

# DAOS Development Update

# DAOS Development

- ESSIO (completed):
  - DAOS Prototype (Data Model, API)
  - N-Way Replication with online Rebuild
  - Metadata Replication using RAFT
  - Non-blocking Operations using Task Scheduling Engine (TSE)
- Ongoing:
  - NVMe Support with SPDK
  - Persistent service

# I/O Middleware (I)

- HDF5:
  - DAOS Virtual Object Layer (VOL) plugin prototype developed in ESSIO
  - Ported several applications and higher middleware I/O libraries (NetCDF4 & PIO) to work on top of the DAOS plugin.
  - New HDF5 extensions prototyped that are not available to date in an HDF5 release:
    - Asynchronous I/O, Query & Indexing API, Snapshots
  - Future work: Productize the HDF5 prototype & develop Extensions

# I/O Middleware (II)

- MPI-I/O:
  - Prototype ROMIO ADIO driver on top of DAOS
  - Most MPI applications/middleware will work seamlessly when DAOS driver is selected.
  - Unsupported functionality include:
    - Shared File Pointer operations
    - MPI File atomicity
    - Pre-allocate
  - [https://github.com/mchaarawi/mpich/tree/daos\\_adio](https://github.com/mchaarawi/mpich/tree/daos_adio)
- POSIX I/O:
  - We envision two operating modes:
    - Fully encapsulated namespace with relaxed POSIX compliance
    - Leverage Lustre for a strong POSIX compliant namespace and store file data in the DAOS tier

# I/O Middleware (III)

- Dataspaces:
  - Collaboration with Rutgers
  - Prototype an integration of Dataspaces on top of DAOS
- PDC:
  - Collaboration with LBL + The HDF Group
  - Plugin for PDC server to use DAOS as an alternative to POSIX.
- Legion:
  - Data Centric Programming Model (Stanford)
  - ESSIO prototype to support the Legion framework on top of the HDF5 DAOS VOL plugin.
  - Redesigned DAOS Epoch model to support independent distributed access pattern like the Legion Framework
- PIO/NetCDF4:
  - NetCDF4 internally uses HDF5 for its backend I/O
  - Leveraged the HDF5 DAOS VOL plugin to store NetCDF data in DAOS

# Applications

- HACC I/O (ESSIO App):
  - I/O Kernel for the Hardware Accelerated Cosmology Code
  - Developed an HDF5 backend that reduces required internal bookkeeping and metadata as compared to using POSIX I/O or MPI-I/O
  - Ported to use the DAOS HDF5 VOL plugin
- CLAMR (ESSIO App):
  - Cell-Based AMR application
  - Reworked I/O strategy to use parallel I/O with HDF5
  - Ported to use the HDF5 DAOS VOL Plugin.
- ACME (ESSIO App):
  - Accelerated Climate Modeling for Energy
  - Uses PIO / NetCDF4 / HDF5
  - Ported to use the HDF5 DAOS VOL Plugin.

# Useful Tools & Language Bindings

- Go Language Binding for DAOS:
  - Go interface that covers most of the DAOS API
- daosfs for very simple POSIX filesystem semantics
  - Provide a C API that can be used by libraries like pNFS-GANESHA (FS abstraction layer) to provide a user-space file server.
  - or other services like a MYSQL server
- daos-fuse tool to expose a DAOS container via FUSE mount point
- Prototypes:
  - <https://github.com/daos-stack/go-daos>

# Documentation

- Source code:
  - <https://github.com/daos-stack>
- Users Website + Mailing list + Materials
  - <https://daos.groups.io/g/users>
- Installation:
  - <https://wiki.hpdd.intel.com/display/DC/DAOS+Community+Home>