STORAGE DEVELOPER CONFERENCE
SDC 23
BY Developers FOR Developers

Breaking Boundaries: Expanding Ceph's Capabilities with NVMe-oF

Orit Wasserman
Distinguished Engineer
ODF Lead Architect
Ceph NVMe-of Architect
IBM

Kyle Bader
STSM
Principal Portfolio Architect - Ceph offerings
IBM

What Is Ceph?

The buzzwords

- "Software defined storage"
- "Unified storage system"
- "Scalable distributed storage"
- "The future of storage"
- "The Linux of storage"

The substance

- Ceph is open source **software**
- Runs on commodity hardware
  - Commodity servers
  - IP networks
  - HDDs, SSDs, NVMe, NV-DIMMs, ...
- A single cluster can serve **object** , **block** , and **file** workloads

Ceph is Free and Open Source

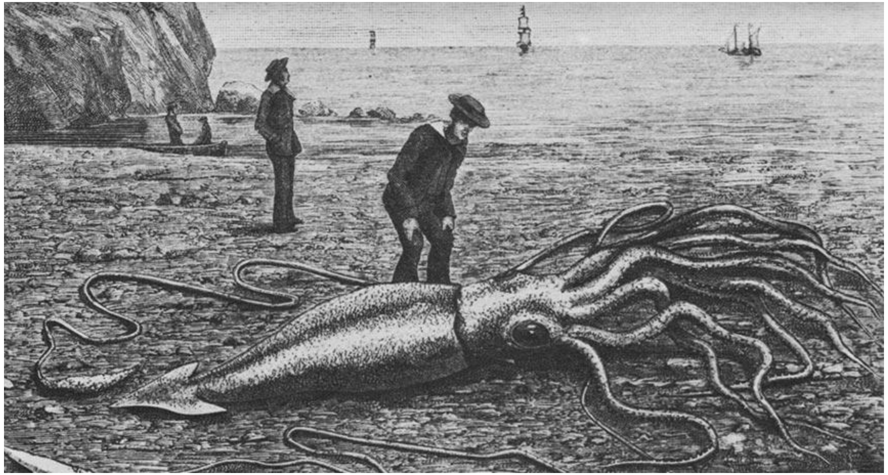
- Freedom to use (free as in beer)
- Freedom to introspect, modify, and share (free as in speech)
- Freedom from vendor lock-in
- Freedom to innovate

Cep is Reliable

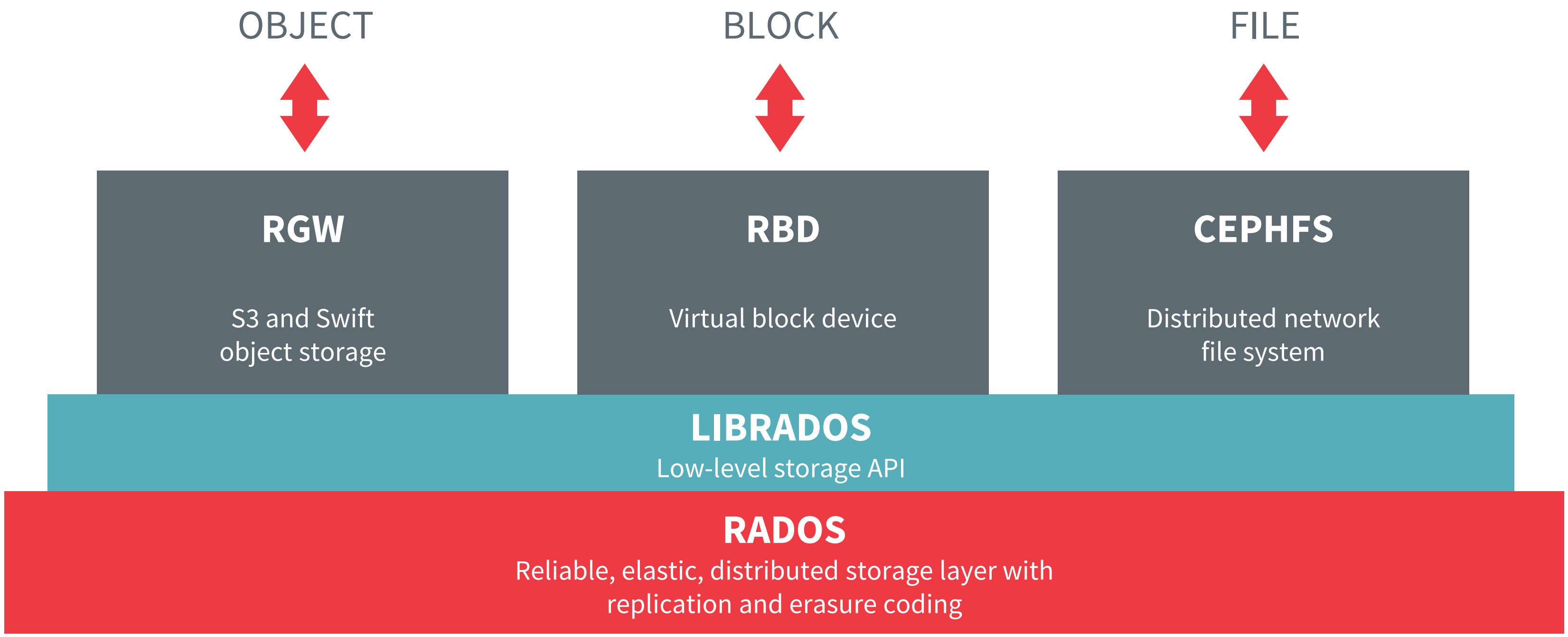
- **Reliable storage** service out of **unreliable components**
  - No single point of failure
  - Data durability via replication or erasure coding
  - No interruption of service from rolling upgrades, online expansion, etc.
- Favor consistency and correctness over performance

Ceph is Scalable

- Ceph is elastic storage infrastructure
  - Storage cluster may grow or shrink
  - Add or remove hardware while system is online and under load
- Scale **up** with bigger, faster hardware
- Scale **out** within a single cluster for capacity and performance
- **Federate** multiple clusters across sites with asynchronous replication and disaster recovery capabilities



Ceph is a Unified Storage System



OBJECT

BLOCK

FILE

RGW

S3 and Swift object storage

RBD

Virtual block device

CEPHFS

Distributed network file system

LIBRADOS

Low-level storage API

RADOS

Reliable, elastic, distributed storage layer with replication and erasure coding

RADOS

RADOS

- **R**eliable **A**utonomic **D**istributed **O**bject **S**torage
  - Common storage layer underpinning object, block, and file services
- Provides low-level data object storage service
  - Reliable and highly available
  - Scalable (on day 1 and day 1000)
  - Manages all replication and/or erasure coding, data placement, rebalancing, repair, etc.
- Strong consistency
  - CP, not AP
- Simplifies design and implementation of higher layers (file, block, object)

LIBRADOS API

- **Efficient key/value storage inside an object (OMAP)**
- Atomic single-object transactions
  - update data, attr, keys together
  - atomic compare-and-swap
- Object-granularity snapshot infrastructure
- Partial overwrite of existing data
- Single-object compound atomic operations
- RADOS classes (stored procedures)
- **Watch/Notify on an object**

RADOS Components

ceph-mon

Monitor

- Central authority for authentication, data placement, policy
- Coordination point for all other cluster components
- Protect critical cluster state with Paxos
- 3, 5, 7 per cluster

ceph-mgr

Manager

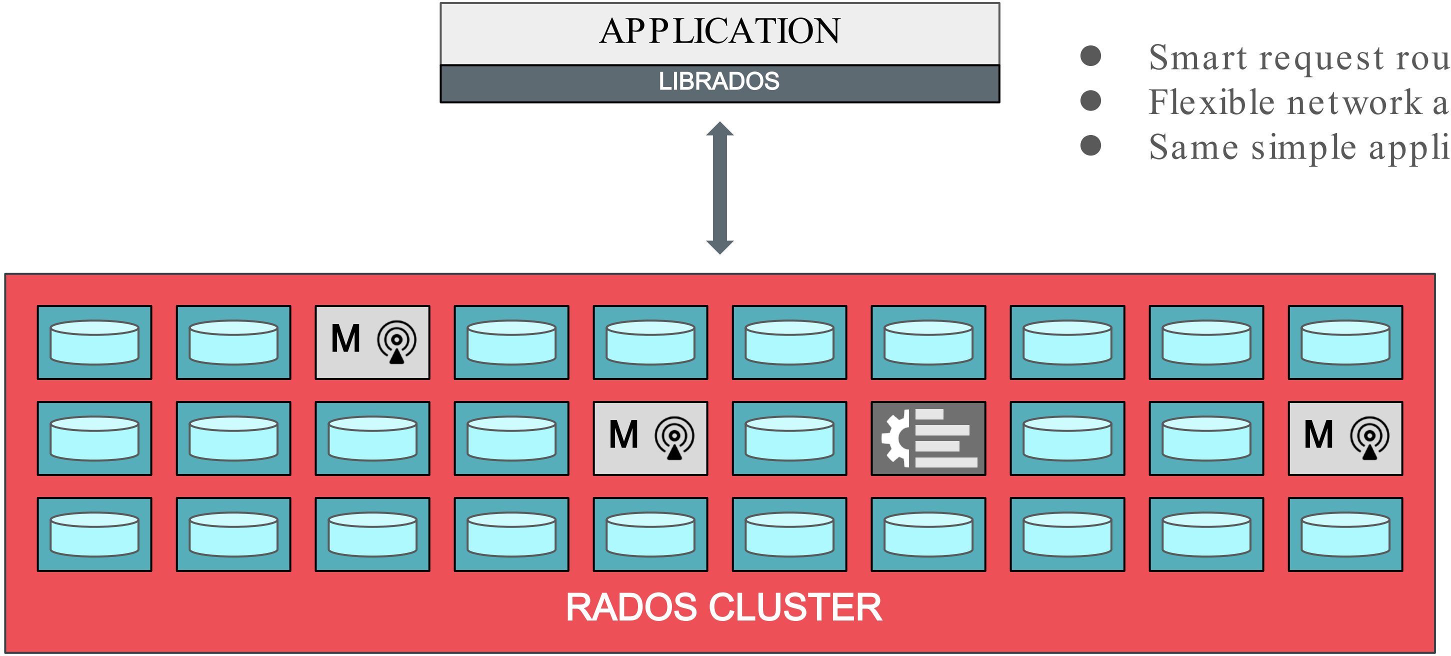
- Aggregates real-time metrics (throughput, disk usage, etc.)
- Host for pluggable management functions
- 1 active, 1+ standby per cluster

ceph-osd

OSD (Object Storage Daemon)

- Stores data on an HDD or SSD
- Services client IO requests
- Cooperatively peers, replicates, rebalances data
- 10s-1000s per cluster

Client/ Cluster Architecture

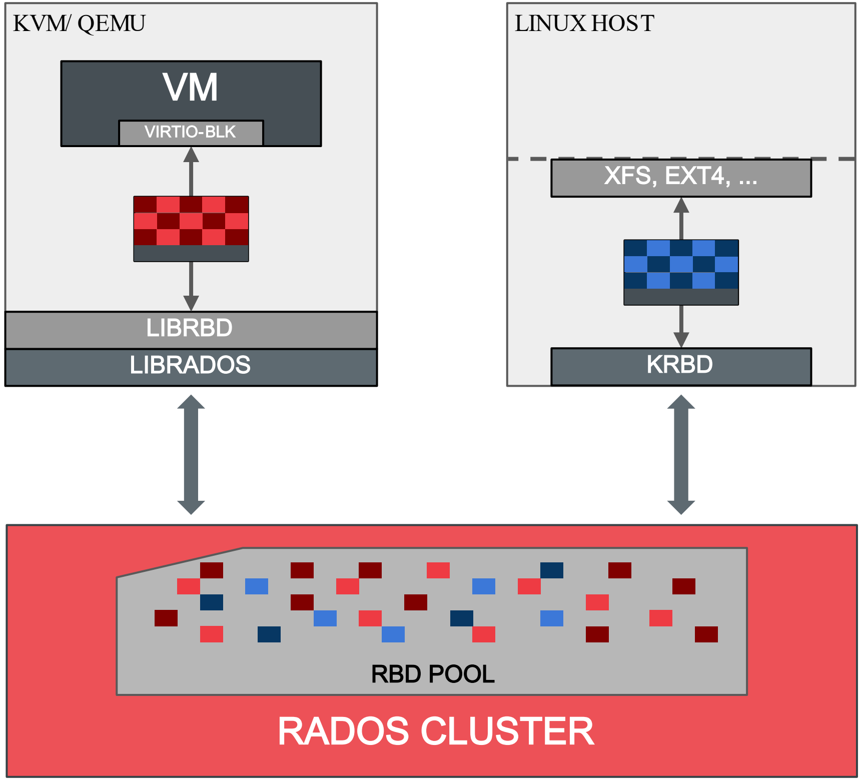


- Smart request routing
- Flexible network addressing
- Same simple application API

RBD: BLOCK STORAGE

RBD: RADOS Block Device

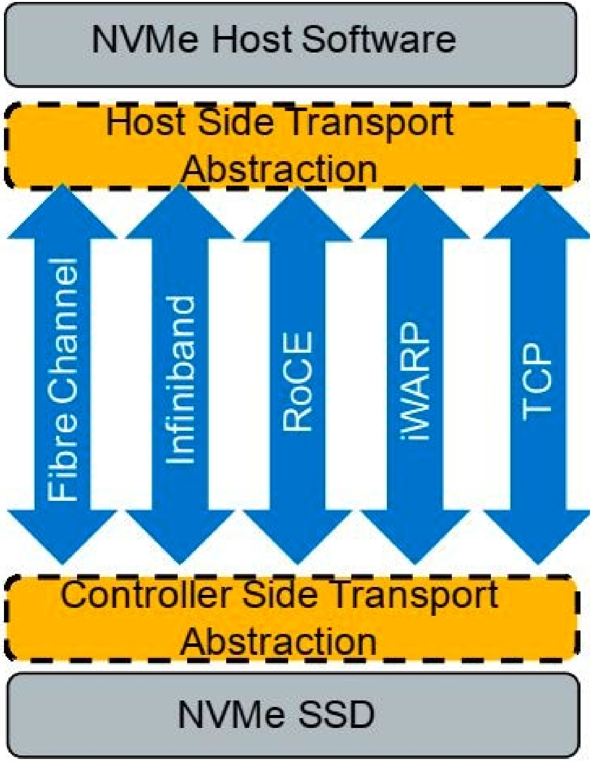
- Virtual block device
  - Store disk images in RADOS
  - Stripe data across many objects in a pool
- Storage decoupled from host, hypervisor
  - Analogous to AWS EBS
- Client implemented in KVM and Linux
- Integrated with
  - Libvirt
  - OpenStack (Cinder, Nova, Glace)
  - Kubernetes
  - Proxmox, CloudStack, Nebula, …
- Ceph iSCSI gateway
  - LIO stack + userspace tools to manage gateway configuration



NVMe Over Fabric

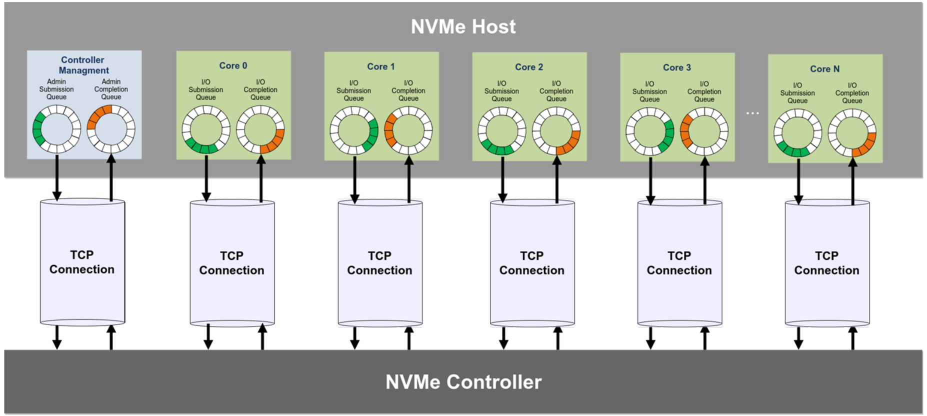
NVMe Over Fabric

- Non Volatile Memory Express (NVMe)
  - Fast PCIe attached storage
  - Local storage
- Expand NVMe efficiency and performance over network fabrics
- Eliminate unnecessary protocol translations
- Enable low-latency and high IOPS remote NVMe storage
- TCP:
  - Well-known and common transport
  - No networking infrastructure requirements and constraints
  - Ratified Nov, 2018


NVMe Host Software
Host Side Transport Abstraction
Fibre Channel
Infiniband
RoCE
iWARP
TCP
Controller Side Transport Abstraction
NVMe SSD

Association Model

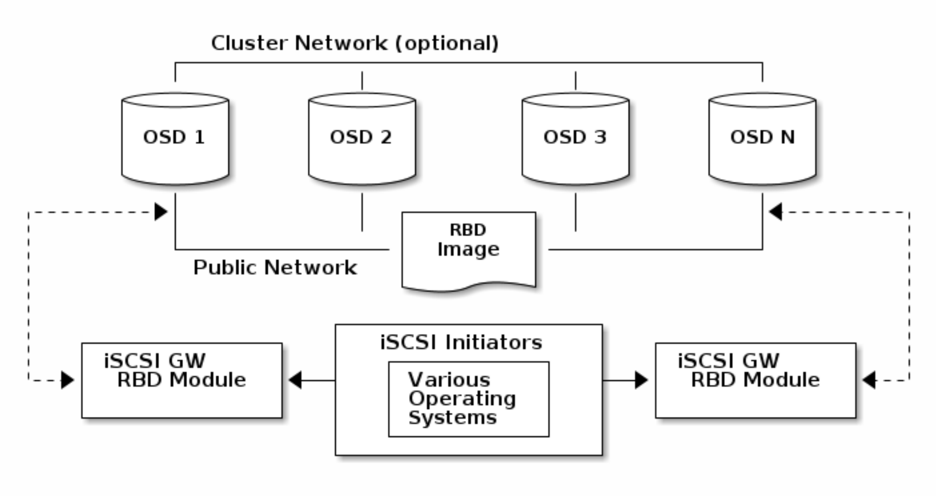
- Controller association maps 1x1 NVMe queue to a TCP connection
- No controller-wide sequencing
- No controller-wide reassembly constraints
- No shared state across NVMe queues and TCP



Connection binding is performed in NVMe-oF connect time (binding queue to controller)

Why not Ceph iSCSI?

- Legacy
- Performance
  - Higher throughput and IOPS
    - 30-70%
  - Reduced latency
    - 30-40%
- Reduced CPU usage
    - 30-40%
- Scalability



Why NVMe-over-Fabrics?

RADOS Block Device (RBD)

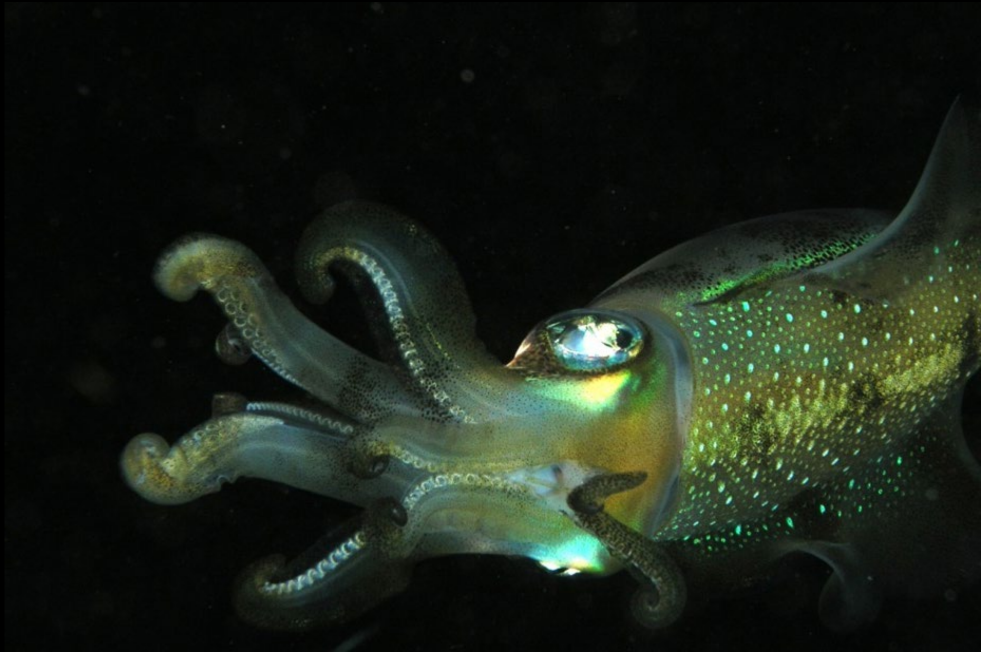
- RADOS protocol
- Distributed n-to-m protocol
- Reliable object access to sharded and replicated/erasure coded storage

Why do we need another protocol to access block storage in Ceph?

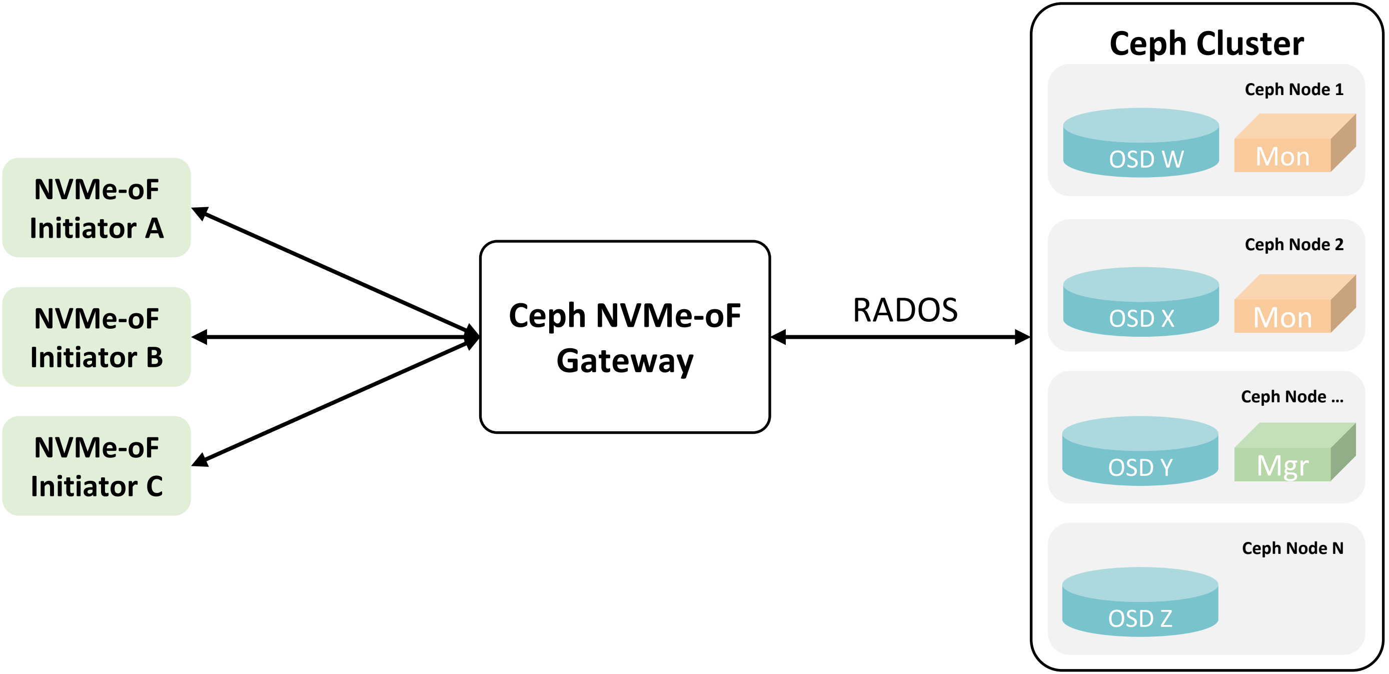
NVMe-over-Fabrics (NVMe-oF)

- Open, widely adopted ***industry standard***
- Enable use-cases where NVMe-oF is already part of ***ecosystem***
- Take advantage of NVMe-oF ***offloading*** in DPUs

Ceph NVMe-of Gateway



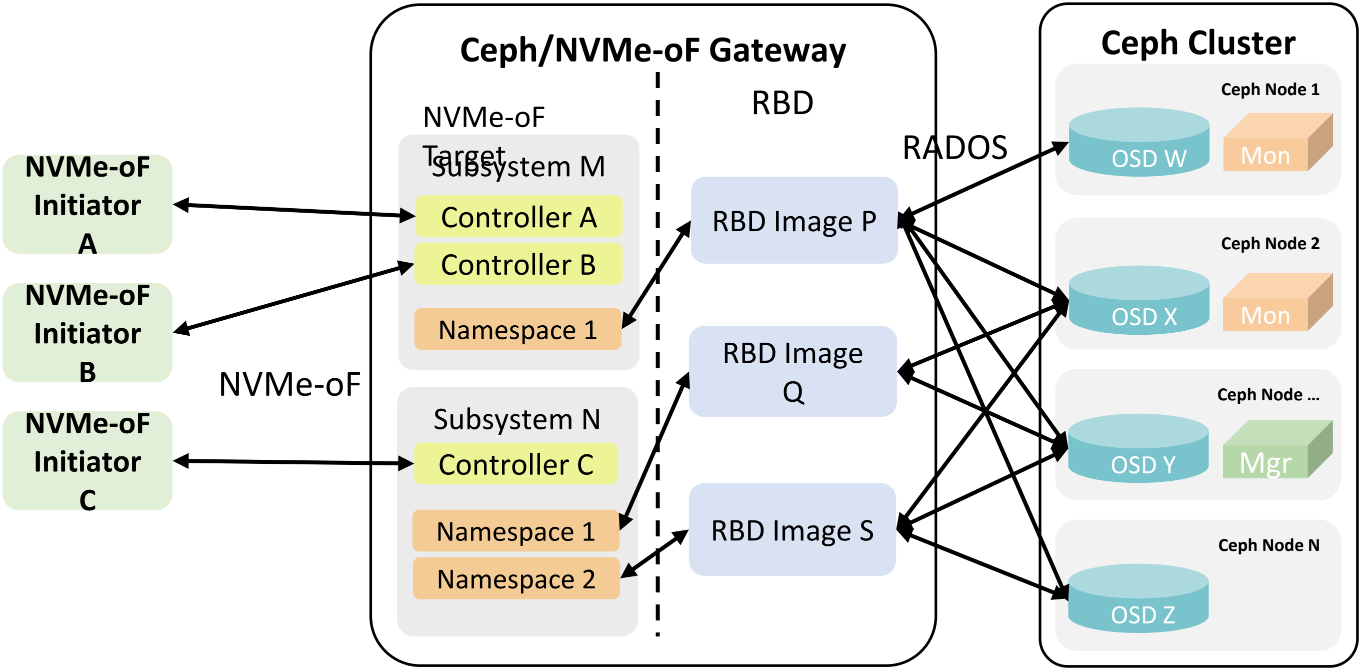
Overview



Fabrics:

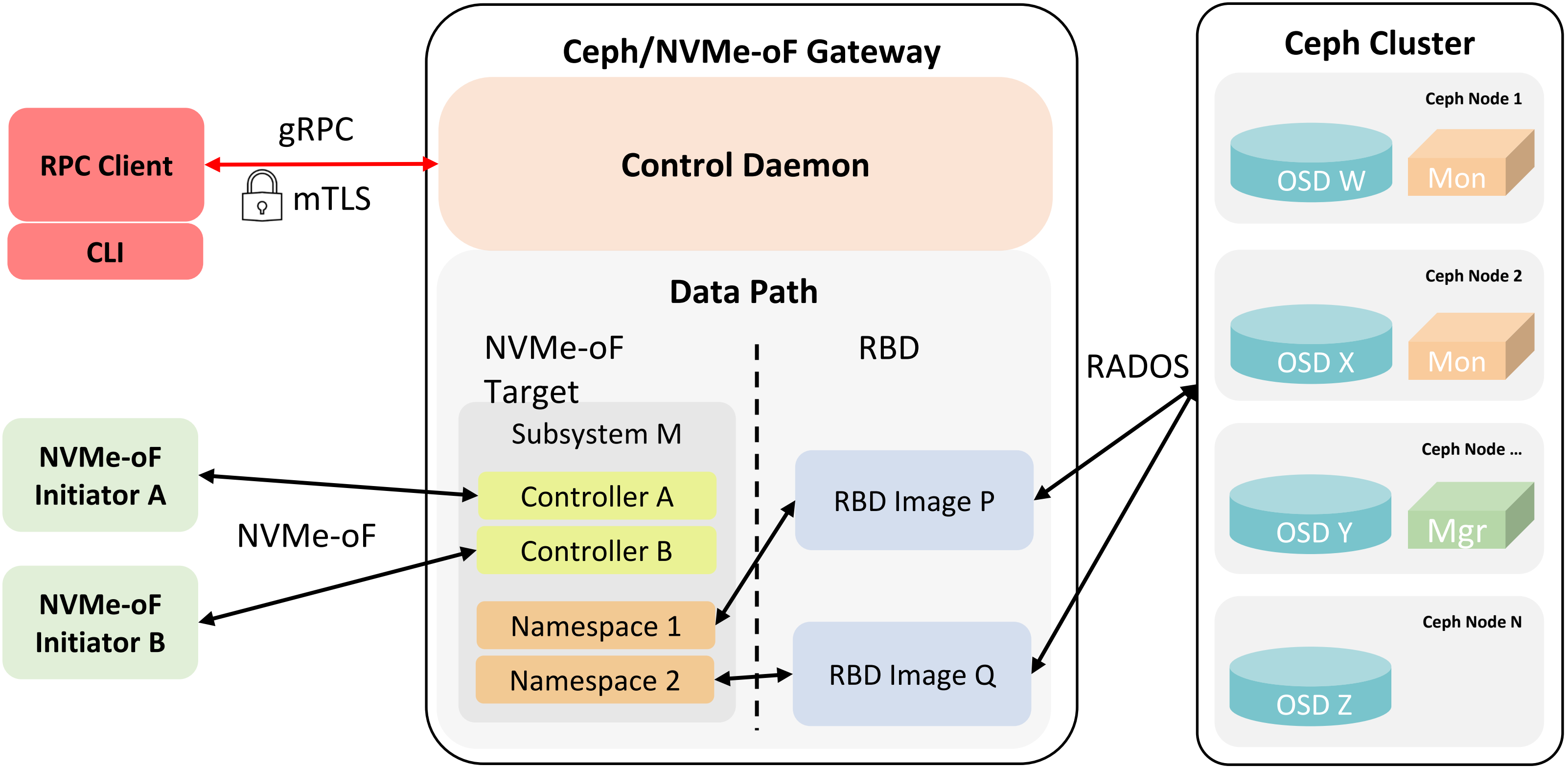
- **TCP**
- RDMA
- FC

NVMe-oF and Ceph

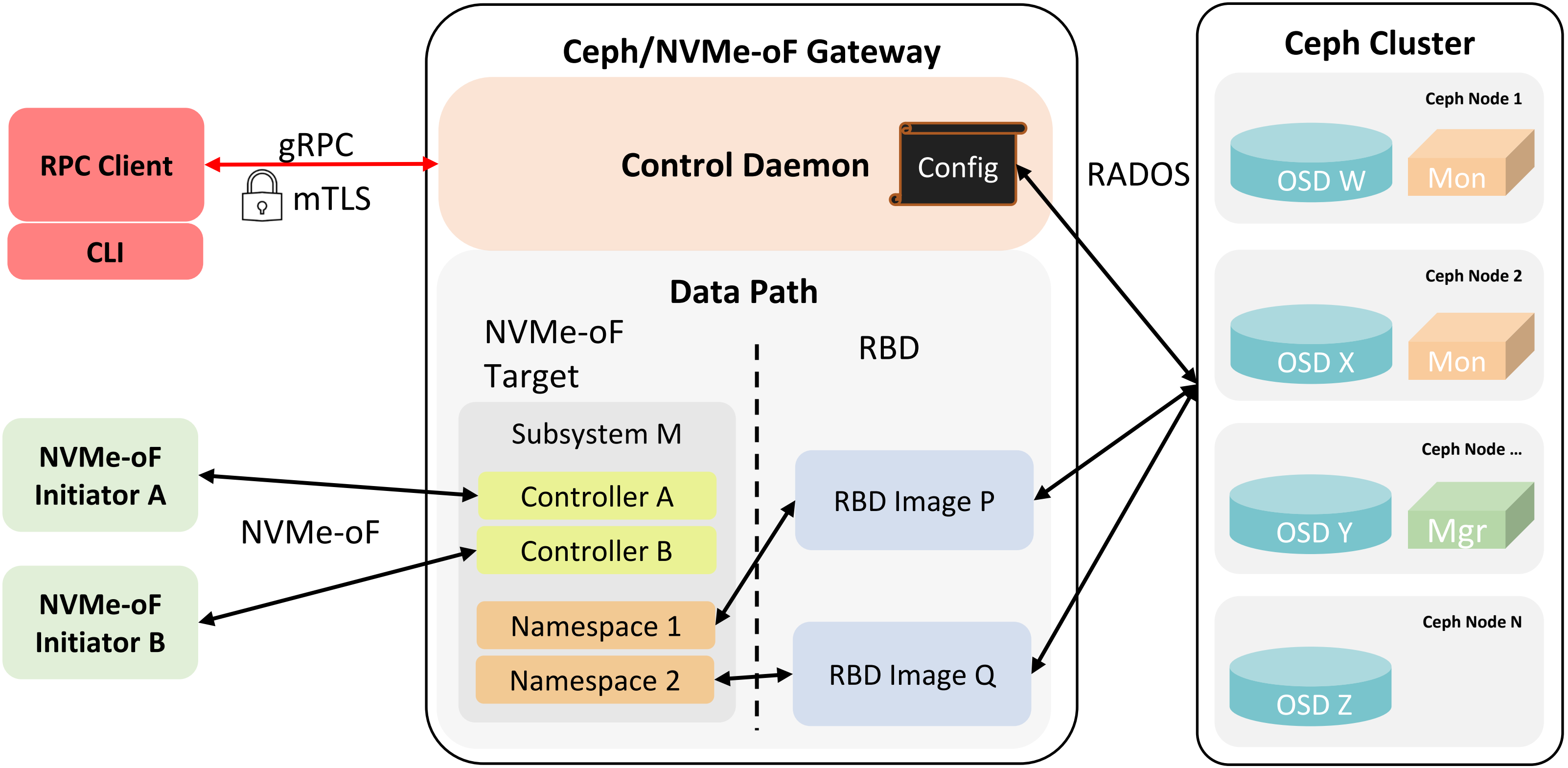


- Namespace mapped to an RBD image
- Subsystems logical grouping of Namespaces
- Each initiator get a Controller

Gateway Control Plane

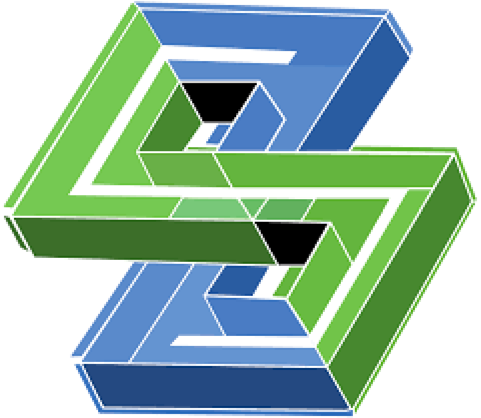


Gateway Configuration

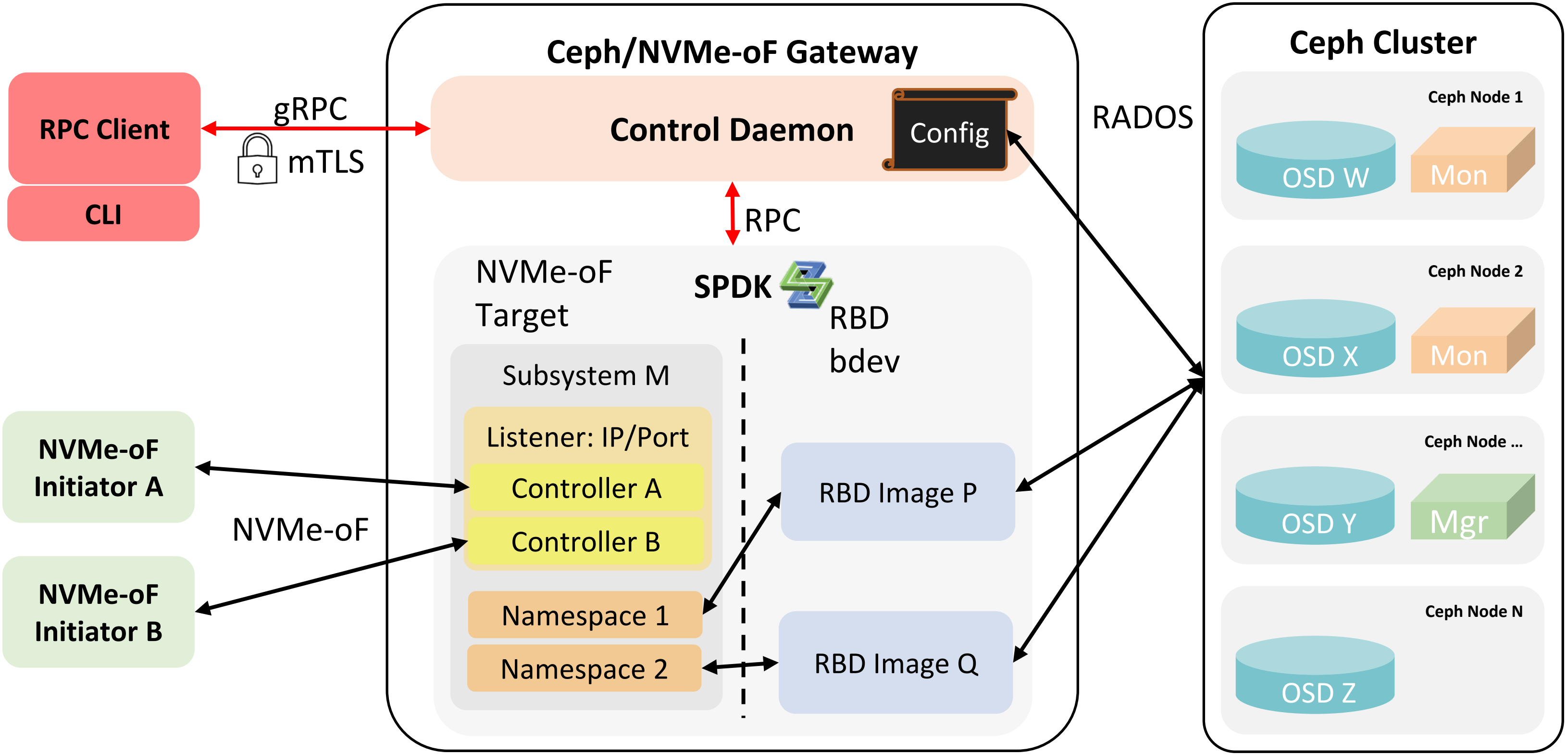


SPDK

- Storage Performance Development Kit (SPDK)
- https://spdk.io/
- Provides a tools and libraries for writing high performance, scalable, user-mode storage applications
- Userspace NVMe Over Fabric target
- Support for Ceph RBD with bdev_rbd
- Open source (BSD)

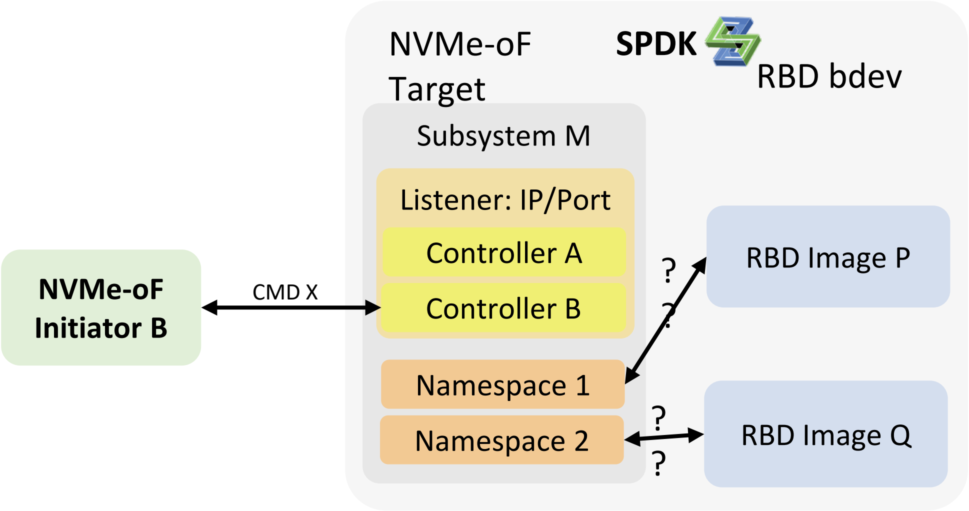


Data Path: SPDK

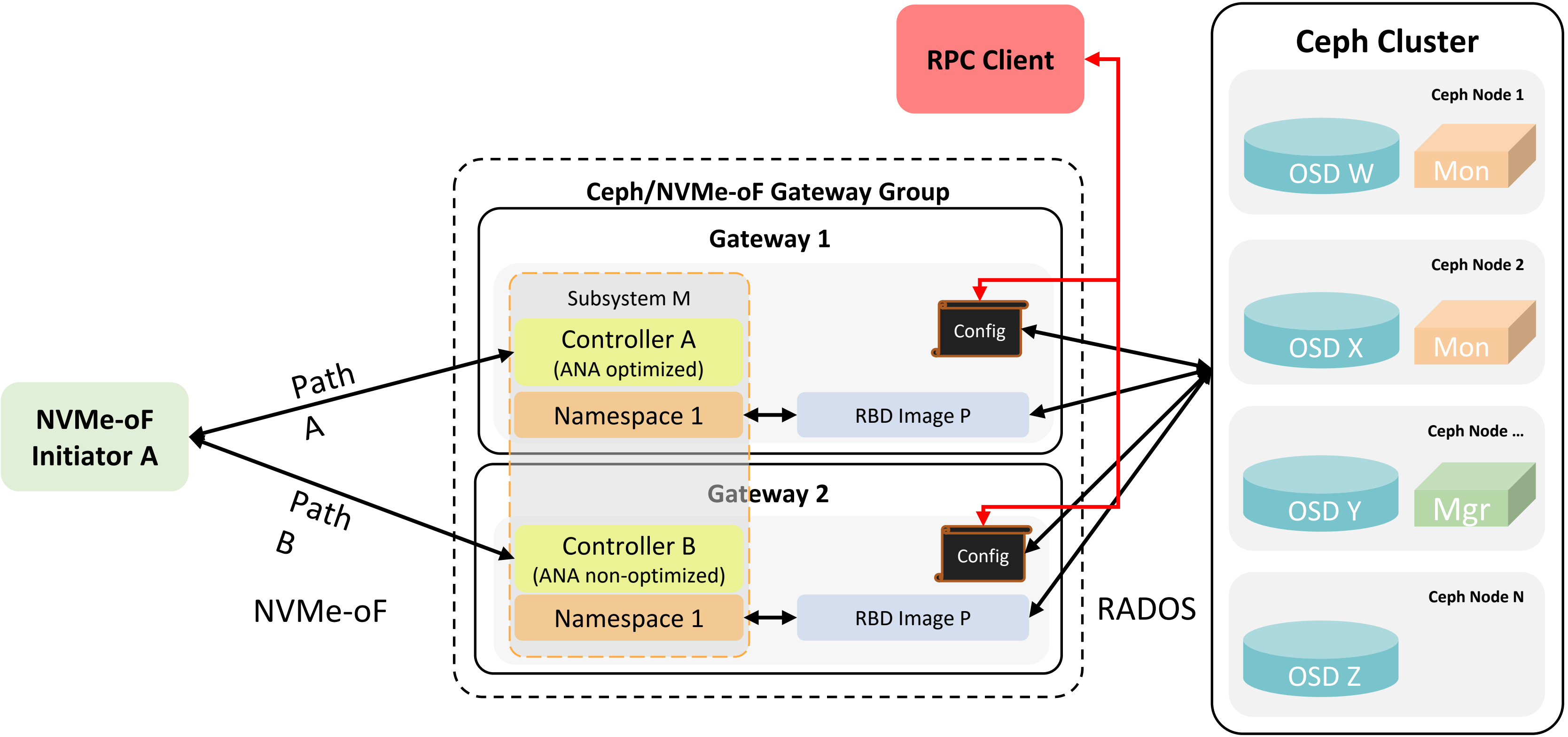


NVMe to RBD Commands Mapping

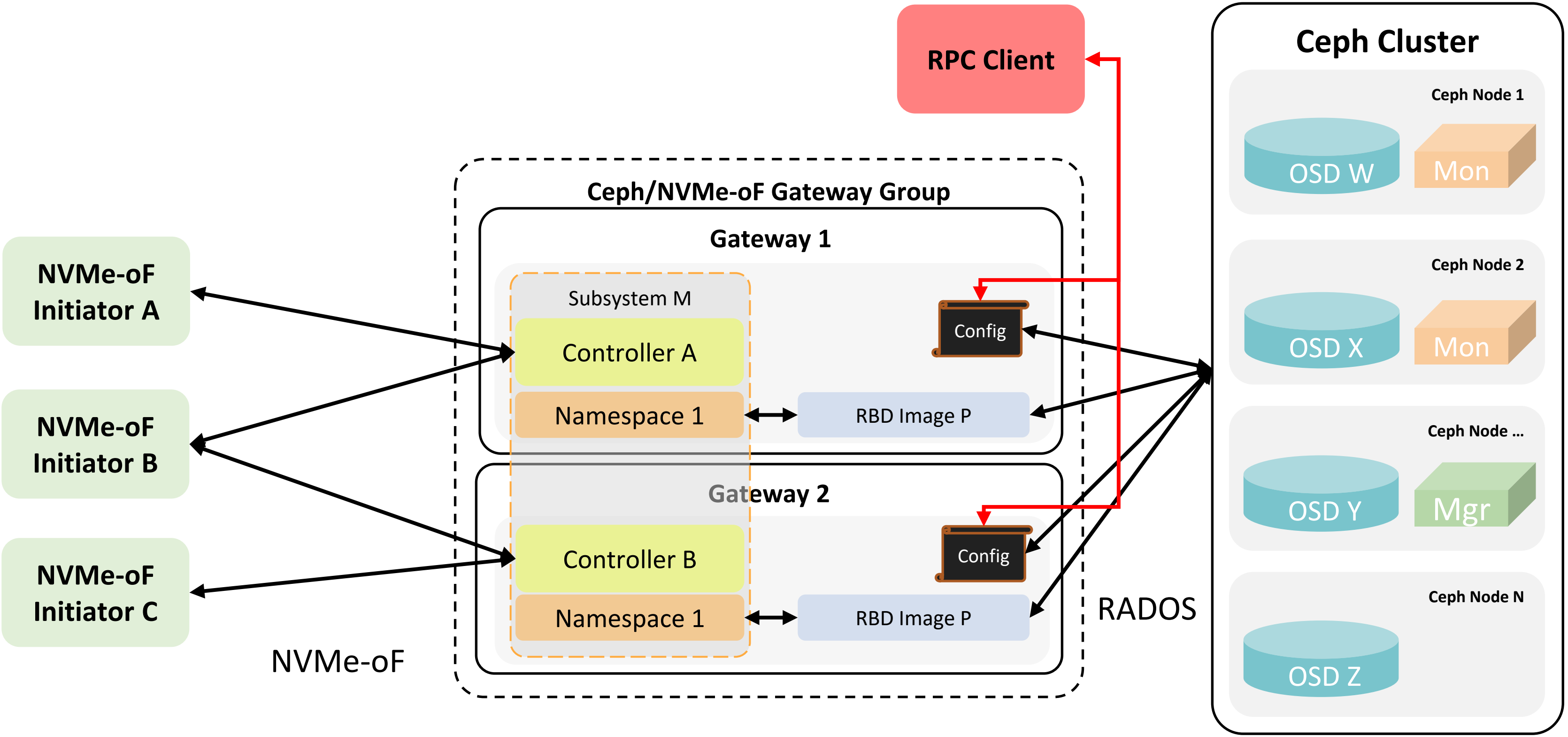
- RBD backend in SPDK maps NVMe operations to RBD API
- Natively supported
  - Read
  - Write
  - Unmap
  - Flush
  - Write zeroes
  - Compare and write
- Emulated
  - Compare
  - Copy
  - Abort**



Gateway Groups & Multi-pathing

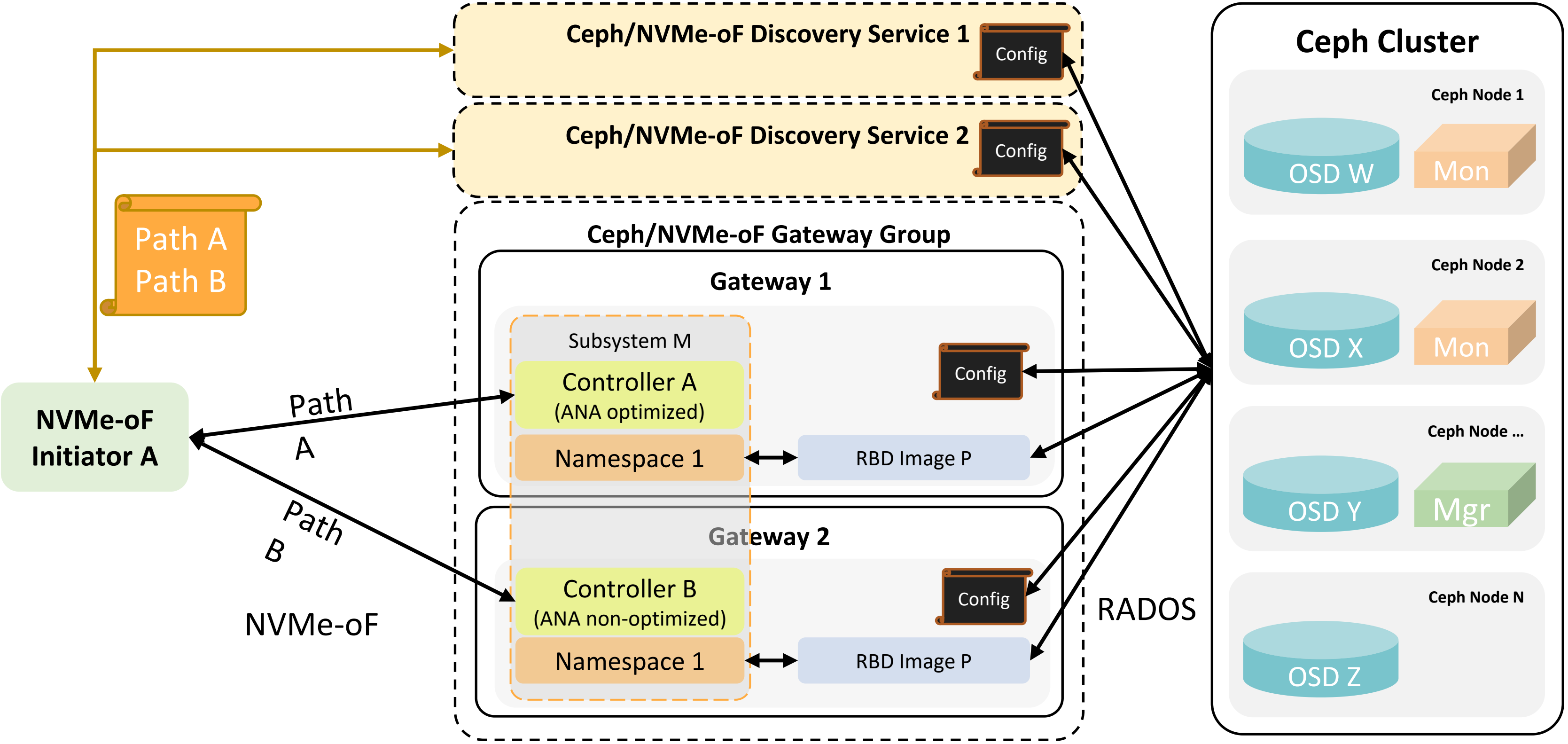


Gateway Groups: Load Balancing / Scaling



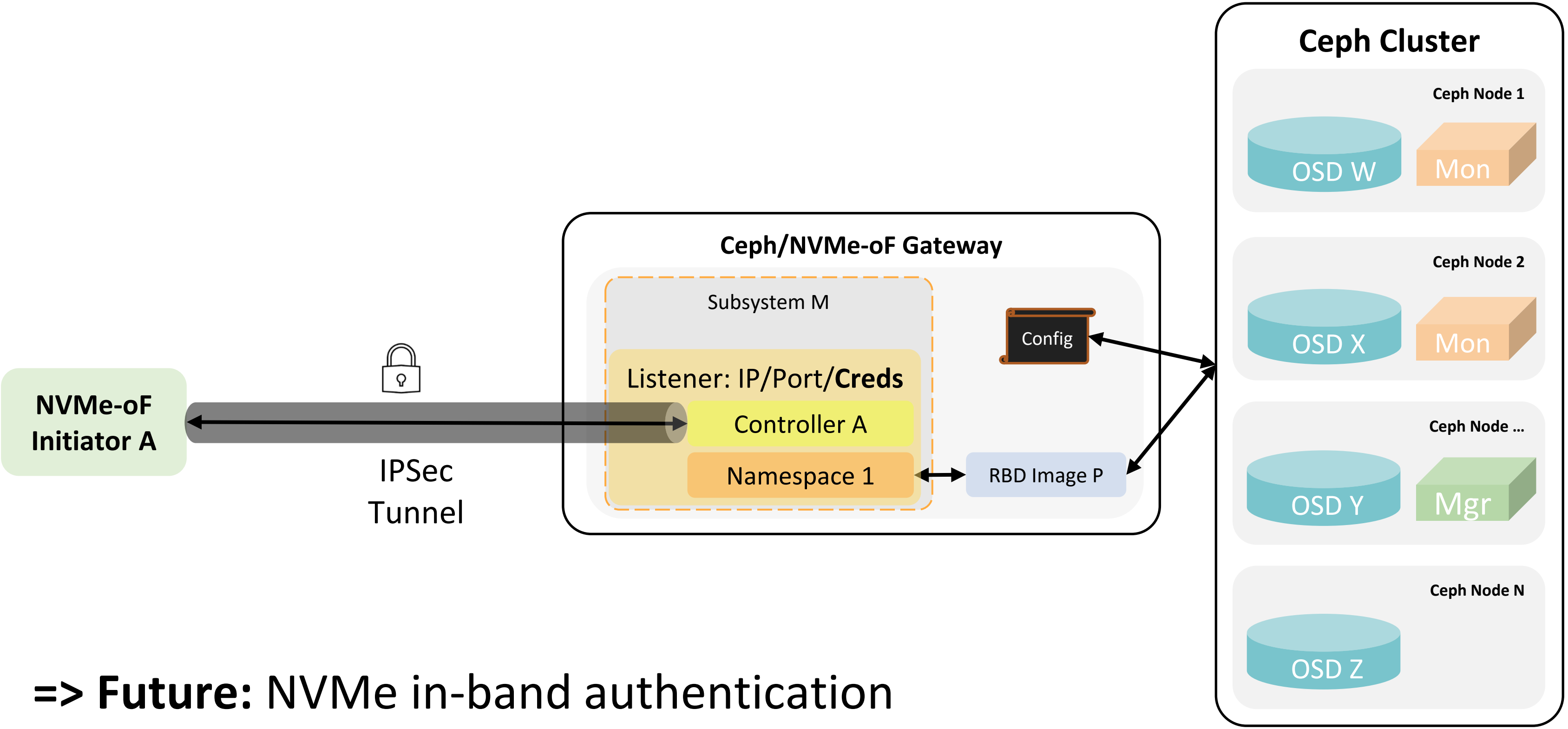
Planned/Future Features

Discovery



Ceph/NVMe-oF Discovery Service 1
Config
Ceph/NVMe-oF Discovery Service 2
Config
Path A
Path B
NVMe-oF Initiator A
Path A
Path B
NVMe-oF
Ceph/NVMe-oF Gateway Group
Gateway 1
Subsystem M
Controller A
(ANA optimized)
Namespace 1
Config
RBD Image P
Gateway 2
Controller B
(ANA non-optimized)
Namespace 1
Config
RBD Image P
RADOS
Ceph Cluster
Ceph Node 1
OSD W
Mon
Ceph Node 2
OSD X
Mon
Ceph Node ...
OSD Y
Mgr
Ceph Node N
OSD Z

Authentication / Encryption: IPSec

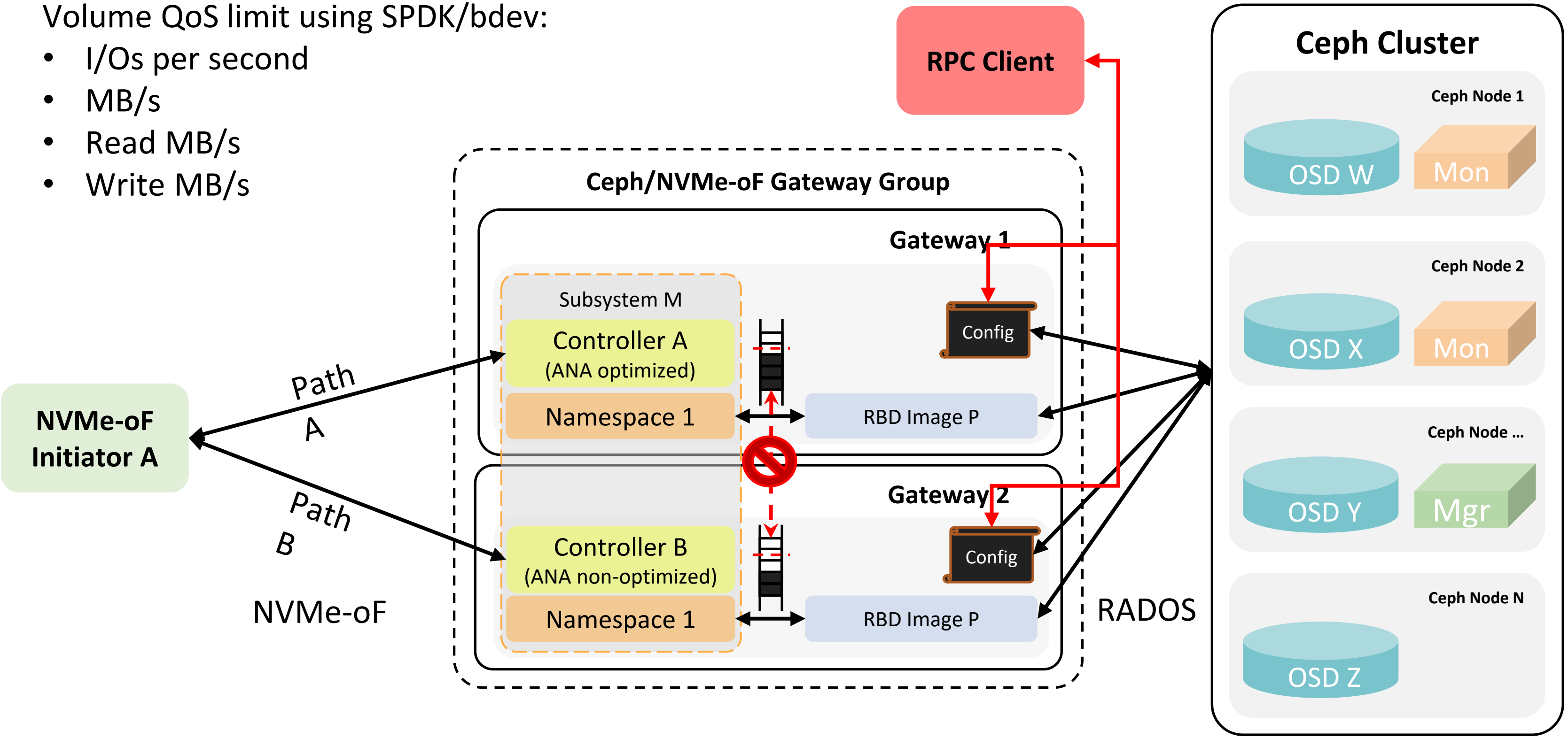


=> **Future:** NVMe in-band authentication

Quality of Service (QoS)

Volume QoS limit using SPDK/bdev:

- I/Os per second
- MB/s
- Read MB/s
- Write MB/s

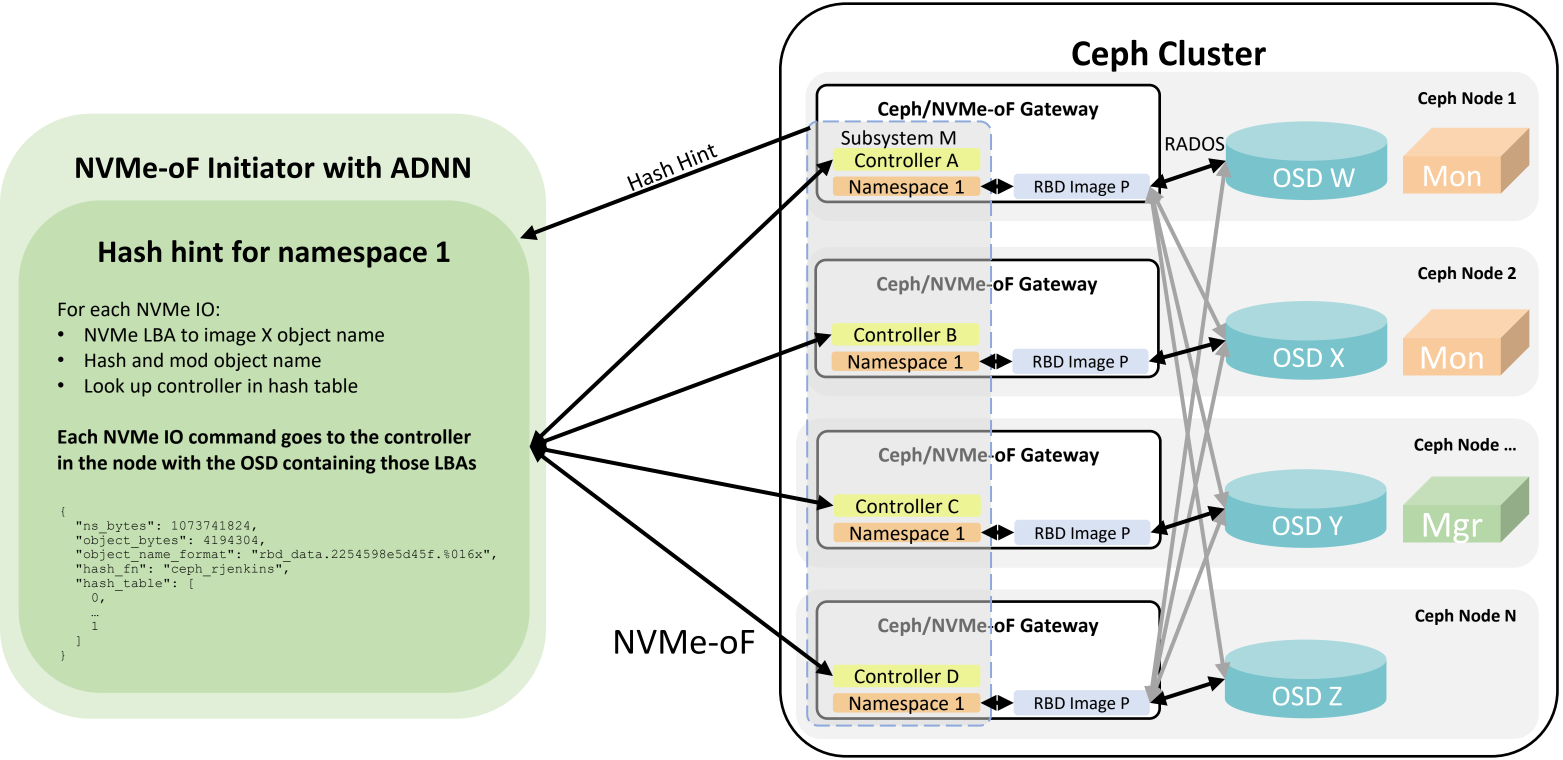


*Global QoS across gateways in a group is not planned

VMWare vSphere/ VAAI Support

- Use shared volumes to create single storage pool
- VMware vSphere Storage APIs – Array Integration (VAAI): Set of storage primitives that enable storage offloading
  - Atomic Test & Set (ATS)
    - Support in using NVMe compare & write fused operation
    - Cmp & write limited to RBD object size resp. stripe size (alignment*) => only 4K/ 1 block required
  - XCOPY (extended copy)
    - Copy NVMe command is supported in SPDK but QoS difficult
  - Write same (zero)
    - Write zeroes NVMe command => maps directly to RBD operation
  - Unmap (delete)
    - Supported as dataset management command => discard in RBD

Adaptive Distributed NVMe-oF Namespaces (ADNN)



NVMe-oF Initiator with ADNN

Hash hint for namespace 1

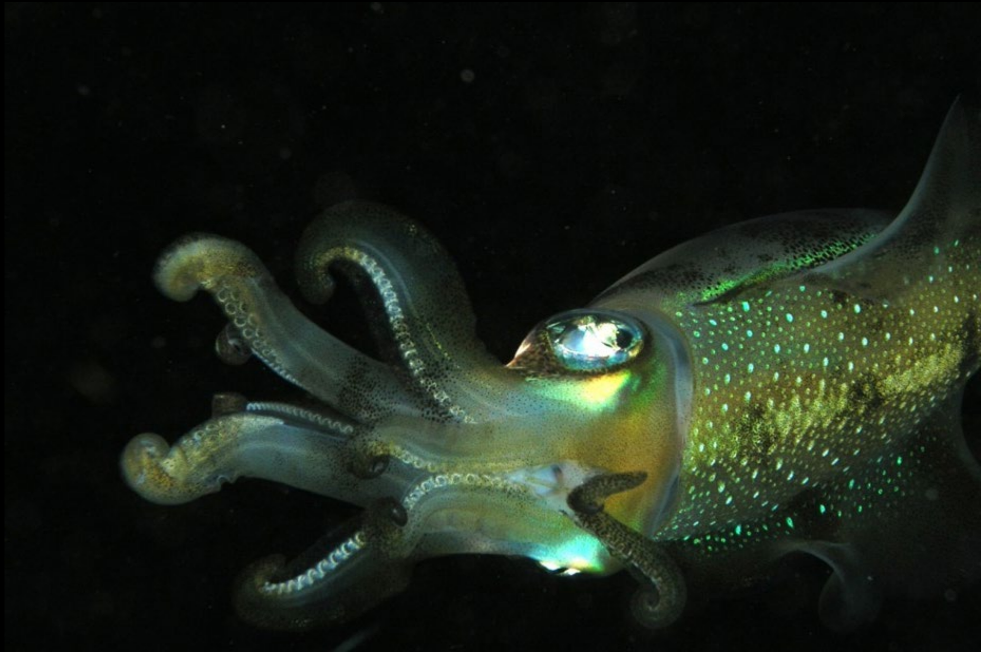
For each NVMe IO:
- NVMe LBA to image X object name
- Hash and mod object name
- Look up controller in hash table

Each NVMe IO command goes to the controller in the node with the OSD containing those LBAs

```
{
  "ns_bytes": 1073741824,
  "object_bytes": 4194304,
  "object_name_format": "rbd_data.2254598e5d45f.%016x",
  "hash_fn": "ceph_rjenkins",
  "hash_table": [
    0,
    …
    1
  ]
}
```

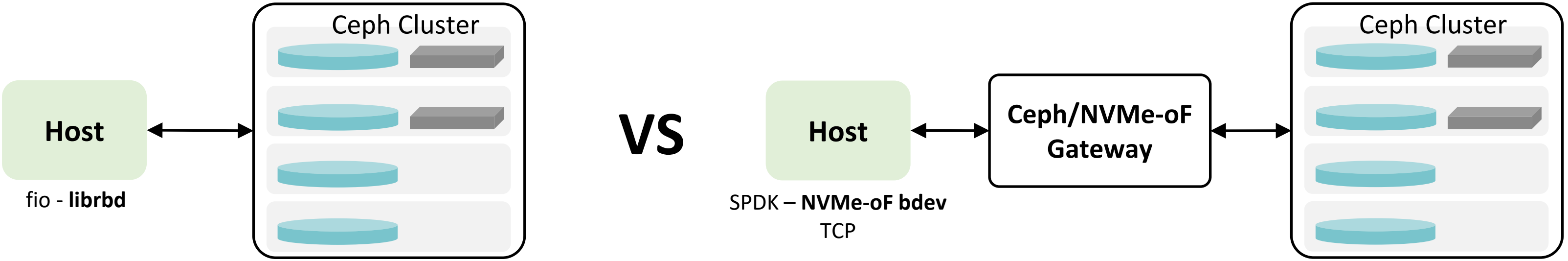
- Move gateway group into OSD nodes
- Hosts route each NVMe IO to the correct node with hash function
- Dedicated gateways (and extra hop) eliminated
- Host overhead much less than librbd (more offload friendly)

Performance



Performance: Setup

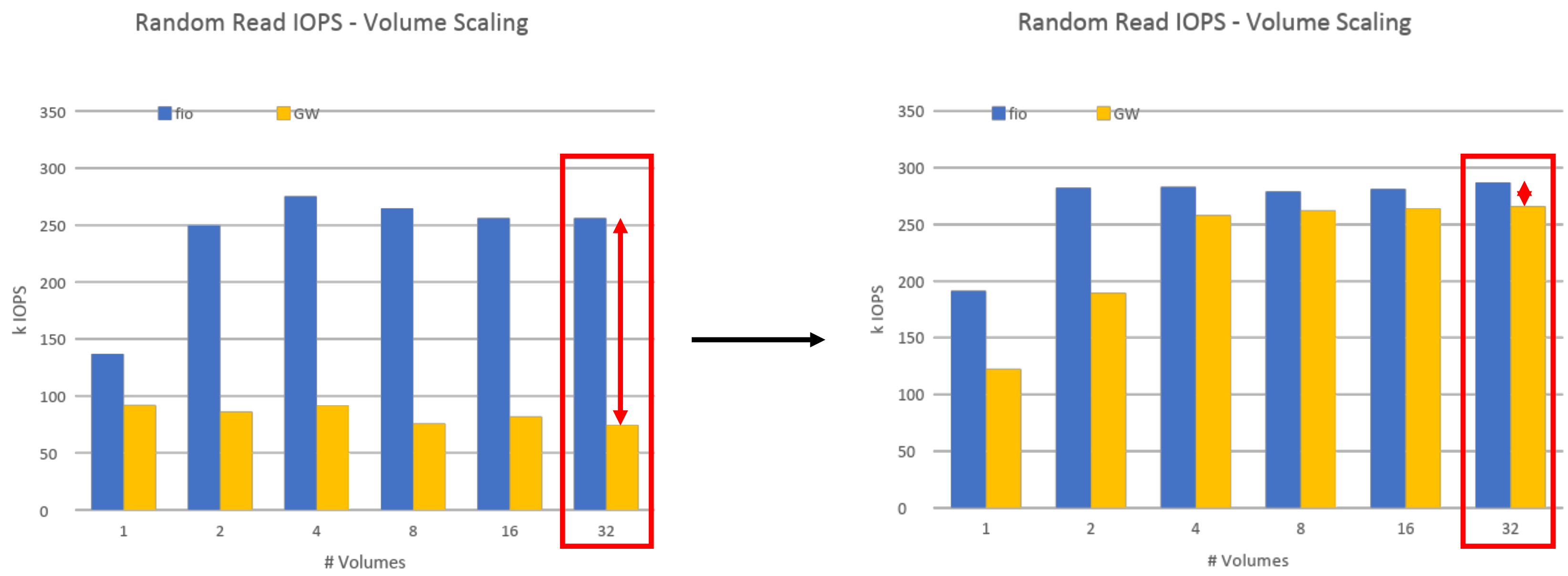
Goal: As close as possible to non-gateway performance



***Nodes:** 2x Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz (**28 cores**), **100 Gbit/s** Mellanox ConnectX-5, Samsung PM1725a*
***3-node Ceph cluster:** Pacific & Quincy with rbd_cache=FALSE*

https://ci.spdk.io/download/2022-virtual-forum-prc/D2_4_Yue_A_Performance_Study_for_Ceph_NVMeoF_Gateway.pdf

Performance: Improvements



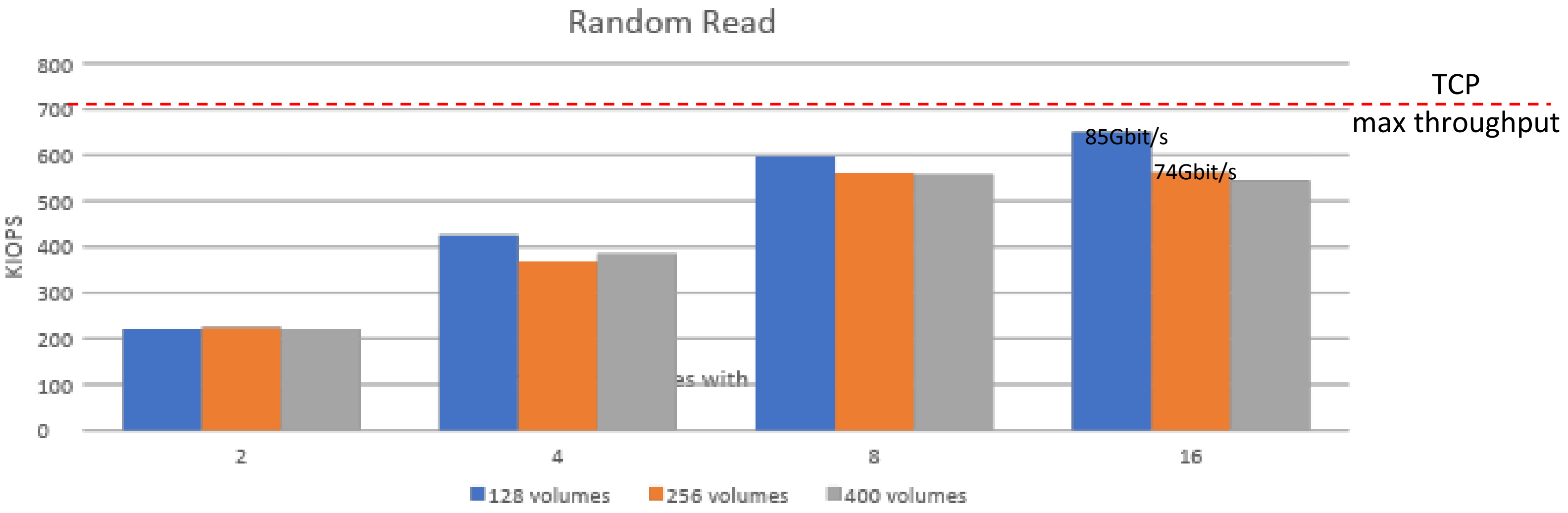
Random Read IOPS - Volume Scaling

Random Read IOPS - Volume Scaling

IO size = 16KiB, **Total QD** = 256
NVMe backed OSDs

Performance: Volume Scaling

- Use multiple Ceph client instances to improve performance in SPDK

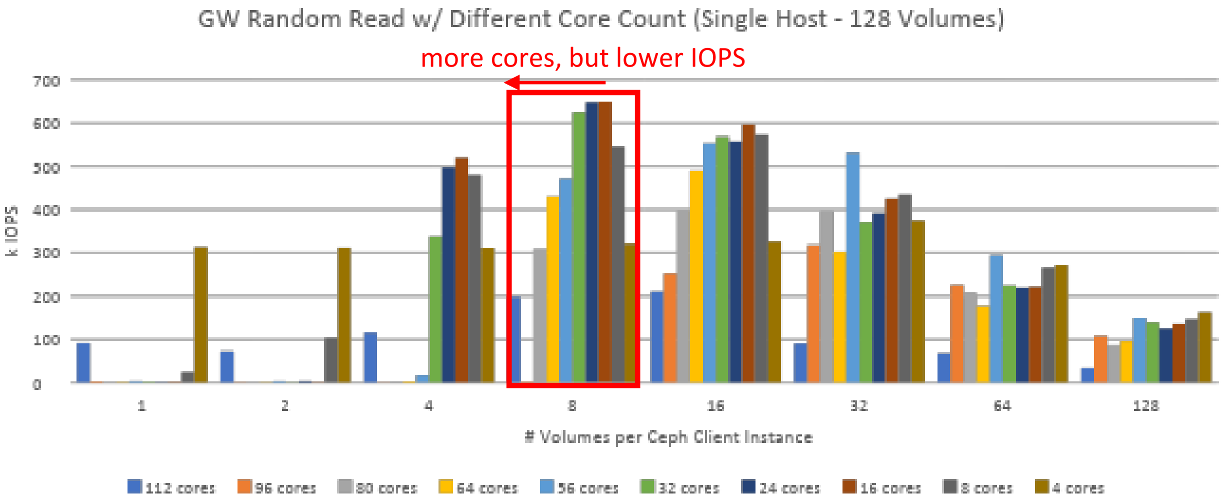


IO size = 16KiB, **Total QD** = 1024, **SPDK core mask** = 16 cores

RAMDisk backed OSDs

Performance: 128 volumes

- Use multiple Ceph client instances to improve performance in SPDK
- Check how core count effects performance



IO size = 16KiB, **Total QD** = 256

RAMDisk backed OSDs

Thank You!

Jonas Pfefferle, Danny Harnik, Scott Peterson, Yue Zhu, Ernesto Puerta, Bharti Wadhwa, Ilya Dryomov, Josh Durgin, Sandy Kaur, Rebecca Cloe, Sanjeev Gupta, Brett Niver, Guifeng Tang, Mykola Golub, Congmin Yin, TJ Harris, Adam King, Redouane Kachach, Rahul Lepakshi, Aviv Caro, Alexander Indenbaum, Leonid Chernin, Gil Bergman, Barak Davidov, Roy Sahar ….

clyso
cloud system solutions

IBM

intel®

Red Hat

Join the Community

https://github.com/ceph/ceph-nvmeof

https://pad.ceph.com/p/rbd_nvmeof

Ceph Slack channel: **#nvmeof**

Weekly meeting: every Tuesday at 7am PT
https://meet.jit.si/ceph-nvmeof

Please take a moment to rate this session.

Your feedback is important to us.