DAOS: Storage Innovations Driven by Intel® Optane™

Liang.zhen@intel.com

Principal Engineer, Intel

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.

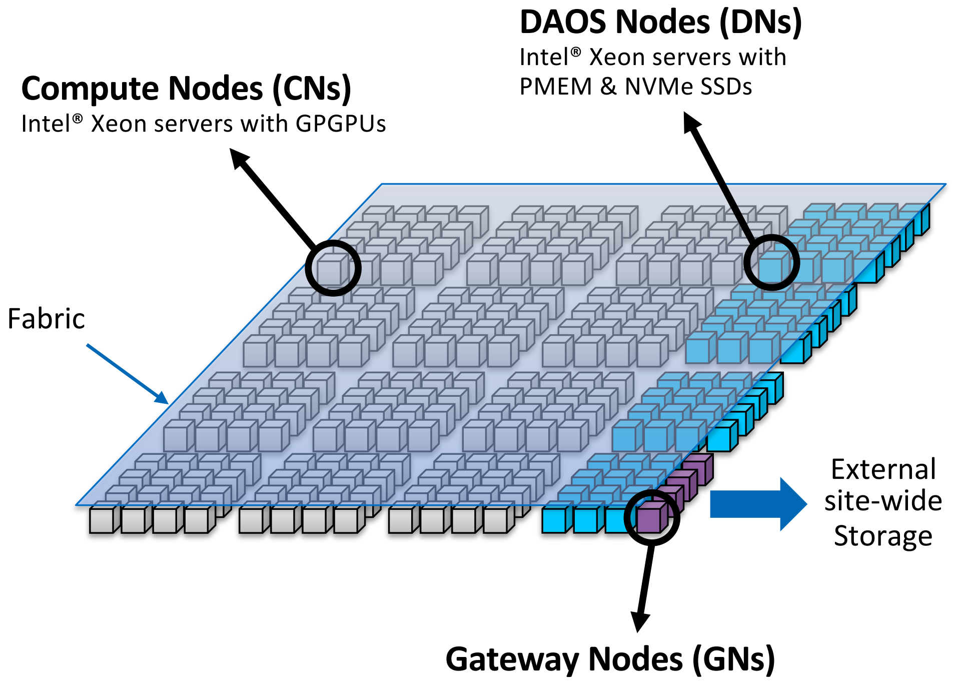
Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

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 overview

What is DAOS?

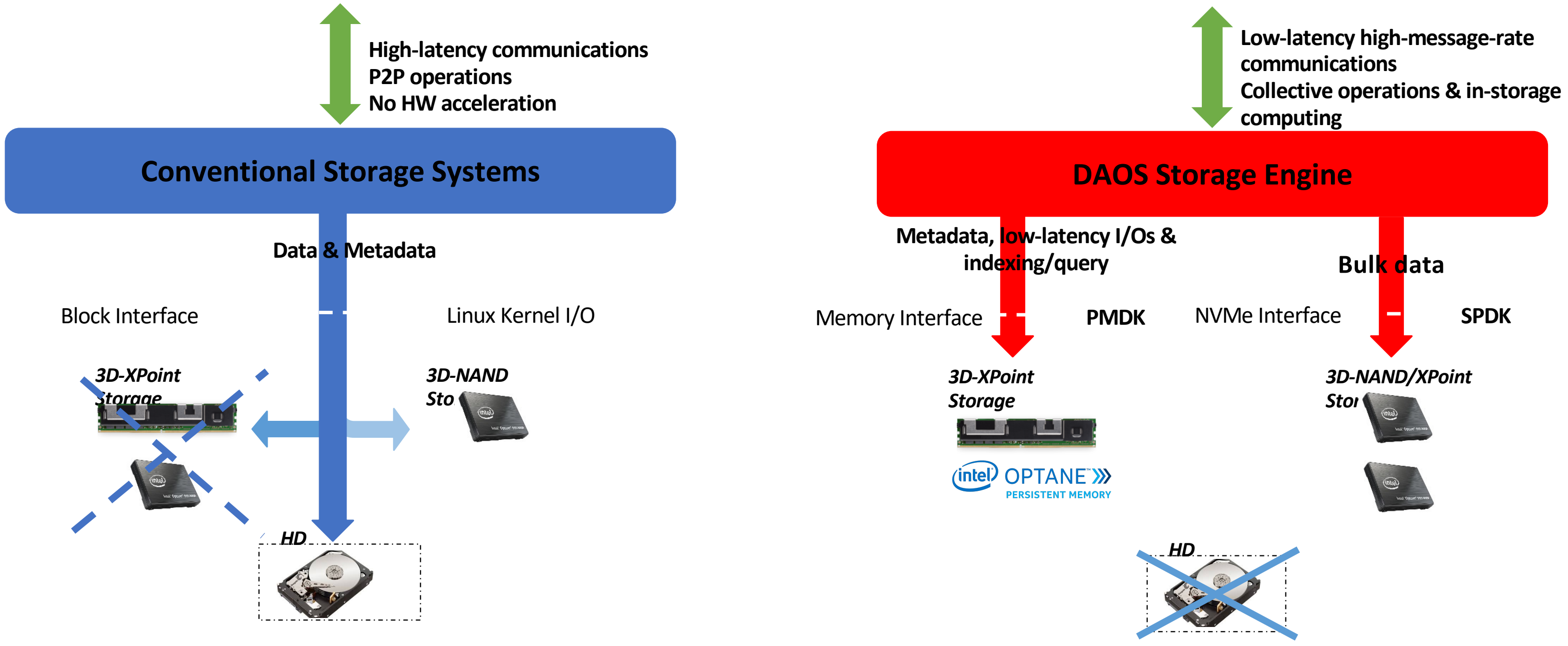
- Storage pools globally accessible over the fabric
- Overcomes industry bottlenecks by leveraging Intel® Optane™ Persistent Memory and NVMe SSDs
- Delivers exceptionally high bandwidth and IOPS, meeting the demands of HPC and AI
- Strong distributed consistency (Database like)
- Tightly integrated with Applications
- Can be deployed as either a standalone file system, or a performance tier integrated with existing storage systems



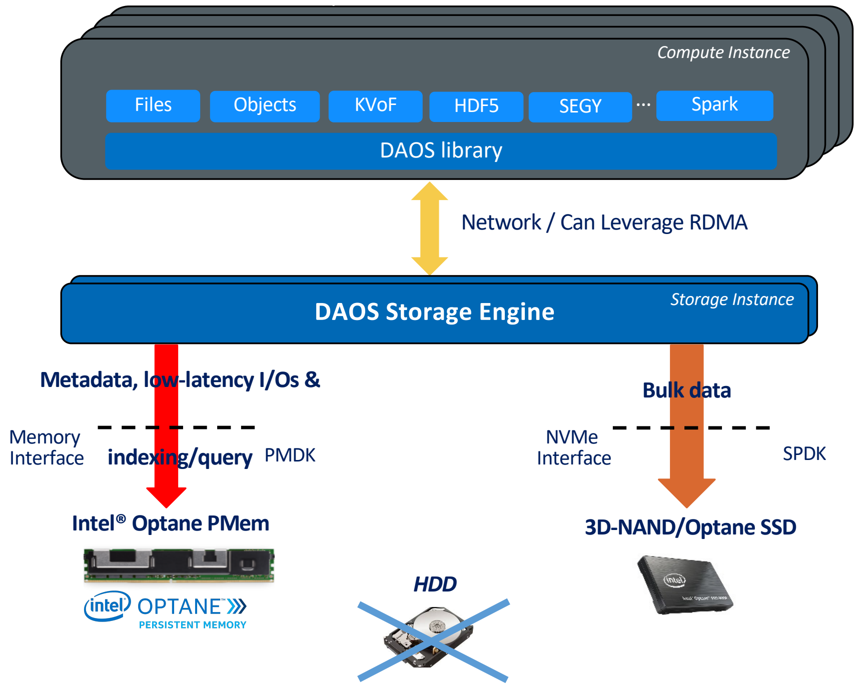
DAOS Nodes (DNs)
Intel® Xeon servers with PMEM & NVMe SSDs
Compute Nodes (CNs)
Intel® Xeon servers with GPGPUs
Fabric
External site-wide Storage
Gateway Nodes (GNs)

Generational leap forward in storage performance

DAOS architecture

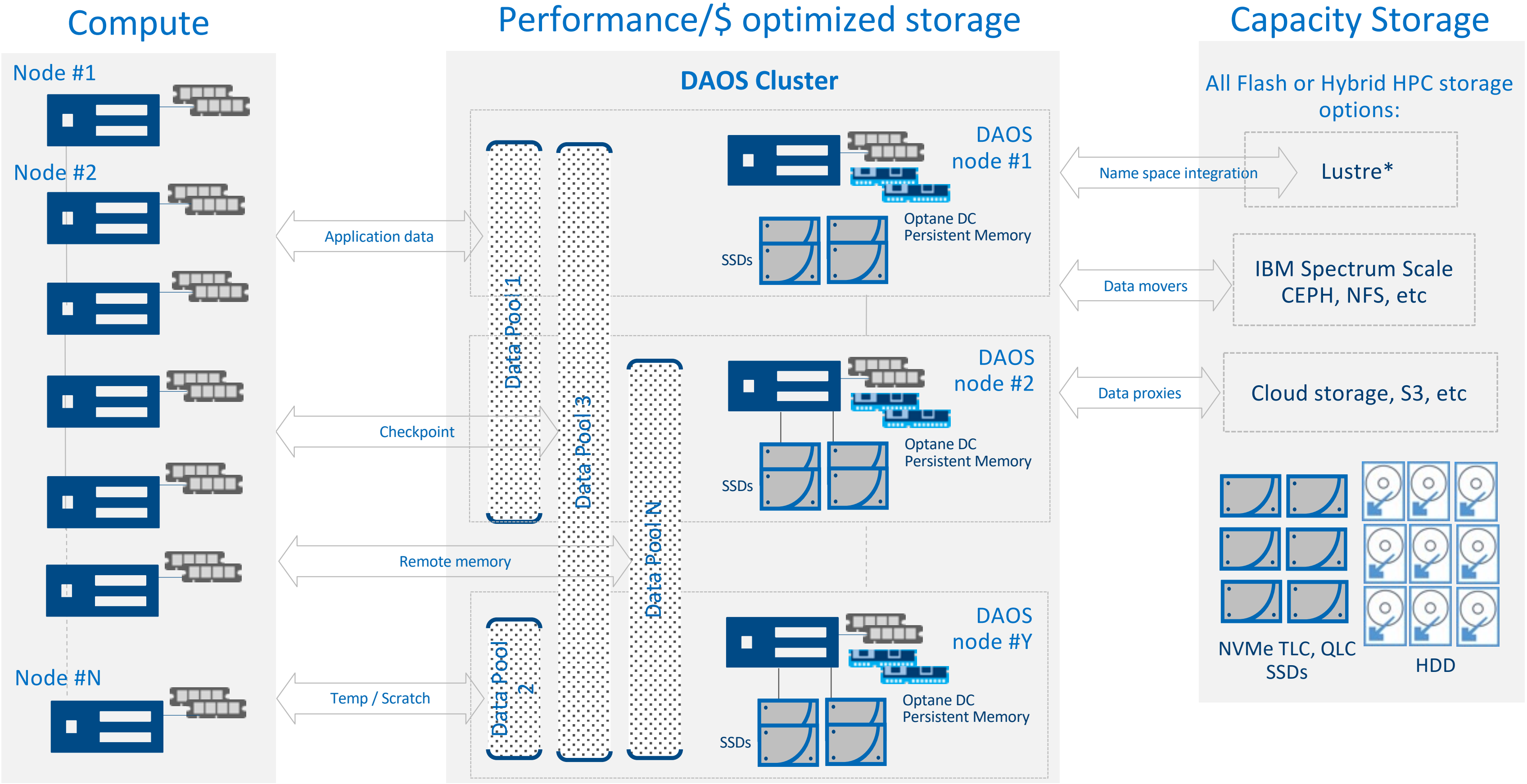


DAOS stack overview

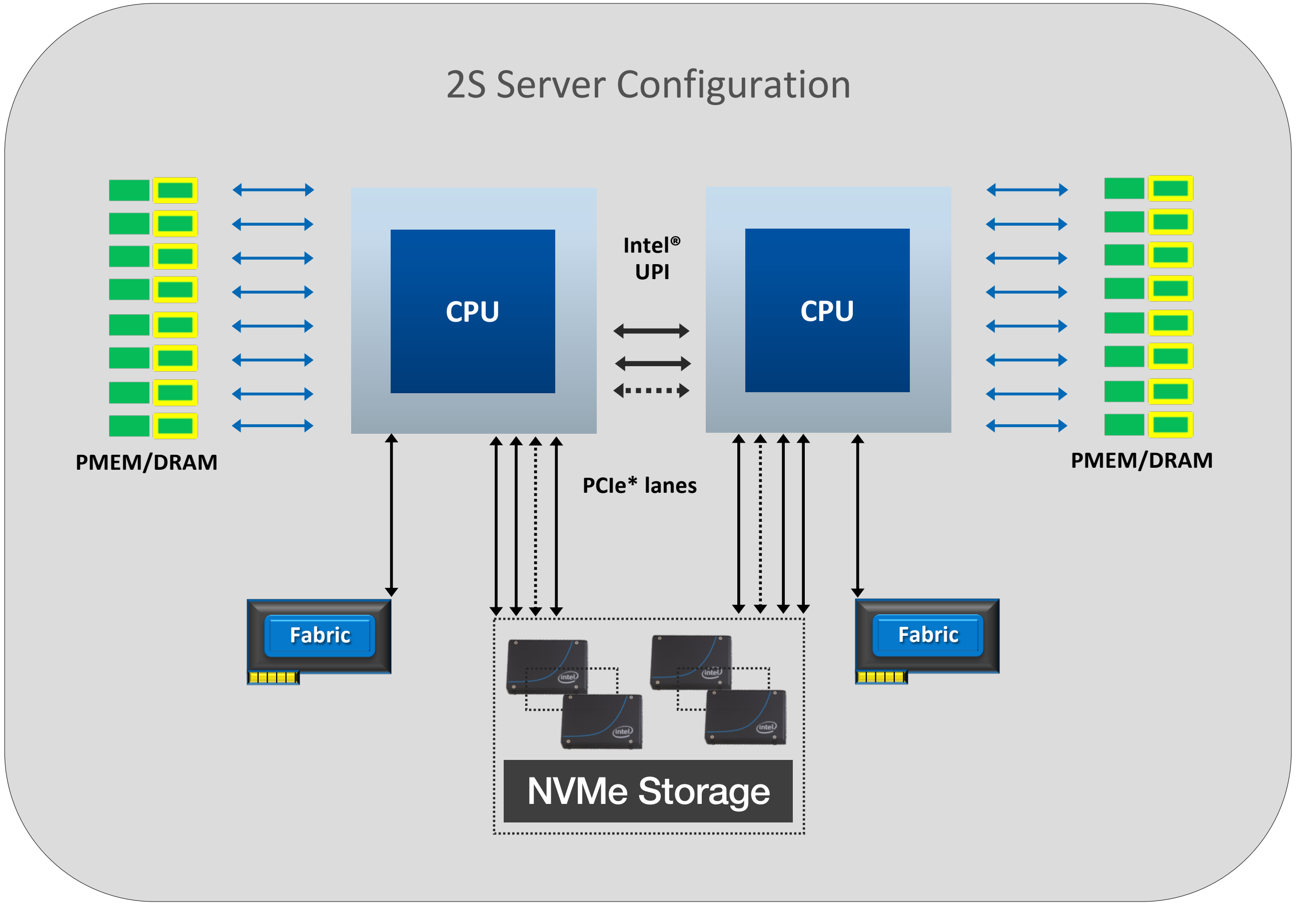


- High **throughput/IOPS @arbitrary** alignment/size (2M IOPS in 1U)
- **Low-latency fine-grained** I/O
- Data access time **orders of magnitude faster** (µs vs ms)
- Highly **scalable**
- Operates in **userspace**
- **Rich** storage semantics
- Support **smooth migration** with support in common frameworks such as Apache Spark*, MPI-IO, HDF, POSIX, etc

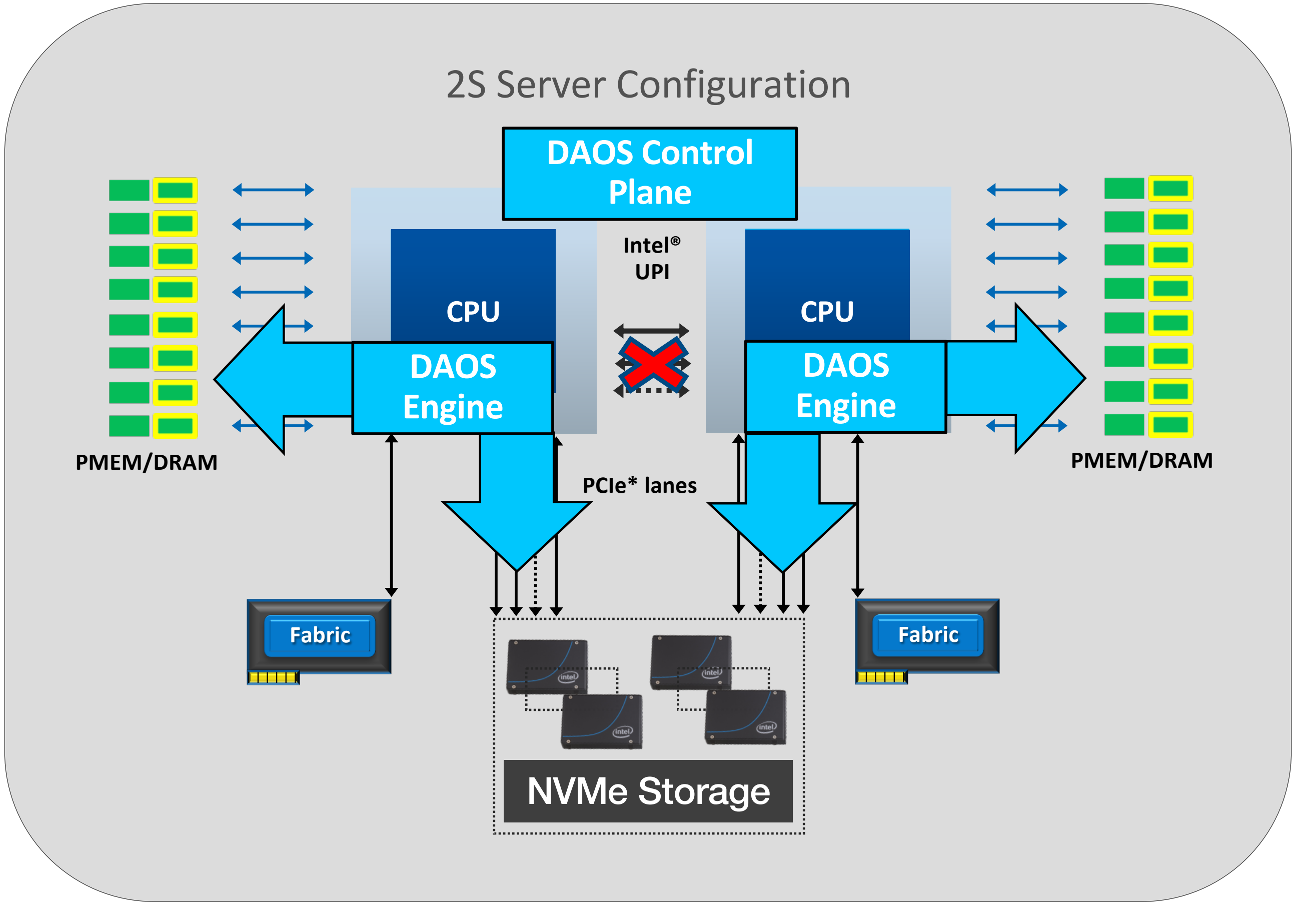
DAOS in the Overall Cluster Architecture



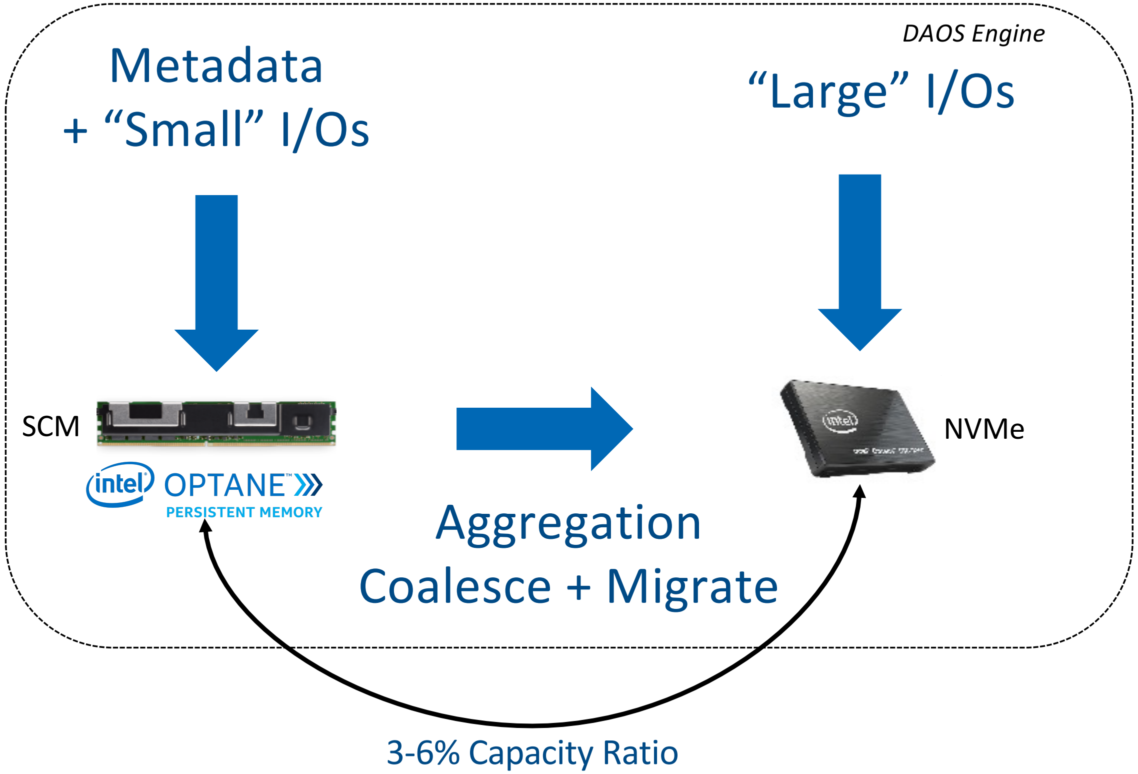
DAOS Node Design



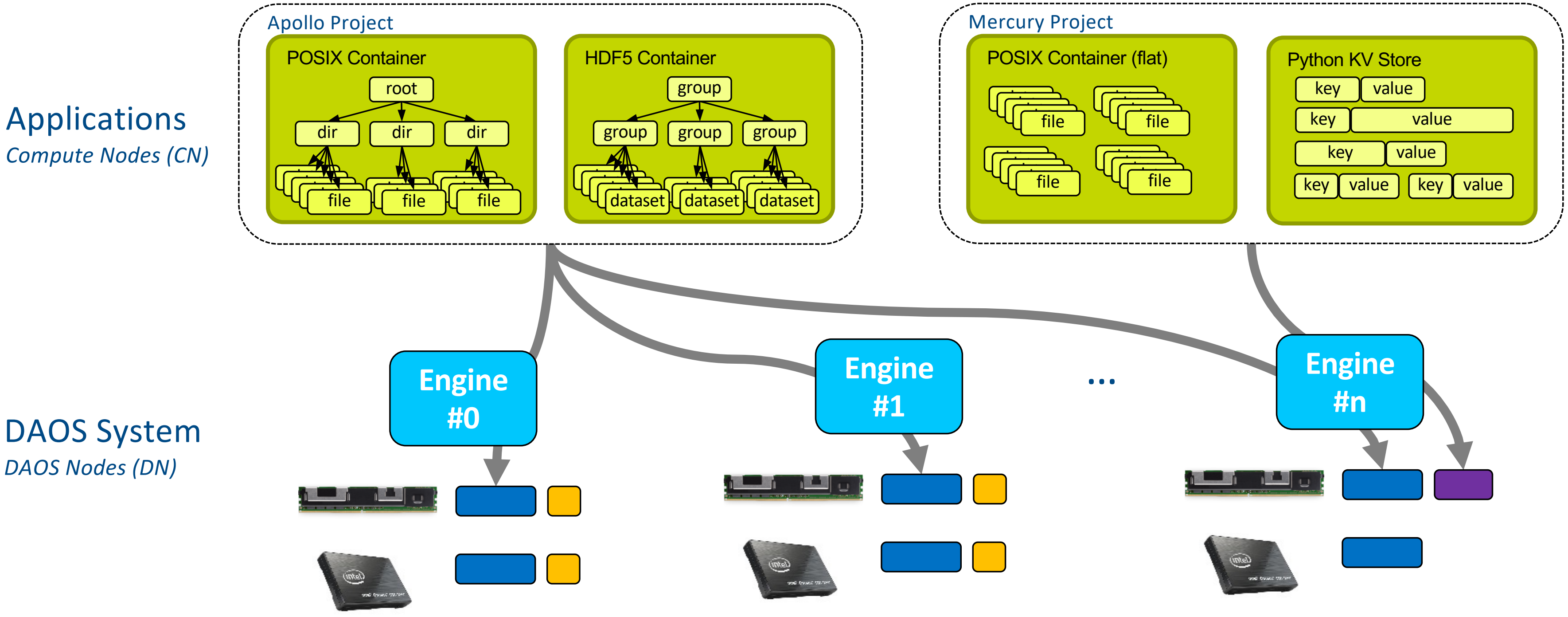
DAOS Node Design



Engine: Media Management



DAOS Data Model: Storage Pooling



Applications

Compute Nodes (CN)

Apollo Project

POSIX Container

root

dir

dir

dir

file

file

file

HDF5 Container

group

group

group

group

dataset

dataset

dataset

Mercury Project

POSIX Container (flat)

file

file

file

file

Python KV Store

key

value

key

value

key

value

key

value

key

value

DAOS System

DAOS Nodes (DN)

Engine #0

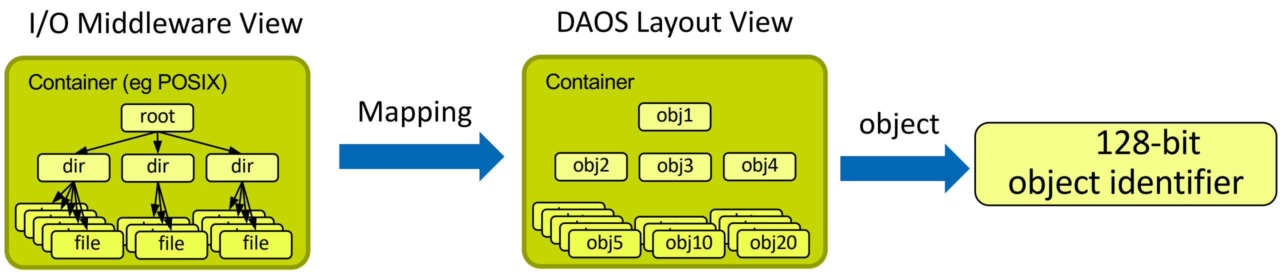
Engine #1

…

Engine #n

Pool 1	🟦	Project Apollo	100PB usable	20TB/s	200M IOPS
Pool 2	🟨	Project Gemini	10PB usable	2TB/s	20M IOPS
Pool 3	🟪	Project Mercury	30TB usable	80GB/s	2M IOPS

DAOS Data Model: Objects



I/O Middleware View

Container (eg POSIX)

root

dir dir dir

file file file

Mapping

DAOS Layout View

Container

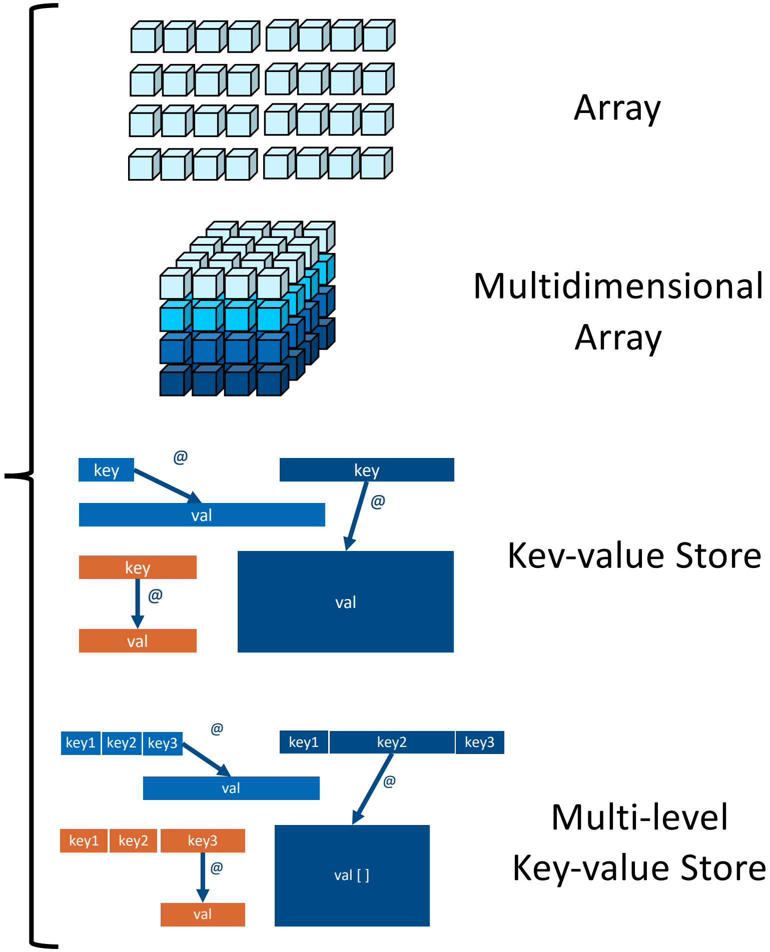
obj1

obj2 obj3 obj4

obj5 obj10 obj20

object

128-bit object identifier



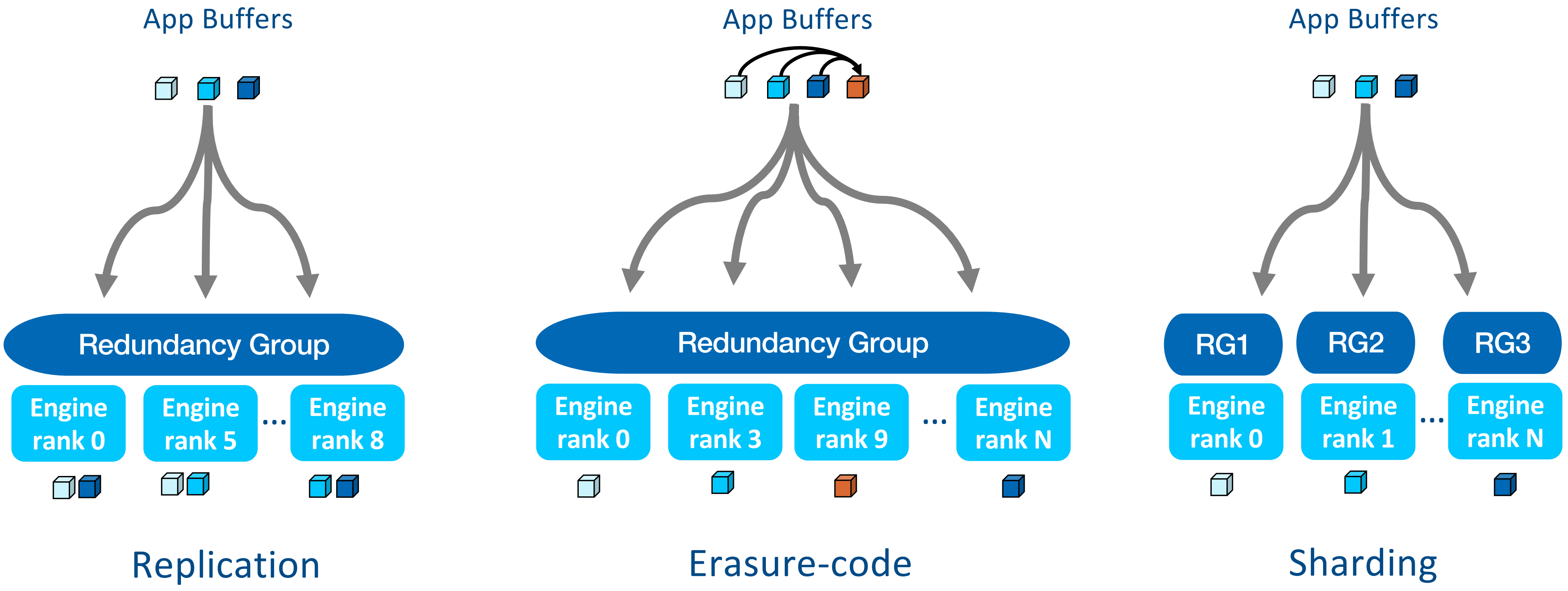
Array

Multidimensional Array

Kev-value Store

Multi-level Key-value Store

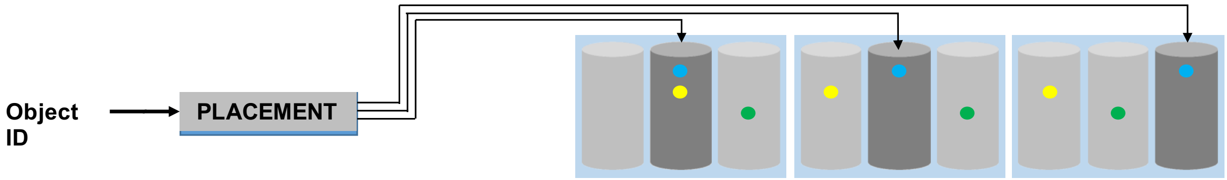
DAOS Data Model: Distribution & Fault Tolerance



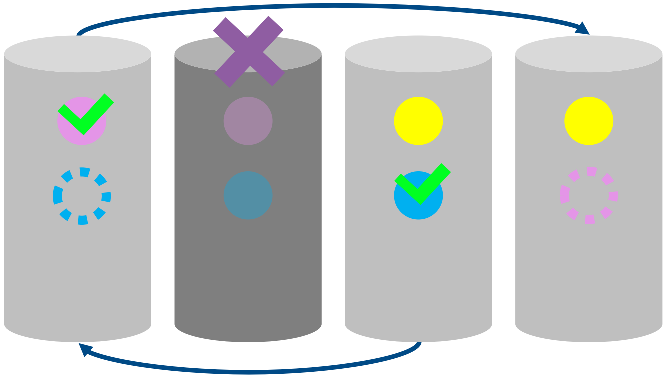
Algorithmic object placement

- No explicit object layout
  - Expensive to maintain
  - Extra round-trip to get the layout
- Object class
  - Data protection method
  - Data distribution requirements
  - Pre-defined attributes, identified by 16-bit integer (class ID)

- Object ID
  - Object class ID + 96-bit ID
- Algorithmic object placement
  - determines where those shards will be stored on the physical system (node, target) based on pool configuration and OID



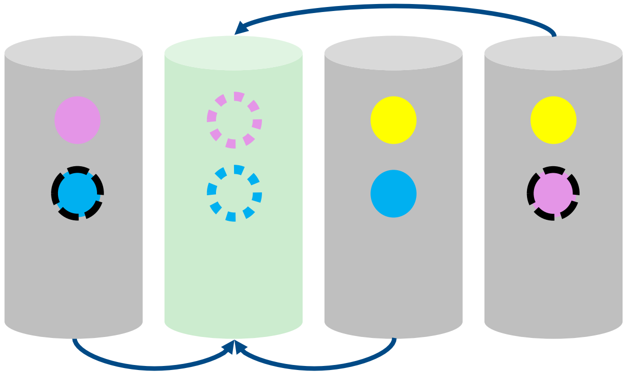
Automated Exclusion & Self-Healing



- Health monitoring
  - Detects failed nodes via SWIM
  - Failed nodes are automatically evicted

- Failure recovery (“rebuild”)
  - All other surviving nodes are notified of the failure
  - Impacted objects are automatically determined and reconstructed on surviving nodes

Reintegration ("reintegrate")

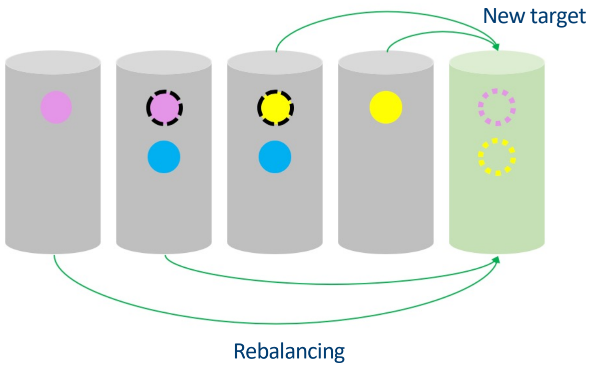


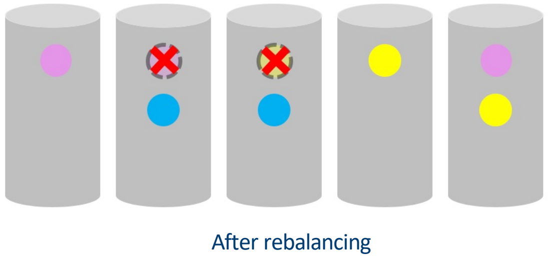
- Restore previously evicted servers/devices to active service
- Data moves back from fallback locations to reintegrated capacity
- Fallback devices service read/write requests until reintegration is complete
- Writes during reintegration also go to reintegrating device to ensure consistency

Extension ("extend")

- Add new servers/devices to the storage system

- Data rebalances automatically across new capacity





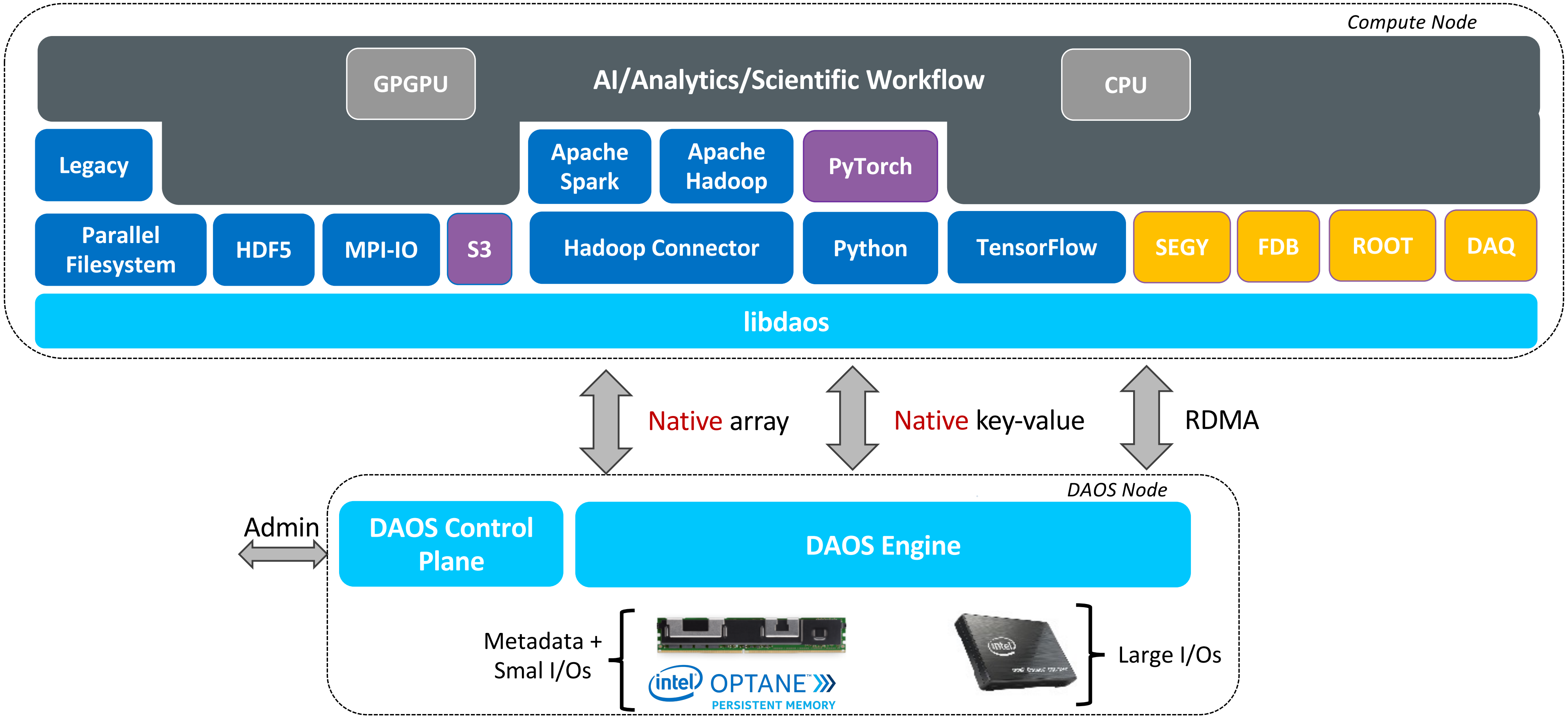
- Original devices service read/write requests until extension is complete

- Writes during extension also go to new devices to ensure consistency

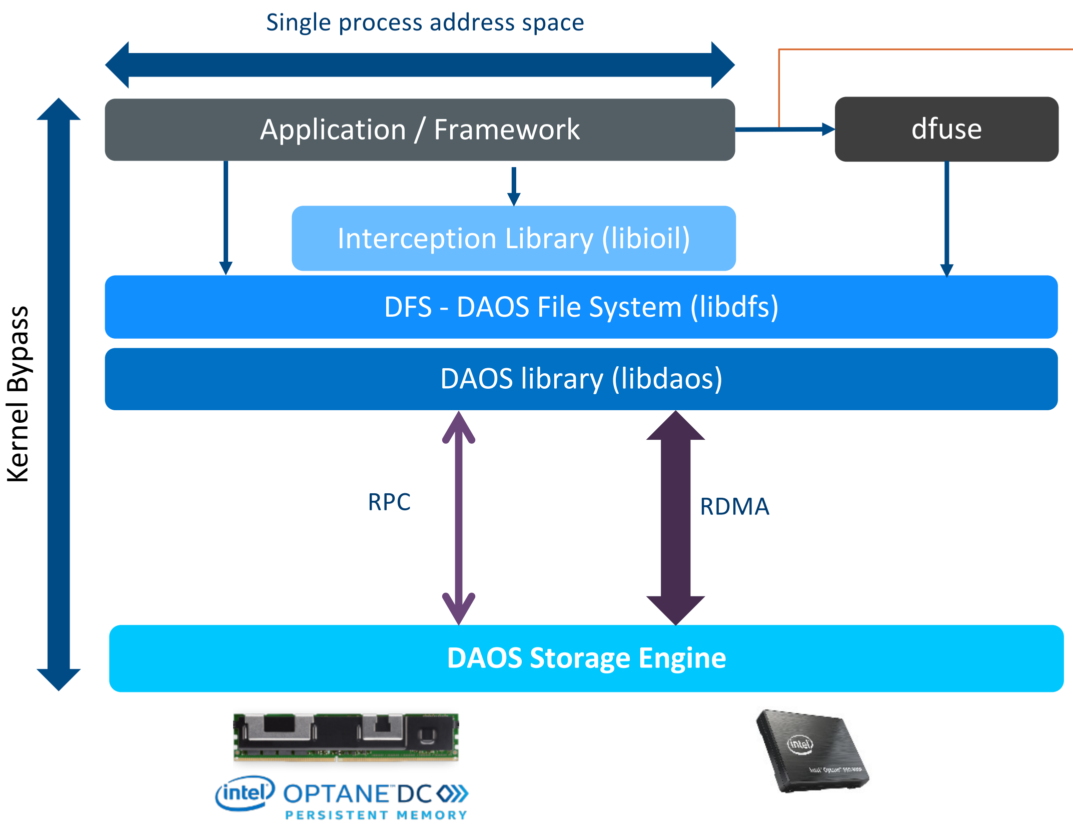
DAOS Software Ecosystem

DAOS Ecosystem

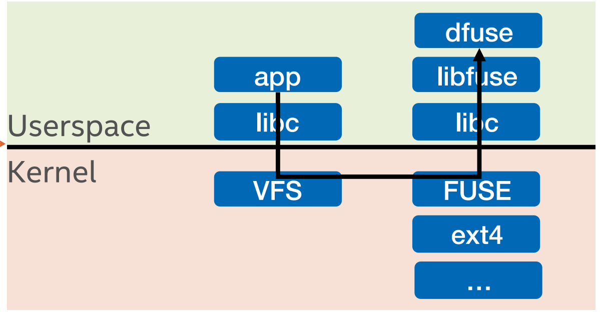
- 🟦 Generic I/O middleware supported today
- 🟨 Domain-specific data models under development in co-design with partners
- 🟪 Enablement in progress



POSIX I/O Support



Single process address space
Kernel Bypass
Application / Framework
dfuse
Interception Library (libioil)
DFS - DAOS File System (libdfs)
DAOS library (libdaos)
RPC
RDMA
DAOS Storage Engine
intel OPTANE DC PERSISTENT MEMORY



- User space DFS library with an API like POSIX.
  - Requires application changes (new API)
- DFUSE plugin to support POSIX API
  - No application changes
  - Limited performance
- DFUSE + IL
  - No application changes, runtime LD_PRELOAD
  - Good raw data I/O performance, limited metadata performance

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.

PyDAOS Primer

- python module primarily written in C
  - Expose DAOS key-value store objects as a python dictionary
    - Support python iterator, direct assignments, …
    - Bulk insert/retrieve
  - Other data structures are under consideration (see later)
- Python objects allocated by PyDAOS:
  - are **persistent**
    - identified by a string name
  - are immediately **visible** upon creation
    - to any process running on the same or a different node.
  - have a **very low memory footprint** since the actual content is stored remotely
    - This allows to manipulate gigantic datasets way bigger than the amount of memory available on the node

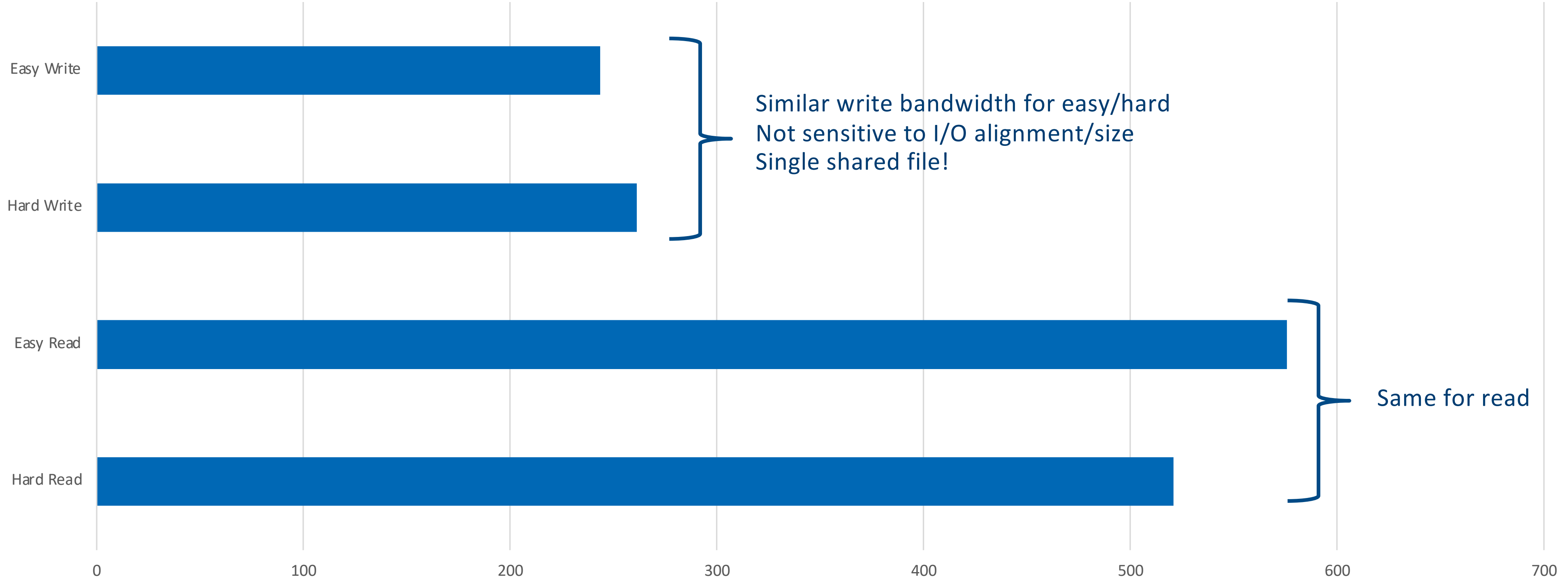
TensorFlow Integration

- Done via TensorFlow-IO (see https://www.tensorflow.org/io)
- Initial integration at the DFS level
  - Use TF-IO filesystem API
  - Compatible with POSIX container
  - Provide full OS bypass
- Development completed & testing underway
  - https://github.com/daos-stack/tensorflow-io-daos/tree/devel
  - https://github.com/daos-stack/tensorflow-io-daos/blob/devel/docs/daos_tf_docs.md

DAOS Performance

DAOS Bandwidth on IO500

IOR Bandwidth (GiB/s)

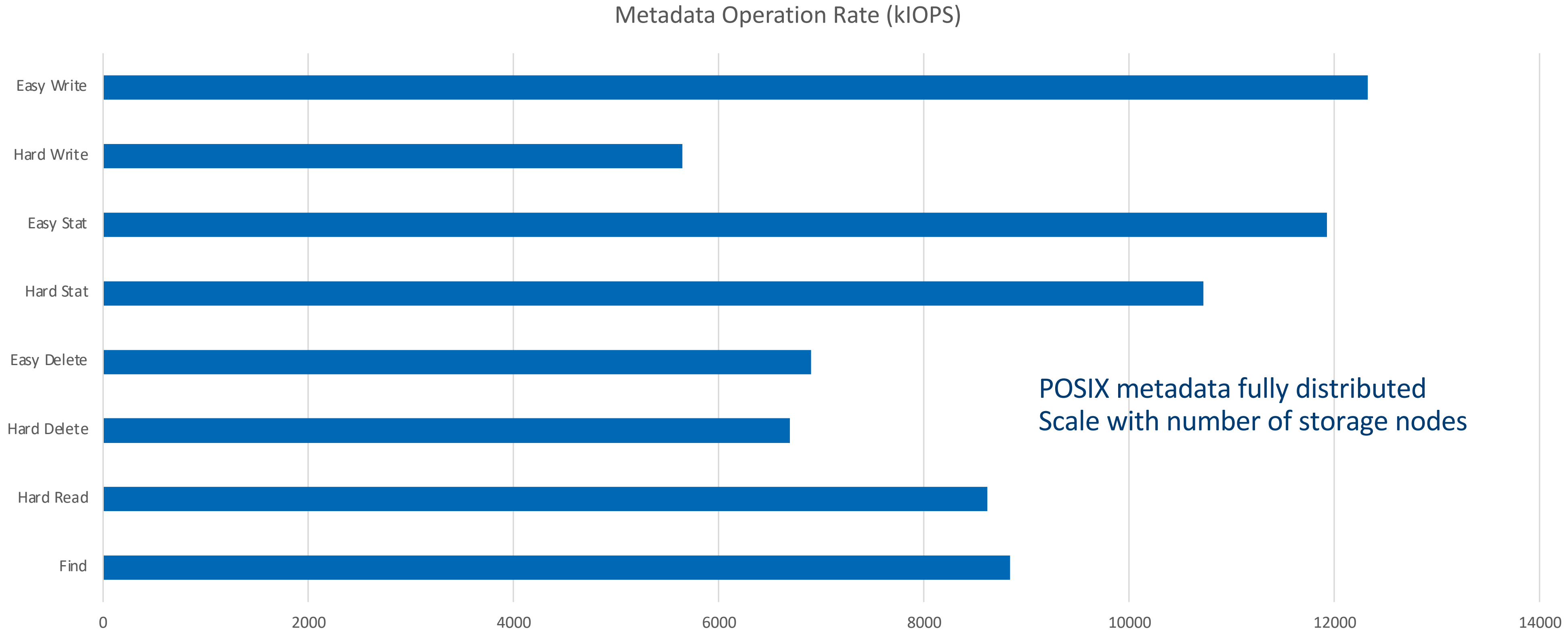


Similar write bandwidth for easy/hard
Not sensitive to I/O alignment/size
Single shared file!

Same for read

Source: https://io500.org/submissions/view/1

DAOS Metadata Performance on IO500

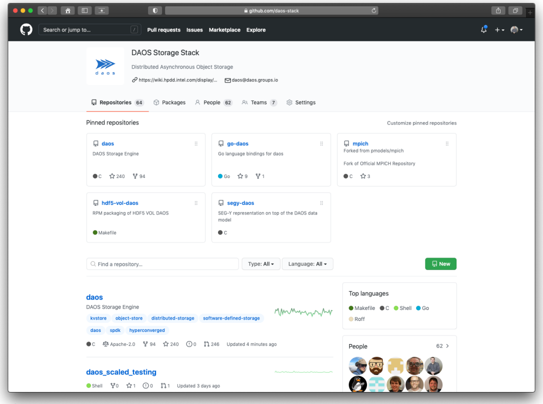


POSIX metadata fully distributed
Scale with number of storage nodes

Source: https://io500.org/site/submissions/view/1

Resources

- Community Resources
  - Github: https://github.com/daos-stack/daos
  - Online doc: http://daos.io
  - Mailing list & slack: https://daos.groups.io
  - YouTube channel: http://video.daos.io
- 5th DAOS User Group (DUG’21)
  - Recordings available at http://dug.daos.io
- Intel landing page
  - https://www.intel.com/content/www/us/en/high-performance-computing/daos.html



DAOS Storage Stack

Distributed Asynchronous Object Storage

https://wiki.hpdd.intel.com/display/... daos@daos.groups.io

Pinned repositories

daos
DAOS Storage Engine
C ☆ 240 94

go-daos
Go language bindings for daos
Go ☆ 9 1

mpich
Forked from pmodels/mpich
Fork of Official MPICH Repository
C ☆ 3

hdf5-vol-daos
RPM packaging of HDF5 VOL DAOS
Makefile

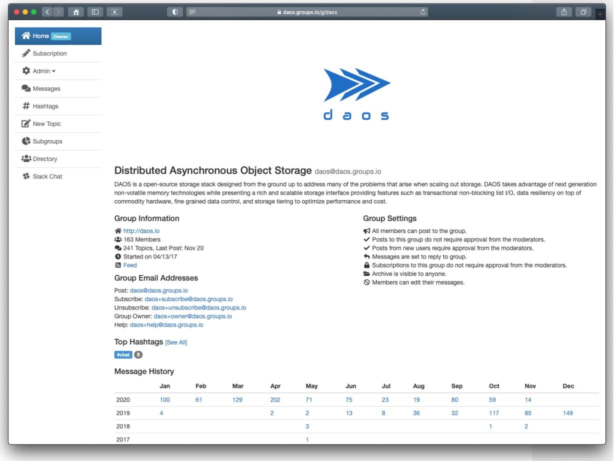
segy-daos
SEG-Y representation on top of the DAOS data model
C

daos
DAOS Storage Engine
kvstore object-store distributed-storage software-defined-storage daos spdk hyperconverged
C Apache-2.0 94 ☆ 240 0 246 Updated 4 minutes ago

daos_scaled_testing
Shell 0 ☆ 1 0 1 Updated 3 days ago

Top languages
Makefile C Shell Go Roff

People 62



Distributed Asynchronous Object Storage daos@daos.groups.io

DAOS is a open-source storage stack designed from the ground up to address many of the problems that arise when scaling out storage. DAOS takes advantage of next generation non-volatile memory technologies while presenting a rich and scalable storage interface providing features such as transactional non-blocking list I/O, data resiliency on top of commodity hardware, fine grained data control, and storage tiering to optimize performance and cost.

Group Information
http://daos.io
163 Members
241 Topics, Last Post: Nov 20
Started on 04/13/17
Feed

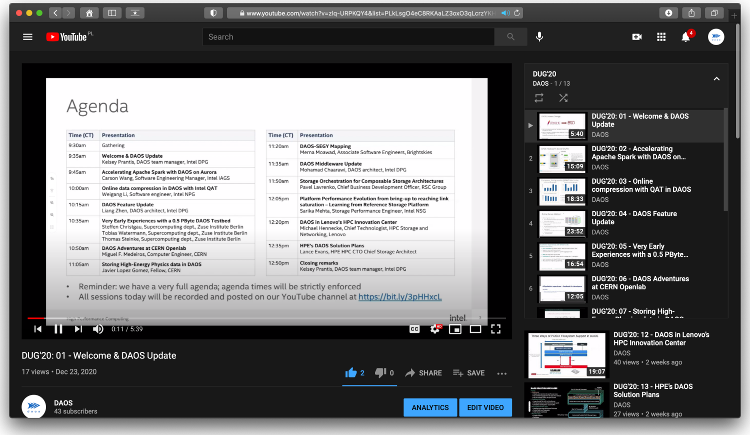
Group Email Addresses
Post: daos@daos.groups.io
Subscribe: daos+subscribe@daos.groups.io
Unsubscribe: daos+unsubscribe@daos.groups.io
Group Owner: daos+owner@daos.groups.io
Help: daos+help@daos.groups.io

Group Settings
All members can post to the group.
✓ Posts to this group do not require approval from the moderators.
✓ Posts from new users require approval from the moderators.
Messages are set to reply to group.
Subscriptions to this group do not require approval from the moderators.
Archive is visible to anyone.
Members can edit their messages.

Top Hashtags [See All]
#chat 5

Message History

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2020	100	61	129	202	71	75	23	19	80	59	14	
2019	4			2	2	13	8	36	32	117	85	149
2018					3					1	2	
2017					1							



Agenda

Time (CT)	Presentation
9:30am	Gathering
9:35am	Welcome & DAOS Update Kelsey Prantis, DAOS team manager, Intel DPG
9:45am	Accelerating Apache Spark with DAOS on Aurora Carson Wang, Software Engineering Manager, Intel IAGS
10:00am	Online data compression in DAOS with Intel QAT Weigang Li, Software engineer, Intel NPG
10:15am	DAOS Feature Update Liang Zhen, DAOS architect, Intel DPG
10:35am	Very Early Experiences with a 0.5 PByte DAOS Testbed Steffen Christgau, Supercomputing dept., Zuse Institute Berlin Tobias Watermann, Supercomputing dept., Zuse Institute Berlin Thomas Steinke, Supercomputing dept., Zuse Institute Berlin
10:50am	DAOS Adventures at CERN Openlab Miguel F. Medeiros, Computer Engineer, CERN
11:05am	Storing High-Energy Physics data in DAOS Javier Lopez Gomez, Fellow, CERN

Time (CT)	Presentation
11:20am	DAOS-SEGY Mapping Merna Moawad, Associate Software Engineers, Brightskies
11:35am	DAOS Middleware Update Mohamad Chaarawi, DAOS architect, Intel DPG
11:50am	Storage Orchestration for Composable Storage Architectures Pavel Lavrenko, Chief Business Development Officer, RSC Group
12:05pm	Platform Performance Evolution from bring-up to reaching link saturation – Learning from Reference Storage Platform Sarika Mehta, Storage Performance Engineer, Intel NSG
12:20pm	DAOS in Lenovo's HPC Innovation Center Michael Hennecke, Chief Technologist, HPC Storage and Networking, Lenovo
12:35pm	HPE's DAOS Solution Plans Lance Evans, HPE HPC CTO Chief Storage Architect
12:50pm	Closing remarks Kelsey Prantis, DAOS team manager, Intel DPG

- Reminder: we have a very full agenda; agenda times will be strictly enforced
- All sessions today will be recorded and posted on our YouTube channel at https://bit.ly/3pHHxcL

DUG'20: 01 - Welcome & DAOS Update
17 views • Dec 23, 2020

DAOS
43 subscribers

DUG'20
DAOS - 1 / 13

DUG'20: 01 - Welcome & DAOS Update
DAOS

2 DUG'20: 02 - Accelerating Apache Spark with DAOS on...
DAOS

3 DUG'20: 03 - Online compression with QAT in DAOS
DAOS

4 DUG'20: 04 - DAOS Feature Update
DAOS

5 DUG'20: 05 - Very Early Experiences with a 0.5 PByte...
DAOS

6 DUG'20: 06 - DAOS Adventures at CERN Openlab
DAOS

DUG'20: 07 - Storing High-

DUG'20: 12 - DAOS in Lenovo's HPC Innovation Center
DAOS
40 views • 2 weeks ago

DUG'20: 13 - HPE's DAOS Solution Plans
DAOS
27 views • 2 weeks ago

Q & A

intel®