Lustre User Group 2013 | China and Japan
Hosted by OpenSFS
Beijing - October 15 Tokyo - October 17
Sponsored by: intel

I/O Dispatcher (IOD)

-Transactional Burst-Buffer above DAOS/Lustre

Xuezhao Liu
EMC FastData Group

Agenda

- EMC aBBa interposed into HPC storage stack
- IOD enabled transactional BurstBuffer above DAOS/Lustre
- Summary

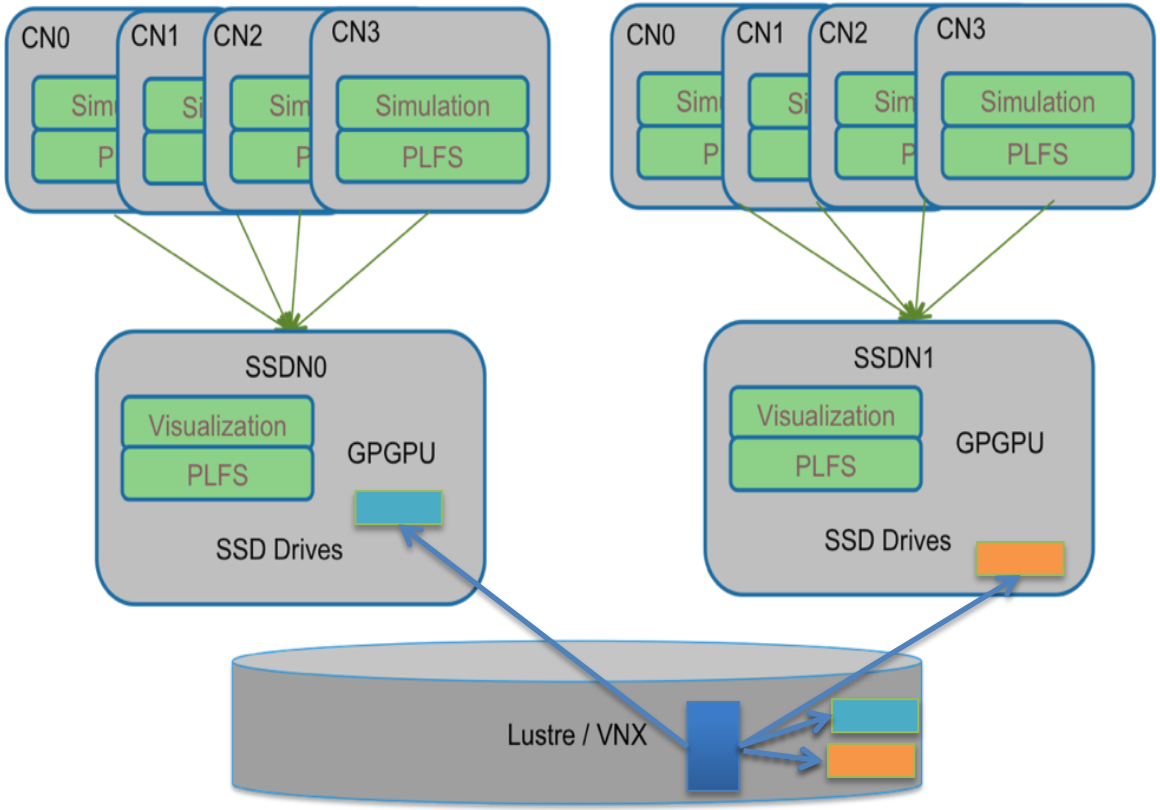
Shared and Bursty I/O of HPC APP

- Shared or partially shared file usage becomes the predominant method
- Burstiness of I/O
  - For 98% of the time, the I/O system was utilized at less than 33% of peak I/O bandwidth. (ANL, BG/P, 2011)
  - High peak bandwidth mainly for checkpointing

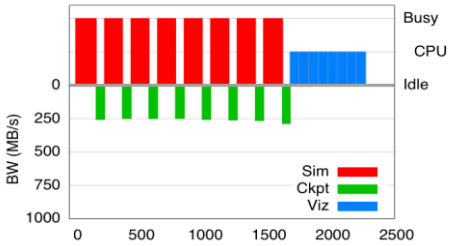
- EMC aBBa fully fits the gap between APP and PFS

A New Storage Tier -- Burst Buffer

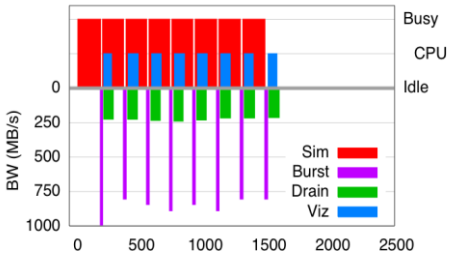
- Existing disk-based storage stack will not scale
  - Disk-only too expensive for bandwidth
  - Flash-only too expensive for capacity
- EMC built prototype burst buffer systems at SC11/SC12/SC13
- 4x faster checkpoint
- Jitter-free co-processed analysis
- 30% total time to completion reduction



Workflow without burst buffers

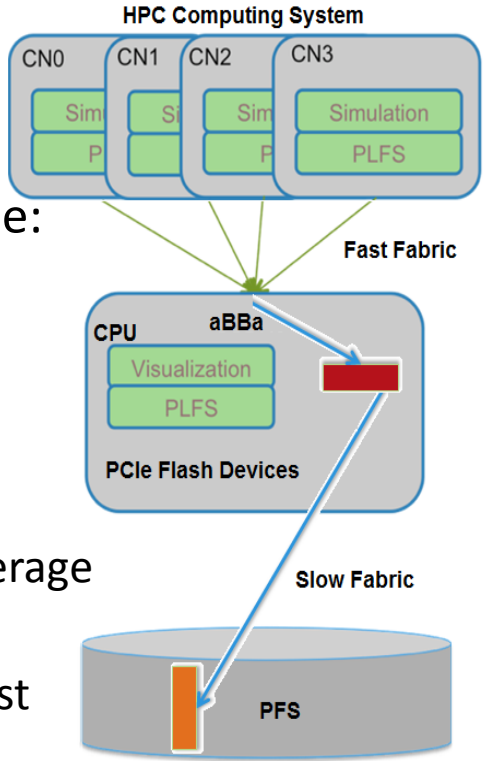


30% faster workflow with burst buffers

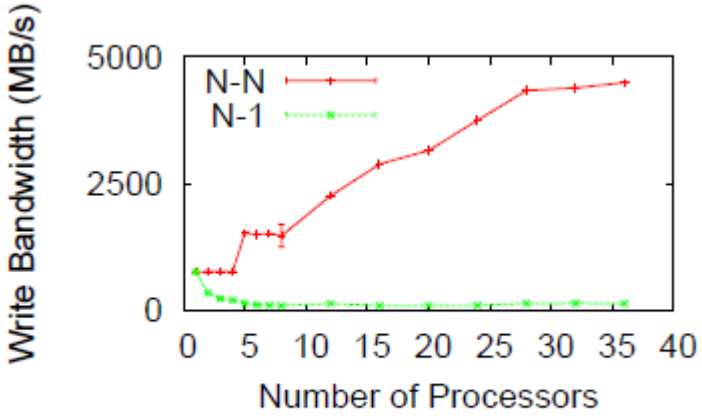


aBBa – Network Flash Appliance

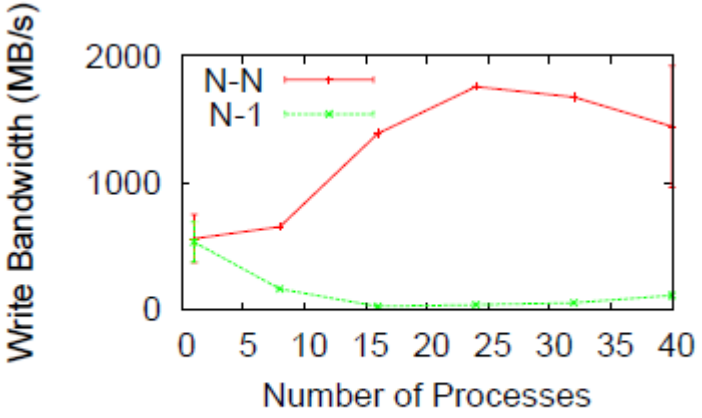
- Active Burst Buffer Appliance - aBBa
  - ABBA = Flash + Compute in the network
  - File System Client Offload to Dedicated Burst Buffer Node: ABBA
  - Flash component as cache write-back
    - Deliver the high BW needed by CN's
    - Allow use of disks for capacity with low disk BW
    - Allow provisioning the storage system for capacity and average BW not peak
    - Allow store checkpoint data until drained slowly to PFS, fast recovery in case of failure
  - ABBA is File System agnostic (Lustre, Isilon, PanFS, HDFS, etc.)
  - **PLFS** (Parallel Log-structured FS) is key
  - Application offload to ABBA: Visualization, analytics co-processing .



PLFS' Motivation: N-1 Concurrent Writing doesn't Scale

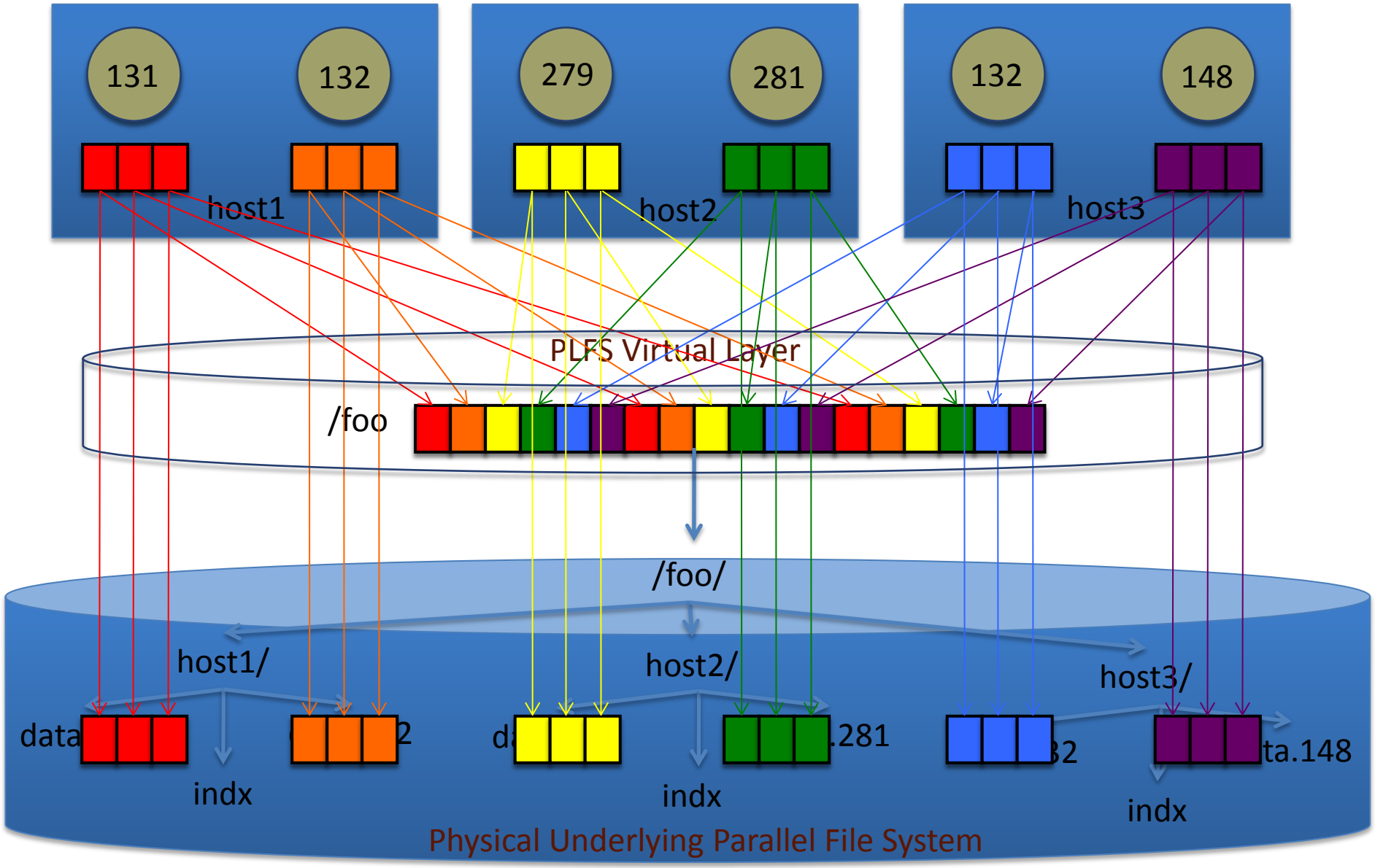


GPFS



Lustre

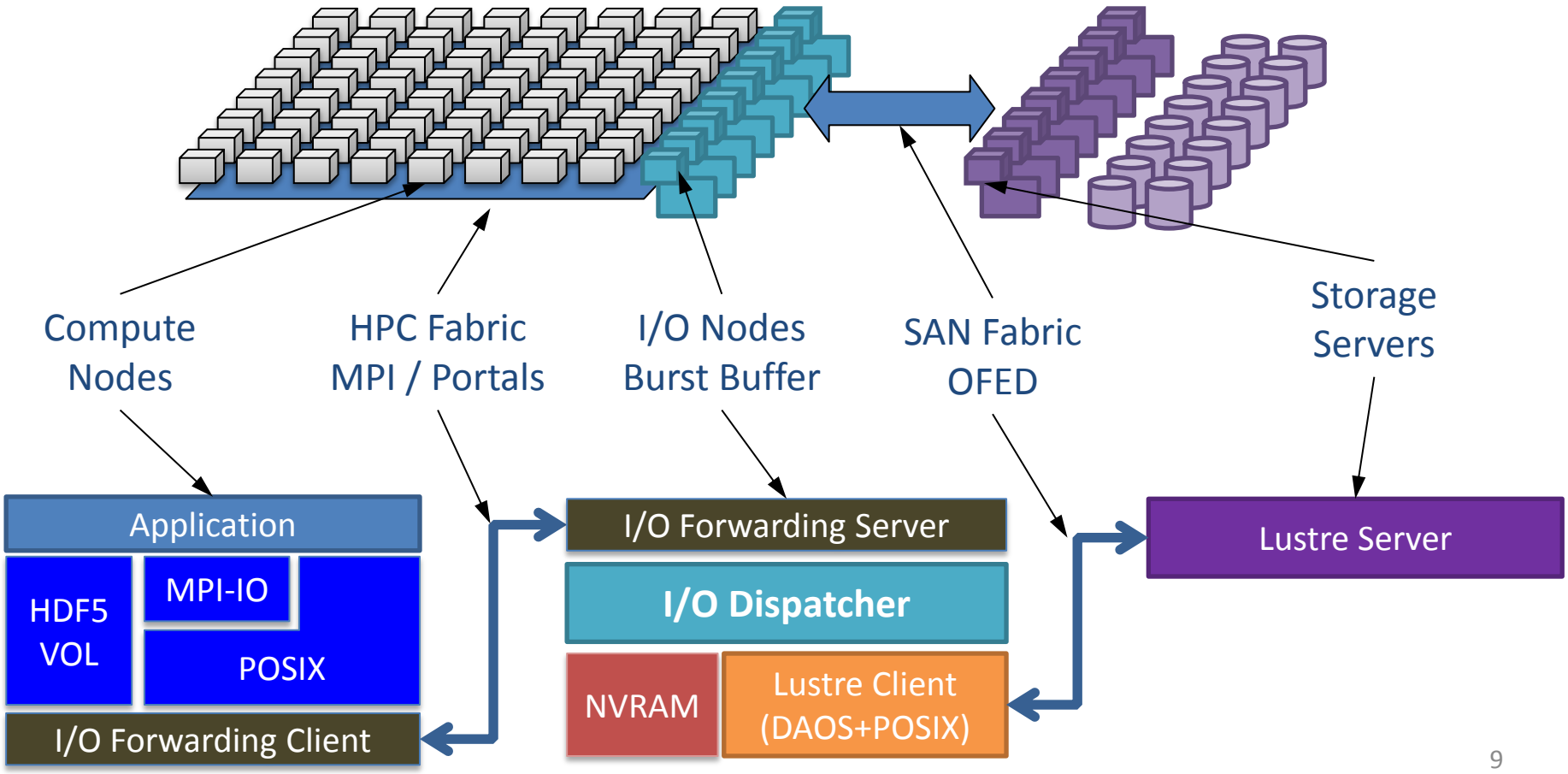
- Many scientific applications create checkpoints using small, strided, concurrent writes to a shared file (N-1 checkpoint)
- Small strides, small writes
  - May be un-aligned
    - Read-modify-write
  - Lock contention
  - Disk seeking


131
132
279
281
132
148
host1
host2
host3
PLFS Virtual Layer
/foo
/foo/
host1/
host2/
host3/
data
indx
indx
indx
Physical Underlying Parallel File System

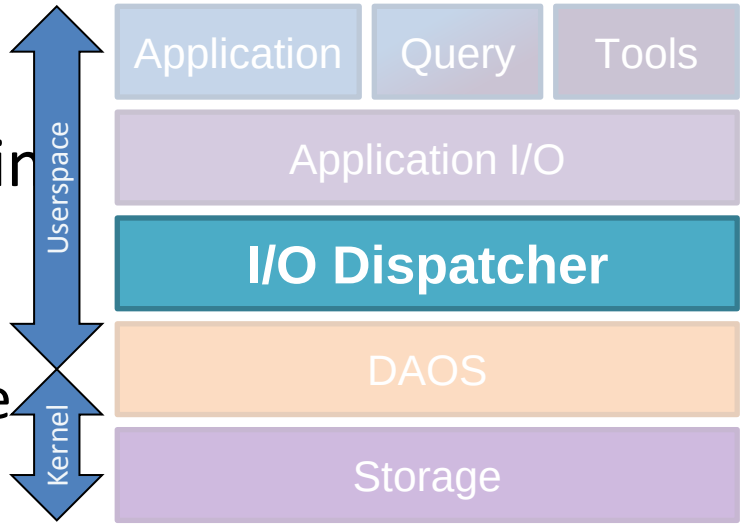
Agenda

- EMC aBBa interposed into HPC storage stack
- IOD enabled transactional BurstBuffer above DAOS/Lustre
  - General introduction
  - Innovative semantics (transactional object storage, semantics awareness)
  - Interactions with DAOS
- Summary

Fast Forward I/O Architecture



I/O Dispatcher



I/O rate/latency/bandwidth matchin
- Burst buffer / prefetch cache
- Absorb peak application load
- Sustain global storage performance

Layout optimization
- Application object aggregation / sharding
- Upper layers provide expected usage

Higher-level resilience models
- Exploit redundancy across storage objects

Scheduler integration
- Pre-staging / Post flushing

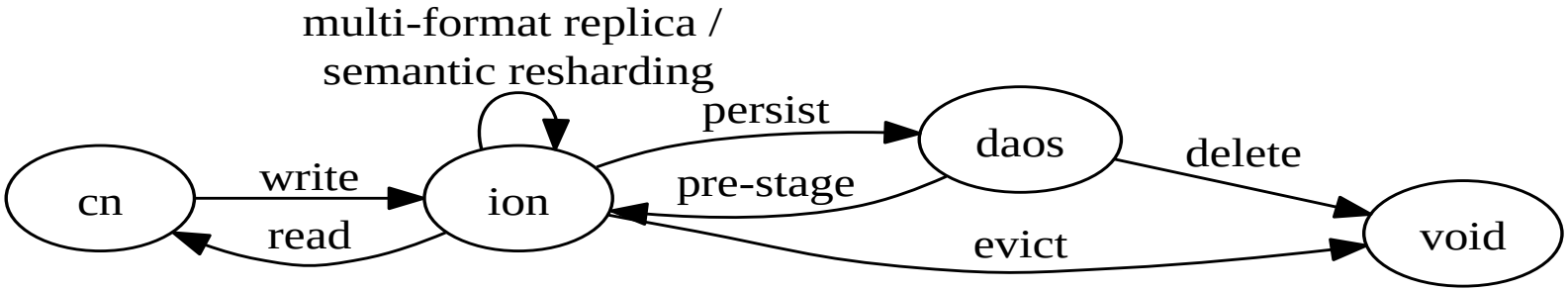
Characteristics of Exascale Application I/O

- Application I/O will be object-oriented, not file-based
  - Instantiate and persist rich distributed data structures and application metadata
- Application I/O will be asynchronous
  - Non-blocking operations initiate I/O
  - Event queues signal completion
- Applications responsible for managing I/O conflicts
  - I/O system provides, but does not impose, appropriate and scalable mechanisms to resolve conflicting operations
  - Avoids unnecessary serialization.
- Applications use transactional I/O model
  - All operations in a given transaction will succeed or fail
  - Failures in components and subsystems will occur

Abstraction Translation

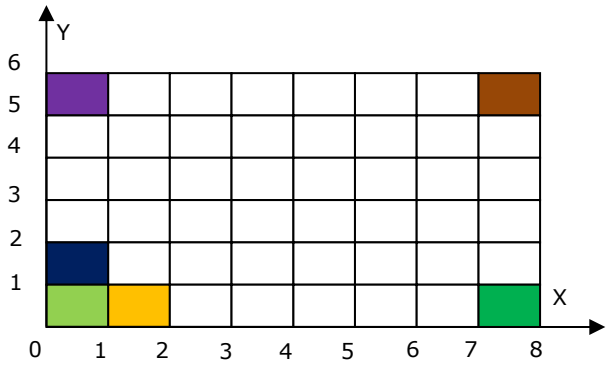
HDF5 Abstraction	IOD Abstraction	DAOS Abstraction
H5File	Container	Container
H5Group	KV object	Set of DAOS objects
H5DataType H5DataSpaces H5Attribute H5Properties H5Reference H5Link	KV pair in KV object	Data in a DAOS object
H5Dataset	Array object	Set of DAOS objects
H5CommittedDatatype	Blob object	Set of DAOS objects

IOD Managed Data Movement



- Transparent writing/reading between CN and BB (ION)
- Explicit, on-demand data movement between BB and DAOS
  - BB to BB
    - Multi-format replica is for blobs and KVs.
    - Semantic resharding is the same idea but for arrays.
  - BB to DAOS – persist
  - DAOS to BB – pre-stage with user preferred layout
  - Purge and punch

IOD Array-Object Layout and Re-Organization

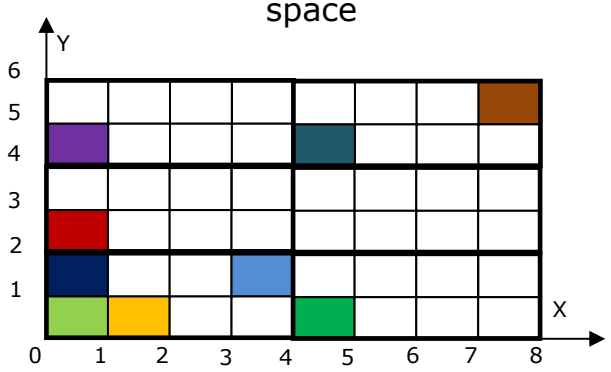


Layout sequence -- YX

Layout sequence -- XY

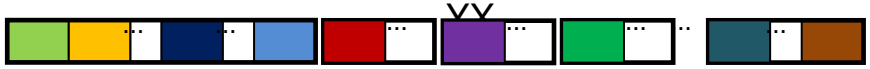
Contiguous layout

Dataset's logical multi-dimensional space



Array object's physical one-dimension space

Layout sequence -- YX



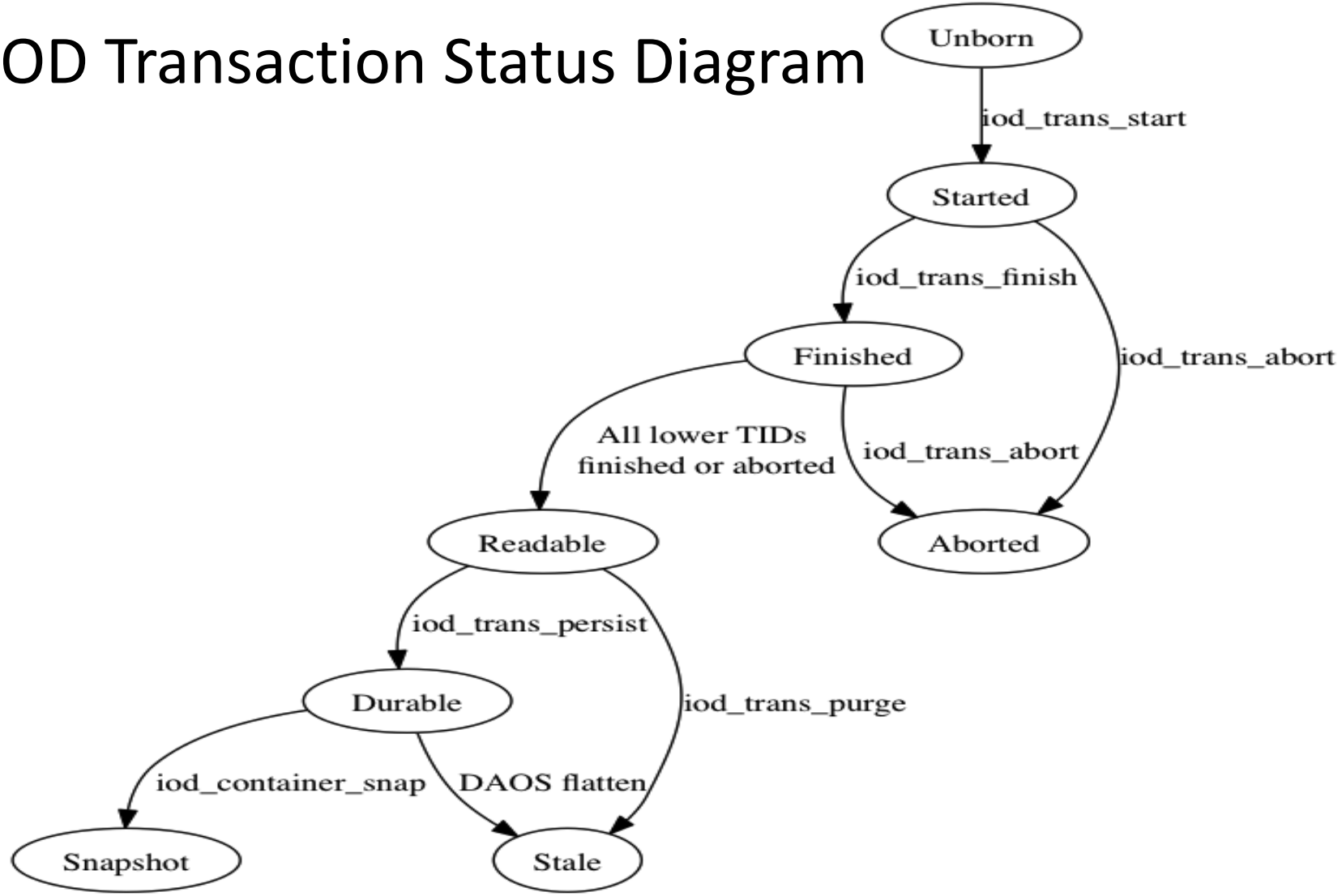
Layout sequence -- XY

Chunked layout

IOD Transaction Properties

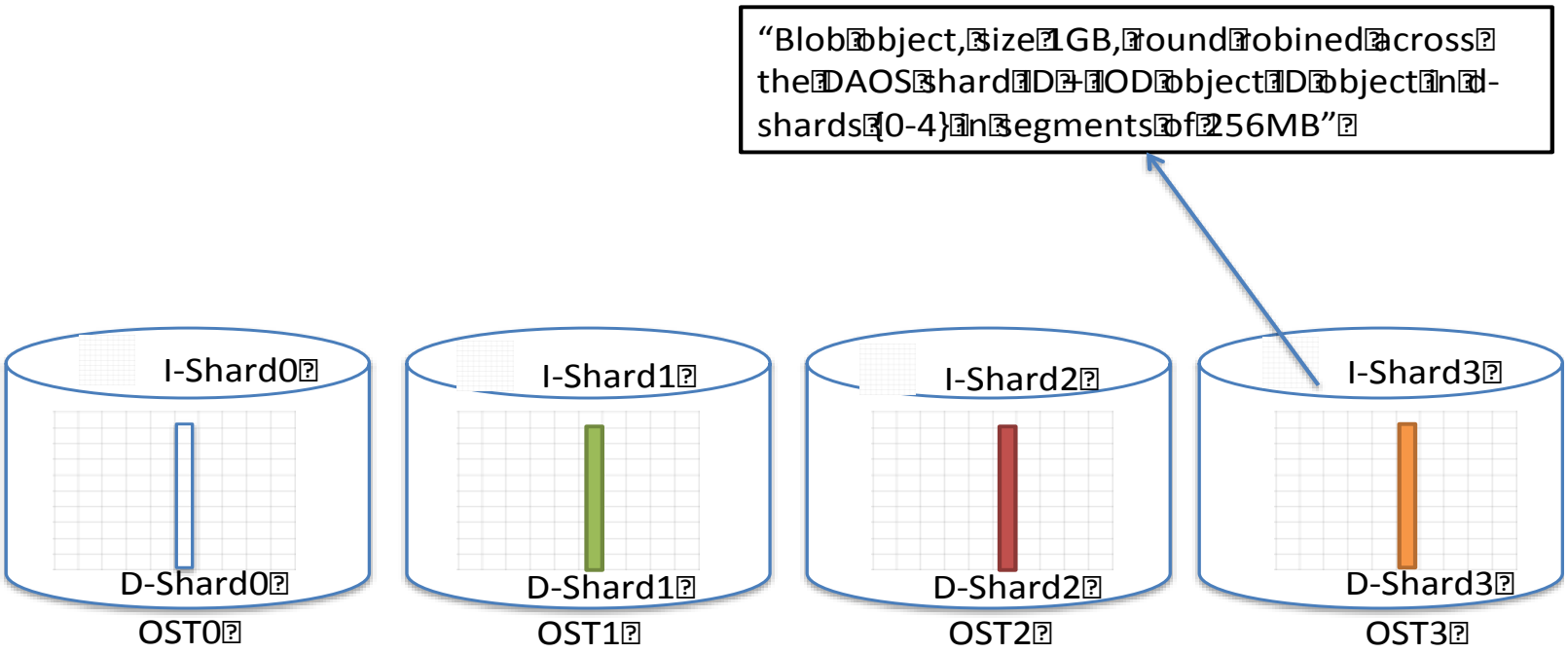
- **Atomic writes** – either all writes in a transaction are applied or none of them are.
- **Commutative writes** – concurrent writes are effectively applied in TID order, not time order.
- **Consistent reads** – all reads in a transaction may "see" the same version data even in the presence of concurrent writers.
- **Multiple objects** – any number of IOD objects within one container may be written in the same transaction.
- **Multiple threads** – any number of threads and/or processes may participate in the same transaction.

IOD Transaction Status Diagram

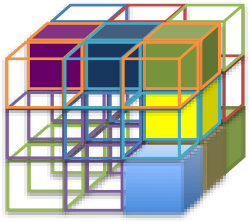


IOD Blob Object Storage on DAOS

- Virtual View:



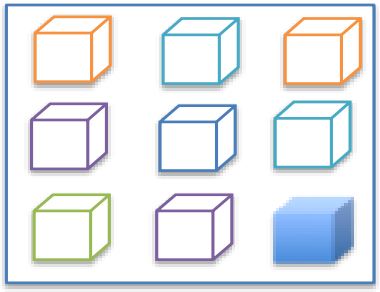
IOD Array Object storage on DAOS

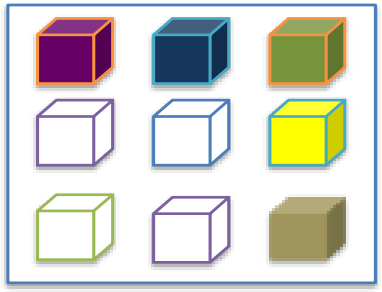


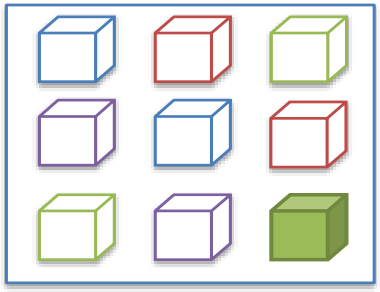
HDF Dataset
IOD Array Object

IOD Array Object on ION's, stream per writer, migrate semantically to DAOS

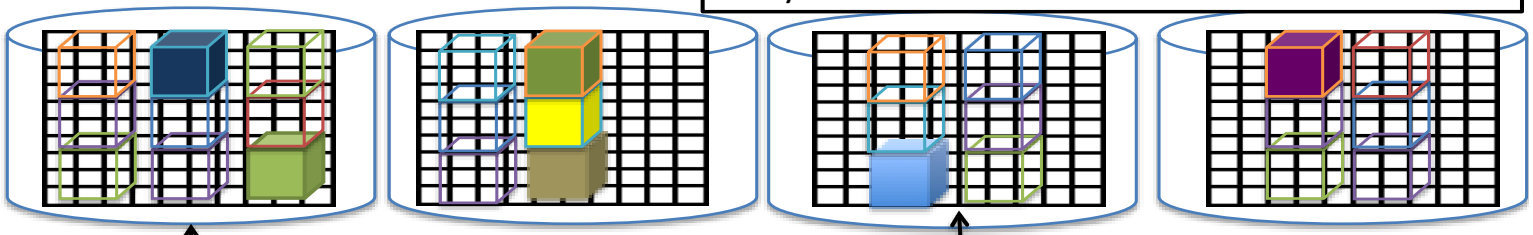
ION's







IOD Metadata: {3x3x3} Arry object, cell size 1048576, round robined across d-shards {0-4} in array semantic columns ordered from "x=0 .. x=2"



DAOS Storage Target

DAOS Shard

Agenda

- EMC aBBa interposed into HPC storage stack
- IOD enabled transactional BurstBuffer above DAOS/Lustre
- Summary

Summary

- EMC aBBa (Active Burst Buffer Appliance)
  - Matches the bursty I/O needs of the SC
  - Match the available slower disk system to the checkpoint draining
  - Allow to reverse the trend of disk for BW to disk for capacity
  - Application offloading: visualization, analytics co-processing
- I/O Dispatcher extended transactional burst-buffer
  - A buffering/optimizing layer above DAOS/Lustre
  - Richer/closer mechanisms provided to application
  - Transactional object storage with 3 major object types (KV, blob, array)
  - Semantics awareness and data re-organization
  - Fully asynchronous APIs
  - Natural well support for random writing (PLFS)
  - **Developing in progress, demonstrated internally …**
    will be open-source (LGPL)

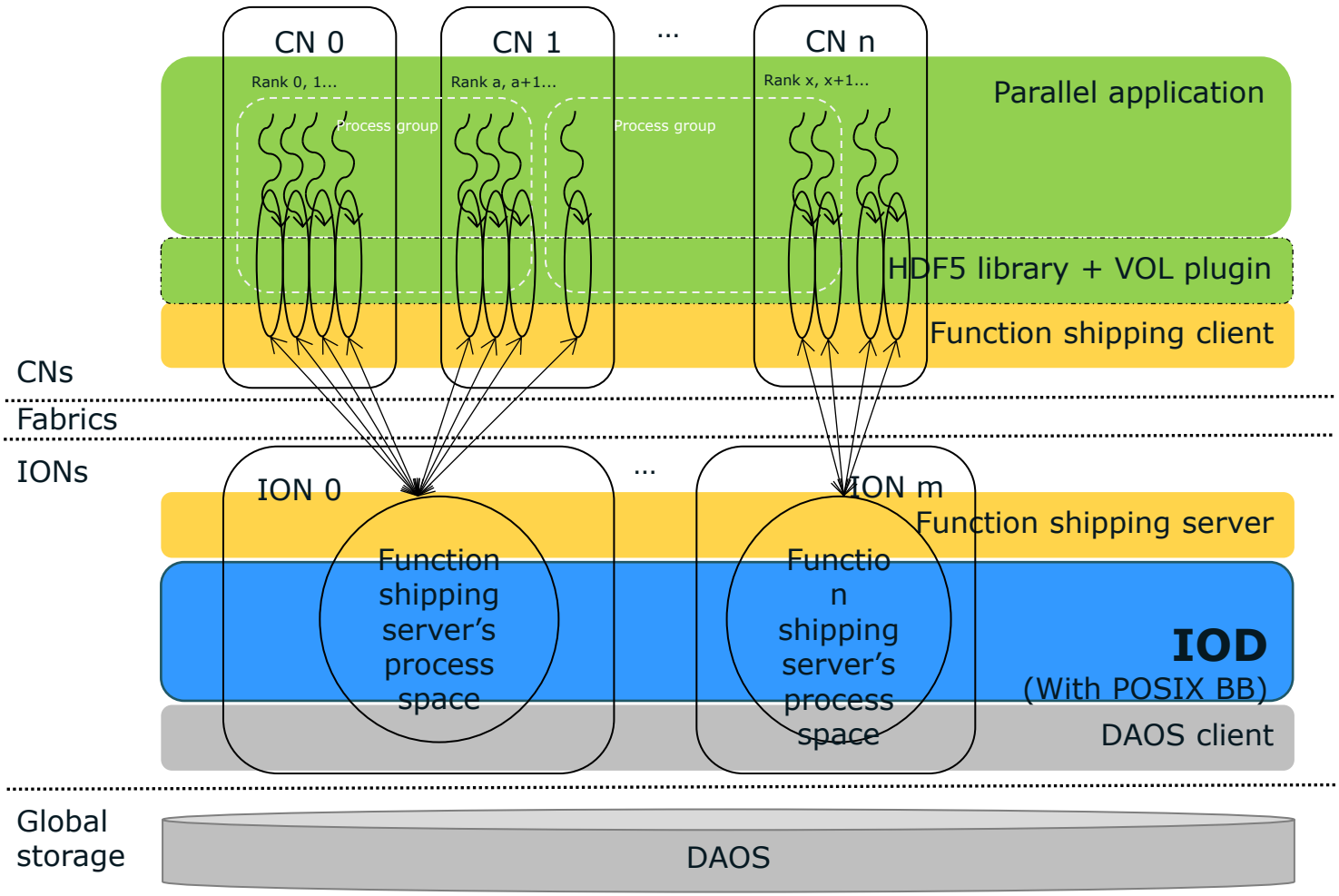
Q&A

Thanks

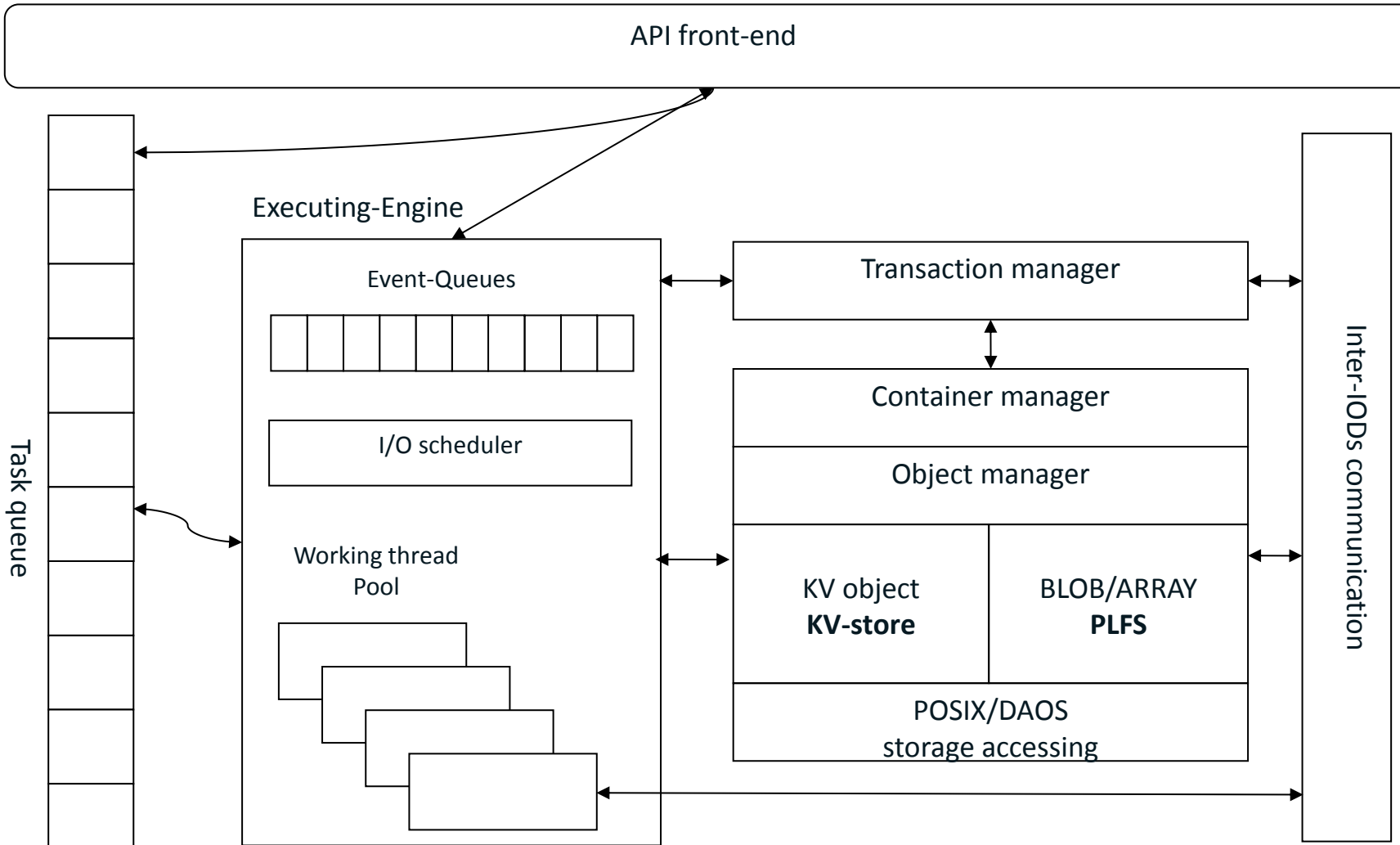
For IOD related information please contact:

John.Bent@EMC.com
Xuezhao.Liu@EMC.com

IOD in FF stack



IOD Sub-modules Overview



Parallel Log-structured FS (PLFS)

- Middleware SW management for ABBA
  - PLFS manages data movement on ABBA
  - PLFS organizes data for efficient bidirectional read/write CN ←→ ABBA ←→ Lustre
  - PLFS guarantees *just in time* data movement between ABBA←→Lustre (drains data between checkpoints; pre-fetches data when needed)
  - PLFS + ABBA guarantee fast checkpoint, removes bursty disk access behavior
  - PLFS + ABBA co-processing offload visualization from CN's and guarantee jitter free compute

Other Benefits with PLFS not Shown

- Better data organization
  - Faster reads
  - Applications no longer need to tune IO
- Directory sharding
  - More file metadata op/s
  - N-N create phase much faster
  - (Directory metadata ops like mkdir are slower)
  - Federate multiple filesystems into one namespace
- About to show seamless integration of flash
- Many other possibilities for future research
  - e.g. data services like dedup, data integrity, compression, etc

EMC Lustre activities: VNX HPC Series

- High IOPS/ Throughput
- Small Form Factor
- High Density
- Best Price/ Performance
- Enterprise Reliability, Availability and World class Service

EMC Lustre activities: VNX HPC Series

- Base Configuration is a Single Rack offering
  - 720 TB Capacity, 8 GB/s Performance
  - Pre-racked and configured VNX5100 and VNX7500
  - Servers for Management and File System
- Single Point of Management via Management Console from Terascala
- Application Ready - Pre-configured and tuned Lustre Parallel File System
- Infini-band (QDR) interface to computational node

IOD's Methodologies

- Objects instead of files
  - Blob objects for traditional sequences of bytes
  - Key-value stores for smaller get/put operations (MDHIM)
  - Array objects for semantic storage of multi-dimensional structure data
- Containers instead of directories
  - Snapshots for efficient COW across sets of objects
  - Transactions for atomic operations across sets of objects
- List I/O all the way through the stack
  - Reduce trips across network
- Everything fully asynchronous with distributed transactions
  - Reads, writes, commits, unlink, etc across sets of objects
- Explicit Burst Buffer management exposed to APP
  - Migrate, purge, pre-stage, multi-format replicas, semantics resharding
- End-to-end data integrity
  - Checksums store with data, APP can detect silent data corruption
- Co-processing analysis on in-transit data
  - Query and reorganize the placement of data structures before analysis shipping