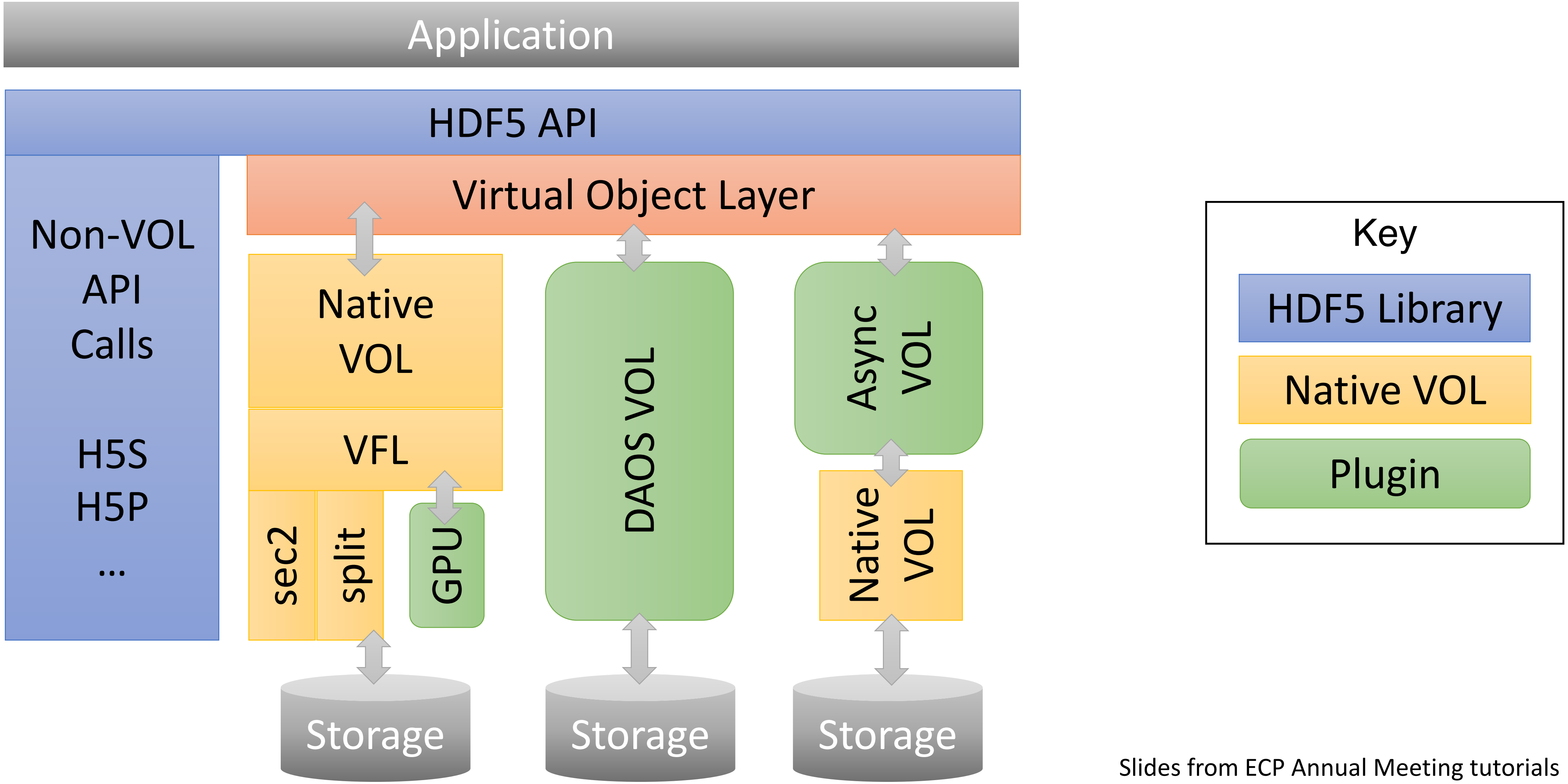


New Layer

Allows you to replace this with your own code!

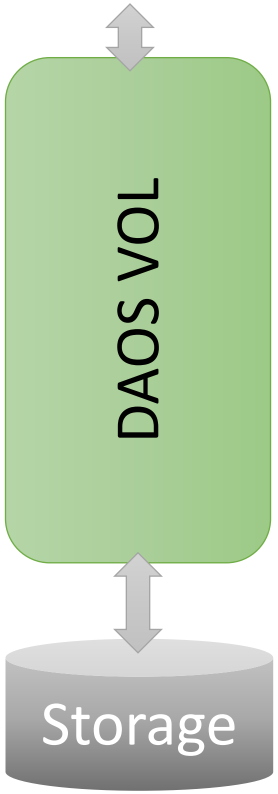
Slides from ECP Annual Meeting tutorials

Current HDF5 Architecture (1.12.0+)



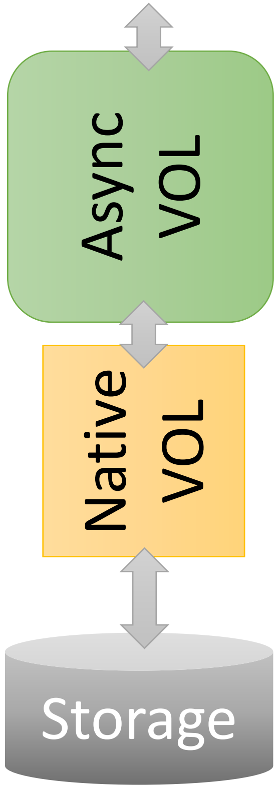
Slides from ECP Annual Meeting tutorials

Two Kinds of VOL Connector



Terminal

Maps HDF5 objects to arbitrary storage schemes

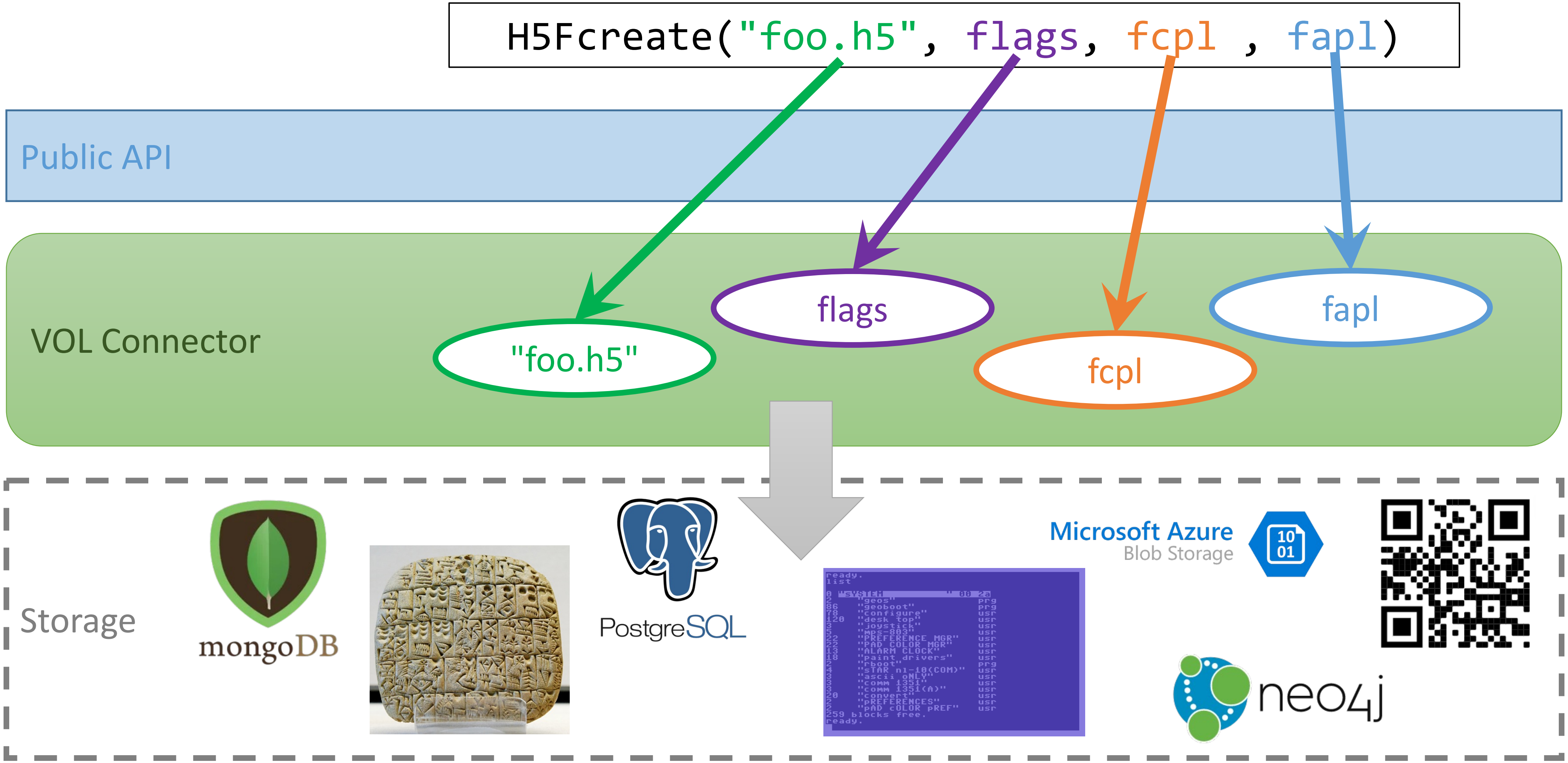


Pass-Through

Perform operations (e.g., caching, logging) before passing the data on to the next connector.

Slides from ECP Annual Meeting tutorials

Terminal VOL Connectors



```
H5Fcreate("foo.h5", flags, fcpl , fapl)
```

```
ready.
list

0 "SYSTEM              " 00 2a
2     "geos"             prg
86    "geoboot"          prg
78    "configure"        usr
120   "desk top"         usr
3     "joystick"         usr
5     "mps-803"          usr
22    "PREFERENCE MGR"   usr
22    "PAD COLOR MGR"    usr
13    "ALARM CLOCK"      usr
18    "paint drivers"    usr
2     "rboot"            prg
4     "sTAR nl-10(COM)"  usr
3     "ascii oNLY"       usr
3     "comm 1351"        usr
3     "comm 1351(A)"     usr
20    "convert"          usr
2     "pREFERENCES"      usr
2     "pAD cOLOR pREF"   usr
259 blocks free.
ready.
```

Slides from ECP Annual Meeting tutorials

VOL Toolkit Repository

- Location: https://github.com/HDFGroup/vol-toolkit
- All your VOL construction needs in a single location
- Does not contain original content
- Designed to bring important content from other repositories together with consistent versioning
- Content is mainly included as git submodules, though the docs are currently copied in
- Tags will identify "HDF5 1.13.0", etc. versions of the toolkit
- Includes an appropriate version of HDF5

Templates

Two template repositories are linked in the toolkit

vol-template (https://github.com/HDFGroup/vol-template)

- Template for building terminal VOL connectors
- Build files + stubs.
- Developed and supported by THG
- Officially a "template repository" on github so you can clone + rename

vol-external-passthrough (https://github.com/hpc-io/vol-external-passthrough)

- Template for constructing pass-through connectors
- Has no-op, pass-through stubs for all callbacks
- Developed and supported by NERSC

Production Connectors (NOT in Toolkit)

When developing your own connector, it can be VERY helpful to see what others have done

Examples:

vol-daos (https://github.com/HDFGroup/vol-daos)

- Terminal VOL connector based on Intel's DAOS developed by THG
- Largely complete coverage of the HDF5 API
- Supports parallel HDF5 and async I/O

vol-async (https://github.com/hpc-io/vol-async)
vol-cache (https://github.com/hpc-io/vol-cache)

- Pass-through VOL connectors developed by LBNL
- Support parallel HDF5 and async I/O

Find a full list here: https://portal.hdfgroup.org/display/support/Registered+VOL+Connectors

Slides from ECP Annual Meeting tutorials

Test Suite

A subset of the HDF5 library tests has been collected in a separate repository

vol-tests (https://github.com/HDFGroup/vol-tests)

- Requires CMake
- Supports parallel connectors and async
- No Windows support
- Tests a lot of the HDF5 API
- Tests the HDF5 command-line tools
- Expect a lot of failed tests until you have significant HDF5 API coverage in your connector
- Instructions for use located in the repository's README

Tutorial and Associated VOL Connector

Feb 2022 VOL tutorial

- Watch here: https://www.youtube.com/watch?v=7XEbm-__QuM
- "Hello, world!" of VOL creation
- Builds a simple connector from scratch using the template terminal VOL connector as a starting point
- Tutorial connector:

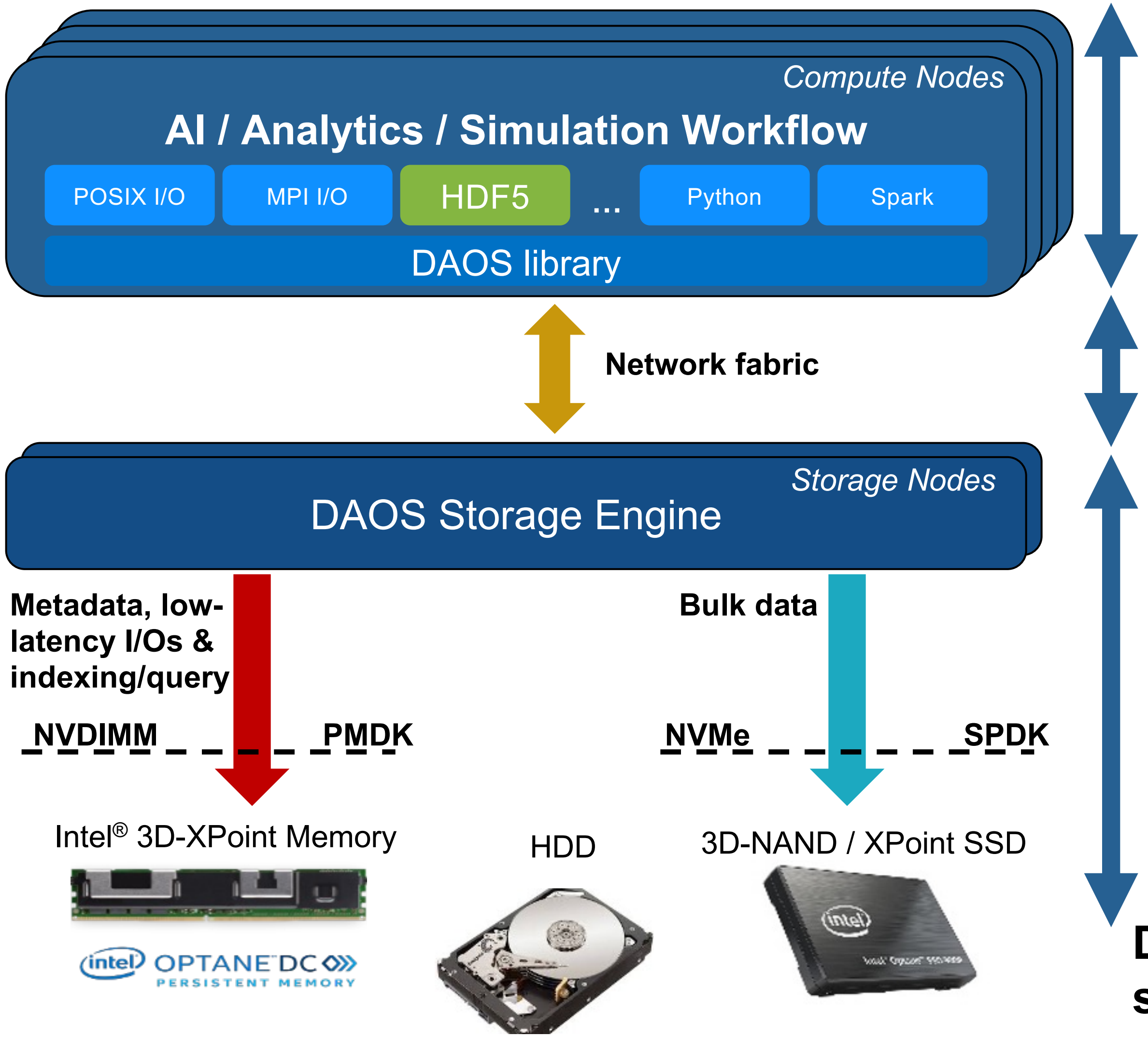
vol-tutorial (https://github.com/HDFGroup/vol-tutorial.git)

Slides from ECP Annual Meeting tutorials

Intel Distributed Asynchronous Object Storage (DAOS)

Credit: Mohamad Chaarawi (Intel Corporation)

More details: https://www.hdfgroup.org/wp-content/uploads/2021/10/Accelerating-HDF5s-Parallel-IO-for-Exascale-using-DAOS.pdf
Full paper: "Accelerating HDF5 I/O for Exascale using DAOS," in IEEE Transactions on Parallel and Distributed Systems, doi: 10.1109/TPDS.2021.3097884



Compute Nodes

AI / Analytics / Simulation Workflow

POSIX I/O | MPI I/O | HDF5 | ... | Python | Spark

DAOS library

Network fabric

Storage Nodes

DAOS Storage Engine

Metadata, low-latency I/Os & indexing/query

Bulk data

NVDIMM **PMDK**

NVMe **SPDK**

Intel® 3D-XPoint Memory

HDD

3D-NAND / XPoint SSD

- DAOS library directly linked with the applications
- No need for dedicated cores
- Low memory/CPU footprint
- End-to-end OS bypass
- KV API, non-blocking, lockless, snapshot support

- Low-latency & high-message-rate communications
- Native support for RDMA & scalable collective operations
- Support for Infiniband, Slingshot, etc through OFI libfabric

- Fine-grained I/O with media selection strategy
- Only application data on SSD to maximize throughput
- Small I/Os aggregated in pmem & migrated to SSD in large chunks
- Full user space model with no system calls on I/O path
- Built-in storage management infrastructure (control plane)
- NFSv4-like ACL

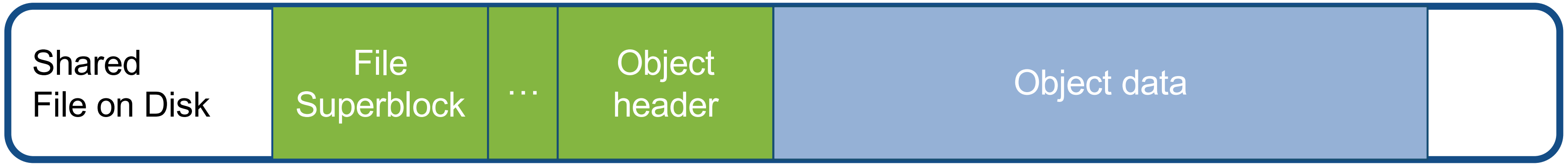
Delivers high-IOPs, high-bandwidth and low-latency storage with advanced features in a single tier

Slides from ECP Annual Meeting tutorials from The HDF Group, Original from Intel Corporation

From “Native” to DAOS Representation

More details: https://www.hdfgroup.org/wp-content/uploads/2021/10/Accelerating-HDF5s-Parallel-IO-for-Exascale-using-DAOS.pdf

- POSIX I/O was designed for **disk-based** storage
  - High-latency to write data at random offsets because of mechanical aspects
  - Current native HDF5 file format inherited POSIX I/O block-based model (serial)



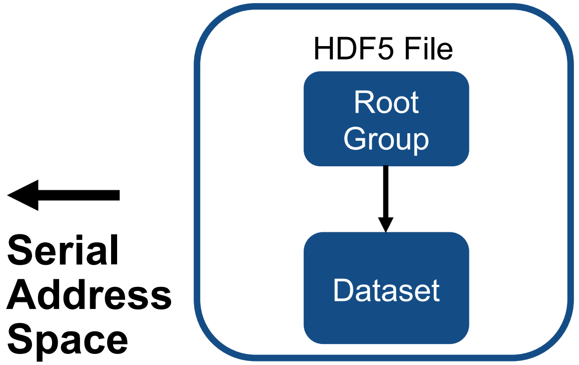
Shared File on Disk

File Superblock

…

Object header

Object data



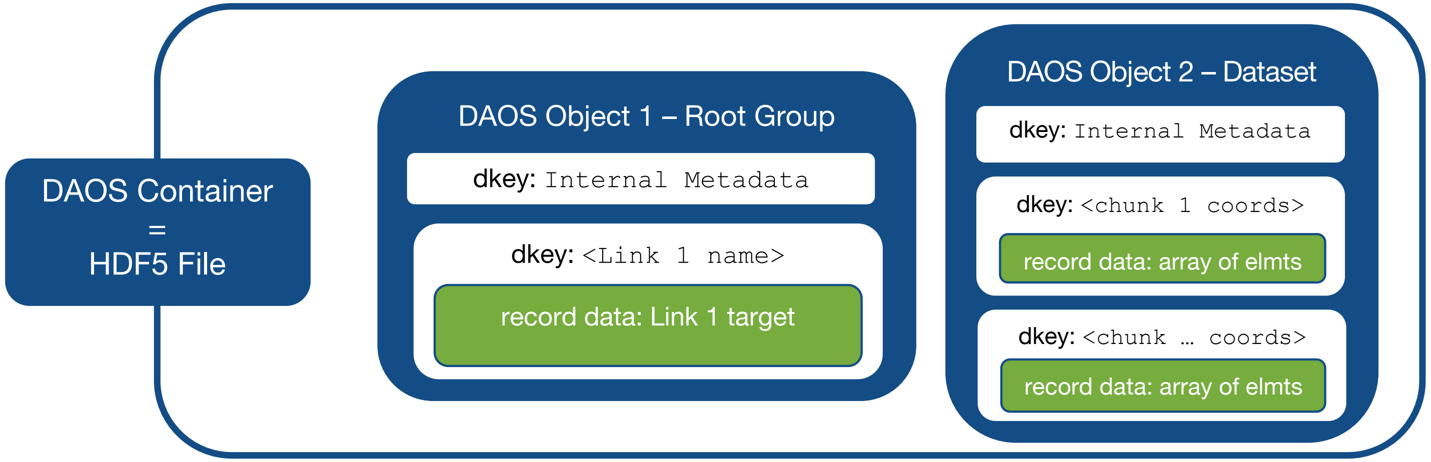
Serial Address Space

HDF5 File

Root Group

Dataset

- DAOS is designed for **object-based** storage



DAOS Container = HDF5 File

DAOS Object 1 – Root Group

dkey: `Internal Metadata`

dkey: `<Link 1 name>`

record data: Link 1 target

DAOS Object 2 – Dataset

dkey: `Internal Metadata`

dkey: `<chunk 1 coords>`

record data: array of elmts

dkey: `<chunk … coords>`

record data: array of elmts

1-to-1 Mapping

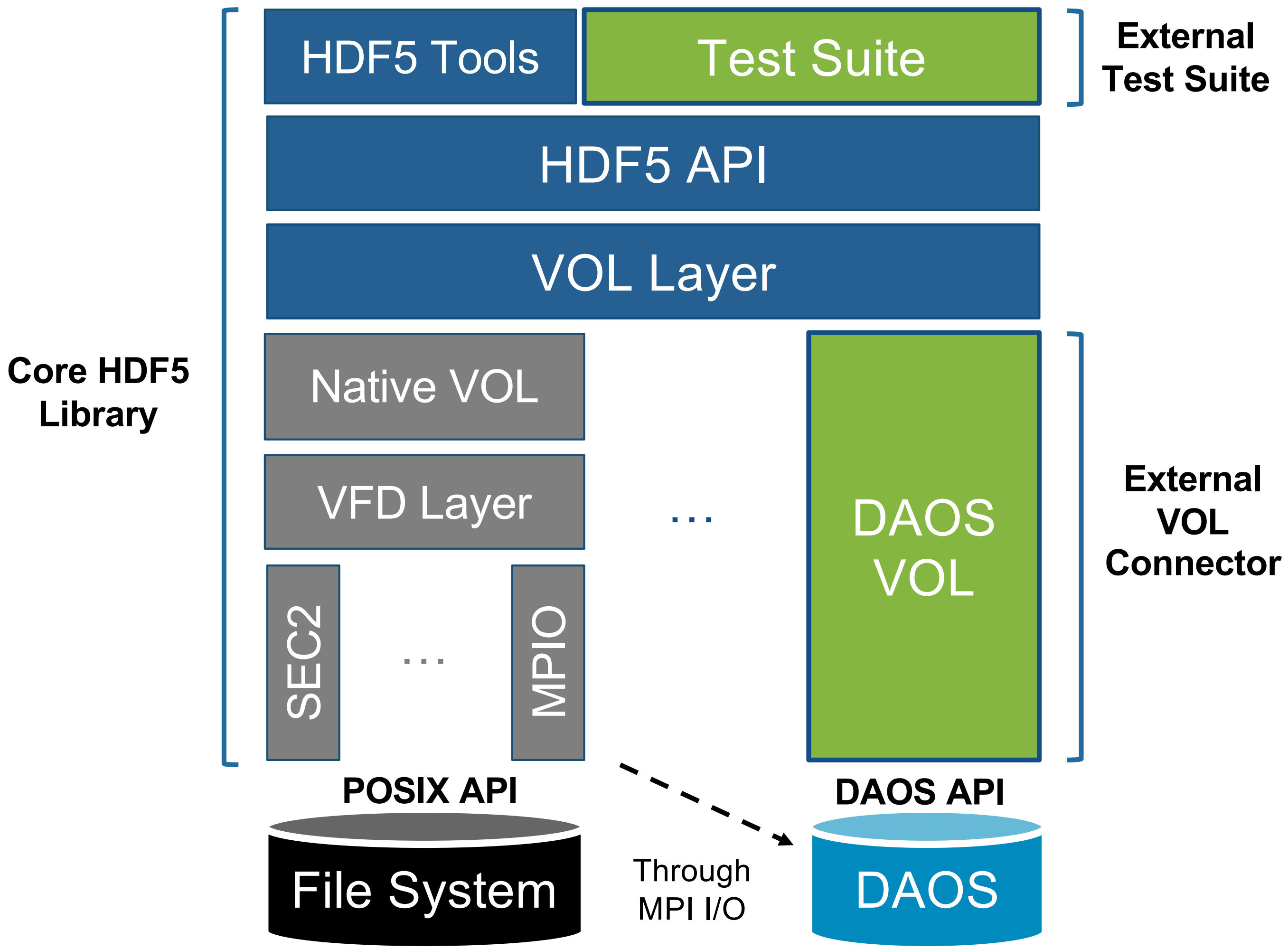
- Parallel I/O and chunking first-class citizens
  - Implicit chunking
- Independent object creation

Slides from ECP Annual Meeting tutorials from The HDF Group, Original slide form Intel Corporation

HDF5 VOL Architecture and DAOS VOL

More details: https://www.hdfgroup.org/wp-content/uploads/2021/10/Accelerating-HDF5s-Parallel-IO-for-Exascale-using-DAOS.pdf

- New Component
- Enhanced Component
- Native Component
- New HDF5 features:
  - Maps (enabled by K/V objects)
  - File deletion
  - Independent metadata
    - HDF5 objects can be created independently
  - Asynchronous I/O
- Tools support:
  - h5dump, h5ls, h5diff, h5repack, h5copy, etc



Slides from ECP Annual Meeting tutorials from The HDF Group, Original slide form Intel Corporation

DAOS VOL Usage

More details: https://www.hdfgroup.org/wp-content/uploads/2021/10/Accelerating-HDF5s-Parallel-IO-for-Exascale-using-DAOS.pdf

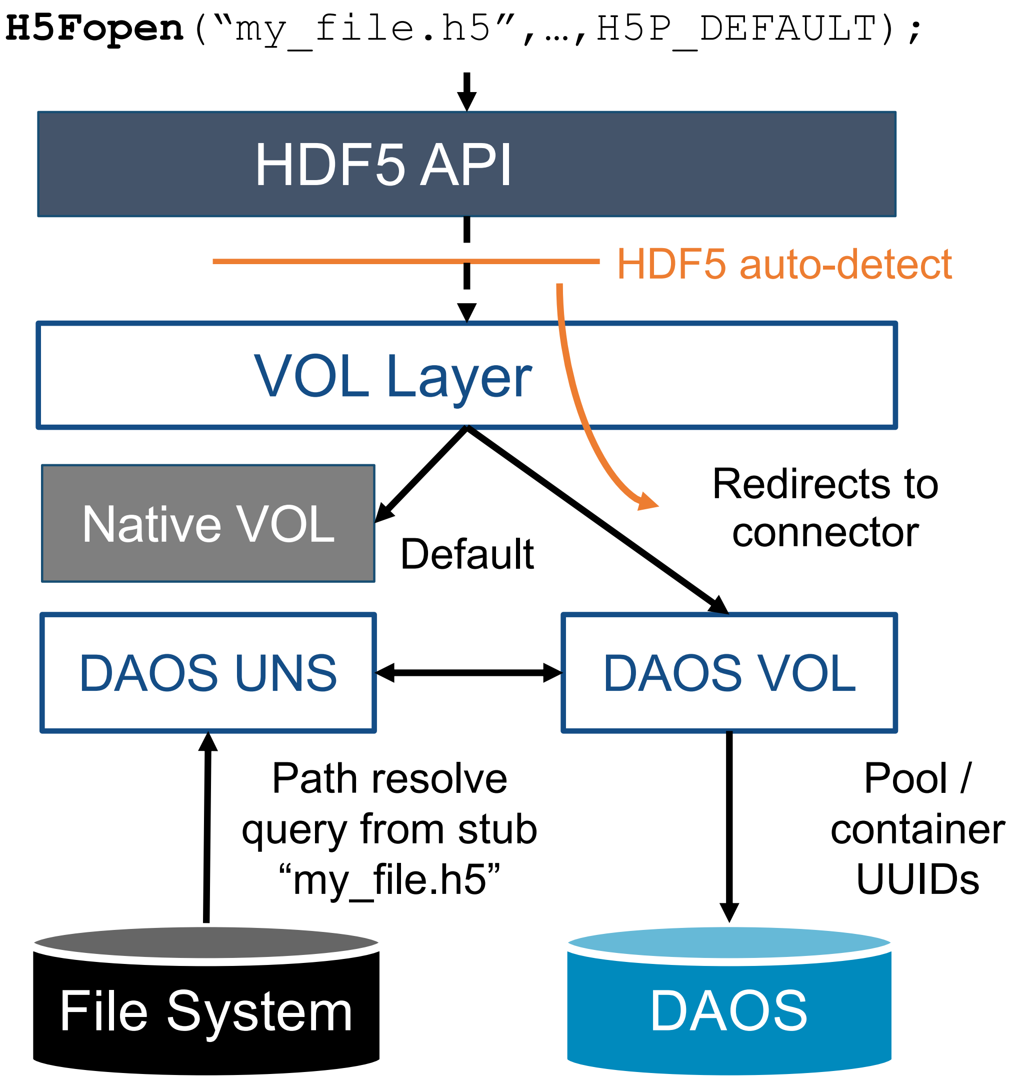
- Minimal or no code changes for application developer (if only looking for compatibility)
- Two ways to tell which connector to use
  - HDF5 file access property list (**recommended for new files or when manipulating multiple VOLs**)

```
herr_t H5Pset_fapl_daos(hid_t fapl_id,
const char *pool, const char *sys_name)
```

  - Environment variable

```
HDF5_VOL_CONNECTOR=daos
HDF5_PLUGIN_PATH=/path/to/connector/folder
```

- **Auto-detect and Unified Namespace** component facilitates opening of DAOS files with the DAOS connector (embedded DAOS metadata through extended attributes)

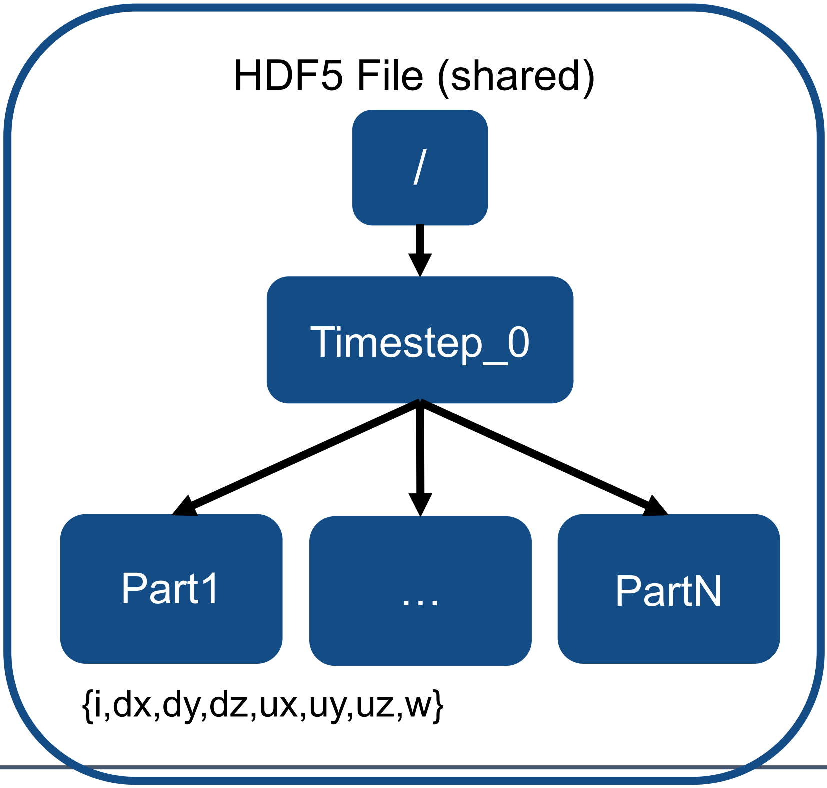


Slides from ECP Annual Meeting tutorials from The HDF Group , Original slide form Intel Corporation

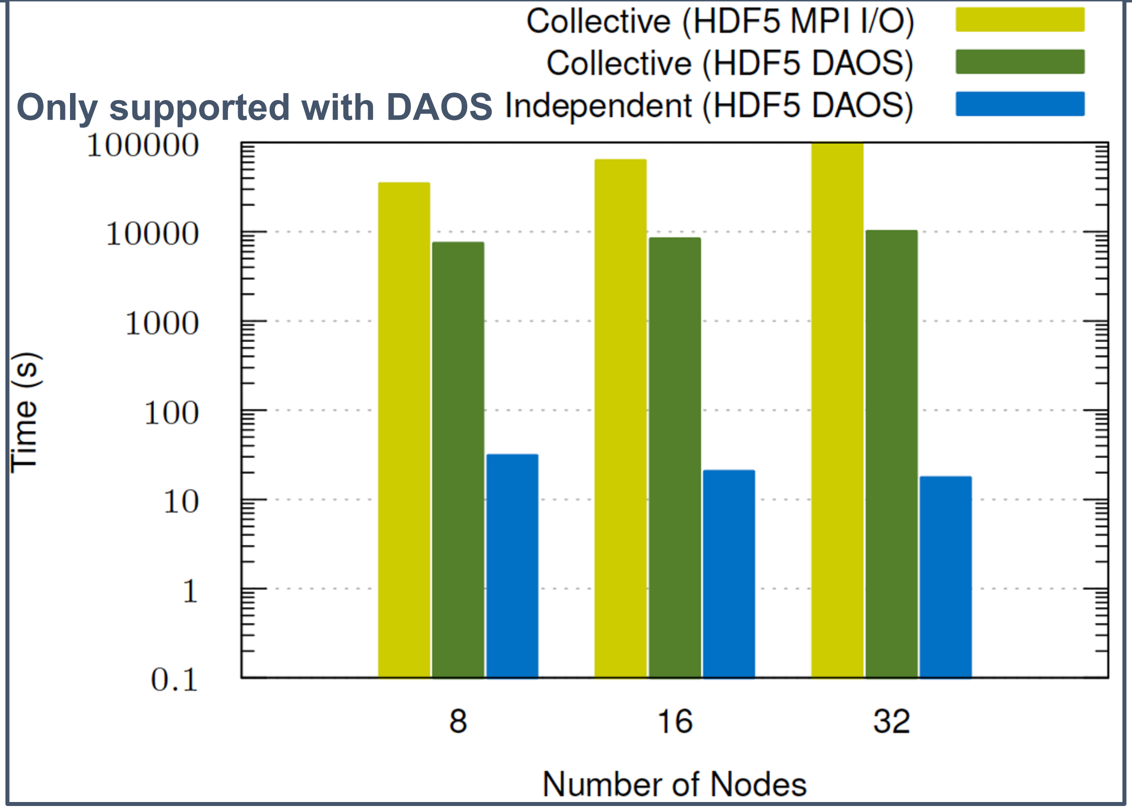
Evaluation – Example w/VPIC (metadata operations)

More details: https://www.hdfgroup.org/wp-content/uploads/2021/10/Accelerating-HDF5s-Parallel-IO-for-Exascale-using-DAOS.pdf

Re-defined VPIC file structure for electron particle (N particles)



VPIC I/O performance using collective and independent group creation



Only supported with DAOS

Slides from ECP Annual Meeting tutorials from The HDF Group , Original slide form Intel Corporation

Summary of today’s class

- Virtual Object Layer (VOL) and DAOS VOL connector

- Next Class – Asynchronous I/O

- Class project –
  - Status update on Apr 4th
  - Final presentation on Apr 20th
  - Final exam on Apr 25th