October 10-12, 2023

ALCF Hands-on HPC Workshop

Argonne NATIONAL LABORATORY

DAOS is your Future

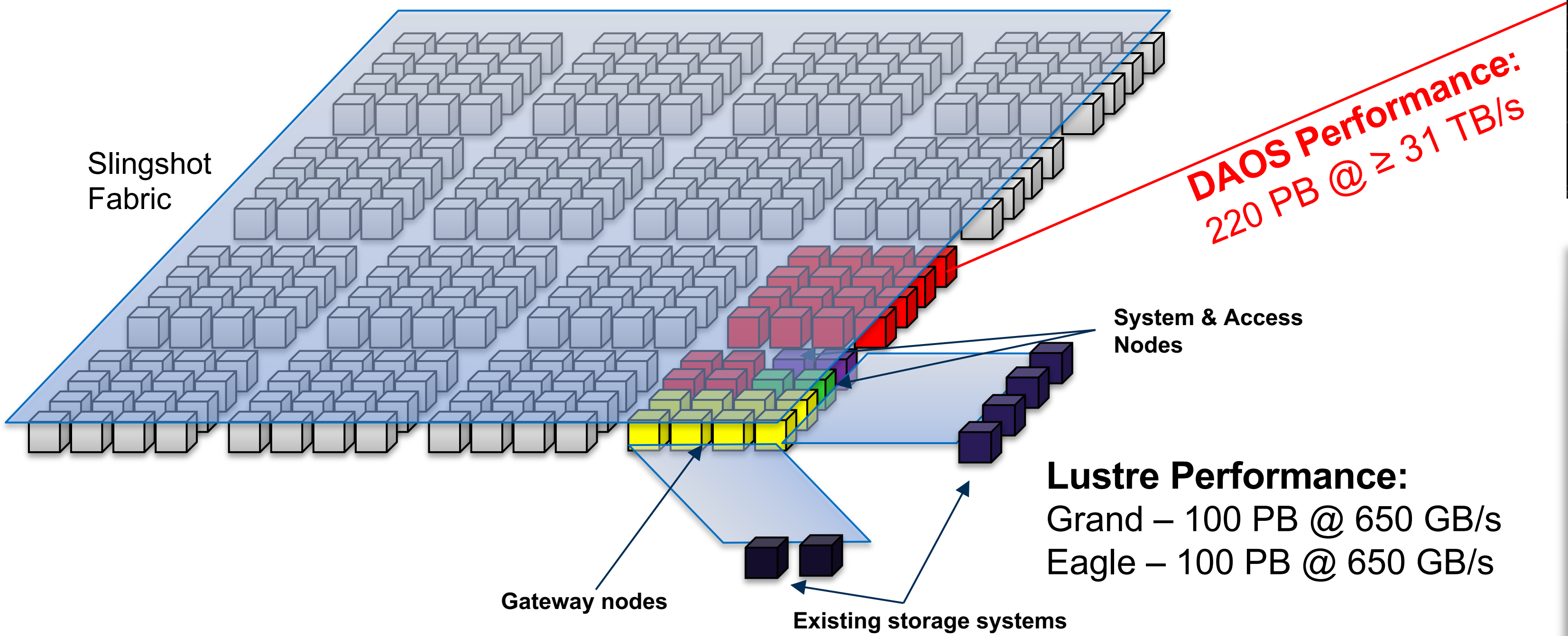
Distributed Asynchronous Object Store

Kevin Harms
Performance Engineering Team Lead

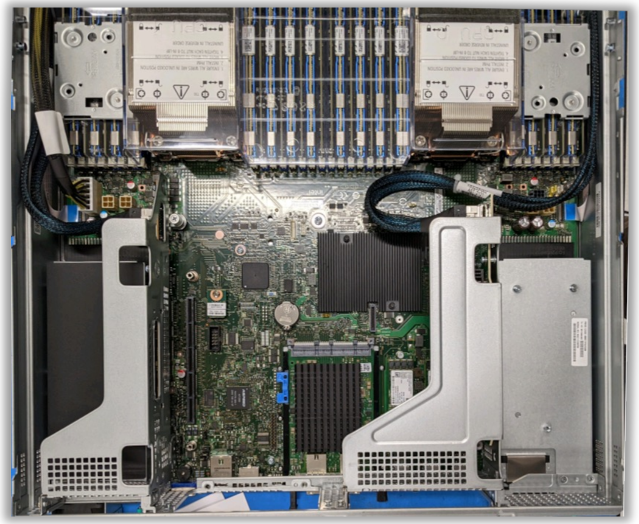
Storage Systems (for Performance)

System	Capacity	Performance	System
DAOS	220 PB @ EC16+2 ▪ 250 PB NVMe ▪ 8 PB Optane PMEM	≥ 25 TB/s Read & Write	Aurora
Eagle	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write	Aurora and Polaris
Grand	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write	Aurora and Polaris
Local	3.2 TB/node • 1.8 PB agg	~ 3 GB/s Read & Write per node 1.7 TB/s aggregate	Polaris

Aurora Storage Architecture



Lustre Performance:
Grand – 100 PB @ 650 GB/s
Eagle – 100 PB @ 650 GB/s



The Aurora open-source storage strategy strongly favors cooperation:

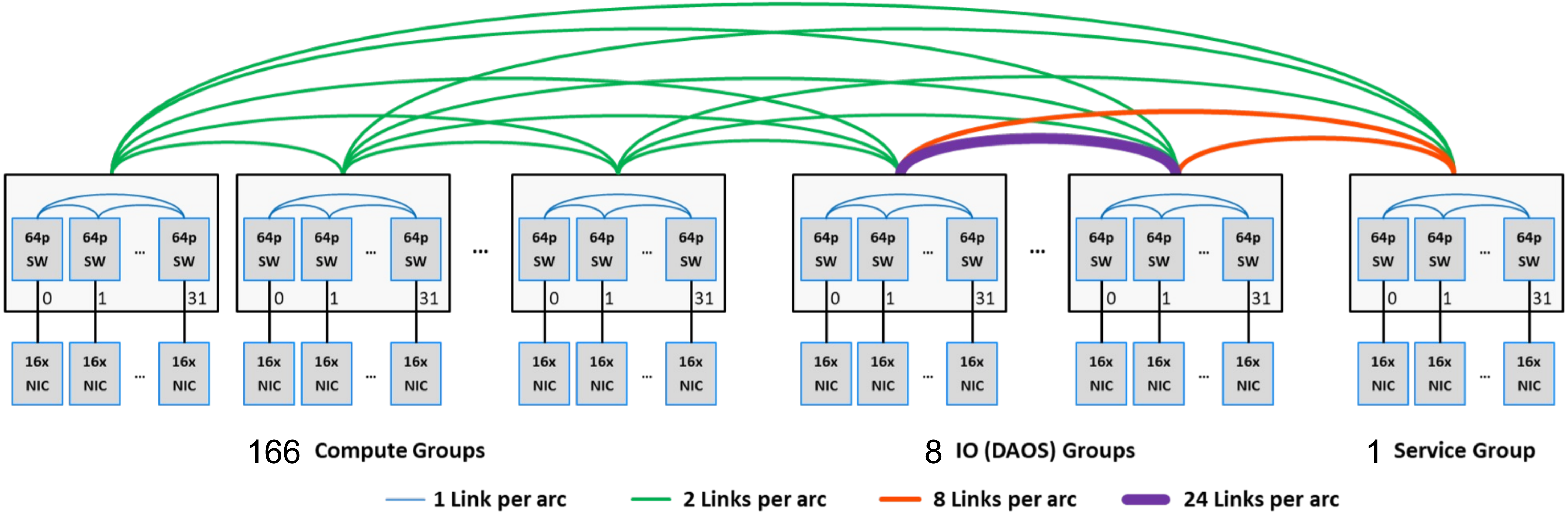
- DAOS: object storage system for in-fabric high-performance platform storage (the first of its kind on a DOE leadership system!)
- Lustre: parallel file systems for facility-wide access and data sharing

Namespace integration will make it easier for users to manage data.

•1024 DAOS server nodes, each with:

- —16 x 512GB persistent memory
- —16 x 15.3TB NVMe drives
- —2 x HPE Slingshot NICs
- —Dual CPU with 512 GB RAM

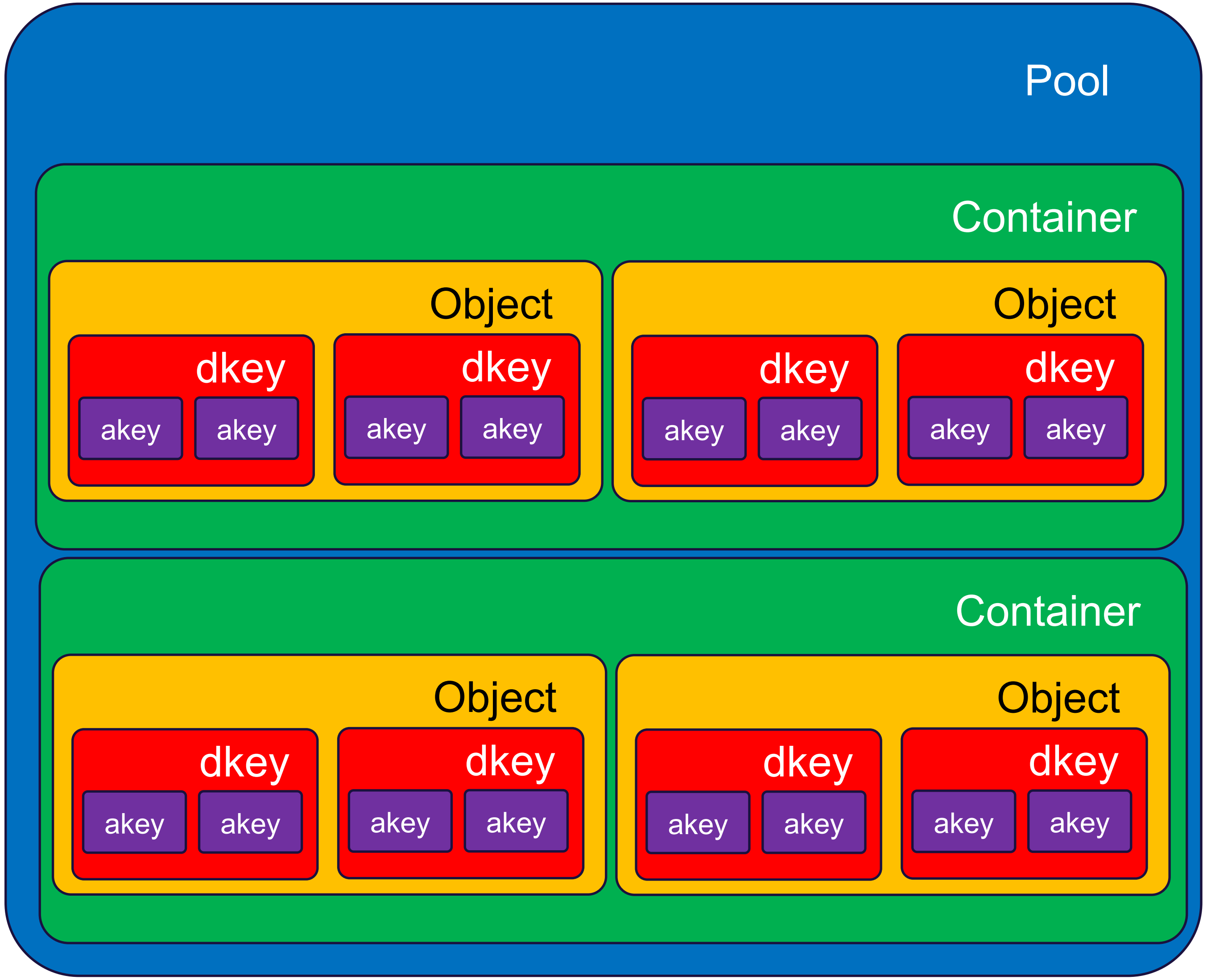
Aurora Network Architecture



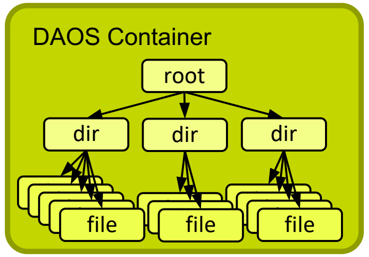
- Increased DAOS inter-group bandwidth
  - Support rebuilding and inter-server communication
  - Prevent DAOS server traffic interfering with application communication
- Increased bandwidth to service group
  - Support off-cluster access and data-movement to other storage systems

Data Model

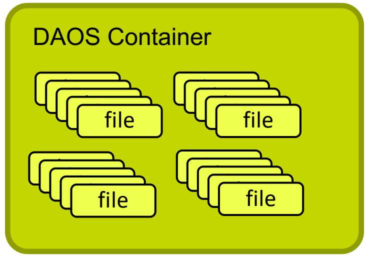
DAOS Data Model



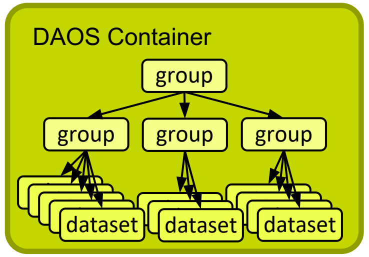
Examples



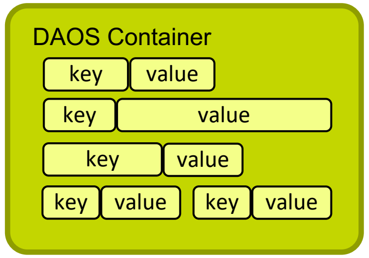
Encapsulated POSIX Namespace



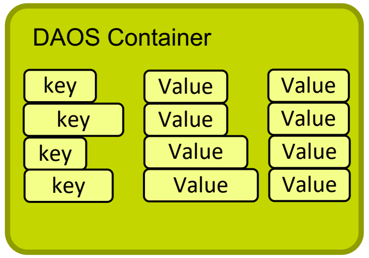
File-per-process



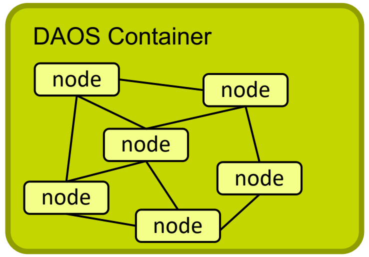
HDF5 « File »



Key-value store



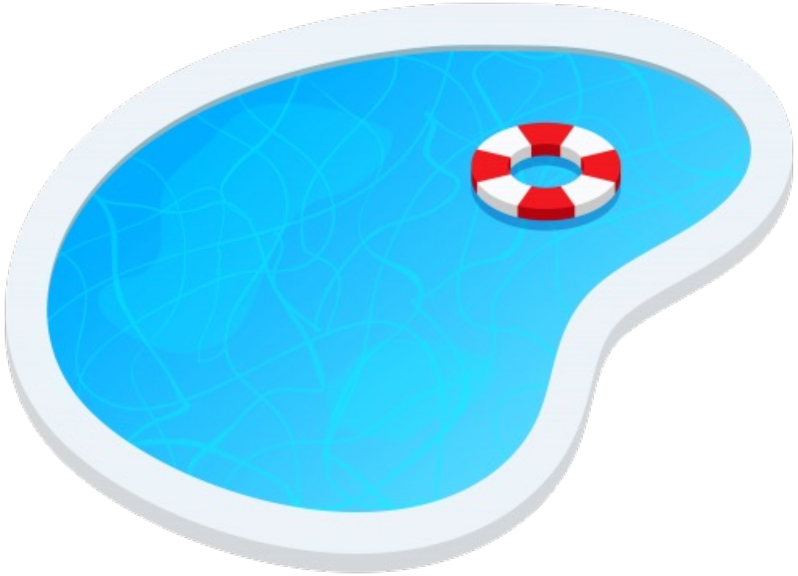
Columnar Database



Graph

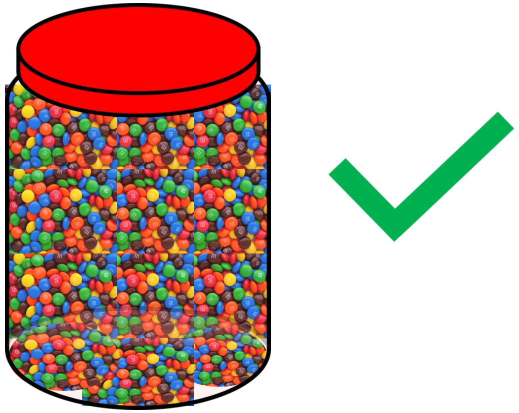
DAOS Pools

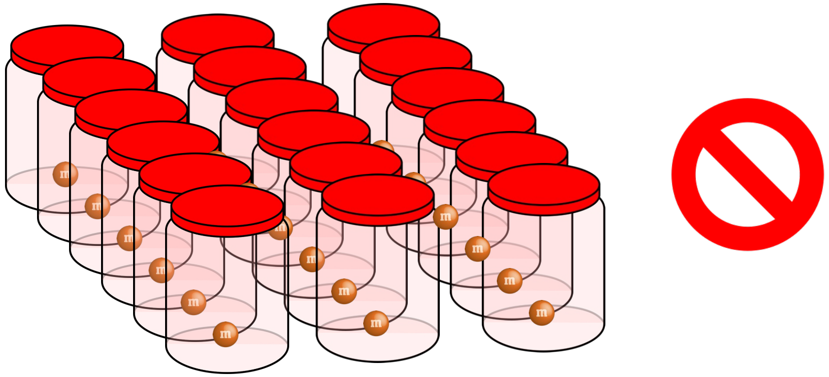
- Pools
  - A system contains *hundreds*
  - Physically allocated storage
    - Decided at pool creation time
  - Equal storage allocated per storage target
  - Contain list of Access Control Lists (ACLs)
  - Contains default parameters for containers



DAOS Containers

- Containers
  - —A pool contains *thousands* of containers
  - —Basic unit of storage from user perspective
  - —Containers have a type (POSIX, HDF5, pyDAOS, SEGY, …)
  - —POSIX containers can have many *millions* of files/directory/data
  - —Configuration for object class/redundancy, checksums, cell size, etc.
    - ▪ Many options
    - ▪ Determines distribution across pool
  - —ACLs
    - ▪ Determine access rights, not POSIX permissions





Usage

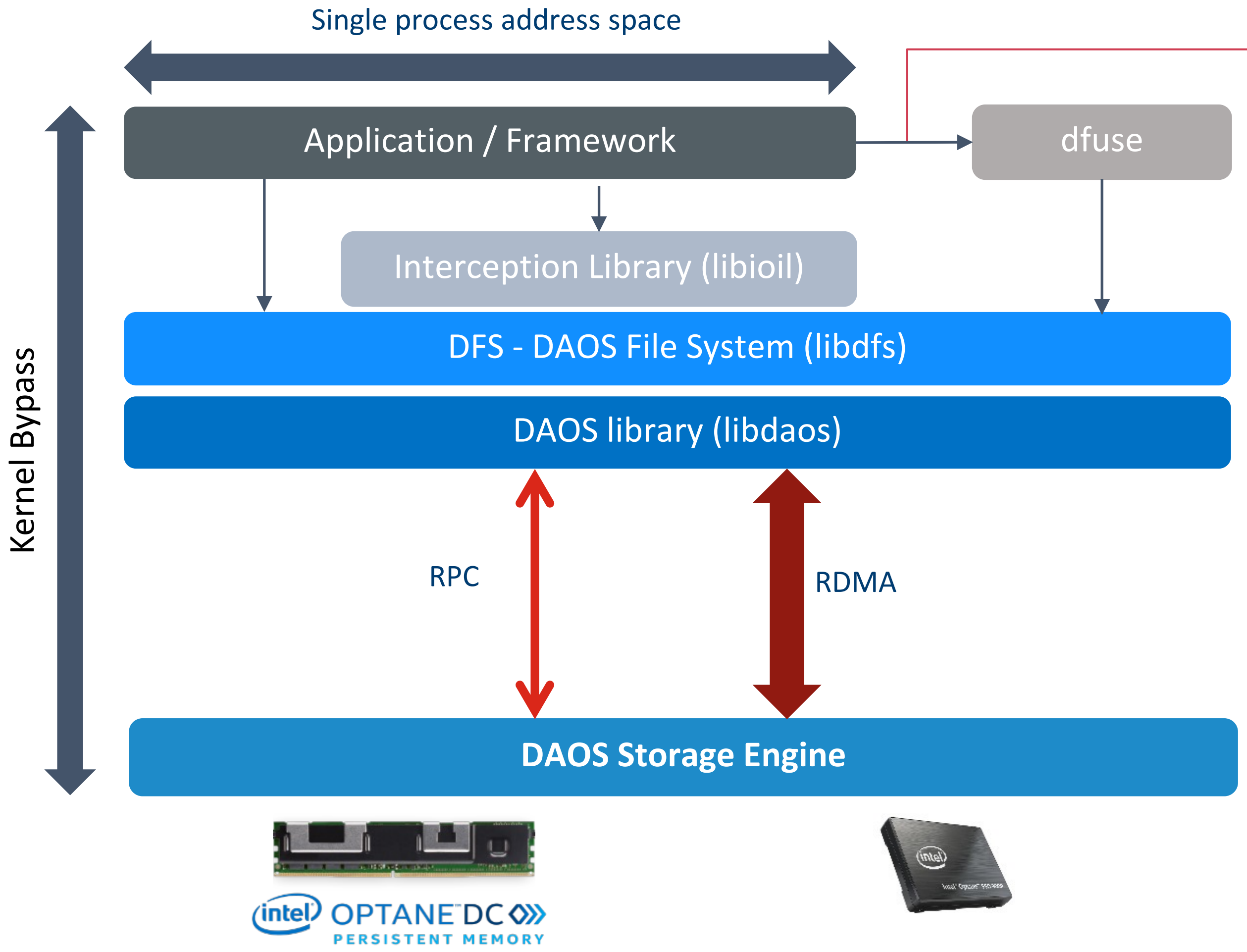
Pools and Containers

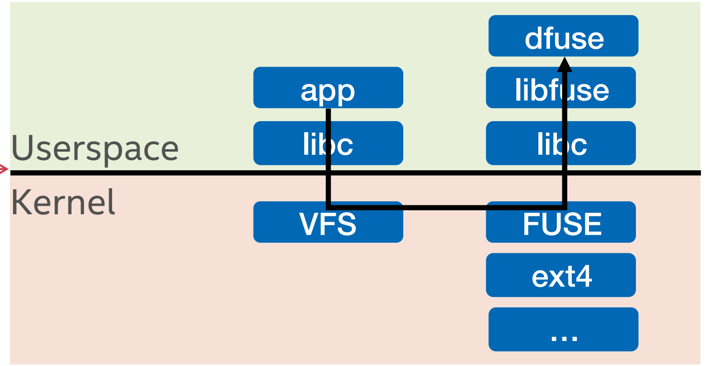
- Aurora will assign pools to projects
  - Large allocations will receive pools with ~80% of available targets
    - Number of targets proportional to performance
  - Pools are a physical allocation (guaranteed allocation of storage)
  - Users of the project will be given full rights to the pool
  - Users create their own containers with their desired settings
    - The initial pool will have the suggested defaults from ALCF
- POSIX containers can be mounted on Aurora via a scheduler flag
  - Dfuse will be started running as the user
  - Container mounted at known mount point
  - Will be able run applications from DAOS
- Lustre/DAOS integration should allow easy POSIX container access
  - Access DAOS POSIX containers via existing Lustre mount points

Containers

- User created
- Ability to select
  - Data protection
    - Checksums
    - Redundancy factor
  - EC Cell size
  - Features
    - compression
    - encryption
  - Security
    - ACLs

POSIX I/O Support





- User space DFS library with an API like POSIX.
  - —Requires application changes (new API)
  - —Kernel Bypass, no client cache
- DFUSE plugin to support POSIX API
  - —No application changes
  - —Fuse Kernel Supports data (wb and ra) & metadata caching (stat, open, etc.)
- DFUSE + IL
  - —No application changes, runtime LD_PRELOAD
  - —Kernel Bypass for raw data IO only.

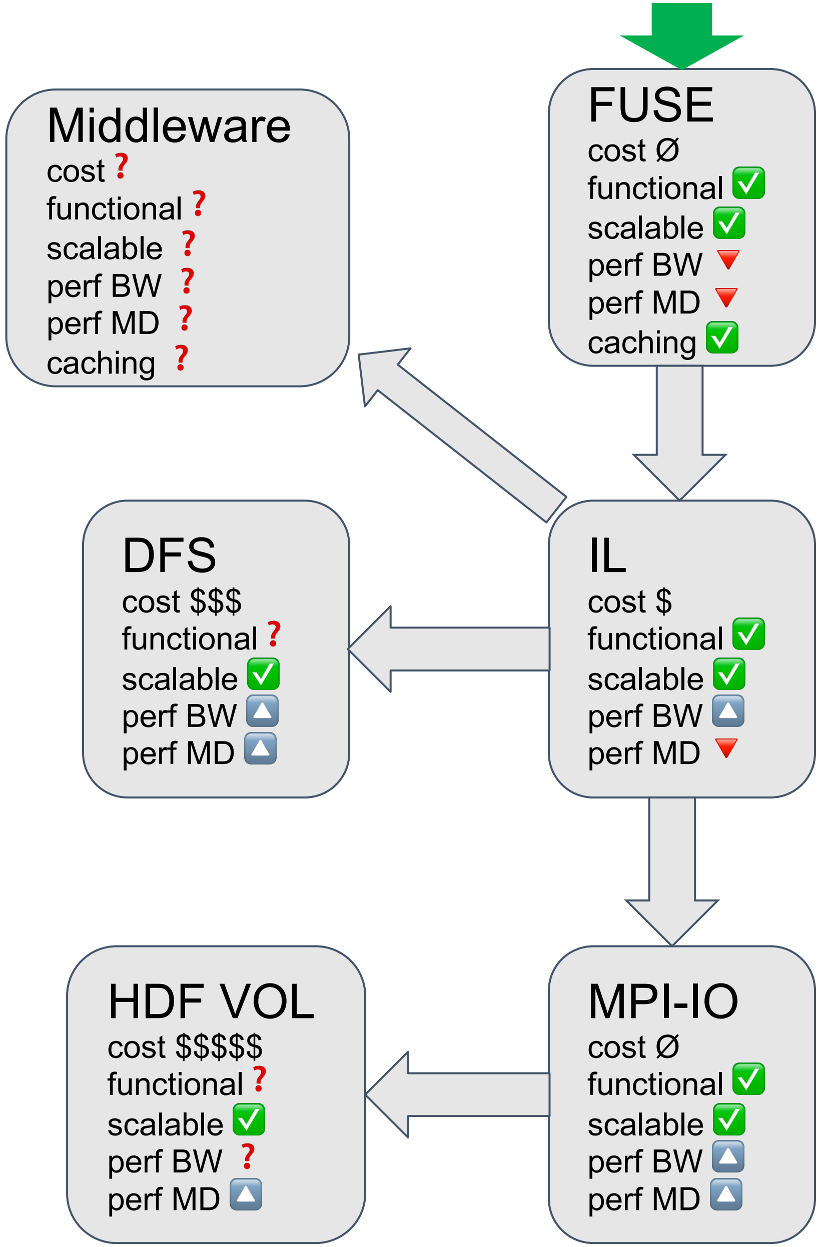
MPI-IO Driver for DAOS

- The DAOS MPI-IO driver is implemented within the I/O library in MPICH (ROMIO).
  - —Added as an ADIO driver
  - —Available in Intel MPI
  - —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.
- How to use this driver?
  - —MPICH: append “daos:” to the file name/path or set env variable:
    - ▪ `ROMIO_FSTYPE_FORCE="daos:"`

Other middleware with custom DAOS backends...

DFS API very similar to POSIX but requires porting all your I/O code. Allows ability for low level control of objects.

HDF VOL usage will require (potentially) significant rework. Need to understand usage before making changes



Middleware
cost ?
functional ?
scalable ?
perf BW ?
perf MD ?
caching ?

FUSE
cost Ø
functional ✅
scalable ✅
perf BW 🔻
perf MD 🔻
caching ✅

DFS
cost $$$
functional ?
scalable ✅
perf BW 🔼
perf MD 🔼

IL
cost $
functional ✅
scalable ✅
perf BW 🔼
perf MD 🔻

HDF VOL
cost $$$$$
functional ?
scalable ✅
perf BW ?
perf MD 🔼

MPI-IO
cost Ø
functional ✅
scalable ✅
perf BW 🔼
perf MD 🔼

dFuse provides no effort path. Performance will be suspect, but does provide caching and buffering which can be positive for performance.

Interception Library low effort with large performance upside on BW. Potential issue with functionality if more esoteric interfaces used.

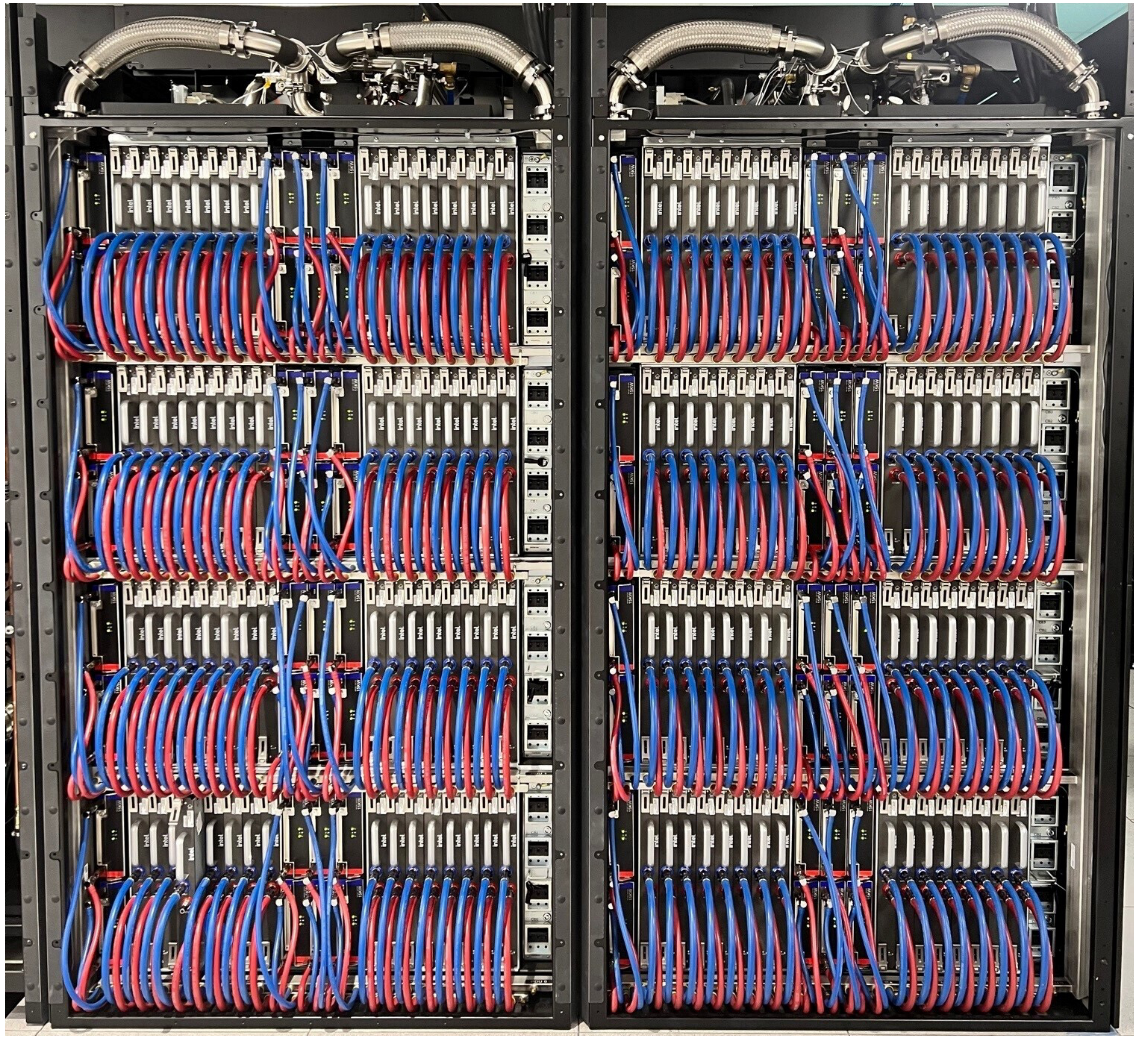
MPI-IO with DAOS ADIO Should be transparent and provide best possible performance.

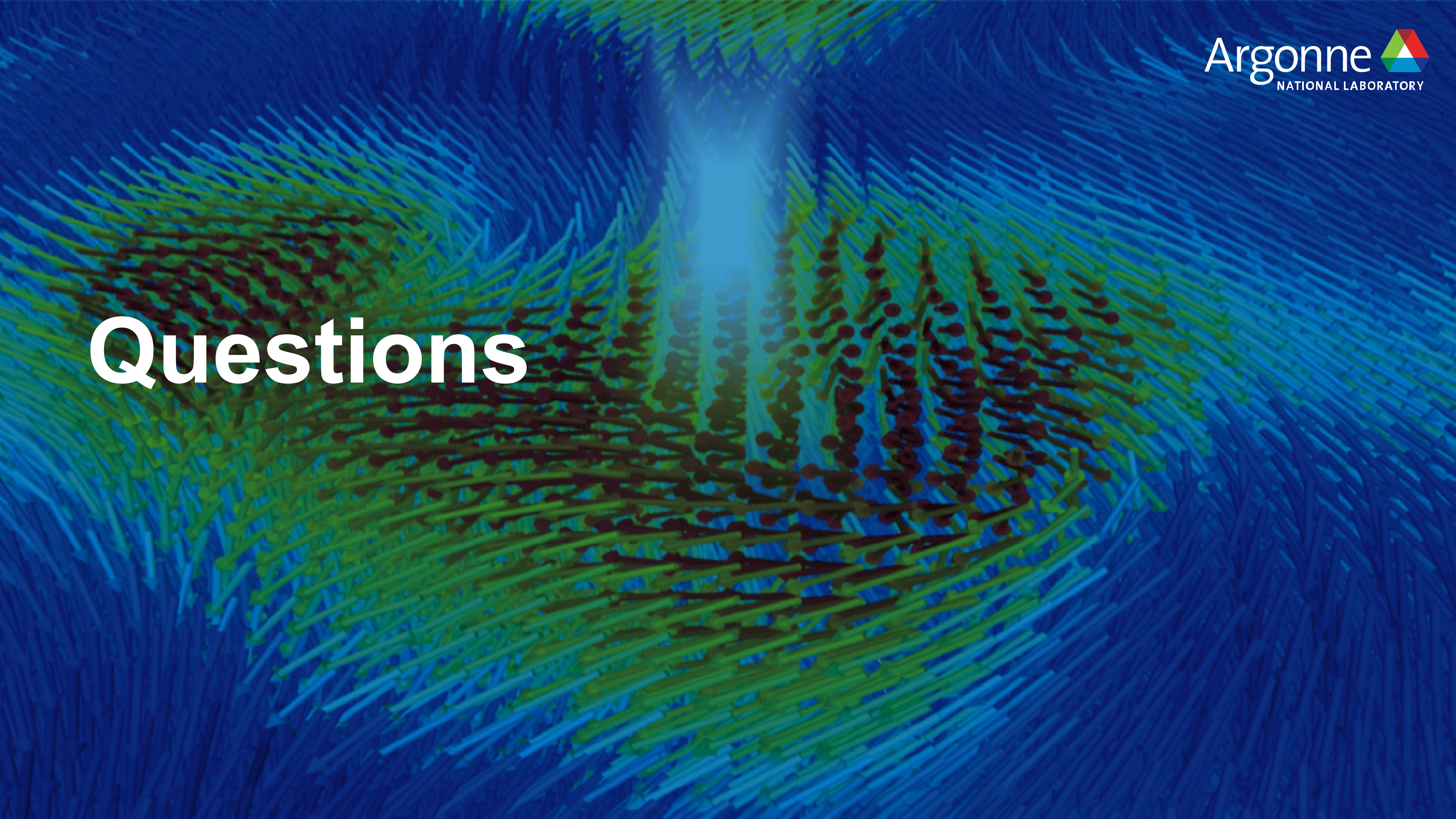
Sunspot

Sunspot

https://www.alcf.anl.gov/support-center/aurora/getting-started-sunspot

- ALCF's Test and Development system
  - Think of it as a baby Aurora
- Two compute racks / groups
  - 128 compute nodes
- DAOS deployment
  - 20 DAOS nodes
  - Identical server configuration to Aurora
  - Allows running EC16+2 – 18 nodes with 2 nodes for failover
- Production environment for DAOS at ALCF
  - Follow pool and container usage plan for Aurora
  - 1 pool per project
    - ACL limits pool to project members
    - Users create containers
  - Suggested default data protection of EC16+2 on containers





Questions