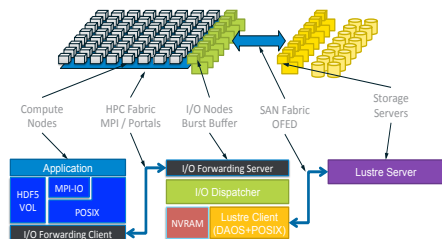


An Innovative Storage Stack Addressing Extreme Scale Platforms and Big Data Applications

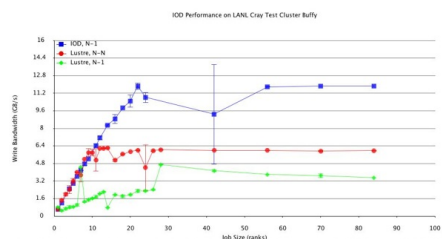
Jay Lofstead, Ivo Jimenez, Carlos Maltzahn, Quincey Koziol, John Bent, Eric Barton
glofst@sandia.gov, ivo@cs.ucsc.edu, carlosm@soe.ucsc.edu, koziol@hdfgroup.org, john.bent@emc.com, eric.barton@intel.com



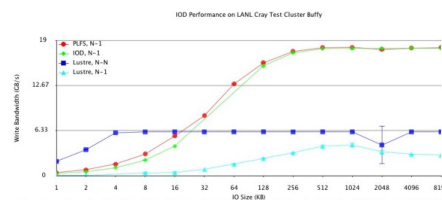
Example Architecture



Write Hosts



Write Size



Burst Buffers

Assuming SSDs

- Absorb high rate output to continue computation without blocking

More efficient data manipulation

- Reorder array dimensions without seek and rotational latency
- Perform data reduction/sorting with high density
- Preload large dataset for faster small reads

Multiple Layers

End-User API

- Mostly hides complexity of underlying layers

I/O Forwarding

- Aggregation
- Function Shipping

I/O Dispatcher (optional)

- Burst Buffers
- Transactions

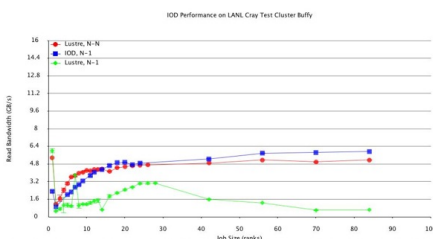
Distributed Asynchronous Object Storage (DAOS)

- Epochs
- Translates user view to storage view

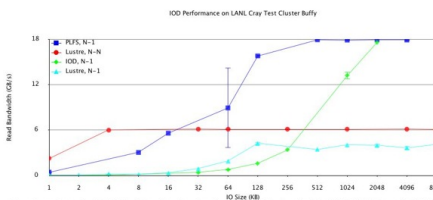
Versioning Object Storage Device (VOSD)

- Device interface

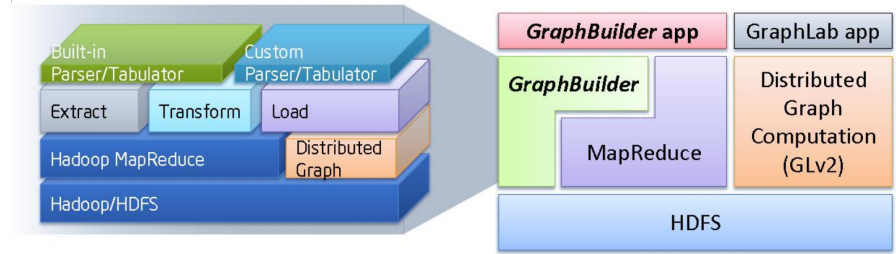
Read Hosts



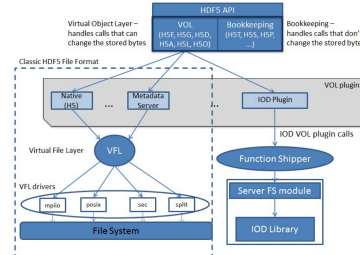
Read Size



Big Data Support



HDF5 VOL Architecture



Data Model

Container (like HDF file)

- Transaction/Epoch granularity
- Collection of related objects
- Also can hold global system state

Key-Value Store (KVS)

- Represent hierarchy level with nested KVSs
- Attributes
- Scalars
- Links to all objects in hierarchy level

Multi-Dimensional Array

- A slice (hyperslab) for an array (think one process worth)
- Opportunities for more intelligent handling

BLOB

- Everything else

Transactions vs. Epochs

"Same" thing at different layers

- IOD for transient data
- DAOS for long-term persistent data
- Copy-on-write semantics for compact storage
- "Flattening" to combine transactions into epochs
- Multiple open transactions at once