Distributed Asynchronous Object Storage (DAOS): A New Storage Paradigm

Kalyana Chadalavada
High Performance Data Division, Intel
Intel HPC Developer Conference, Nov 2017
Notices

Acknowledgment: This material is based upon work supported by Lawrence Berkeley National Labs subcontracts 7078611 and 7216501 and Lawrence Livermore National Labs subcontract B608115.

Disclosure Notice: This presentation is bound by Non-Disclosure Agreements between Intel Corporation, the Department of Energy, and DOE National Labs, and is therefore for Internal Use Only and not for distribution outside these organizations or publication outside this Subcontract.

USG Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Intel Disclaimer: Intel makes available this document and the information contained herein in furtherance of DesignForward, FastForward and the Extreme Scale Initiative. None of the information contained therein is, or should be construed, as advice. While Intel makes every effort to present accurate and reliable information, Intel does not guarantee the accuracy, completeness, efficacy, or timeliness of such information. Use of such information is voluntary, and reliance on it should only be undertaken after an independent review by qualified experts.

Access to this document is with the understanding that Intel is not engaged in rendering advise or other professional services. Information in this document may be changed or updated without notice by Intel.

This document contains copyright information, the terms of which must be observed and followed.

Reference herein to any specific commercial product, process or service does not constitute or imply endorsement, recommendation, or favoring by Intel or the US Government.

Intel makes no representations whatsoever about this document or the information contained herein. IN NO EVENT WILL INTEL BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS DOCUMENT, INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, OR OTHERWISE, EVEN IF INTEL IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Copyright © 2017 Intel Corporation. All rights reserved.
Agenda

• Storage Challenges and DAOS overview
• Exascale Storage Vision
• Exascale Storage Stack
• Middleware I/O & Applications
• DAOS/Lustre* Integration
Today’s HPC Storage System Pain Points

• HPC storage systems perform poorly with random, unaligned or non-large I/Os
  • Require larger & larger well-aligned sequential I/Os

• Scientific data models limited by POSIX®
  • One-size-fits-all POSIX data model
  • **Worst-case** concurrency control mechanism

• Hitting scalability limits of traditional PFS
Challenge: I/O Latency & IOPS

- HDD
- Software stack
Challenge: I/O Latency & IOPS

- HDD
- Software stack

- NAND
- Software stack
Challenge: I/O Latency & IOPS

Traditional storage stack entirely masks low latency of 3D XPoint™!
Challenge: Access Granularity

Traditional storage stack entirely masks low latency & capabilities of 3D XPoint™!
**Distributed Asynchronous Object Storage**

- **Scale-out object store** designed from the ground up for nextgen storage & fabric technologies
  - High **throughput/IOPS**
  - Byte addressable
  - **OS bypass** with lightweight client/server
- **Advanced storage API**
  - New scalable **storage model** suitable for both **structured** & **unstructured** data
  - **Non-blocking** data & metadata operations

---

* I/O middleware not ported to DAOS
** I/O middleware prototyped over DAOS

---

**HPC Apps**
- (HACC, ACME, CLAMR, ...)

**Big Data & AI Apps**
- Spark
- RDF*
- (No)SQL*
- (N)FS* (Ganesha)
- S3* Swift*

**Enterprise & Cloud Apps**
- Legion
- NetCDF

**HPC**
- Legion
- NetCDF

**Big Data & AI**
- (No)SQL*
- (N)FS* (Ganesha)
- S3* Swift*

**Enterprise & Cloud**
- Argobots
- Mercury

**I/O Middleware**
- POSIX I/O*
- MPI-I/O*
- SCR* VeloC* Dataspaces*

---

**DAOS**
- Open Source Apache 2.0 License

---

**Storage Models**
- Byte-granular data/metadata
- Bulk (e.g. checkpoints)

---

**Distributed Synchronous Object Storage**

- **Scale-out object store** designed from the ground up for nextgen storage & fabric technologies
  - High **throughput/IOPS**
  - Byte addressable
  - **OS bypass** with lightweight client/server
- **Advanced storage API**
  - New scalable **storage model** suitable for both **structured** & **unstructured** data
  - **Non-blocking** data & metadata operations
**Distributed Asynchronous Object Storage**

- Scale-out object store designed from the ground up for nextgen storage & fabric technologies
  - High throughput/IOPS
  - Byte addressable
  - OS bypass with lightweight client/server
- Advanced storage API
  - New scalable storage model suitable for both structured & unstructured data
  - Non-blocking data & metadata operations

---

Open source
APACHE 2.0 License
https://github.com/daos-stack

---

DAOS
Open Source Apache 2.0 License

- HPC Apps (HACC, ACME, CLAMR, ...)
- Big Data & AI Apps
- Enterprise & Cloud Apps

- Legion
- NetCDF
- POSIX I/O
- MPI-I/O
- SCR
- FIT
- Veloc
- Dataspace
- Spark RDD
- (No)SQL
- NFS
- S3
- S3
- Swift
- HDFS
- NVRAM
- NVMe

- Byte-granular data/metadata
- Bulk data (e.g. checkpoints)

* I/O middleware not ported to DAOS
** I/O middleware prototyped over DAOS
Exascale Storage Vision

• NVM storage storing datasets
  • **Externally** accessible

• **System namespace**
  • **Global** POSIX namespace
  • **Links** to datasets
  • Binaries, libraries, user files, ...

- System Namespace
  - /projects
    - /posix
    - /Particle
    - /Climate

- POSIX dataset
- ADIOS
- HDF5 File

- Data streaming (e.g. from instrument(s))
- Ephemeral session-based datasets

- S3, MarFS, HSM, ....
Exascale Storage Stack

New storage API (*DAOS*) provides extended capabilities and high bandwidth/IOPS to middleware.
Exascale Storage Stack

Port I/O middleware (HDF5, MPI-I/O, ...) to new storage backend and **augment** API to take advantage of new capabilities

**New** storage API (DAOS) provides extended capabilities and high bandwidth/IOPS to middleware
Exascale Storage Stack

Evaluate applications (HACC, ACME, CLAMR) and new programming model (Legion) over enhanced I/O middleware

Port I/O middleware (HDF5, MPI-I/O, ...) to new storage backend and augment API to take advantage of new capabilities

New storage API (DAOS) provides extended capabilities and high bandwidth/IOPS to middleware
Lightweight Storage Stack

• Mercury user space function shipping
• Applications link directly with DAOS lib
• Userspace DAOS server
  • Mmap non-volatile memory (NVML)
  • NVMe access through SPDK/BlobFS
Storage Model

- **DAOS Tier**
  - Pool
  - Container
  - Object
  - DKey
  - Akey[i]

- **I/O Middleware**
  - dkey
  - akey

- **Storage Backend**
  - Record size = 1 Byte (Byte Array)
  - Record size = 1024 Bytes
  - Single any-size value

The information on this page is subject to the use and disclosure restrictions provided on the second page to this document.
HDF5

- Mapping HDF5 to DAOS:
  - HDF5 file -> DAOS Container
  - HDF5 Objects -> DAOS KV objects

- HDF5 DAOS VOL Plugin
  - Prototyped in ESSIO
  - All applications or middleware I/O libraries (e.g. NetCDF4, PIO, etc.) that use HDF5 would be able to utilize the DAOS tier with minimal changes.
  - Newly developed applications or I/O libraries can utilize new extensions to HDF5 that are not available to date without the DAOS VOL plugin (some might be added to the POSIX HDF5 plugin in the future):
    - Asynchronous I/O for both metadata and raw data operations
    - Query, Indexing, & Analysis shipping
    - Container Snapshots
    - User controlled transactions
    - End to End data integrity
POSIX® I/O

• POSIX Encapsulation
  • Each DAOS container encapsulates a namespace.
  • Highly scalable I/O to single shared file or file per process with full OS bypass.
  • Relaxed POSIX compliance
    • OK for most applications
    • Strong compliance comes at the price of complexity and performance.

• POSIX Extensions
  • Asynchronous I/O operations.
  • POSIX namespace snapshots

• Not yet implemented

POSIX® is a registered trademark of the Institute of Electrical and Electronic Engineers, Inc.
**MPI-I/O**

- Mapping MPI-I/O to DAOS:
  - 1 DAOS container to hold 1-n MPI-I/O files (single shared file, file per process, etc.).
  - File striped across multiple object D-Keys

- **MPI-I/O Support**
  - Implement an ADIO driver in ROMIO (widely used as the de-facto MPI-I/O implementation in most MPI libraries).
  - Minimal application modification (set a hint to use the DAOS driver) + Supports middleware libraries that use MPI-I/O but have not implemented a DAOS driver as a backend.
  - Scalable mapping of an MPI file to a DAOS object with implicit stripping across multiple Distribution Keys.
  - Consistency and Recoverability features of DAOS epochs can be exposed through `MPI_File_sync()` that advances the container epoch.

- Not yet implemented
Application Evaluation

- **Legion**
  - Data Centric programming model

- **Hardware/Hybrid Accelerated Cosmology Code**
  - Improved fault tolerance by storing transactional checkpoints

- **Cell-Based Adaptive Mesh Refinement**
  - Use HDF5 instead of POSIX I/O

- **Accelerated Climate Modeling for Energy**
  - Ported NetCDF & PIO to HDF5 DAOS VOL plugin
DAOS/Lustre* Integration

• DAOS Tier
  • **Checkpoint/defensive I/O**
  • Advanced data **analytics**
  • New data intensive **workflow**
  • New data-centric **programming models**
  • Storage media
    • 3D Xpoint™ **NVDIMMs**
      • byte-granular data & metadata
    • **Intel® 3D NAND SSD** or 3D Xpoint™ **SSDs**
      • bulk data, including checkpoint data

• Lustre* Tier
  • **Robust** system namespace
    • Mature & scalable POSIX namespace
    • Rich feature sets
  • **Smooth** migration path
    • Lustre* directly accessible through Mercury IOF
    • Slowly migrate applications to DAOS
    • APPs with strong POSIX requirements
  • Storage media
    • Dual-ported **JBOD**
    • Dual-ported **JBOF**
Questions?

Contact:
- kalyana.chadalavada@intel.com
- johann.lombardi@intel.com

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