Hewlett Packard Enterprise

HPC·AI ADVISORY COUNCIL
NETWORK OF EXPERTISE

Converging HPC and AI Storage: The DAOS Storage Architecture

Michael Hennecke (HPE Cray Supercomputing Storage Systems)

09-Apr-2025

daos

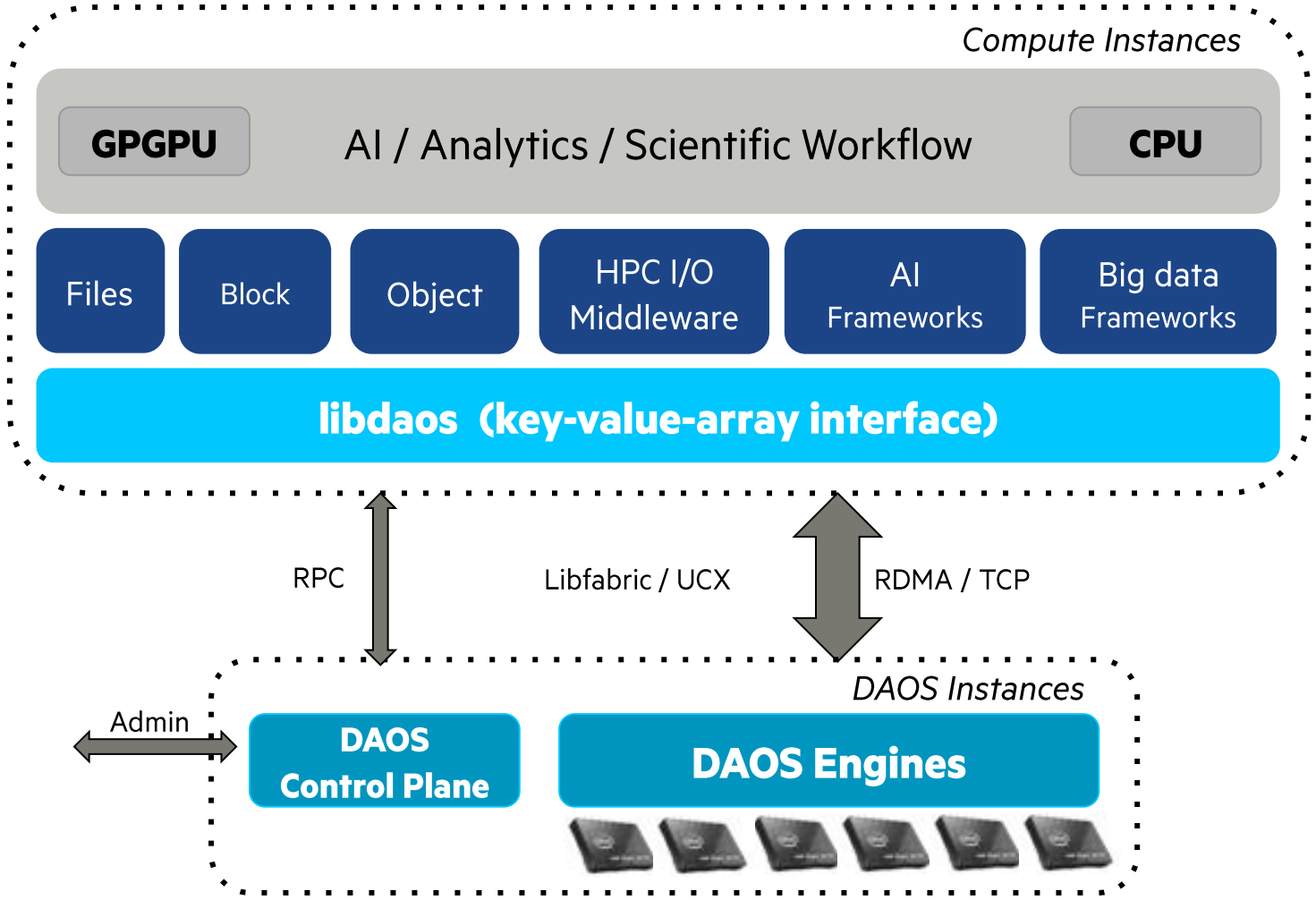
DAOS Foundation – A Project of the Linux Foundation

Learn more about how to join the DAOS Foundation here: https://daos.io/how-to-join-the-daos-foundation.

DAOS: Nextgen Open Source Storage Platform

Distributed Asynchronous Object Storage

- Platform for innovation
- Files, blocks, objects and more
- Full end-to-end userspace
- Flexible built-in data protection
  - EC / replication with self-healing
- Flexible network layer
- Efficient single server
  - O(100) GB/s and O(1M) IOPS per server
- Highly scalable
  - Tens of TB/s, billions of IOPS of aggregated performance
  - O(1M) client processes
- Time to first byte in O(10) µs



Hewlett Packard Enterprise

daos

DAOS Design Fundamentals

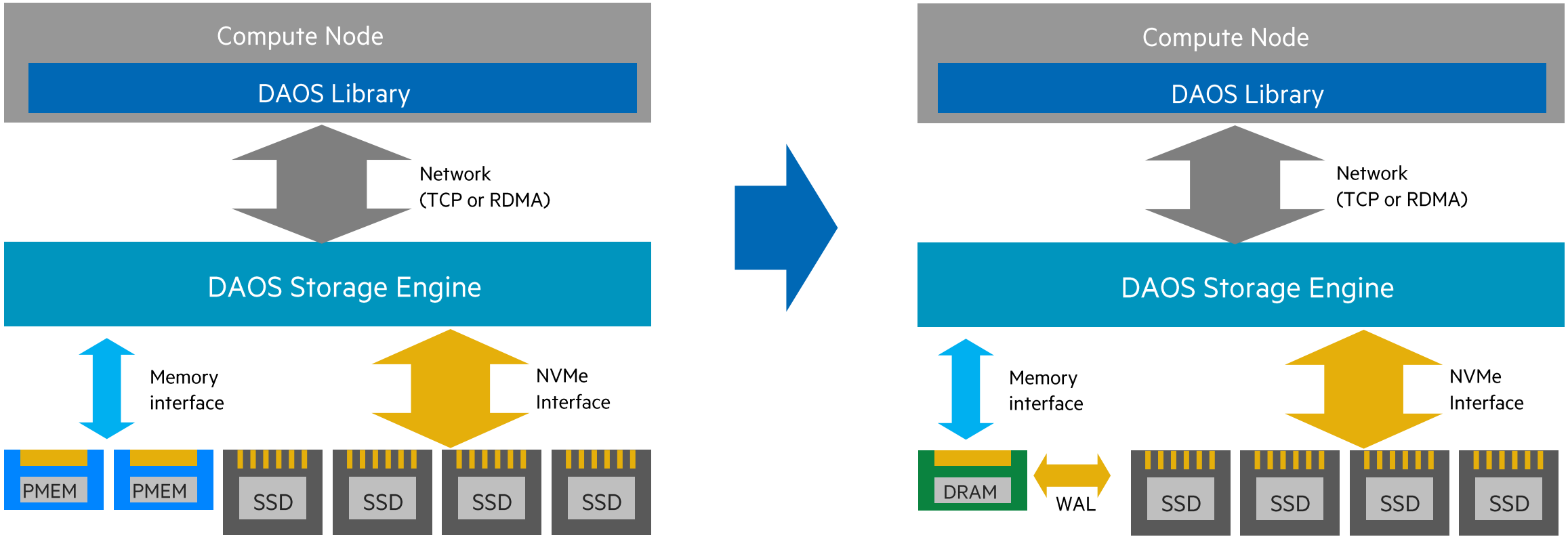
- No read-modify-write on I/O path (use versioning)
- No locking/DLM (use MVCC)
- No client tracking or client recovery
- No centralized (meta)data server
- No global object table
- Non-blocking I/O processing
- Serializable distributed transactions
- Built-in multi-tenancy
- User snapshots

Scalability & Performance

High IOPS

Unique Capabilities

DAOS Architecture Evolution (with / without PMem)

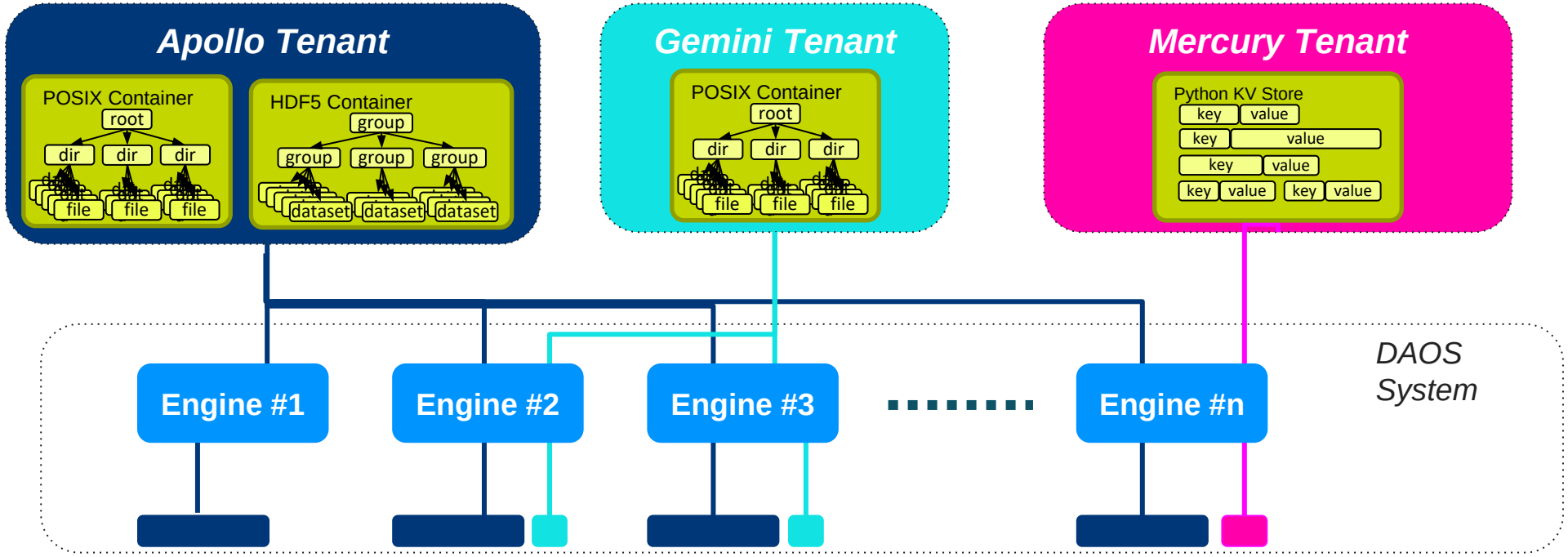


With Persistent Memory

Without Persistent Memory

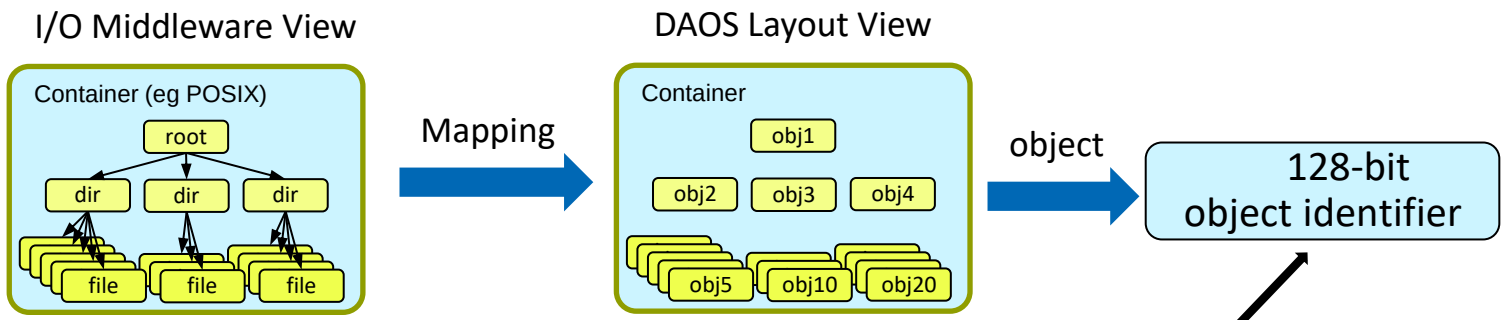
See the ISC23 workshop paper ...

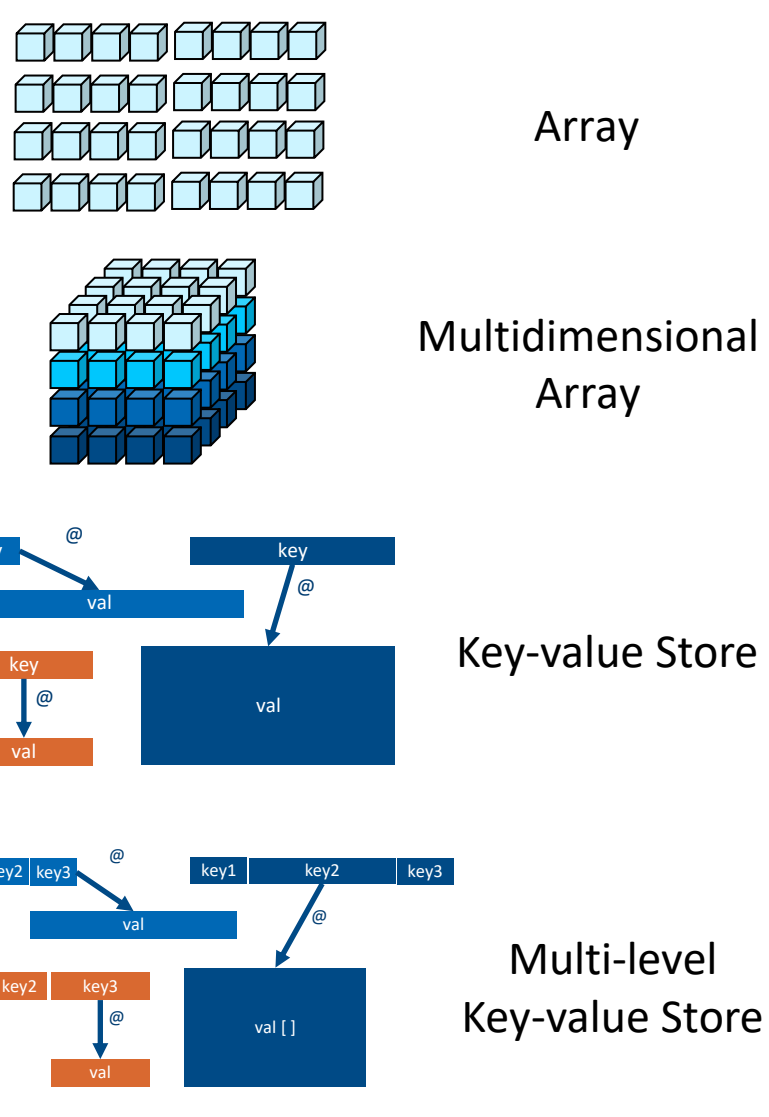
DAOS Storage Pooling and Multi-Tenancy



Pool 1	Apollo Tenant	100PB	20TB/s	200M IOPS
Pool 2	Gemini Tenant	10PB	2TB/s	20M IOPS
Pool 3	Mercury Tenant	30TB	80GB/s	2M IOPS

DAOS Object Model

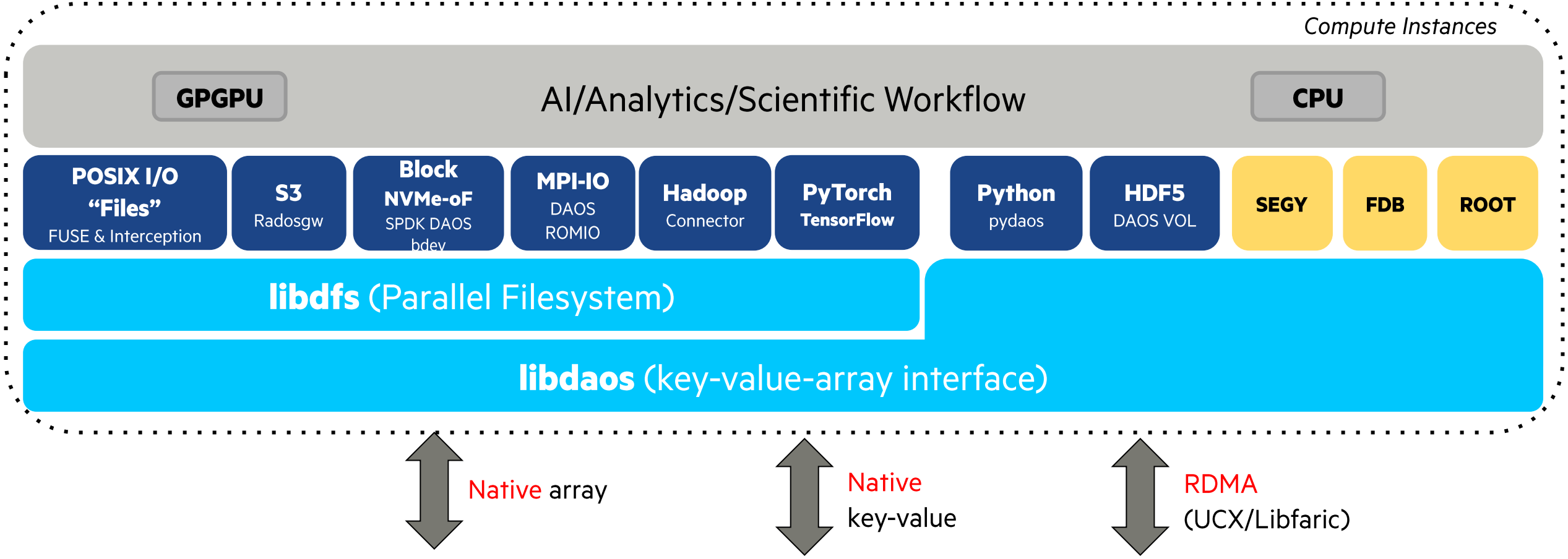




The 128-bit DAOS OID includes:

- Lower 96-bit: user's object ID
- Upper 32-bit used by DAOS to specify:
  - DAOS object type: KV, array, multi-level KV
  - DAOS object class: data distribution and protection

DAOS Software Ecosystem



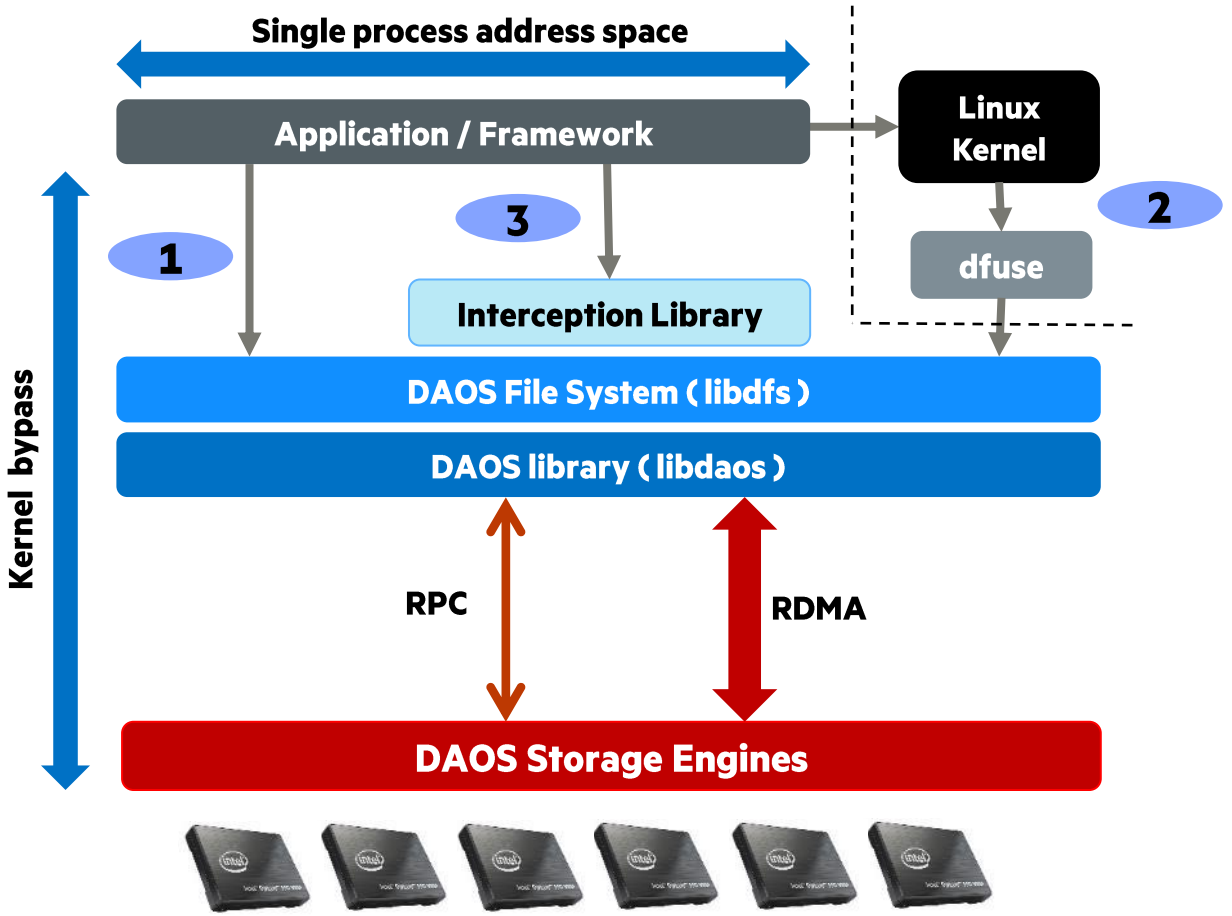
Hewlett Packard Enterprise | daos

Generic I/O Middleware/frameworks

Domain-specific data models under development in co-design with partners

POSIX I/O Support

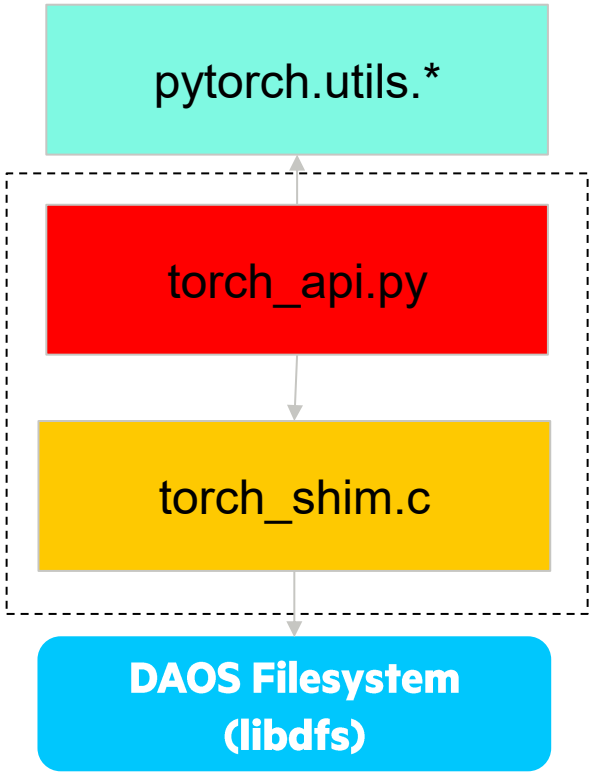
- DAOS File System (**libdfs**) 1
  - Implements POSIX namespace in a container
  - Userspace **libdfs.so** library (**requires** application changes)
  - Full OS bypass, asynchronous I/O; no caching
- FUSE Daemon (**dfuse**) 2
  - Transparent POSIX access (no application changes)
  - VFS mountpoint through dfuse; high latency
  - Caching by Linux kernel
- I/O interception library (**libpil4dfs**) 3
  - Combined with dfuse; data & metadata interception
  - **LD_PRELOAD=/usr/lib64/libpil4dfs.so (or libioil.so)**
  - mmap and binary execution via dfuse

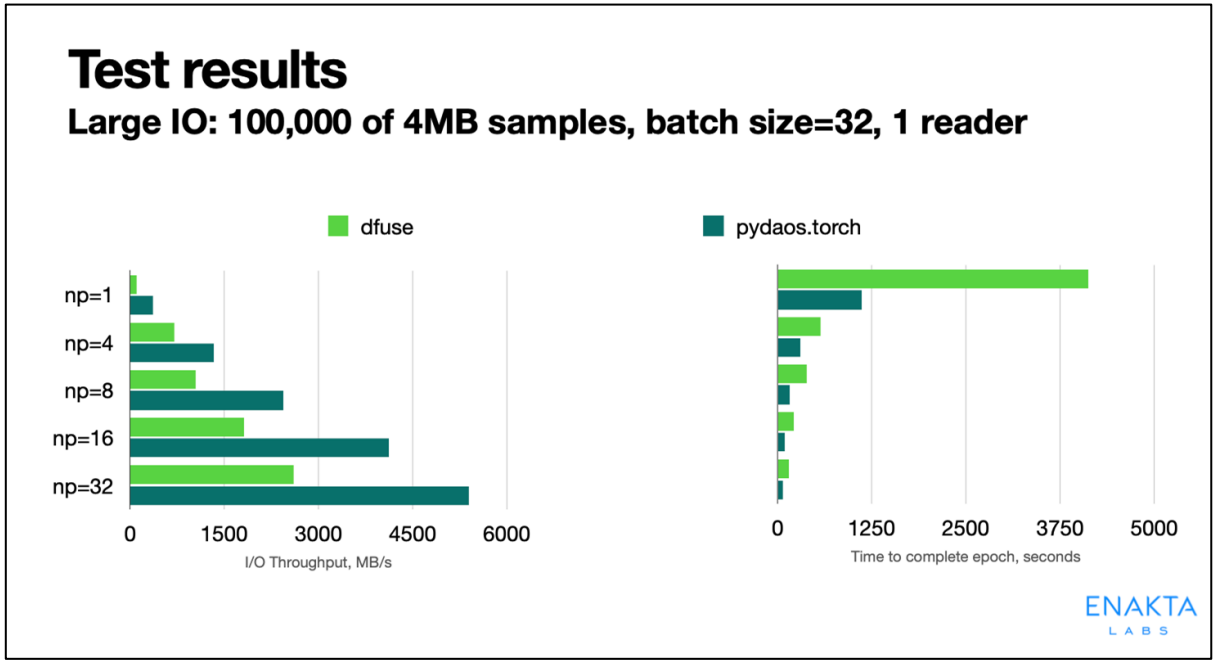


Pytorch Data Modules

ENAKTA LABS

- Collaboration between Enakta Labs and Google
- DataLoader and Checkpoint modules
  - Support for both iterable and map-style datasets
  - High parallelism using several DAOS event queues
  - Parallel namespace scanning using dfs anchor API





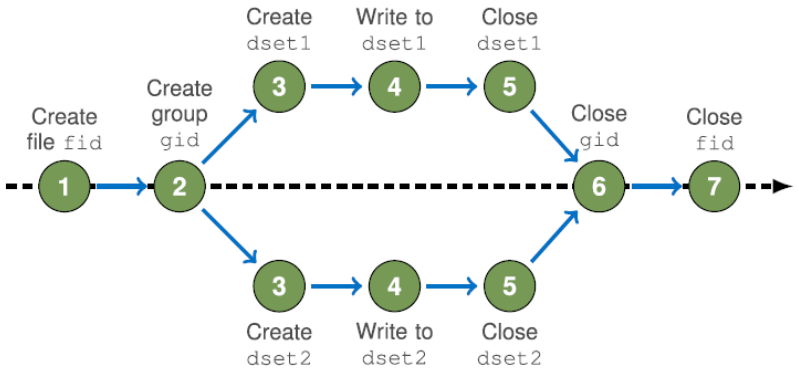
	Time to scan 1.1M Files
Regular scan	**291s**
Optimized scan	**32s**

Hewlett Packard Enterprise | daos

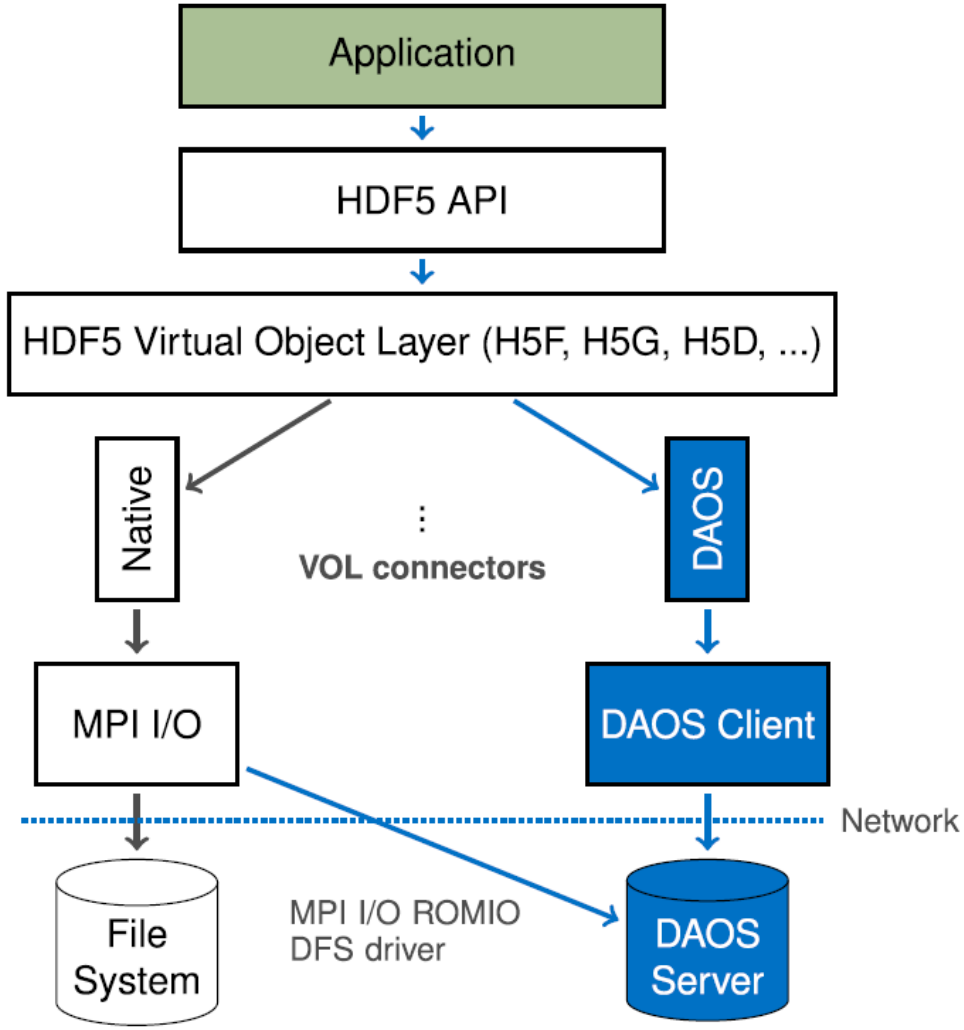
HDF5 VOL Plugin for DAOS

New functionality provided in DAOS VOL plugin:

- Control over object placement/striping
  - H5daos_set_object_class
- Independent object creation
  - H5daos_set_all_ind_metadata_ops
  - Avoids collective synchronisation across ranks
- Maps: application-defined K/V store
  - H5Mput/H5Mget
- Asynchronous I/O




```
es_id = H5EScreate();
fid = H5Fcreate_async(..., es_id);
gid = H5Gcreate_async(fid, ..., es_id);
did1 = H5Dcreate_async(gid, ..., es_id);
H5Dwrite_async(did1, ..., es_id);
did2 = H5Dcreate_async(gid, ..., es_id);
H5Dwrite_async(did2, ..., es_id);
H5Dclose_async(did1, es_id);
H5Dclose_async(did2, es_id);
H5Gclose_async(gid);
H5Fclose_async(fid);
H5ESwait(es_id, ...);
```

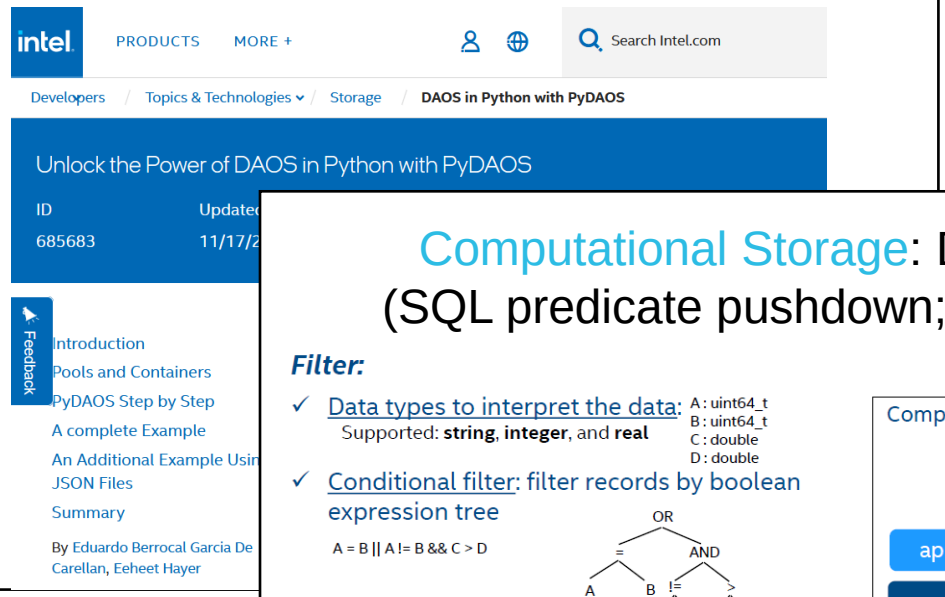


Hewlett Packard Enterprise | daos

See the Accelerating HDF5 I/O for Exascale paper...

DAOS Beyond POSIX: Storage Innovations

Python Container type and PyDAOS API



Computational Storage: DAOS pipeline API (SQL predicate pushdown; optimized pfind, …)

DAOS APIs, incl. transactions

(PMTutorial by A. Jackson and M. Chaarawi)

```
        daos_tx_open(coh, &th, ...);
restart:
        daos_obj_fetch(..., th, ...);
        daos_obj_update(..., th, ...);
        daos_obj_fetch(..., th, ...);
        daos_obj_update(..., th, ...);
        daos_obj_dkey_punch(..., th, ...);
        rc = daos_tx_commit(th, ...);
        if (rc == -DER_RESTART) {
                daos_tx_restart(th, ...);
                goto restart;
        }
        daos_tx_close(th, ...);
```

DAOS Deployments at ALCF (Aurora) and LRZ (SNG2)

Compute Nodes:
2x Intel SPR+**HBM**, **6x** Intel Xe "PVC" GPUs, **8x** HPE Slingshot

1024 DAOS Servers (Intel M50CYP):
- 2x Xeon 5320 26core 2.2GHz CPUs
- 16x 32GB DDR4 DRAM
- 16x **512GB** Intel Optane 200 PMem
- **16x** Samsung PM1733 **15.36TB** NVMe (gen4)
- 2x HPE Slingshot (200Gbps)

➔ 16k NVMe (250PB), 16k PMem (8PB), 2k engines

Compute Nodes:
2x Intel SPR, **4x** Intel Xe "PVC" GPUs, **2x** NVIDIA HDR

42 DAOS Servers (Lenovo SR630v2):
- 2x Xeon 8352Y 32core 2.2GHz CPUs
- 16x 32GB DDR4 DRAM
- 16x **128GB** Intel Optane 200 PMem
- **8x** Intel P5500 **3.84TB** NVMe (gen4)
- 2x NVIDIA HDR InfiniBand (200Gbps)

➔ 336 NVMe (1.3PB), 672 PMem (84TB), 84 engines

IO500-SC23 Performance per Server (Production List – bigger is better)

IO500-SC23 Production List	#1 ANL (DAOS 2.5.0) 642 ICX server @ 16 NVMe		#2 LRZ (DAOS 2.4.0) 42 ICX server @ 8 NVMe		#3 KAUST (ClusterStor 6.4) MDS: 20 AMD srv @ 12 NVMe OSS: 96 AMD srv @ 12 NVMe		#5 MSKCC (WekaFS 4.2.4) 22 ICX server @ 8 NVMe		#7 ICL (GPFS-ECE 5.1.7.1) 14 ICX server @ 10 NVMe	
	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**
MDTEST	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
Easy Write	60985	**95,0**	6324	**150,6**	780	**39,0**	662	**30,1**	246	**17,6**
Easy Stat	225295	**350,9**	29403	**700,1**	1739	**87,0**	9852	**447,8**	723	**51,6**
Easy Delete	57648	**89,8**	3442	**82,0**	702	**35,1**	882	**40,1**	177	**12,6**
Hard Write	33827	**52,7**	2644	**63,0**	359	**18,0**	120	**5,5**	52	**3,7**
Hard Read	141467	**220,4**	17023	**405,3**	633	**31,7**	3622	**164,6**	574	**41,0**
Hard Stat	230086	**358,4**	23242	**553,4**	2495	**124,8**	8056	**366,2**	684	**48,9**
Hard Delete	62196	**96,9**	3112	**74,1**	206	**10,3**	89	**4,0**	45	**3,2**
IOR	GiB/s	GiB/s	GiB/s	GiB/s		GiB/s	GiB/s	GiB/s	GiB/s	GiB/s
Easy Write	20693	**32,2**	896	**21,3**	2340	**24,4**	174	**7,9**	131	**9,4**
Easy Read	12122	**18,9**	1872	**44,6**	3949	**41,1**	327	**14,9**	137	**9,8**
Hard Write	4216	**6,6**	252	**6,0**	23	**0,2**	44	**2,0**	7	**0,5**
Hard Read	9706	**15,1**	718	**17,1**	1157	**12,1**	47	**2,1**	29	**2,1**
FIND	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
Find	229672	**357,7**	12733	**303,2**	3684	**184,2**	262	**11,9**	3709	**264,9**
SCORE										
IO500 Score	32165	**50,1**	2508	**59,7**	797	***18,2***	308	**14,0**	119	**8,5**
IO500 BW	10066	**15,7**	742	**17,7**	709	**7,4**	104	**4,7**	44	**3,1**
IO500 MD	102785	**160,1**	8472	**201,7**	895	**44,8**	910	**41,4**	320	**22,9**

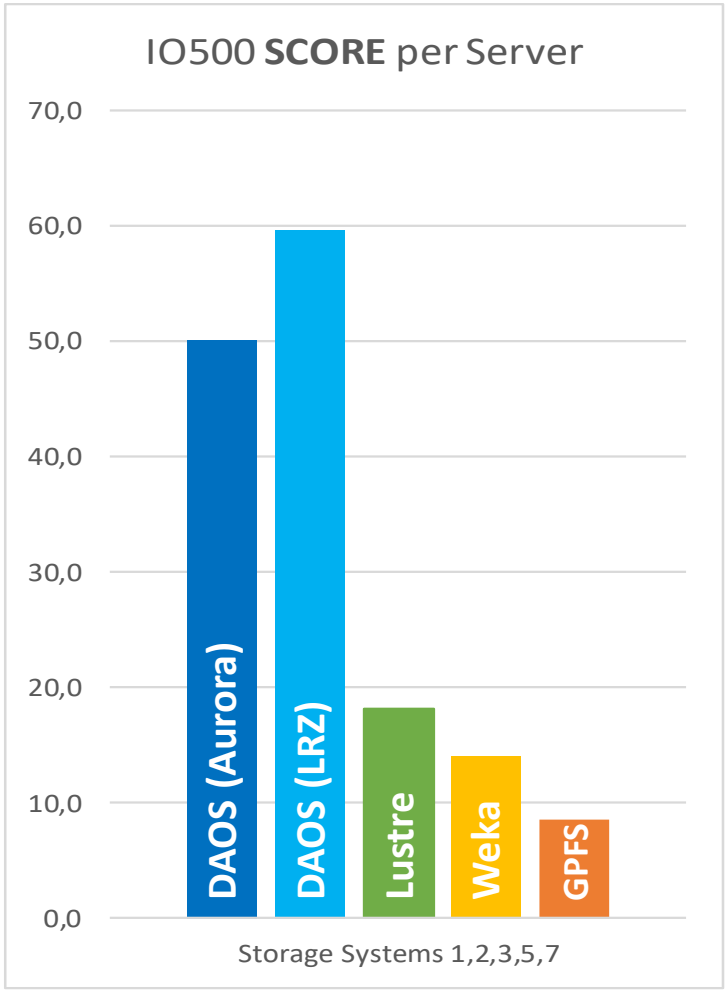
Hewlett Packard Enterprise

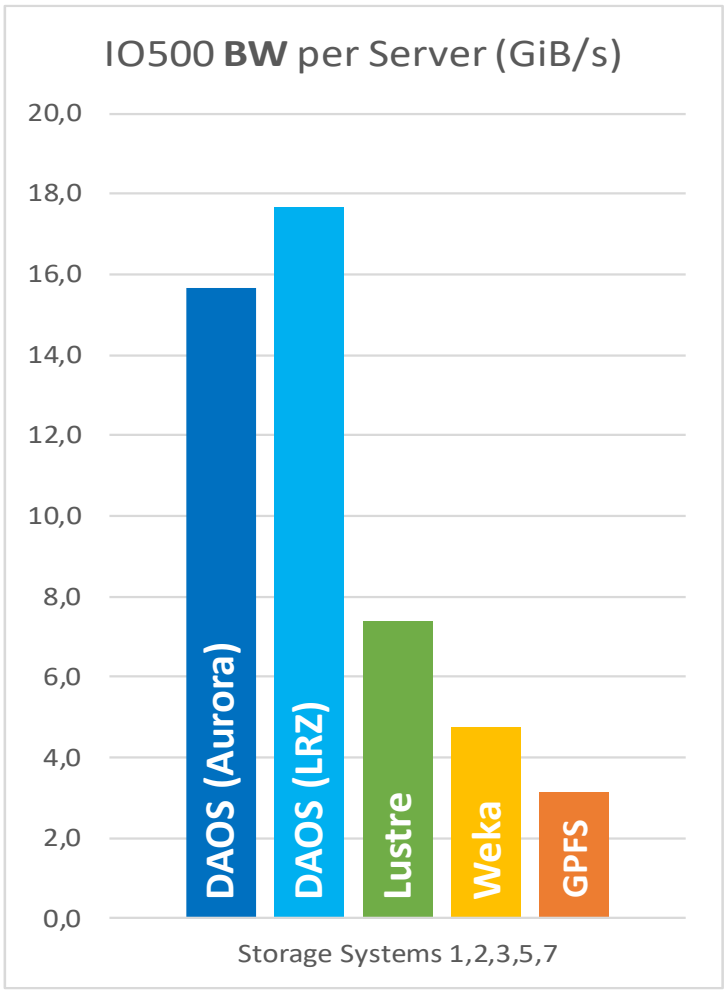
daos

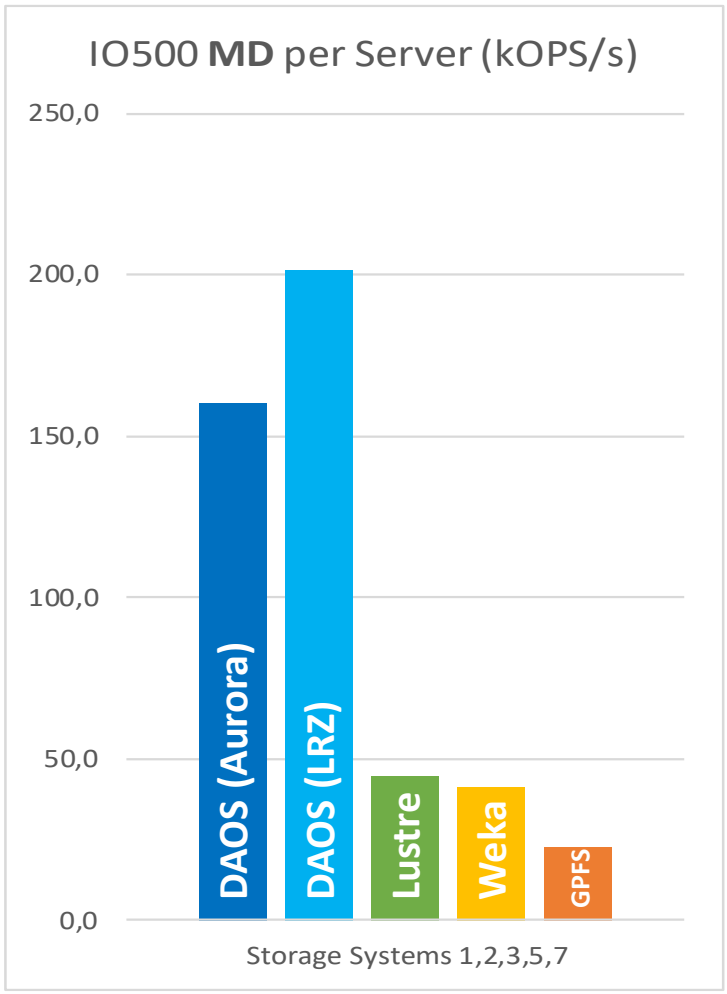
IO500-SC23 Performance per Server (Production List – bigger is better)

IO500-SC23 Production List	#1 ANL (DAOS 2.5.0) 642 ICX server @ 16 NVMe		#2 LRZ (DAOS 2.4.0) 42 ICX server @ 8 NVMe		#3 KAUST (ClusterStor 6.4) MDS: 20 AMD srv @ 12 NVMe OSS: 96 AMD srv @ 12 NVMe		#5 MSKCC (WekaFS 4.2.4) 22 ICX server @ 8 NVMe		#7 ICL (GPFS-ECE 5.1.7.1) 14 ICX server @ 10 NVMe	
	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**	absolute	**per-server**
MDTEST	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
Easy Write	60985	**95,0**	6324	**150,6**	780	**39,0**	662	**30,1**	246	**17,6**
Easy Stat	225295	**350,9**	29403	**700,1**	1739	**87,0**	9852	**447,8**	723	**51,6**
Easy Delete	57648	**89,8**	3442	**82,0**	702	**35,1**	882	**40,1**	177	**12,6**
Hard Write	33827	**52,7**	2644	**63,0**	359	**18,0**	120	**5,5**	52	**3,7**
Hard Read	141467	**220,4**	17023	**405,3**	633	**31,7**	3622	**164,6**	574	**41,0**
Hard Stat	230086	**358,4**	23242	**553,4**	2495	**124,8**	8056	**366,2**	684	**48,9**
Hard Delete	62196	**96,9**	3112	**74,1**	206	**10,3**	89	**4,0**	45	**3,2**
IOR	GiB/s	GiB/s	GiB/s	GiB/s		GiB/s	GiB/s	GiB/s	GiB/s	GiB/s
Easy Write	20693	**32,2**	896	**21,3**	2340	**24,4**	174	**7,9**	131	**9,4**
Easy Read	12122	**18,9**	1872	**44,6**	3949	**41,1**	327	**14,9**	137	**9,8**
Hard Write	4216	**6,6**	252	**6,0**	23	**0,2**	44	**2,0**	7	**0,5**
Hard Read	9706	**15,1**	718	**17,1**	1157	**12,1**	47	**2,1**	29	**2,1**
FIND	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s	kIOP/s
Find	229672	**357,7**	12733	**303,2**	3684	**184,2**	262	**11,9**	3709	**264,9**
SCORE										
IO500 Score	32165	**50,1**	2508	**59,7**	797	***18,2***	308	**14,0**	119	**8,5**
IO500 BW	10066	**15,7**	742	**17,7**	709	**7,4**	104	**4,7**	44	**3,1**
IO500 MD	102785	**160,1**	8472	**201,7**	895	**44,8**	910	**41,4**	320	**22,9**

IO500-SC23 Performance per Server (Production List – bigger is better)

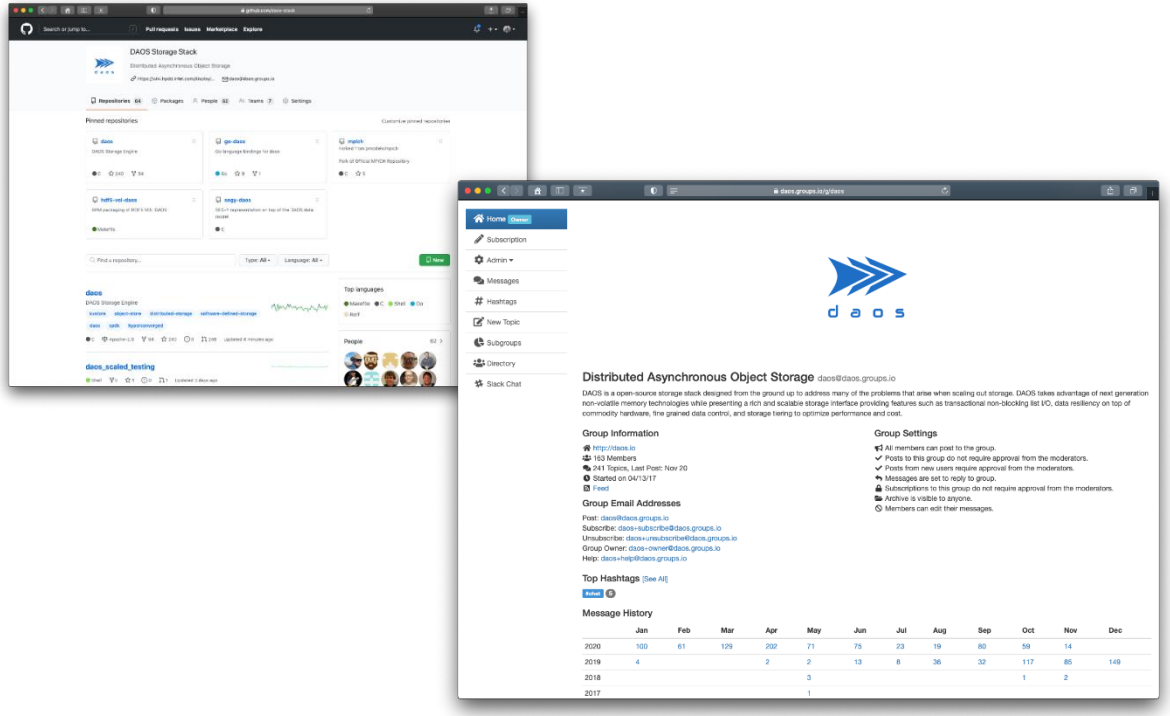


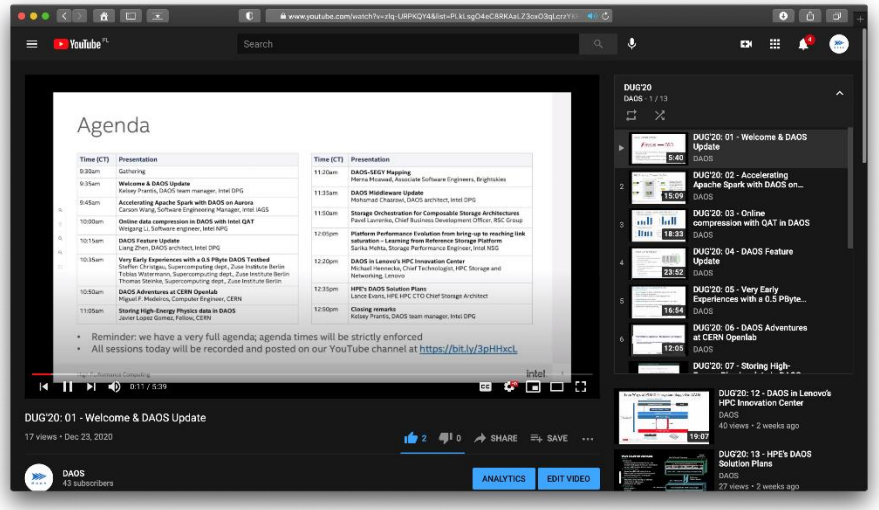




DAOS Resources

- DAOS Foundation landing page
  - https://daos.io/
- Community Resources
  - Github: https://github.com/daos-stack/daos
  - Online doc: https://docs.daos.io/
  - Mailing list & slack: https://daos.groups.io/
  - YouTube channel: https://video.daos.io/
- Virtual DAOS User Group on 22-May-2025:
  - https://daos.io/event/virtual-dug-25





Thank you

(michael.hennecke@hpe.com)

Hewlett Packard Enterprise | daos