Bringing AI Everywhere in HPC

High Performance Fabric Support in DAOS

Michael Hennecke

Principal Engineer, HPC Storage

ISC High Performance 2024

Legal Notices and Disclaimers

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

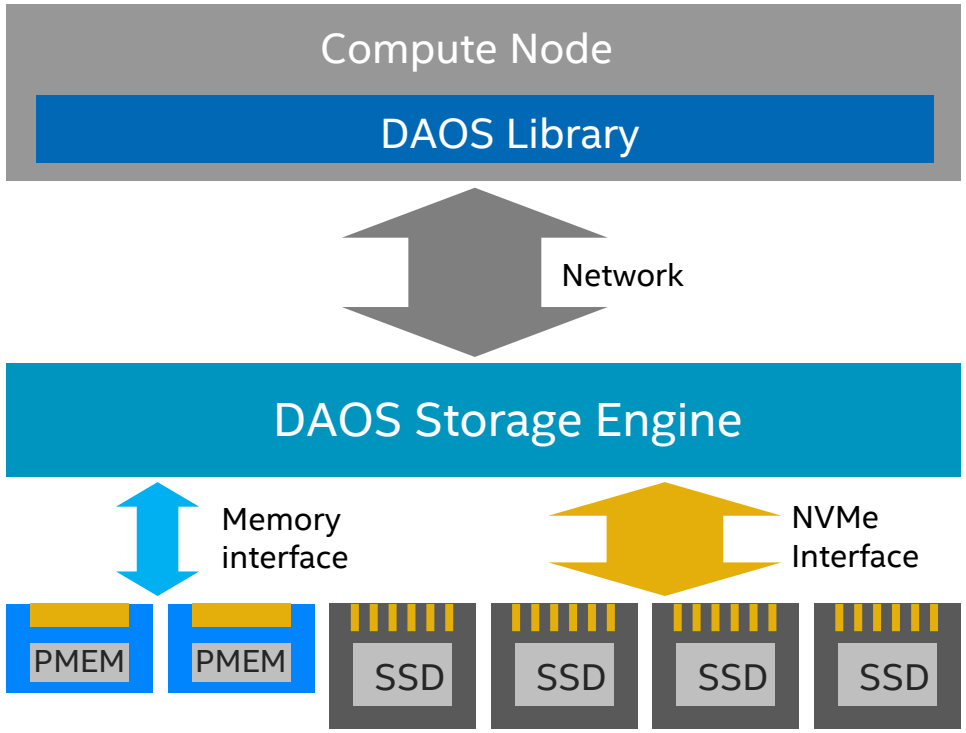
Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

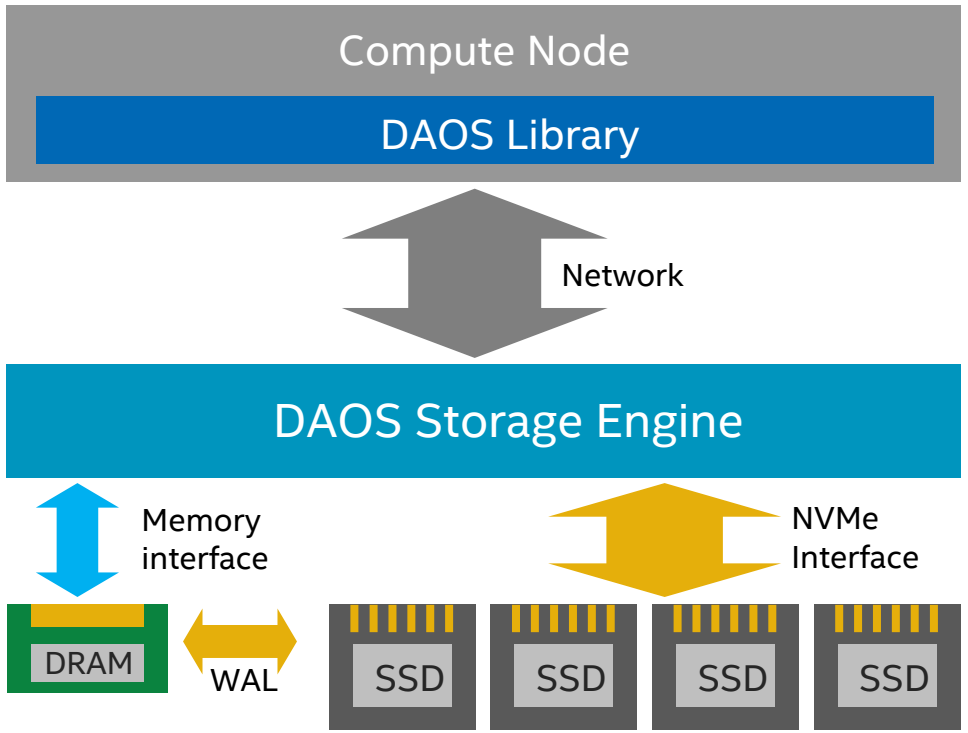
Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

DAOS Architecture (PMem → non-PMem)

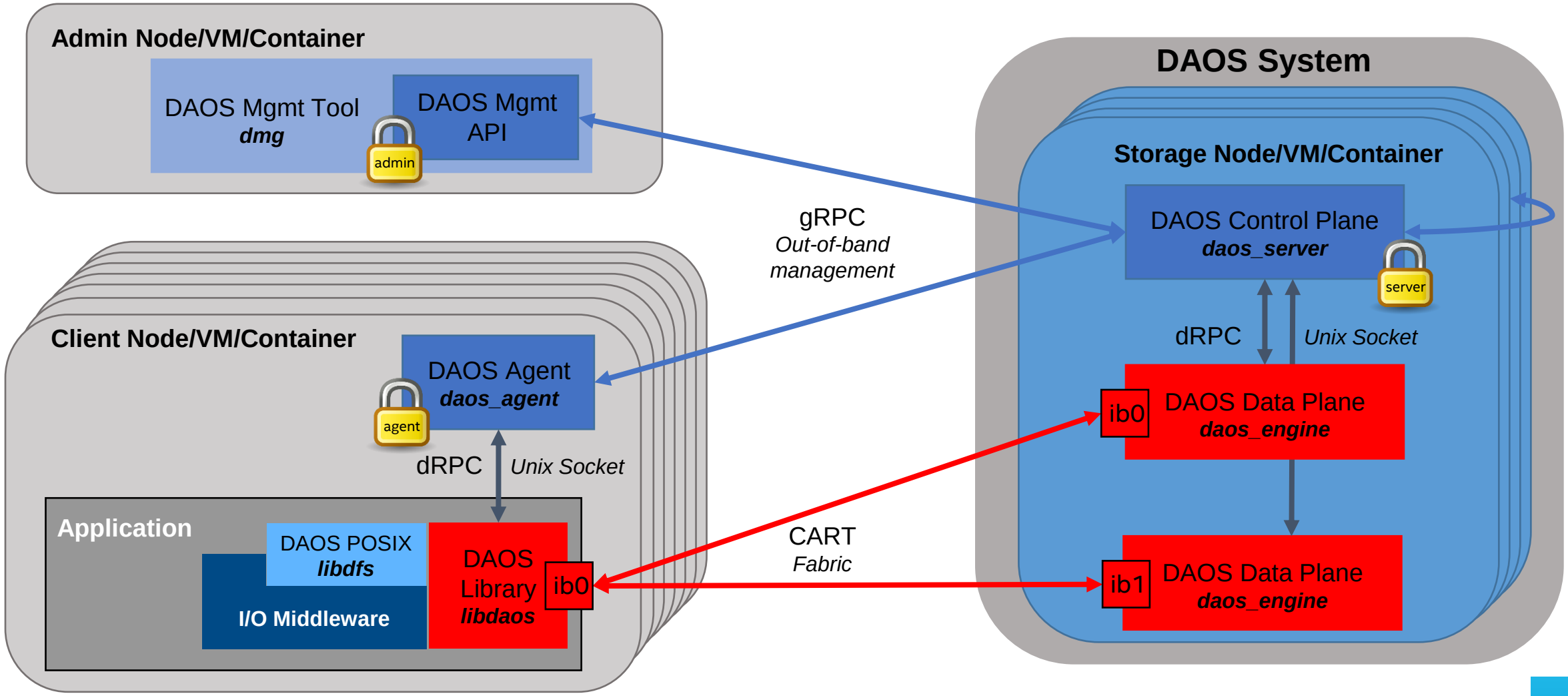


With Persistent Memory



Without Persistent Memory

DAOS Software Components



Networking in the DAOS Configuration Files

daos_server.yml

```
# To operate, DAOS will need a quorum
# of access point nodes to be available.

access_points:

- daos[01-03]


provider: ofi+verbs;ofi_rxm


engines:

-

  fabric_iface: ib0

  fabric_iface_port: 20000

-

  fabric_iface: ib1

  fabric_iface_port: 21000
```

daos_agent.yml

```
# Management service access points
# Must have the same value for all
# agents and all servers in a system.

access_points:

- daos[01-03]
```

daos_control.yml

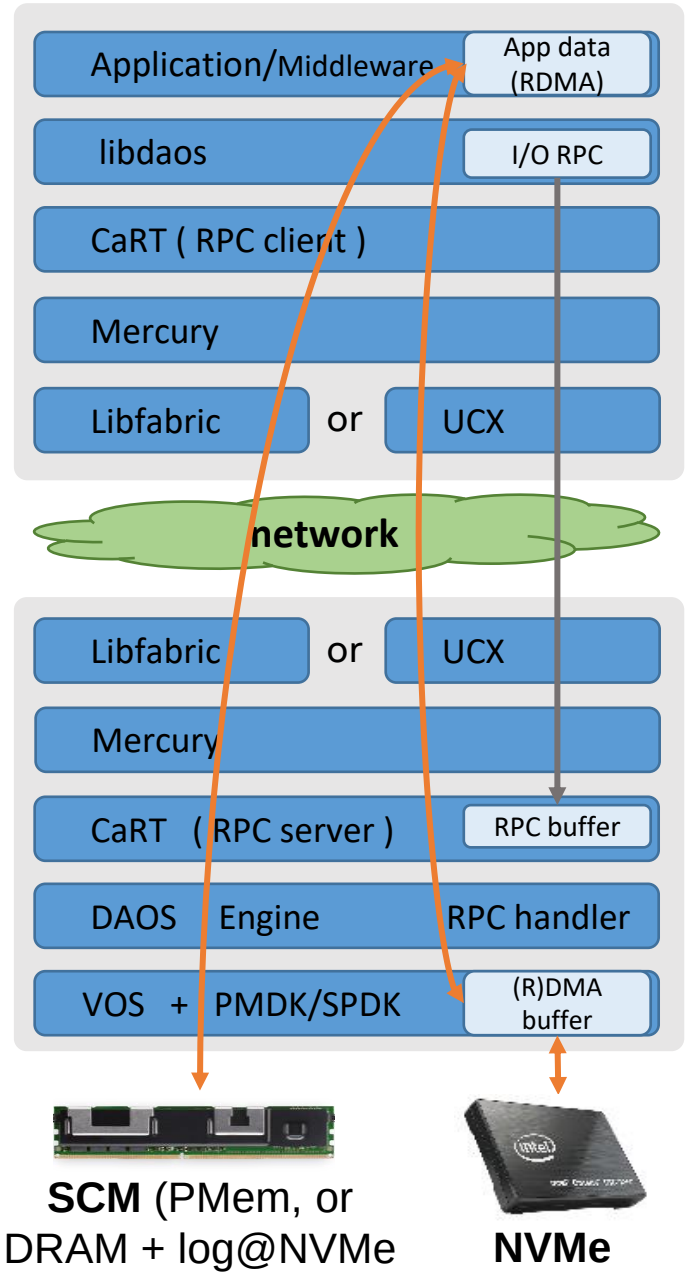
```
# Hostlist,
# a comma separated list of addresses
# (hostnames or IPv4 addresses).

hostlist:

- daos[01-42]
```

DAOS I/O Flow

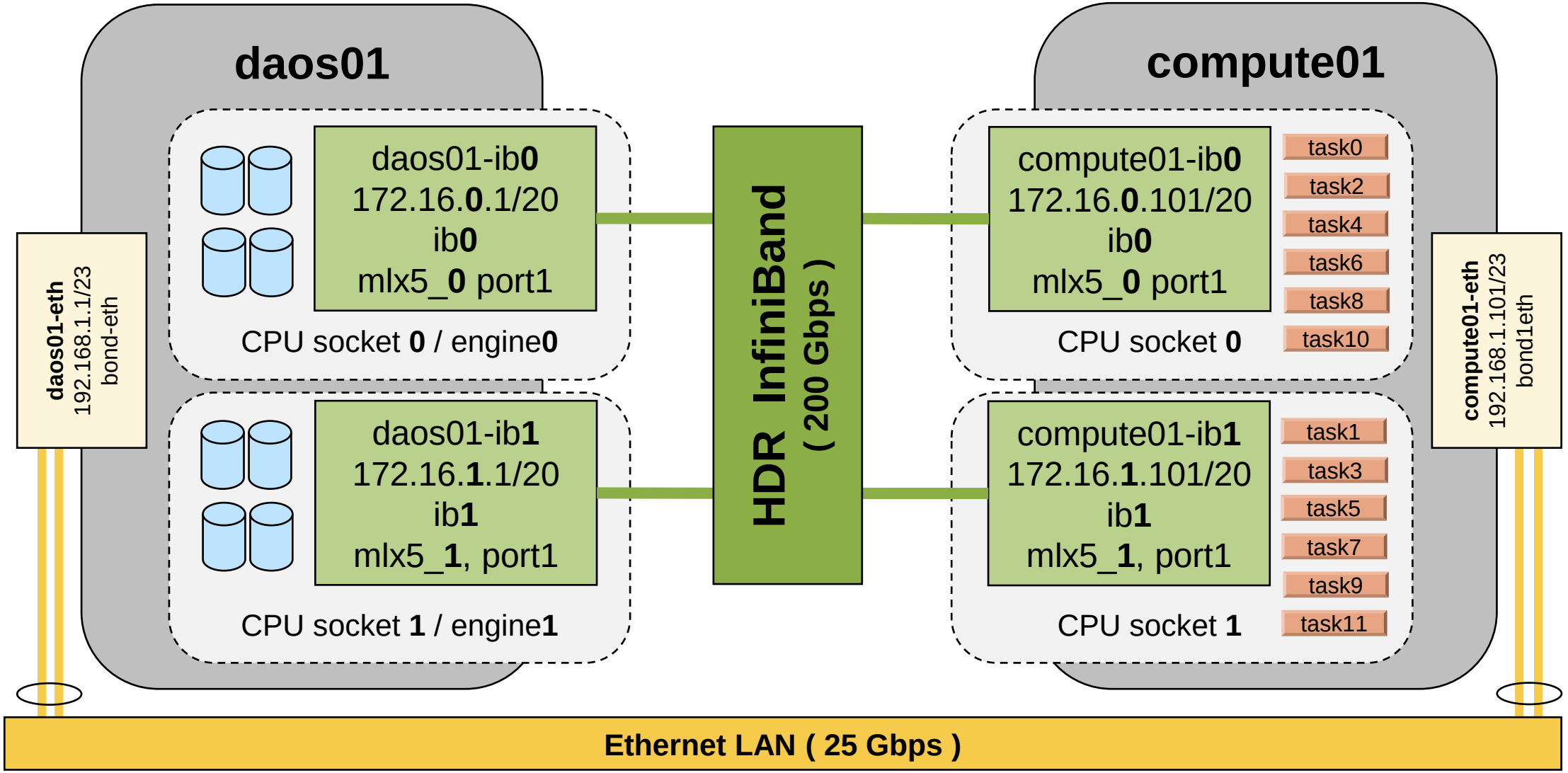
- OFI **Libfabric** – point-to-point messaging
  - Thin send/receive messaging layer, no RPCs
  - Back-end providers: tcp; verbs; cxi [; opx]
- Or **UCX** on InfiniBand (ucx+ud_x for best scaling)
- **Mercury** – p2p RPC layer over libfabric or UCX
- DAOS **C**ollective **a**nd **R**PC **T**ransport (**CaRT**)
  - Collective operations (e.g. broadcast)
  - Timeouts for RPCs; Detection of server failures
  - Flow control (managing the number of inflight RPCs)



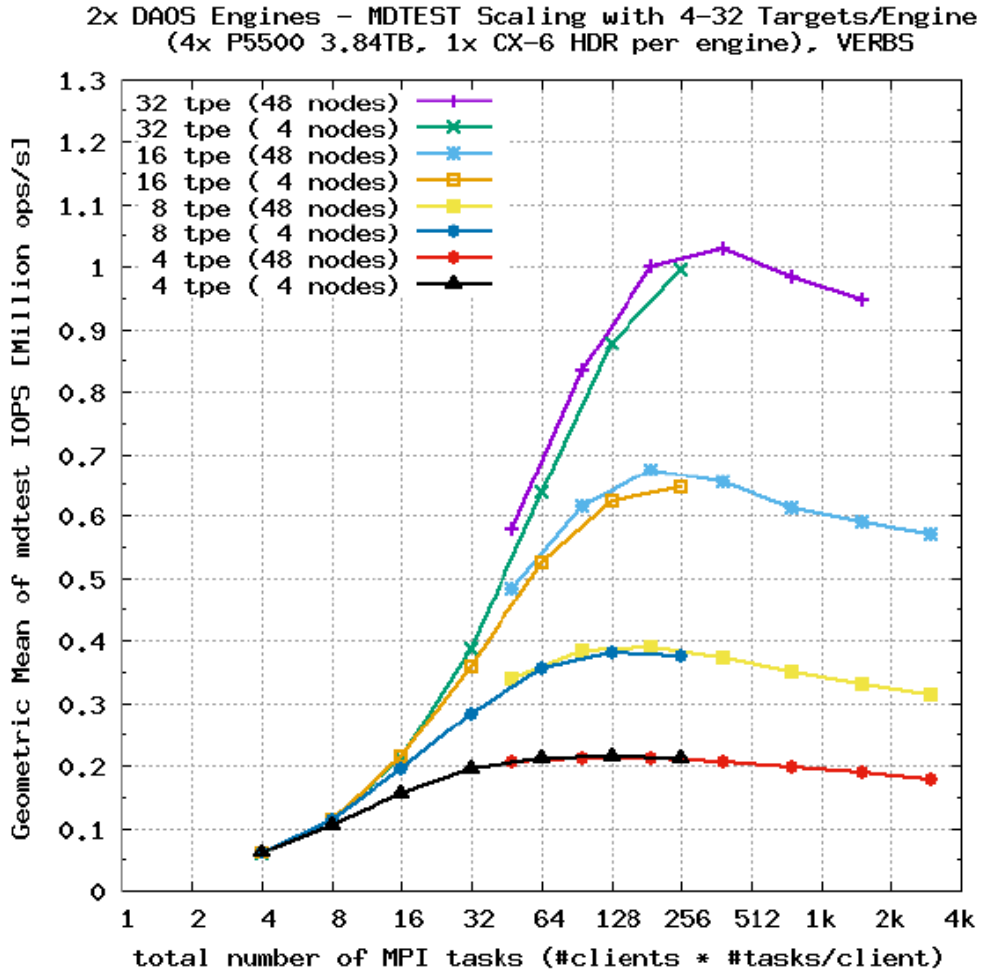
Supported High Performance Fabrics

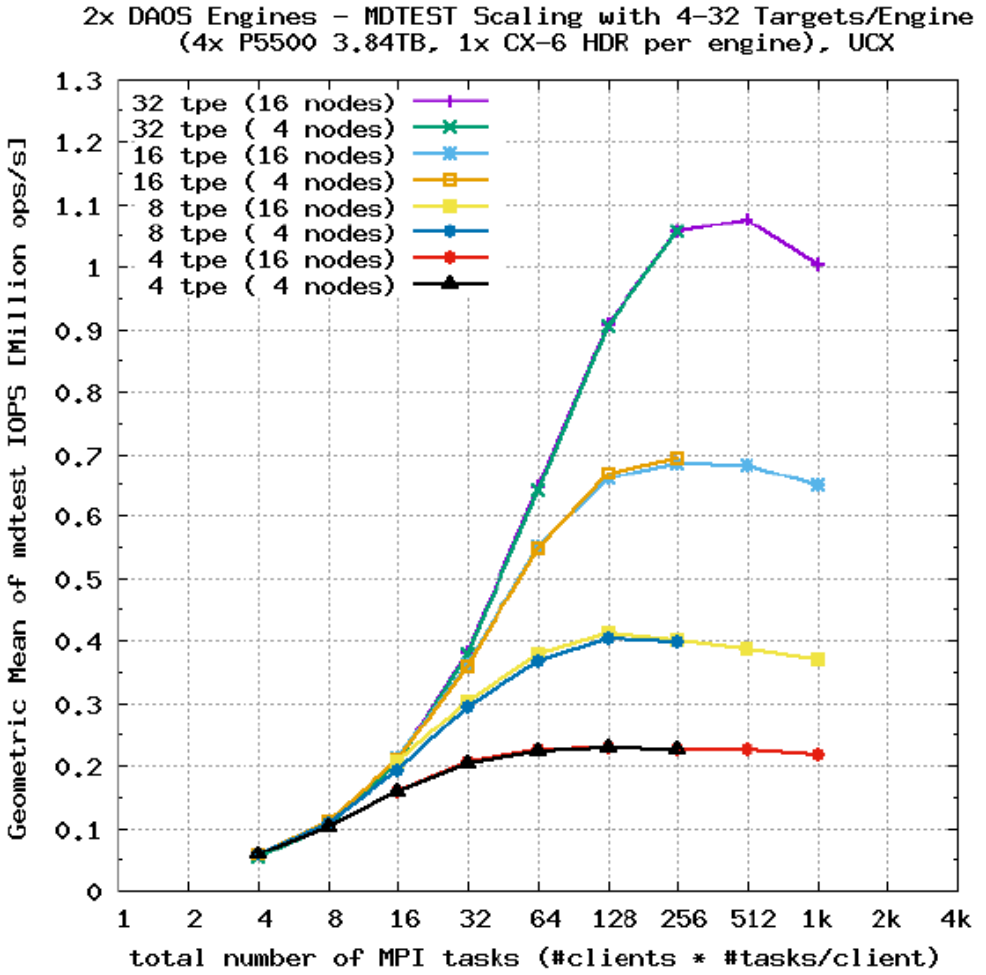
- Ethernet
  `provider: ofi+tcp [;ofi_rxm]`
- Ethernet with RoCE v2; “small” InfiniBand fabrics (see below)
  `provider: ofi+verbs;ofi_rxm`
- “Large” InfiniBand fabrics
  `provider: ucx+ud_x`
- Slingshot fabrics
  `provider: ofi+cxi`
  (using HPE-provided version of libfabric with cxi=Slingshot support)
- Omni-Path fabrics
  `provider: ofi+tcp [;ofi_rxm]`
  (Cornelis is working on DAOS support @ ofi+opx provider; not upstreamed yet)

Multiple Fabric Ports in DAOS Servers and Clients



Metadata Performance and Number of Storage Targets





mdtest scaling with #targets: ofi+verbs (left) and ucx+dc_x (right). From https://doi.org/10.1145/3581576.3581577

intel

How many InfiniBand Queue Pairs does RC need?

- On a DAOS client, each MPI task needs to connect to all storage targets on all daos_engine instances:
  - 56 cores/client-port * (42 servers * 2 engines/server * 4 targets/engine) = 18 816 QPs
- On a DAOS server, each storage target must connect to all MPI tasks on all clients:
  - 4 targets/engine * (240 clients * 2 sockets/client * 56 cores/socket) = 107 520 QPs
  - All targets also connect to all other targets: 4 tgt * (42 srv * 2 eng/srv * 4 tgt) = 1344 QPs
- But NVIDIA ConnectX-6 adapters support only 128k QPs:

```
i20r01c06s10:~ # ibv_devinfo -v | egrep "^hca|max_qp:“
hca_id: mlx5_0
        max_qp:                                 131072
hca_id: mlx5_1
        max_qp:                                 131072
```

➔ For “large” InfiniBand fabrics, it’s better to use UCX **with** UD (not RC or DC)…

How many InfiniBand Queue Pairs does RC need?

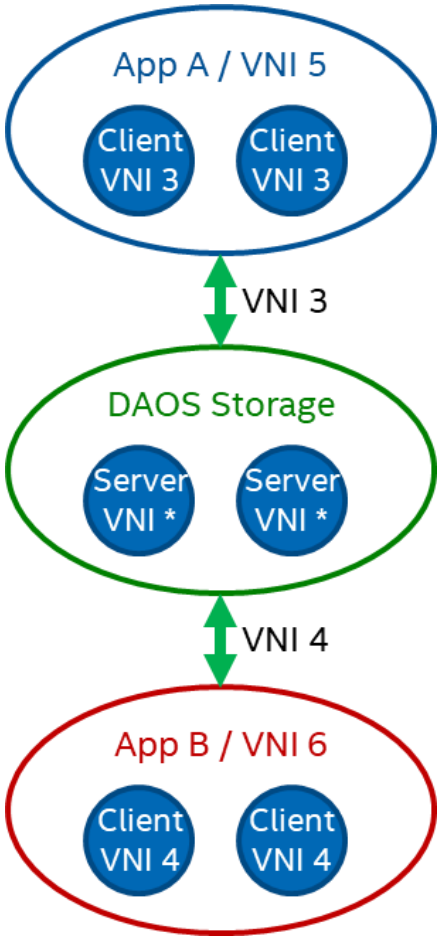
- On a DAOS client, each MPI task needs to connect to all storage targets on all daos_engine instances:
  - 56 cores/client-port * (42 servers * 2 engines/server * 24 targets/engine) = 112 896 QPs
- On a DAOS server, each storage target must connect to all MPI tasks on all clients:
  - 24 targets/engine * (240 clients * 2 sockets/client * 56 cores/socket) = 645 110 QPs
  - All targets also connect to all other targets: 24 tgt * (42 srv * 2 eng/srv * 24 tgt) = 48 384 QPs
- But NVIDIA ConnectX-6 adapters support only 128k QPs:

```
i20r01c06s10:~ # ibv_devinfo -v | egrep "^hca|max_qp:“
hca_id: mlx5_0
        max_qp:                                 131072
hca_id: mlx5_1
        max_qp:                                 131072
```

➔ For “large” InfiniBand fabrics, it’s better to use UCX with UD (not RC or DC)…

Slingshot Security Framework – VNI for Client/Server

- Slingshot uses Virtual Network Identifiers (VNIs) for security isolation (like VLAN in Ethernet, P-Key in IB)
- Each job runs with its own VNI
  - e.g. for MPI messaging
- All jobs must also communicate with the DAOS servers
  - sharing "server VNIs" across jobs would compromise the job isolation
  - solution: make DAOS servers "promiscuous" for a range of VNIs, and assign a separate VNI from this range to each job
- Needed libfabric extension for authorization keyrings
  - original libfabric only supports one auth key/endpoint



For more information on the DAOS Foundation and the DAOS Project, please visit https://daos.io/

intel

intel.