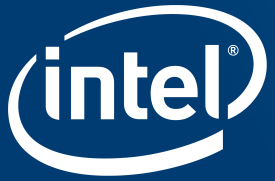


DAOS TUTORIAL

Johann Lombardi, Cloud & Enterprise Solution Group

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No product or component can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

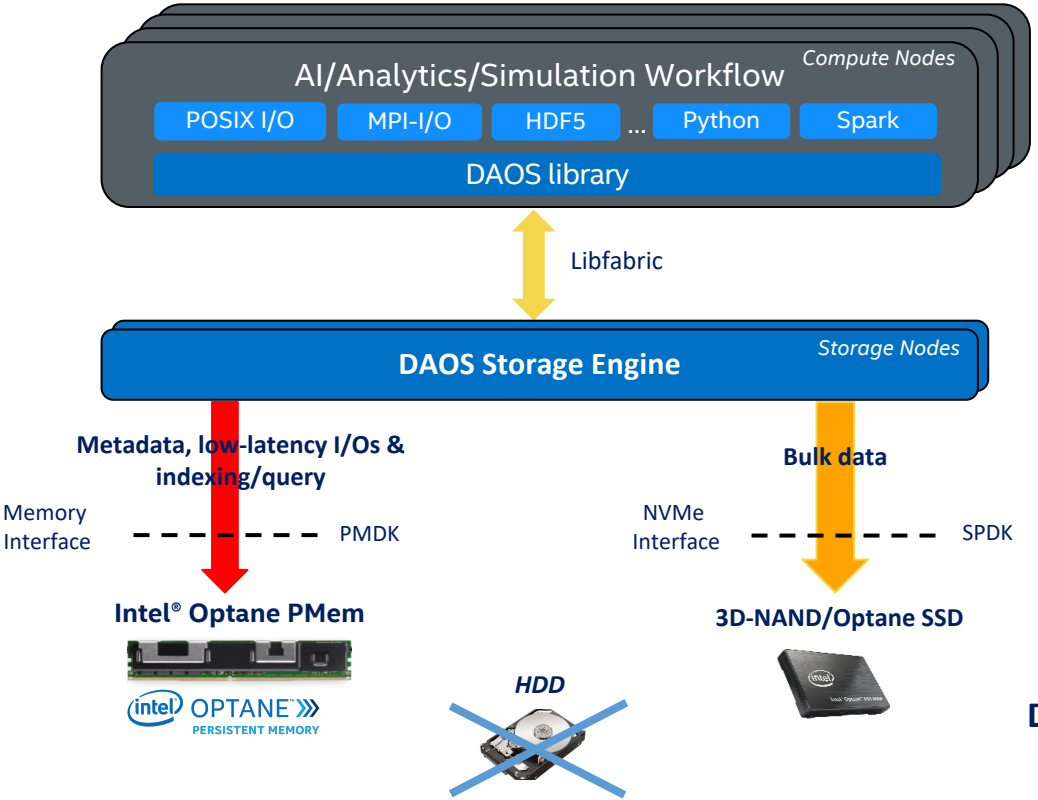
Intel Advanced Vector Extensions (Intel AVX) provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

DAOS Storage Architecture



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

- **Low-latency** & **high-message-rate** communications
- Native support for **RDMA** & **scalable** collective operations
- Support for iWarp, RoCE, Infiniband, OPA, Slingshot, …

- **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 userspace 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

DAOS: Primary Storage on Aurora

Aurora DAOS configuration

- Capacity: 230PB
- Bandwidth: >25TB/s

"Combined in Aurora, the Intel compute system, Cray Slingshot network, and the Intel DAOS storage open new possibilities for accelerating the scientific research needed to solve critical human challenges such as cancer and disease. DAOS enables the creation of new storage data models tailored specifically to applications like the Cancer Distributed Learning Environment (CANDLE) which provide a powerful platform to advance a wide array of scientific challenges using deep learning."

– Rick Stevens, Associate Laboratory Director for Computing, Environment and Life Sciences

"The Argonne Leadership Computing Facility is excited to be the first major production deployment of the DAOS storage system as part of Aurora, an US exascale system coming in 2021. As designed, it will provide us unprecedented levels of metadata operation rates and extremely high bandwidth for I/O intensive workloads."

– Susan Coghlan, ALCF-X Project Director/Exascale Computing Systems Deputy Director

High Value Use Cases

Traditional Modelling & Simulation

- Workloads with small file I/O, large IOPs, or low-latency requirements struggle under existing PFS that are optimized for large streaming reads and writes
- Traditional HPC centers aligning on IO-500 benchmarks are placing more and more requirements in RFPs on unaligned I/Os
- Direct integration of the domain-specific data models (e.g. oil & gas, meteorology, animation…) over the DAOS API to accelerate applications

High Performance Data Analytics (HPDA)

- HPDA generates large volumes of small random reads/writes
- DAOS provides new rich storage API with native support for unstructured and semi-structured data
- DAOS stores small writes and metadata into byte-granular persistent memory, removing performance bottlenecks & unleashing higher performance

High Value Use Cases (cont'd)

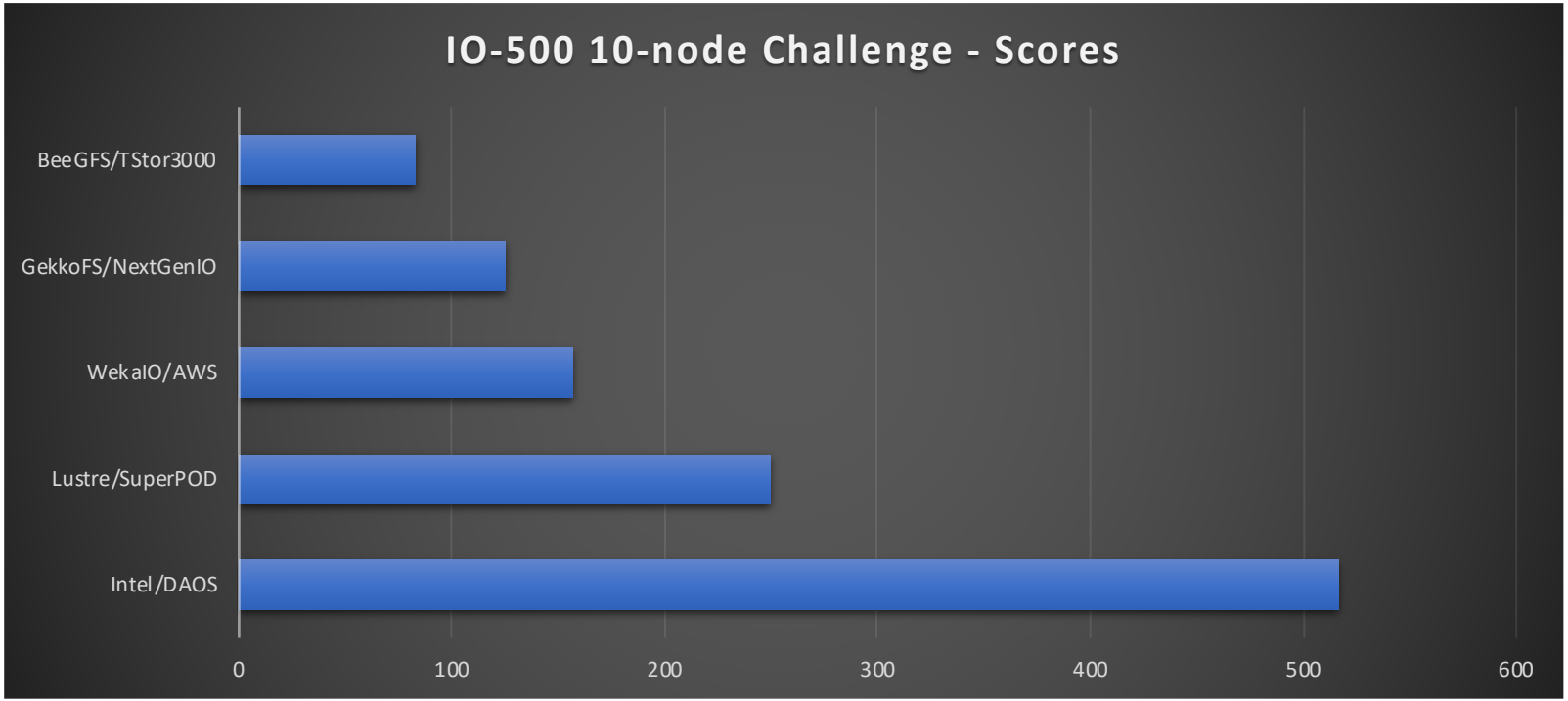
Artificial Intelligence

- AI workloads perform large volumes of reads - data access time becomes critical
- Native AI framework support enables AI workloads
- DAOS provides low latency access to read the data compared to existing solutions

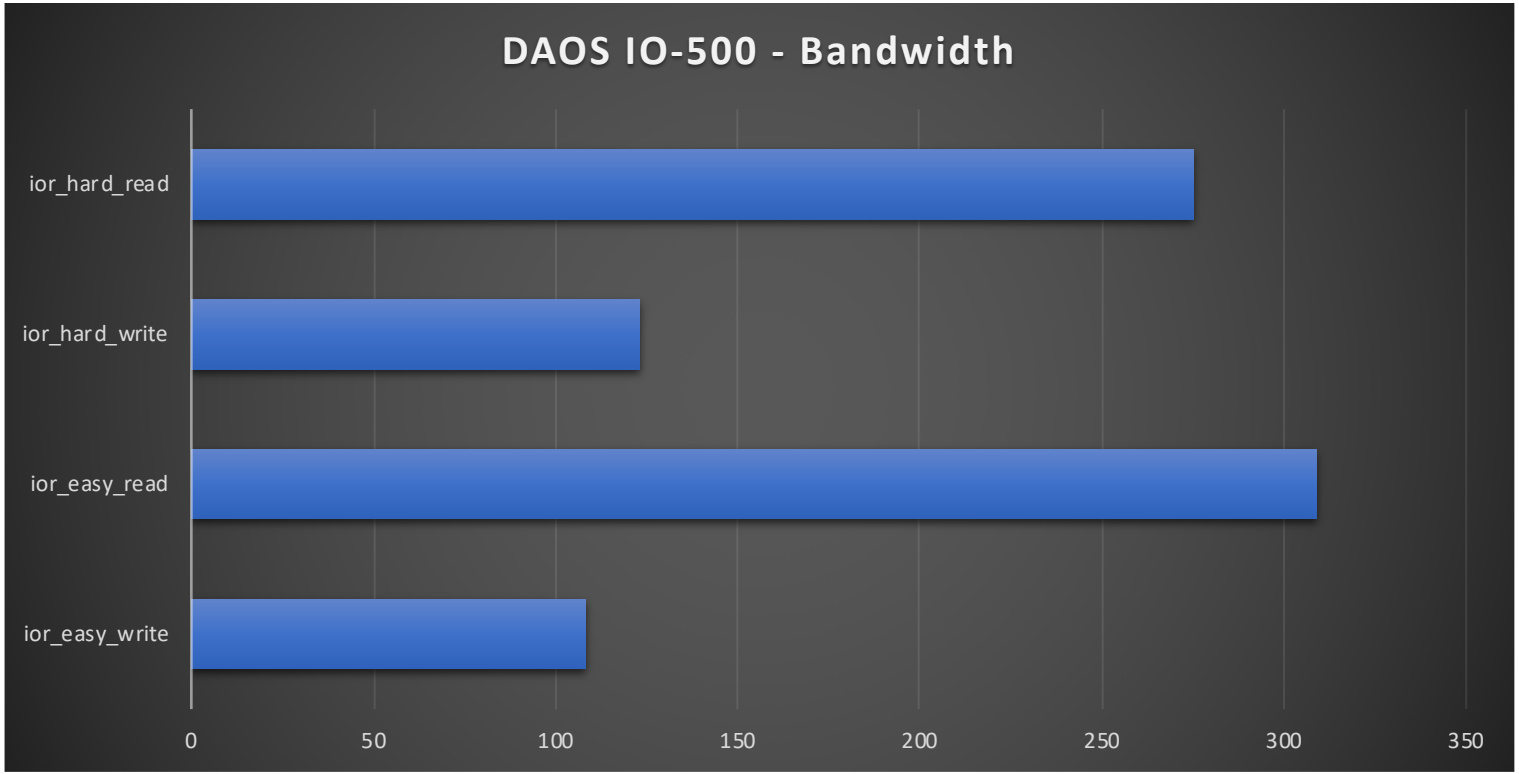
Convergence of AI, HPDA, and traditional HPC

- Need a storage system that can simultaneously support next gen workflows, where the different workflows can exchange data and communicate at high levels of performance
- DAOS is the answer!

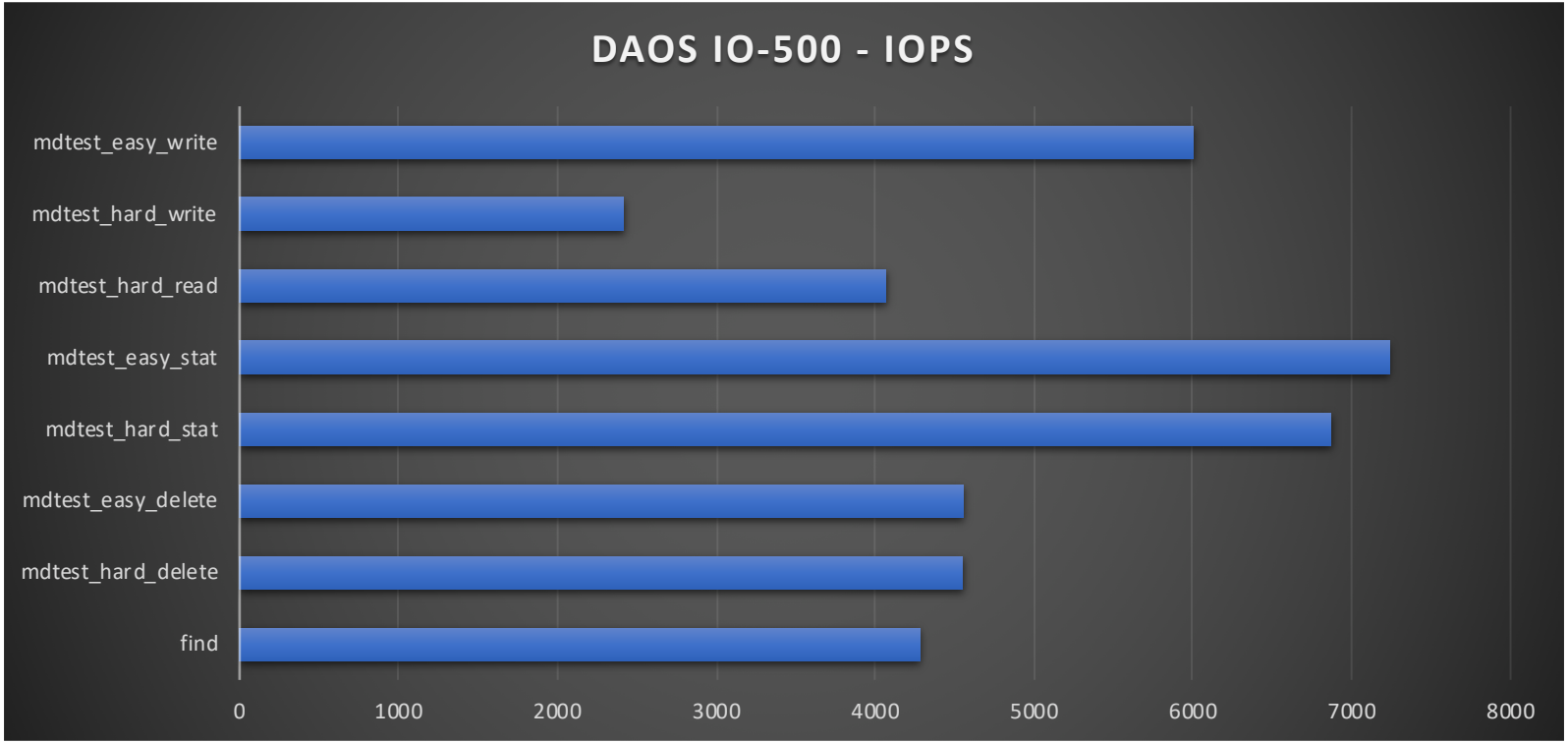
DAOS & IO-500 – 10-node Challenge



DAOS & IO-500 - Bandwidth



DAOS & IO-500 - IOPS



DAOS Tier Anatomy

DAOS Tier

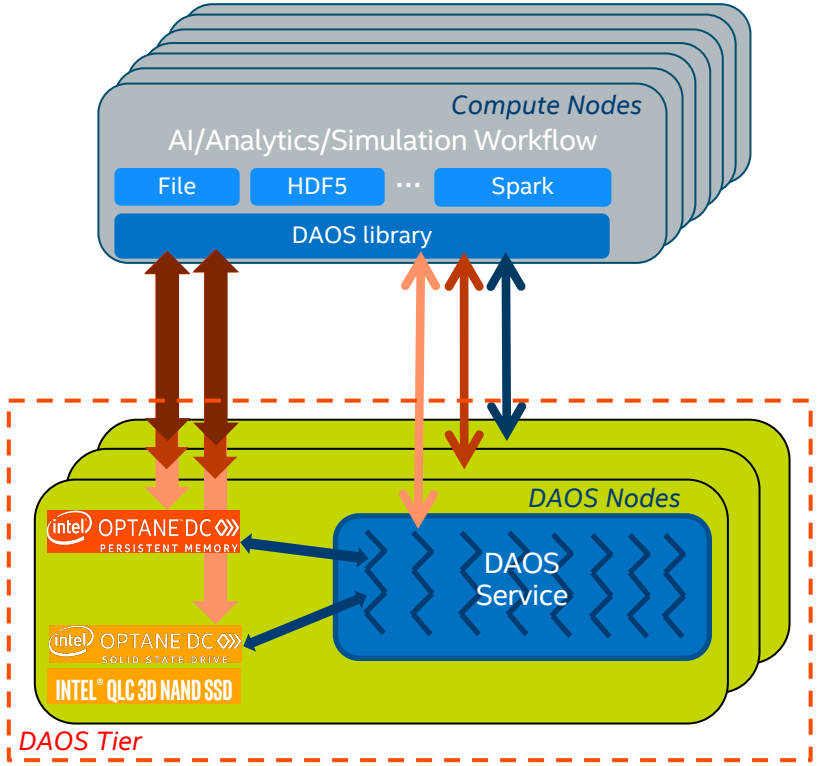
- Globally accessible from any compute nodes
- Large capacity (100's PB)

DAOS Storage Nodes

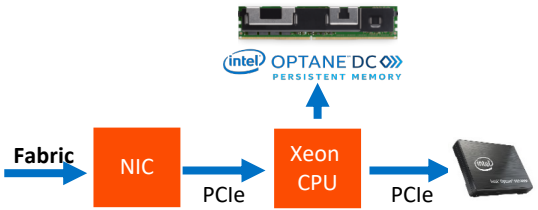
- COTS Intel® Xeon servers running the DAOS service
- RNIC attached for communications
- Support multiple RNICs per server to sustain backend storage IOPS/bandwidth

Mix of storage technologies attached

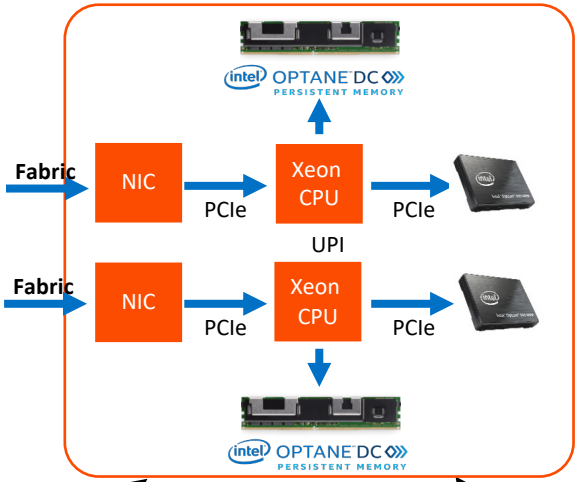
- Intel® Optane™ DC Persistent Memory (DCPMM)
- NVMe SSD (*NAND, Intel® Optane™ SSDs)



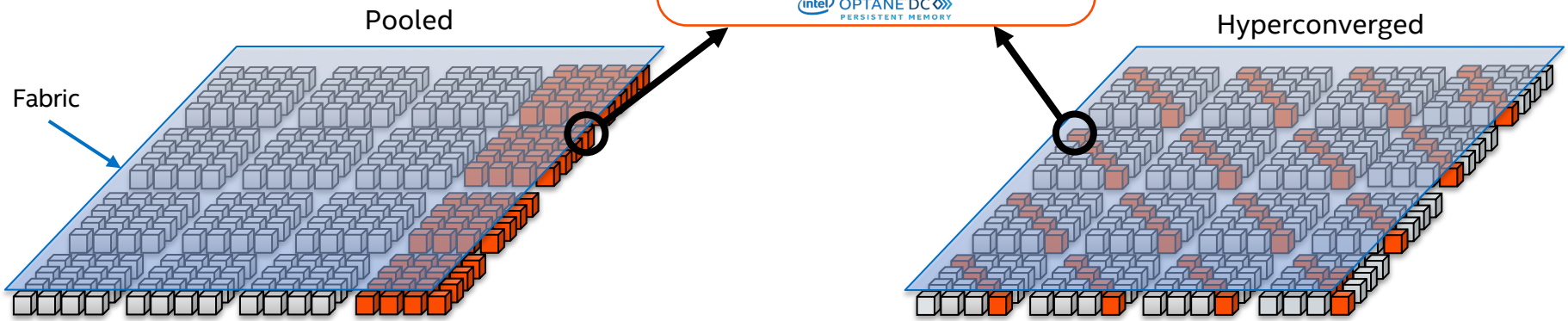
DAOS Deployments



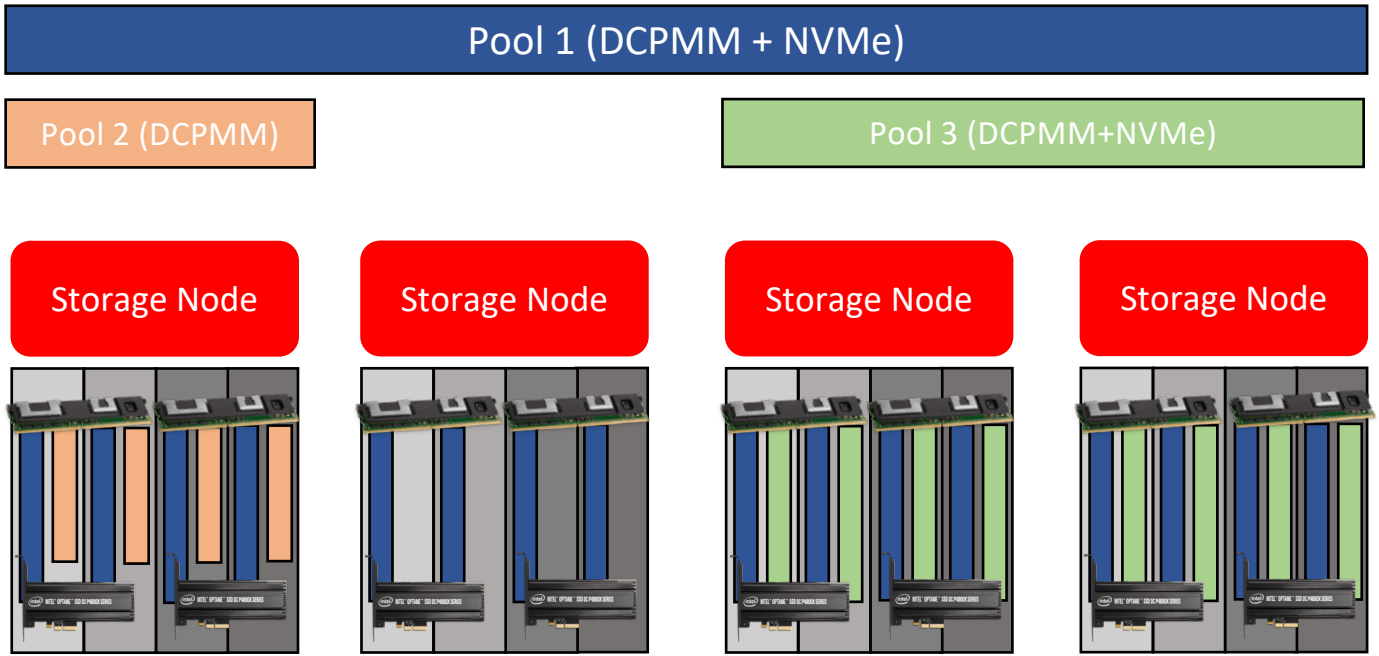
Baseline Building Block
Run one DAOS server instance



Dual-Socket DAOS Nodes (DNs)
Intel® Xeon servers with DCPMM & NVMe SSDs



STaaS: Storage Virtualization & Multi-Tenancy



Pool Specifications

Predicatable capacity

Fast creation/destroy (seconds)

Can be extended

Can be resized

Security ACL

Dedicated properties

- % space reserved for self-healing
- Space reclaim strategy

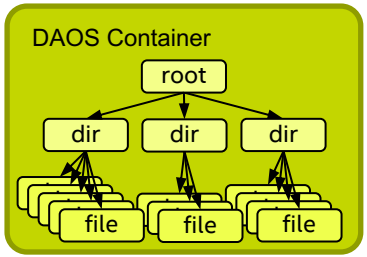
Use cases

- One pool per job
  - Ephemeral pools with short lifetime
  - Integration with resource manager required
- One pool per project
  - Persistent (multiple months/years)
  - Data movement less frequent
- Single pool with a few POSIX containers
  - Use DAOS like a traditional PFS

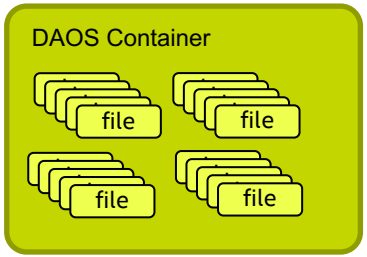
Storage Containers

Aggregate related datasets into manageable and coherent entities

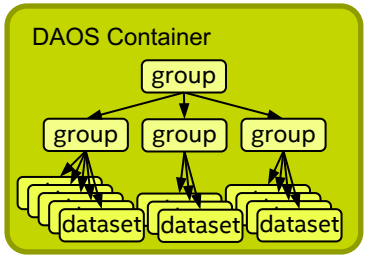
- Distributed consistency & automated recovery
- Full Versioning
- Simplified data management
  - Snapshot
  - Cross-tier Migration
  - Indexing



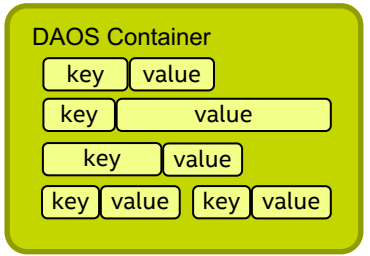
Encapsulated POSIX Namespace



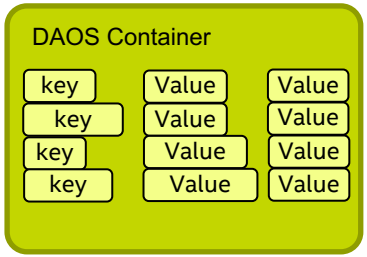
File-per-process



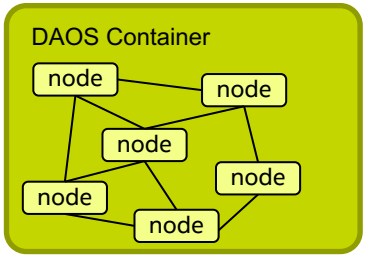
HDF5 « File »



Key-value store



Columnar Database



Graph

Container Specifications

Container = dataset

Share pool space

Security ACL

Dedicated properties

- Layout type/version
- Checksum on/off & type
- Max snapshots
- Redundancy factor
- Future: compression, encryption,…

Use cases

- One container per HDF5 file
- One container for all FPP
- One container as a SSF
- One container per KV store
- One container per compute node

End-to-end Data Integrity

Detect silent data corruption

- Checksum computed by client on write (crc16/32/64 & sha)
- Stored in persistent memory
- Verified by server/client during I/O

Correct data corruption

- Data is protected by replication or Erasure coding

Degraded mode

- Disable I/O to corrupted object shard

Security: Pool/Container ACL

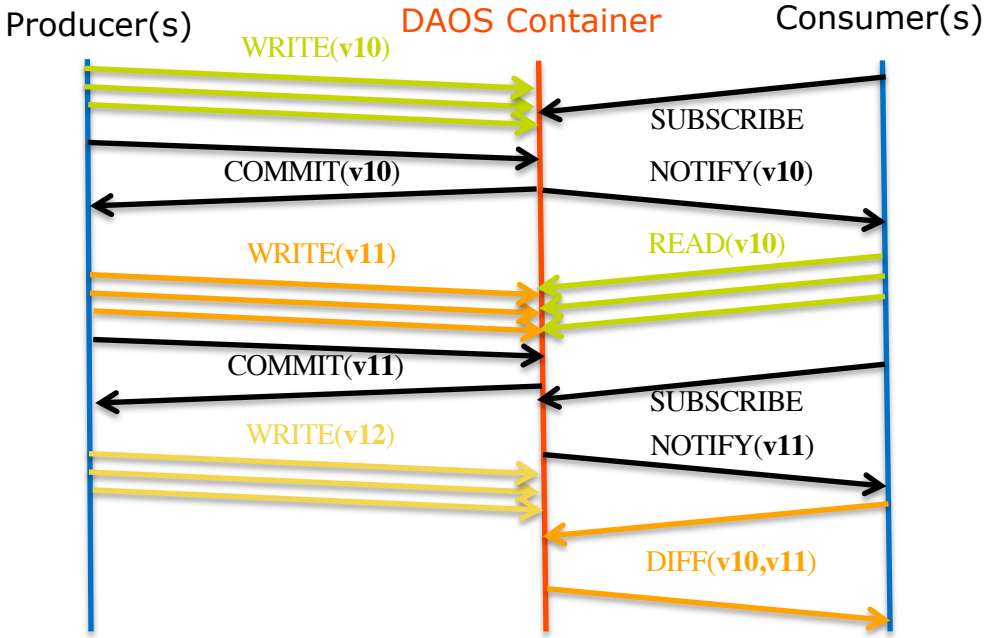
Permission	Pool	Container	Description
r	X*	X	read-data from a resource
w	X*	X	write-data to a resource
c	X		create a container in a pool
d	X	X	delete a container from a pool
t	X	X	read-properties – read resource properties
T		X	write-properties – write resource properties
a		X	read-ACL – read the ACL entries on a resource
A		X	write-ACL – modify the ACL entries on a resource
o		X	write-owner – change ownership of a resource

*Aliases for sets of permissions

Data-driven Workflow

DAOS **simplifies** developing **complex** workflows

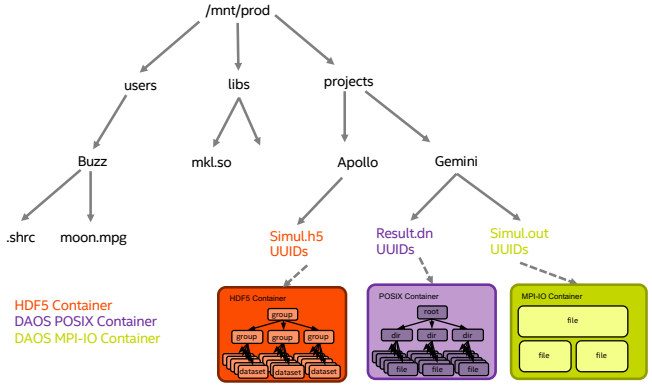
- **Native** producer/consumer **pipeline** support
- **Concurrency** & **dataflow** control
- Container **versioning**
  - Allow incremental update when maintaining external data structures
  - Many use cases like data indexing, visualization, ML, incremental backup, …



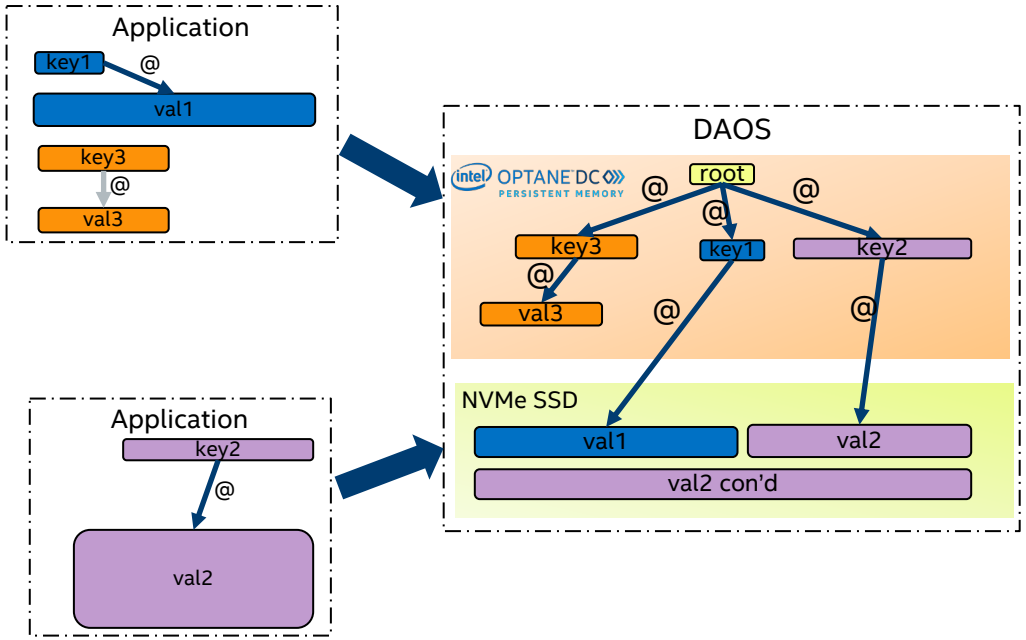
Unified Namespace

Allow users to create links between a file/dir in a system namespace to containers:

- daos container create --path=/mnt/project1/userA/NS1 --pool=uuid --type=POSIX/HDF5/etc.
- Path created above becomes a special file or directory (depending on container type) with an extended attribute with the pool and container information.
- Accessing that path from DAOS aware middleware will make the link on the fly with the DAOS UNS library.

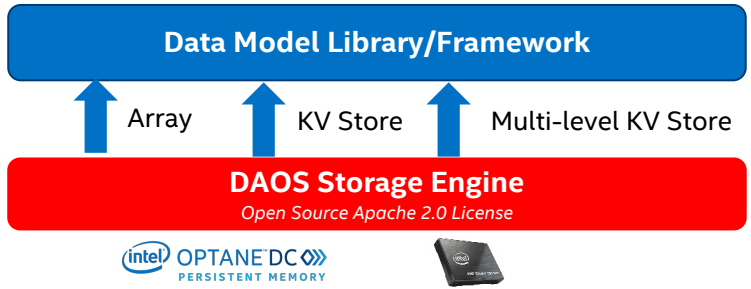


DAOS Objects



Native support for structured, semi-structured & unstructured data models

- Built on top of persistent memory
- Unconstrained by POSIX serialization
- Custom attributes
- Data access time orders of magnitude faster (µs)
- Scalable concurrent updates & high IOPS
- Non-blocking
- Enable in-storage computing



Data Protection and Self-Healing - Replication

Data replication in DAOS

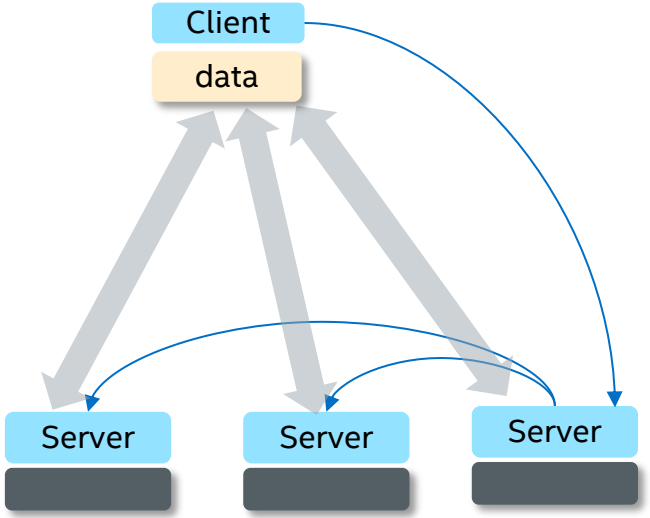
- Primary-slave replication
- Distributed transaction for atomicity

Degraded mode

- Non-blocking protocol for server fail-out

Online data recovery

- Low impact on ongoing I/O
- Declustered data reconstruction



Data Protection and Self-Healing – Erasure Code

Erasure code in DAOS

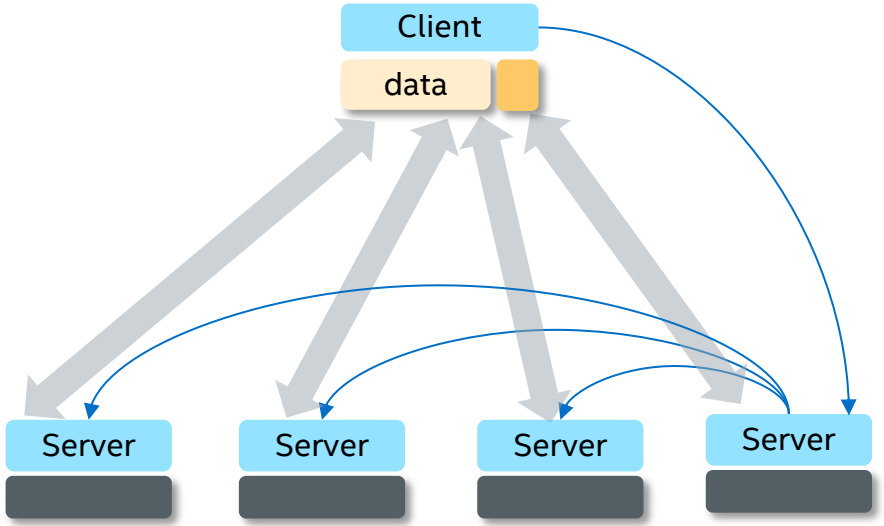
- Computed by client on write
- Distributed transaction for atomicity
- Replication for partial write
  - Merge and encode by server

Degraded mode

- Non-blocking protocol for server fail-out
- Client side inflight data reconstructing

Online data recovery

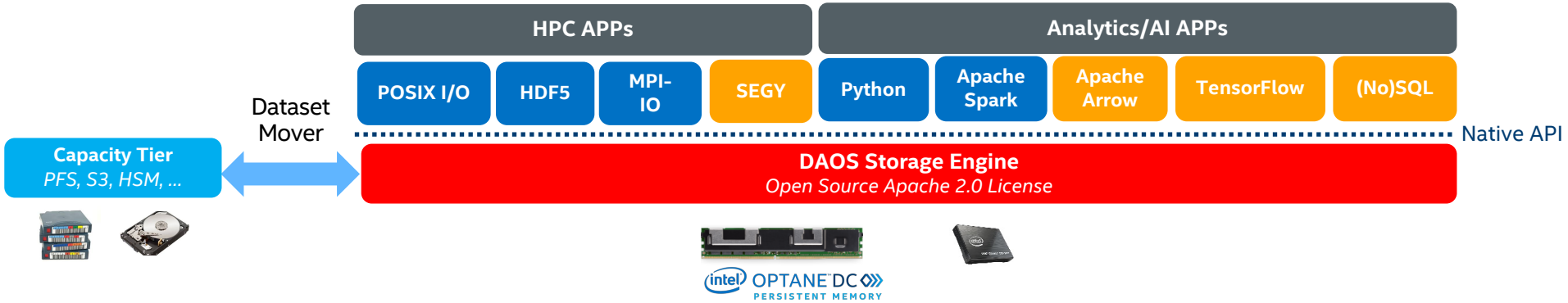
- Server side data exchange and reconstruction



DAOS Tools

Tool	dmg	daos
Target	Administrators	Users
Lustre Equivalent	lctl/mkfs/mount/IML	lfs
Functionality	• Storage provisioning • Burn-in • Firmware update • Data plane mgmt & monitoring • Configure/monitor scrubbing • Pool mgmt • Telemetry	• Pool query • Container mgmt • Unified namespace mgmt • Container user attributes • Snapshots • Object debugging • POSIX container configuration

Application Interface



POSIX I/O Support

DAOS File System (libdfs)

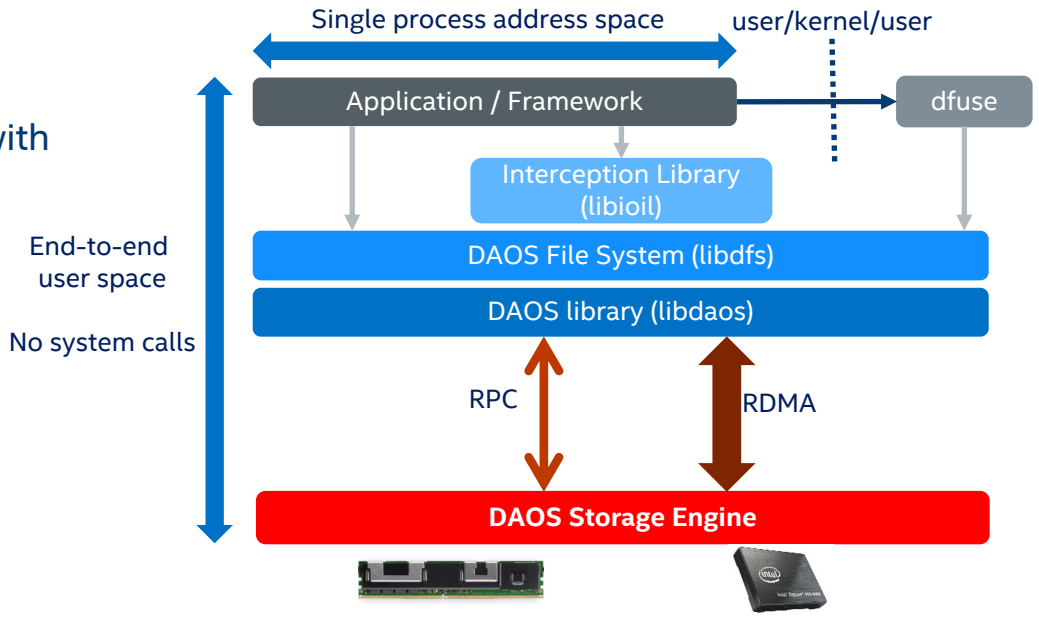
- Encapsulated POSIX namespace
- Application/framework can link directly with libdfs
  - ior/mdtest backend provided
  - MPI-IO driver leveraging collective open
  - TensorFlow, …

FUSE Daemon (dfuse)

- Transparent access to DAOS
- Involves system calls

I/O interception library

- OS bypass for read/write operations



DFUSE

A FUSE daemon provided to mount a POSIX container in the local filesystem tree.

To mount an existing POSIX container with dfuse, run the following command:

- *dfuse --pool pool_uuid --container cont_uuid -m /tmp/daos*

Can access files / directories as any namespace in the container.

Libioil:

- This library works in conjunction with dfuse and allow to interception of POSIX I/O calls and issue the I/O operations directly from the application context through libdaos without any application changes.
- This provides kernel-bypass for I/O data leading to improved performance. To use this set the LD_PRELOAD to point to the shared libray in the DOAS install dir
  - LD_PRELOAD=/path/to/daos/install/lib/libioil.so

DFS Library

DFS Client Library applications can link with.
Encapsulated POSIX Namespace
Like POSIX API for easy changes to IO to use DFS.
Examples:

- IOR & mdtest DFS driver (https://github.com/hpc/ior)

```
…
fd = open(file_name, O_CREAT|O_RDWR, 0600);
/** set up iov */
…
pwritev(fd, iov, 1, offset);
preadv(fd, iov, 1, offset);
close(fd);
```

```
…
dfs_open(dfs, NULL, file_name, 0600, O_CREAT|O_RDWR, 0,
0, NULL, &file);
/** setup sgl (DAOS iov) */
…
dfs_write(dfs, file, &sgl, offset, NULL);
dfs_read(dfs, file, &sgl, offset, &bytes_read, NULL);
dfs_release(file);
```

DFS API

POSIX	DFS
mkdir(), rmdir()	dfs_mkdir(), dfs_rmdir()
open(), close(), access()	dfs_open(), dfs_release(),dfs_lookup()
pwritev(), preadv()	dfs_read/write()
{set,get,list,remove}xattr()	dfs_{set,get,list,remove}xattr
stat(), fstat()	dfs_stat(),ostat()
readdir() …	dfs_readdir() …

Mostly 1-1 mapping from POSIX API to DFS API.
Instead of File & Directory descriptors, use DFS objects.
All calls need a DFS mount which is usually done on initialization with the pool / container access handles.

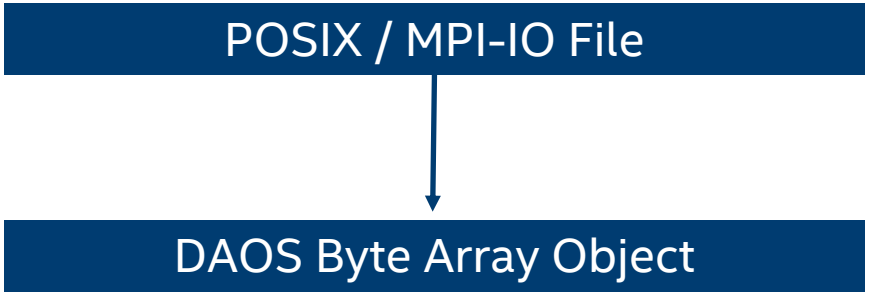
MPI-IO Driver for DAOS

The DAOS MPI-IO driver is implemented within the I/O library in MPICH (ROMIO).

- Added as an ADIO driver
- Portable to Open-MPI, Intel MPI, etc.
- https://github.com/pmodels/mpich

MPI Files use the same DFS mapping to the DAOS Object Model

- MPI Files can be accessed through the DFS API
- MPI Files can be accessed through regular POSIX with a dfuse mount over the container.

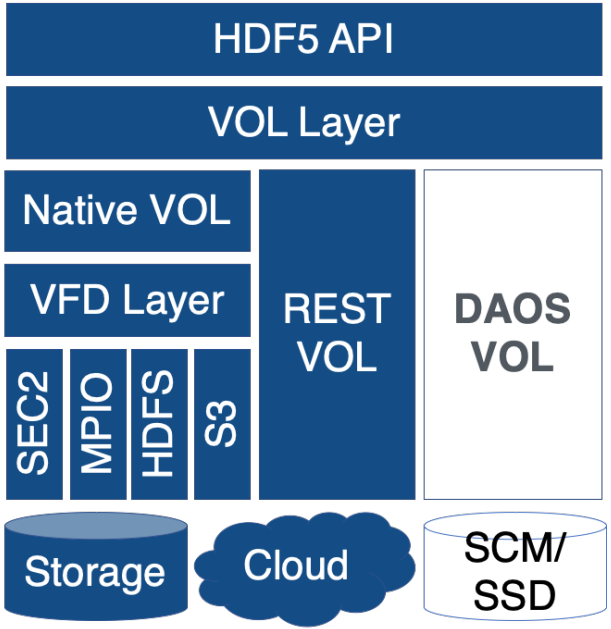


Special DAOS Object:

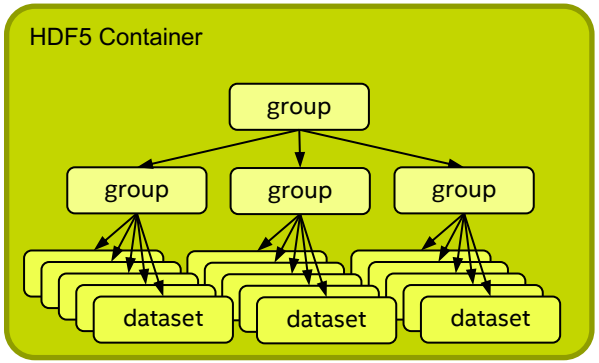
- 1 Level Key
- 1 Byte records
- Configurable Chunk Size

Application works seamlessly by just specifying the use of the driver by appending "`daos:`" to the path.
MPI-IO ROMIO driver (https://github.com/pmodels/mpich/tree/master/src/mpi/romio/adio/ad_daos)

HDF5®



- Developing an HDF5® VOL Connector
- Minimal / No application code changes (including other middleware I/O libraries (e.g. NetCDF4, PIO, etc.)



Adding new extensions to the HDF5® library that are not available to date without the DAOS VOL connector

- Asynchronous I/O for both metadata and raw data operations
- Container Snapshots
- Independent Metadata Operations

HDF5 VOL Connector

Available from: https://git.hdfgroup.org/projects/HDF5VOL/repos/daos-vol/

- See user's guide for more detailed list of supported features

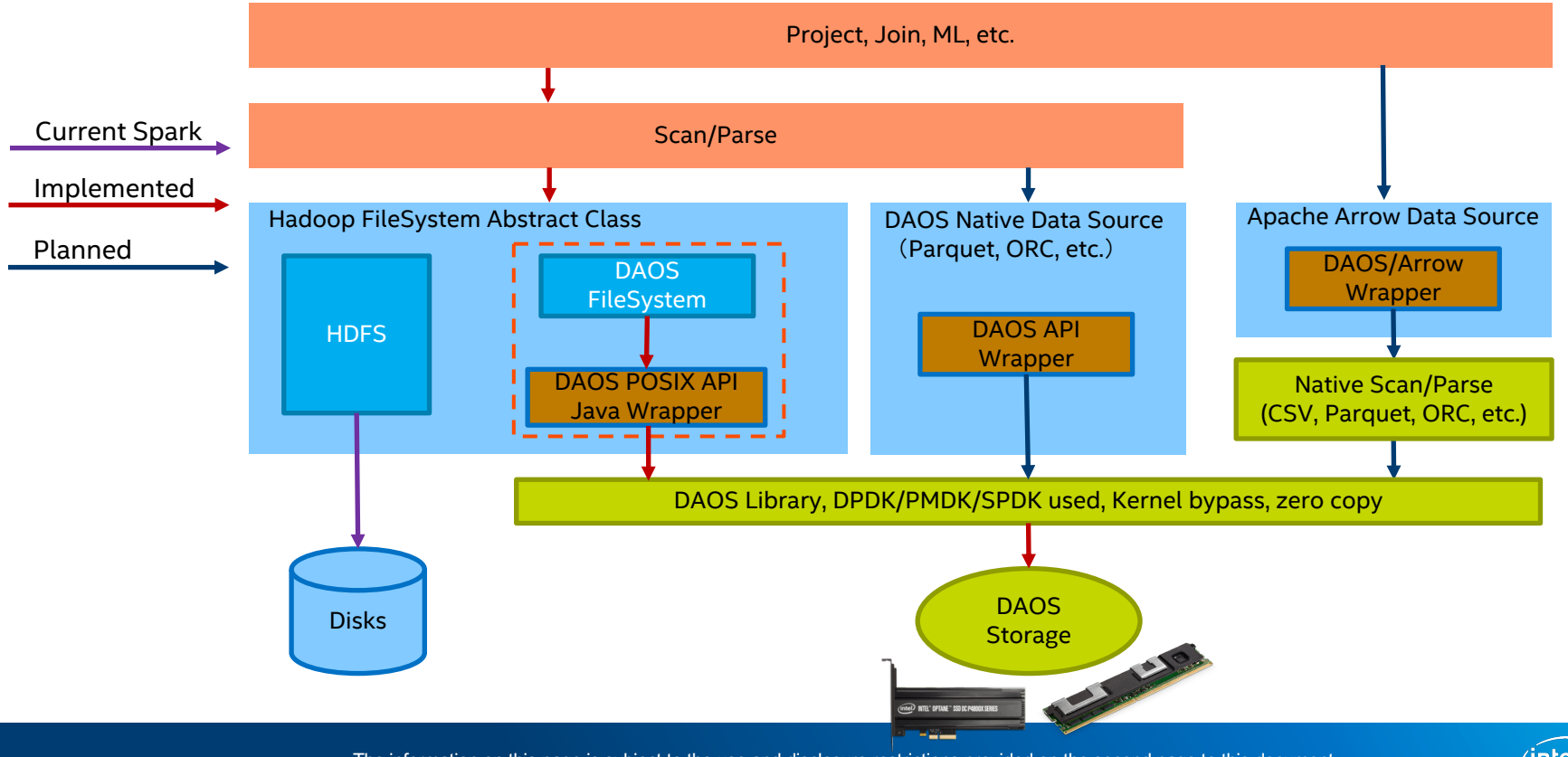
Dynamically loaded plugin **(No change to user application)**:

- export HDF5_PLUGIN_PATH=/path/daos-vol/lib/
- export HDF5_VOL_CONNECTOR=daos

How to set/get pool & container uuid then?

1. Use env variables (User passes those).
2. Unified Namespace with special file storing pool/container as extended attributes (not implemented yet).

Spark Input/Output Support



DAOS/Spark SC19 Demonstration

Python Support

Pythonic bindings called pydaos

- Export key-value store objects
- Integrated with python dictionnaries
  - Support python iterator, direct assignments, …
- Scalable & performant
  - Bulk insert/retrieve
  - Core written in C
- Python 2.7 & 3 support

Python integration

- dbm
- pyprob

TODO

- Expose snapshots
- Integration with NumPy, …

Application Example: Etalumis

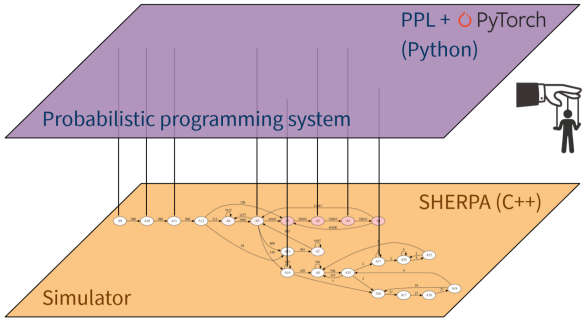
Deep-Learning based Probabilistic Programming System

- PyTorch
- C++ Sherpa Simulator
- pyprob, ppx
- dataset with 15 Million traces, 1.7 TB, 100k traces/file, 150 files total

Contact: Lei Shao <lei.shao@intel.com>

Porting effort

- Integrate pyprob with daosdbm
- Migrator tool
  - One container with all traces



Advanced Usage: Distributed Transactions

daos_tx_open(): creates an transaction handle with a global unique identifier. It is a client local operation without RPC to any server.

daos_tx_commit(): flush all the updates that are cached on that transaction handle to the coordinator with a single compound DAOS RPC.

daos_tx_abort(): abort the transaction. All related modifications will be discarded.

daos_tx_close(): closes the transaction and frees local handle resources. This is a local operation.

daos_tx_open_snap(): open a read-only transaction using an epoch of a persistent snapshot.

Advanced Usage: Transaction Flow

```
daos_tx_open(coh, &th, ...);
restart:
   daos_obj_fetch(..., th, ...);
   daos_obj_update(..., th, ...);
   daos_obj_fetch(..., th, ...);

   daos_obj_update(..., th, ...);
   daos_obj_punch(..., th, ...);

   rc = daos_tx_commit(th, ...);
   if (rc == -DER_RESTART)
       goto restart;
daos_tx_close(th, ...);
```

Just a local operation
Select transaction coordinator

Reads are served from servers
Writes are buffered on client

Coordinator executes:

1. Log transaction START & updates
2. Execute updates
3. Log transaction COMMIT or ABORT
4. Notify all targets of COMMIT or ABORT

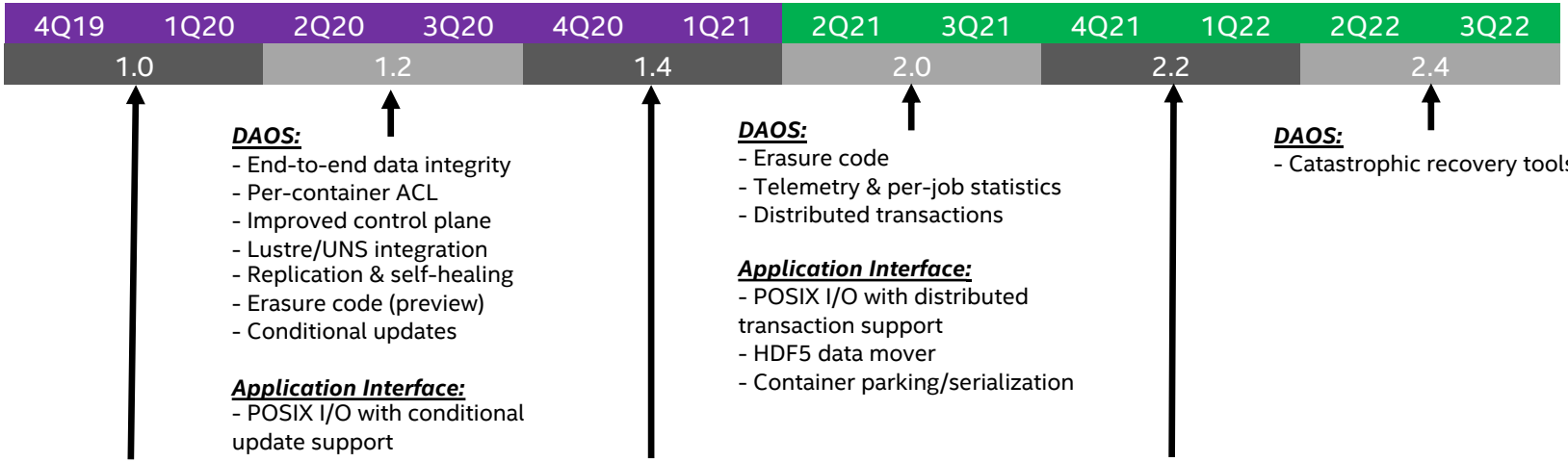
Visibility

Not visible
Cannot read what is modified

Need to check if tx is committed

All updates globally visible

DAOS Community Roadmap



DAOS:
- NVMe & DCPMM support
- Per-pool ACL
- UNS in DAOS via dfuse
- Replication & self-healing (preview)

Application Interface:
- MPI-IO Driver
- HDF5 DAOS Connector
- Basic POSIX I/O support
- Spark

DAOS:
- End-to-end data integrity
- Per-container ACL
- Improved control plane
- Lustre/UNS integration
- Replication & self-healing
- Erasure code (preview)
- Conditional updates

Application Interface:
- POSIX I/O with conditional update support

DAOS:
- Online server addition
- Advanced control plane
- Multi OFI provider support

Application Interface:
- POSIX data mover
- Async HDF5 operations over DAOS

DAOS:
- Erasure code
- Telemetry & per-job statistics
- Distributed transactions

Application Interface:
- POSIX I/O with distributed transaction support
- HDF5 data mover
- Container parking/serialization

DAOS:
- Progressive layout
- Placement optimizations
- Checksum scrubbing

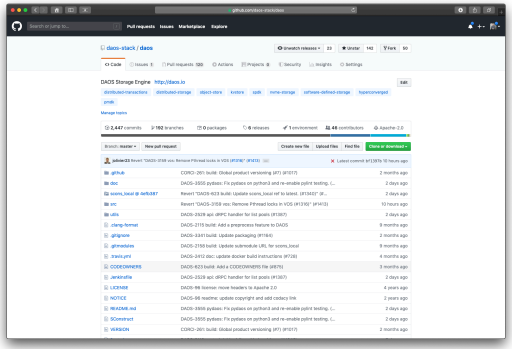
DAOS:
- Catastrophic recovery tools

NOTE: All information provided in this roadmap is subject to change without notice.

Resources

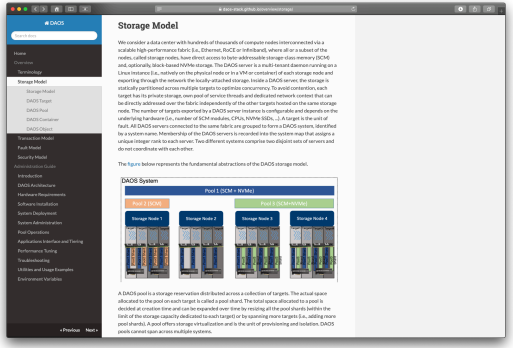
Source code on GitHub

- https://github.com/daos-stack/daos



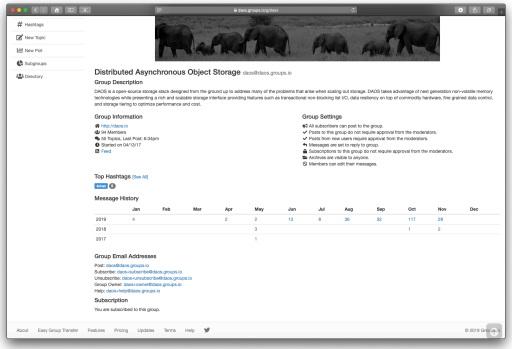
Documentation

- http://daos.io



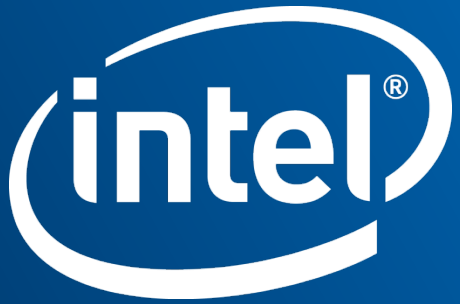
Community mailing list

- https://daos.groups.io



DAOS solution brief

- https://www.intel.com/content/www/us/en/high-performance-computing/


intel®