DAOS DEVELOPMENT UPDATE

Johann Lombardi, Principal Engineer, Intel

DAOS User Group 2018

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/benchmarks .

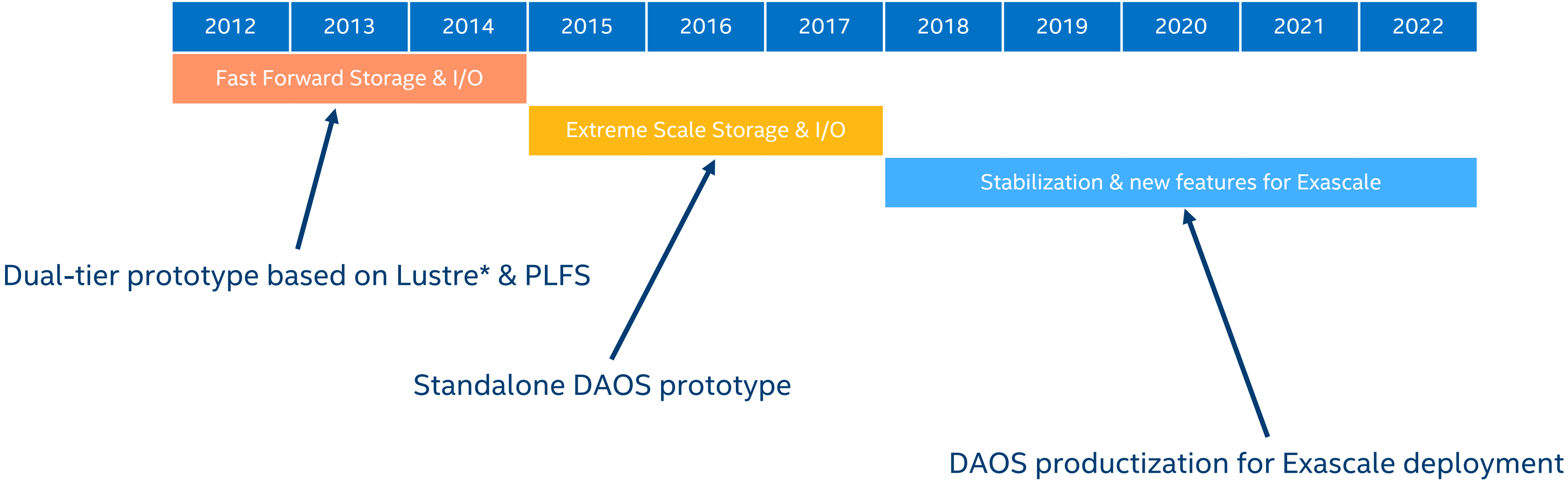
Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
Intel, the Intel logo, and Intel Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as property of others.

PROJECT HISTORY



2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022

Fast Forward Storage & I/O

Extreme Scale Storage & I/O

Stabilization & new features for Exascale

Dual-tier prototype based on Lustre* & PLFS

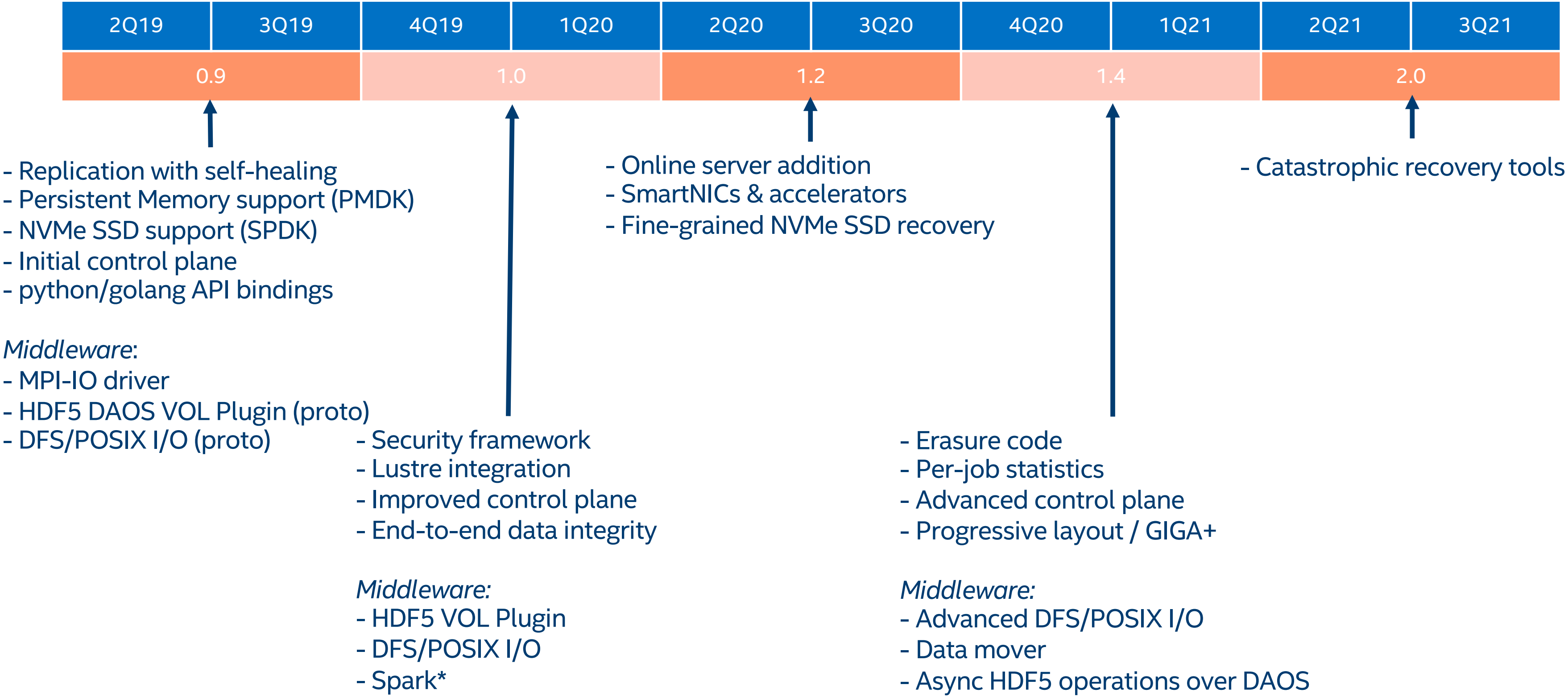
Standalone DAOS prototype

DAOS productization for Exascale deployment

*Other names and brands may be claimed as the property of others.

All information provided in this roadmap is subject to change without notice.

DAOS COMMUNITY ROADMAP – Q4 2018



2Q19	3Q19	4Q19	1Q20	2Q20	3Q20	4Q20	1Q21	2Q21	3Q21
0.9		1.0		1.2		1.4		2.0	

- Replication with self-healing
- Persistent Memory support (PMDK)
- NVMe SSD support (SPDK)
- Initial control plane
- python/golang API bindings

Middleware:
- MPI-IO driver
- HDF5 DAOS VOL Plugin (proto)
- DFS/POSIX I/O (proto)

- Security framework
- Lustre integration
- Improved control plane
- End-to-end data integrity

Middleware:
- HDF5 VOL Plugin
- DFS/POSIX I/O
- Spark*

- Online server addition
- SmartNICs & accelerators
- Fine-grained NVMe SSD recovery

- Erasure code
- Per-job statistics
- Advanced control plane
- Progressive layout / GIGA+

Middleware:
- Advanced DFS/POSIX I/O
- Data mover
- Async HDF5 operations over DAOS

- Catastrophic recovery tools

DAOS STABILIZATION EFFORT (Q1'18-Q1'19)

Increase test coverage & fix resulting bugs

- Unit test improvements & CI integration
- Developed fault injection framework
- Functional test development and integration with Avocado
- Additional semi-automated testing run over psm2
- More to come
  - Scale, performance & soak tests

Address technical debt

- Focuses on a few main areas: rebuild, metadata, VOS & trees.

Develop documentation

- DAOS internals (markdown format)
- DAOS administrative guide

FABRIC SUPPORT

Regular testing

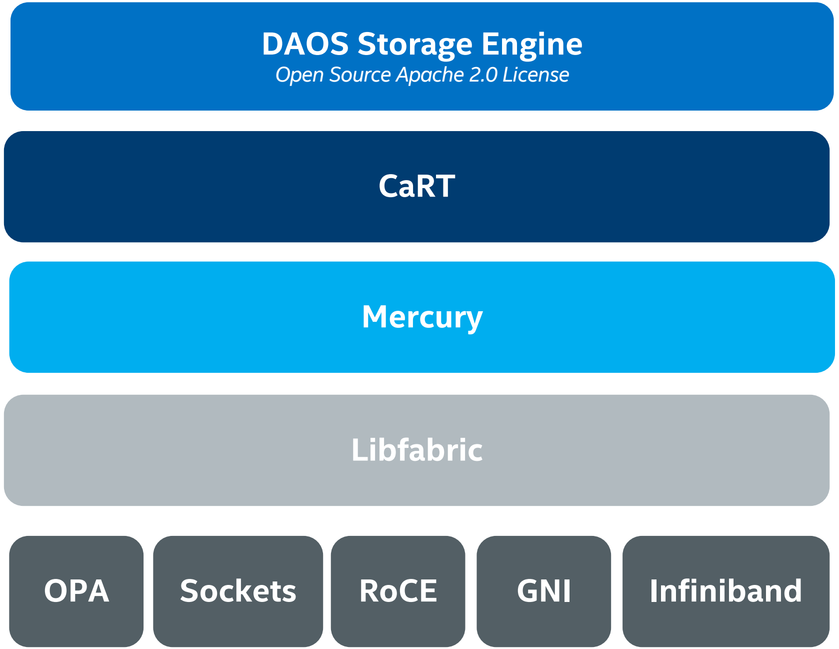
- OPA – PSM2 provider
- Ethernet & IPoFabric – Socket provider

Occasional testing

- GNI
- RoCE
- Infiniband – rxm/verbs provider

CaRT selftest

- Benchmark/validate fabric & comm layer
- Emulate DAOS traffic



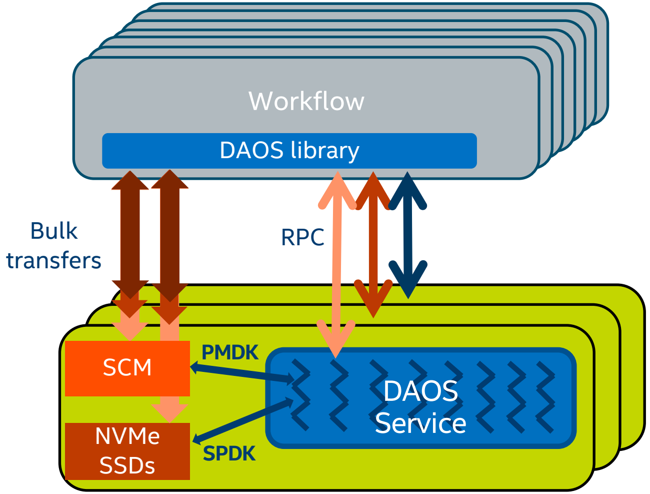
STORAGE BACKEND SUPPORT

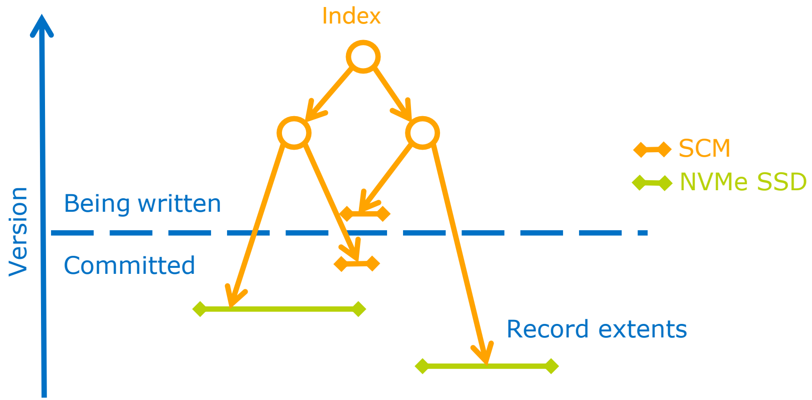
Storage-class memory

- Testing/performance tuning with Optane DC persistent memory
- Working closely with PMDK team
  - Extend PMDK with new reserve/publish API

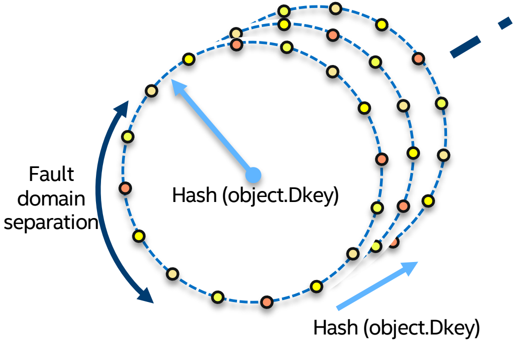
NVMe SSD

- SPDK support is finally there!
- Very basic allocation policies for now
  - All extents >= 4K on NVMe SSDs
- Next steps
  - Single SSD eviction & reintegration
  - Aggregation





DATA MANAGEMENT



Data Distribution

- Algorithmic placement
  - Exploring jump consistent hash
- Progressive layout with GIGA+

Data Protection

- Declustered replication & erasure code
- Fault-domain aware placement
- Self-healing
- End-to-end data integrity

Data Versioning

- Non-destructive write & consistent read
- Native snapshot support

Data Security & Reduction

- Online real-time data encryption & compression (not POR)

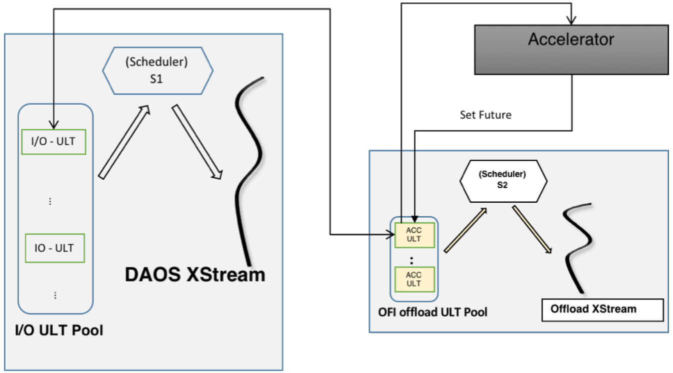
STORAGE ACCELERATION FRAMEWORK

Investigating offload API for client and server

- ISA-L (software) on IA
- Accelerators (hardware)
  - Intel QuickAssist
  - GPGPU
  - SmartNICs (libfabric extensions)
  - …

Possible use cases

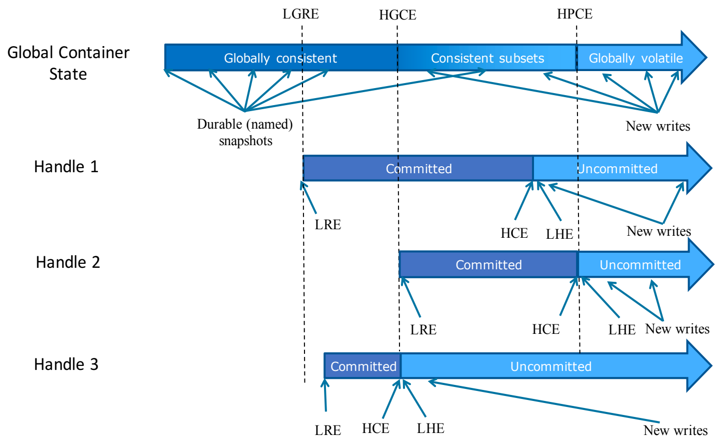
- Erasure code
- Checksums
- Compression
- Encryption
- …



TRANSACTION MODEL EVOLUTION

Retiring original epoch model

- Too complex & coarse grain
- Difficult to implement new features like erasure code



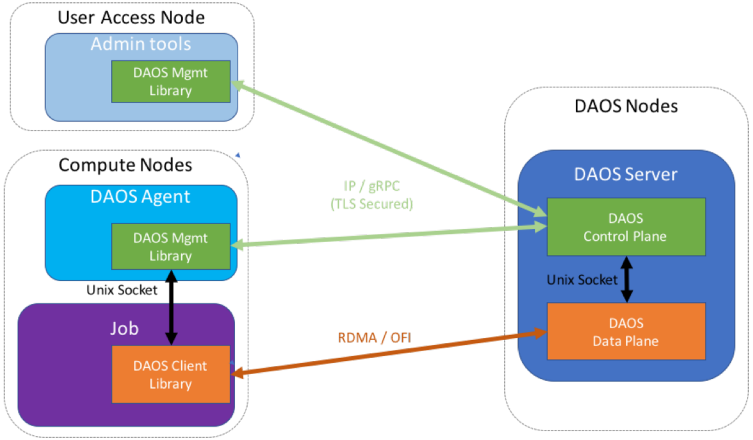
New transaction model

- Used internally to guarantee replication & erasure code consistency
- Transaction exported through the API
  - Used for I/O middleware consistency
  - e.g. POSIX rename, SQL operation, …
- On-demand concurrency control
  - Optimistic conflict detection & resolution
  - Lockless / no serialization
  - Widely used in databases since the 80’s
- Still provide instantaneous global snapshot & time travel

SECURITY

Flexible security framework

- Support different authentication methods
  - Local agent on compute node authenticating process through AUTH_SYS
  - Third party authentication service (e.g. munge)
- TLS-secured channel using certificates
- Very minimal impact expected on I/O path



CONTROL PLANE

Storage provisioning

- Detect SCM & NVMe storage
  - CPU/storage affinity
- Configure/format/mount SCM
  - Interleaved mode
- Configure NVMe SSDs
  - Firmware update
- Integrated storage burn-in capability

Fabric configuration

- Comm layer configuration
- Interface/CPU affinity

DAOS configuration

- zero-conf/auto-conf with device filters/manual-conf
- YAML configuration for admins

DAOS service management

- Manage/monitor/troubleshoot
- Integration with systemd & other frameworks

Telemetry

- Storage/service/fabric activity
- Per-job statistics

Storage API & tools

- CLI tools built over the control plane API

I/O MIDDLEWARE

MPI-IO

- Prototyped ROMIO Driver
  - Not supporting shared file pointer operations
  - Not supporting MPI_File_preallocate()
- Driver successfully tested with:
  - ROMIO & LLNL test suite
  - IOR
  - MACSIO
- Next steps
  - Code improvements & hardening
  - More testing & benchmarking

POSIX I/O

- DFS (DAOS File System) library
  - Basic functionality working
  - No cross-client concurrency control yet
- Application interface
  - DFS backend for MDTest & IOR available
  - FUSE driver available
  - Interception library (through I/O Forwarding)
- Next steps
  - Concurrency control

HDF5

- See next presentation from Elena

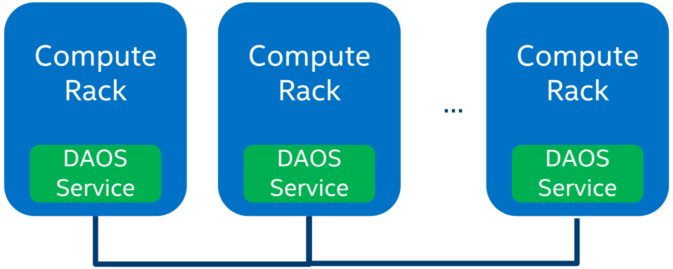
DAOS DEPLOYMENT OPTIONS

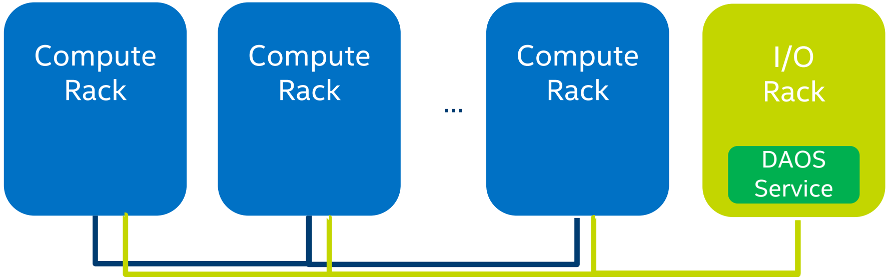
Disaggregated Storage

- Storage integrated in compute rack
- Highly distributed storage
- Non-uniform storage access

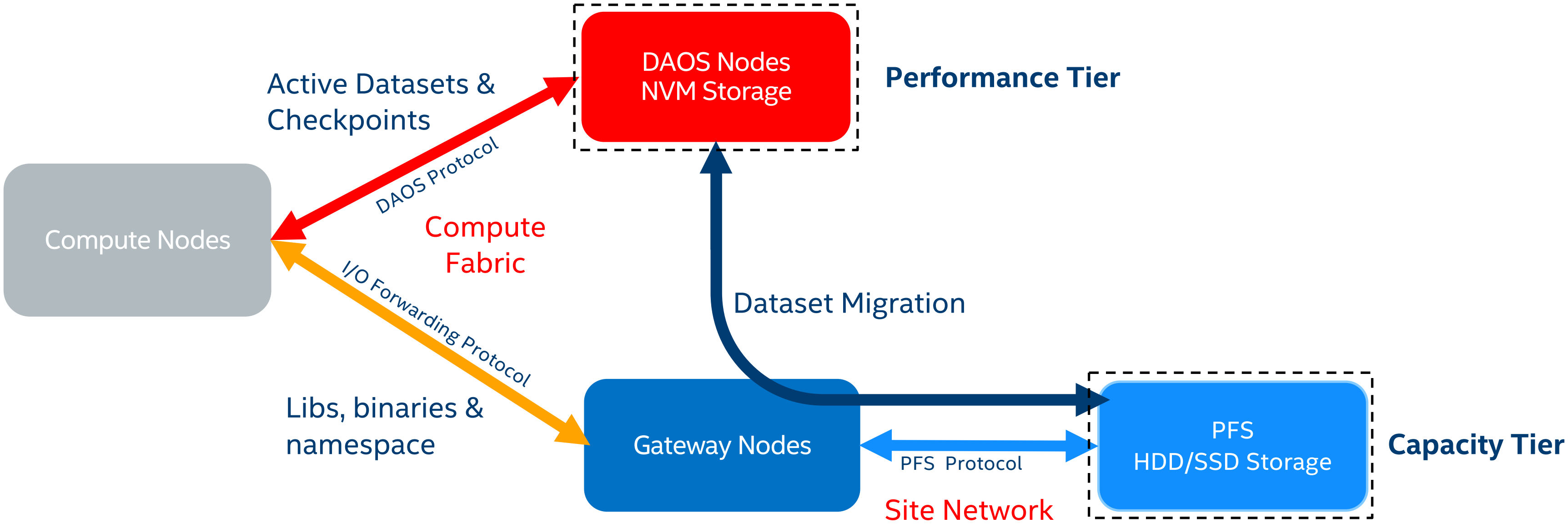
Pooled Storage

- Storage in separate racks
- High density storage servers
- Uniform storage access





INTEL EXASCALE STORAGE ARCHITECTURE



DAOS Nodes
NVM Storage

Performance Tier

Active Datasets &
Checkpoints

DAOS Protocol

Compute Nodes

Compute
Fabric

I/O Forwarding Protocol

Dataset Migration

Libs, binaries &
namespace

Gateway Nodes

PFS Protocol

Site Network

PFS
HDD/SSD Storage

Capacity Tier

TESTING DAOS

Storage requirements

- SCM/NVMe recommended ratio
  - 6% minimum to store internal metadata
- Emulating persistent memory
  - DRAM with tmpfs
- Emulating NVMe SSD
  - SPDK malloc device
  - SPDK AIO bdev

Fabric requirements

- RDMA-capable fabric prefered
  - OPA, Infiniband, GNI, RoCE, …
- TCP/IP
- Shared memory

Supported distributions

- CentOS7.4 and above
- openSuSE 42.2
- Ubuntu 18.04
- Docker files for CentOS & Ubuntu

RESOURCES

Source code on GitHub

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

Community mailing list on Groups.io

- daos@daos.groups.io or https://daos.groups.io/g/daos

Wiki

- http://daos.io or https://wiki.hpdd.intel.com

Bug tracker

- https://jira.hpdd.intel.com

intel®