2022 SPDK PRC Virtual Forum

A Performance Study for Ceph NVMe-over-Fabrics Gateway

Danny Harnik, Scott Peterson, Orit Wasserman, Yue Zhu, Ernesto Puerta, Bharti Wadhwa, Ilya Dryomov, Sandy Kaur, Rebecca Cloe, Sanjeev Gupta, Brett Niver, Guifeng Tang, Mykola Golub, Congmin Yin, TJ Harris, Jonas Pfefferle

IBM Research intel Red Hat SUSE

Why NVMe-over-Fabrics?

RADOS **B**lock **D**evice (RBD)

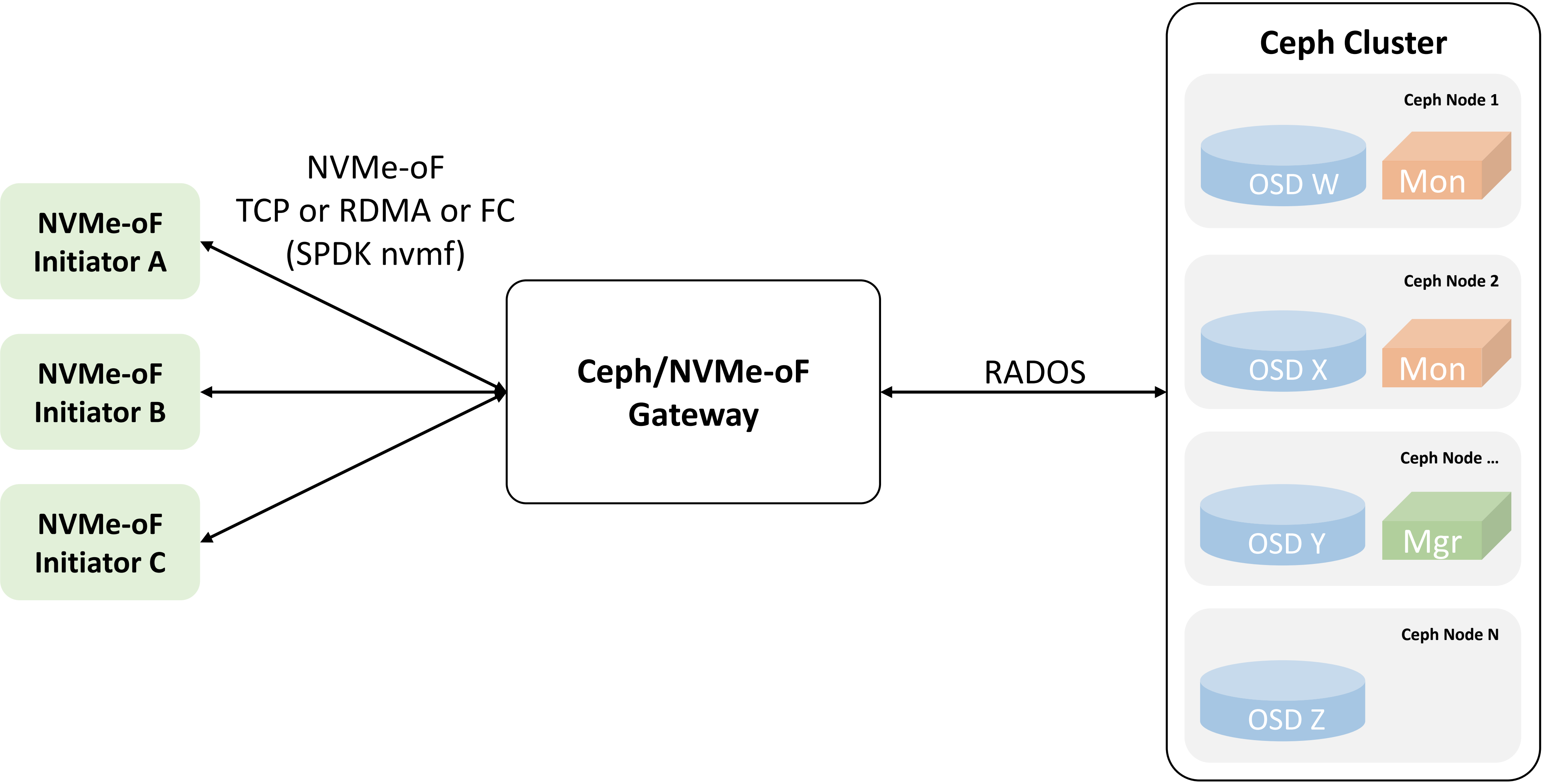
- RADOS protocol
- Distributed n-to-m protocol
- Reliable object access to sharded and replicated/erasure coded storage

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

NVMe-over-Fabrics (NVMe-oF)

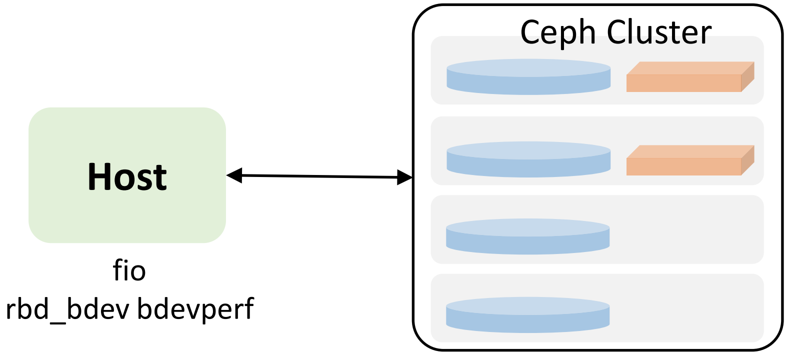
- Open, widely adopted ***industry standard***
- Enable use-cases where NVMe-oF is already part of ***ecosystem***
- Take advantage of NVMe-oF ***offloading*** in DPUs

Ceph NVMeoF Gateway Overview

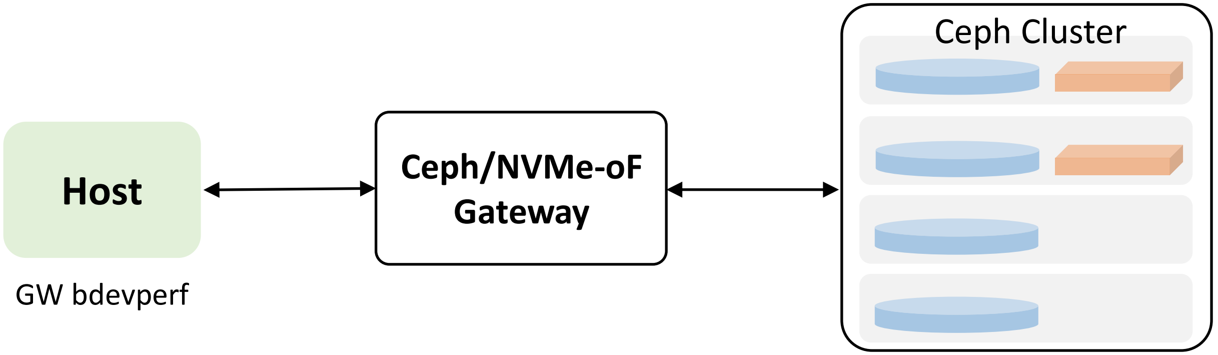


Our Performance Goal

- Target 5~10% performance loss compared to the non-GW solution
  - Non-GW solution: librbd (host ←→ Ceph)
  - Our solution: NVMeoF GW (host ←→ GW ←→ Ceph)



Non-GW Solution



Our GW Solution

- IO benchmarks
  - fio: reports max RBD volume performance with librbd
  - SPDK bdevperf:
    - rbd_bdev: reports SPDK librbd-based rbd_bdev performance (librbd + SPDK)
    - GW: reports SPDK rbd_bdev w/ NVMe-oF performance (our GW solution)

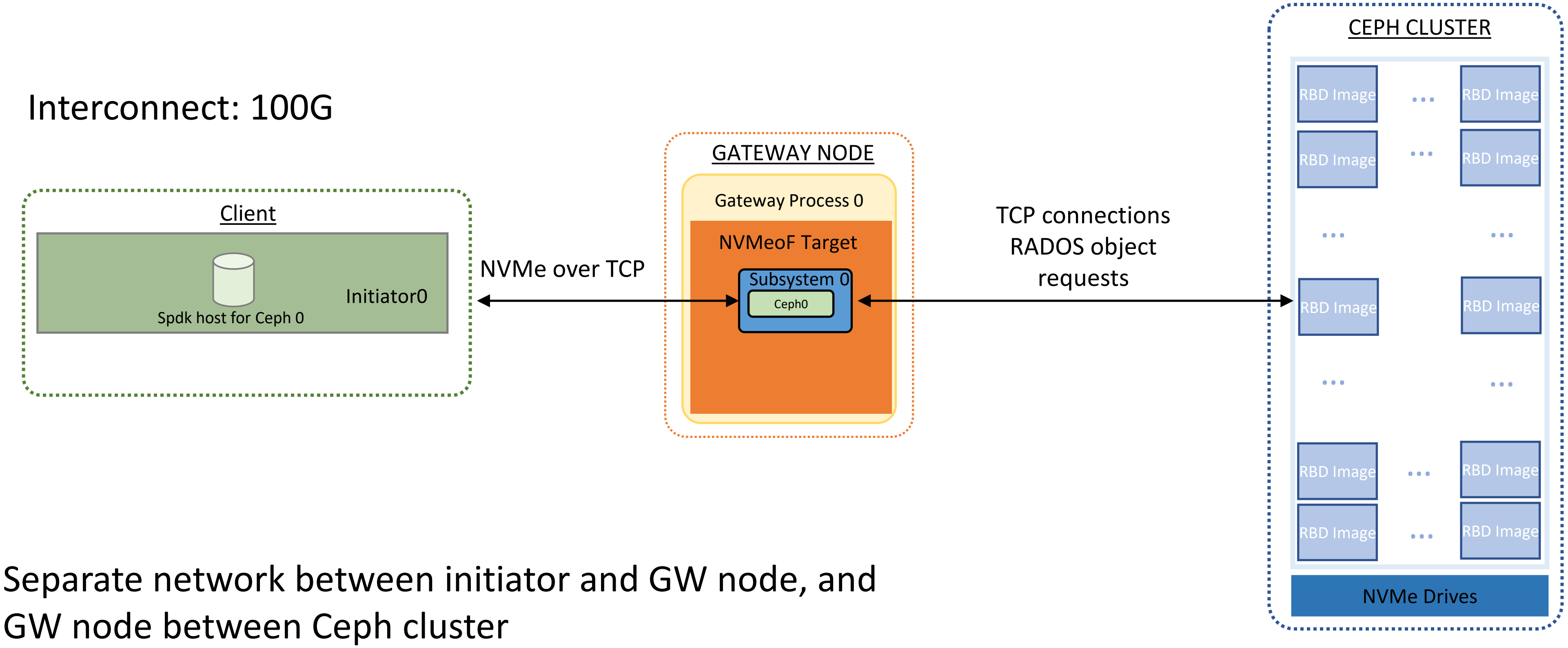
Our Test Setup

- Test environment
  - Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz (**28 cores**)
  - **100 Gbit/s** Mellanox Technologies MT28800 Family [ConnectX-5 Ex] connected via PCIe Gen3
  - Samsung PM1725a NVMe SSD
  - Client: 1 node; Ceph cluster: 3 nodes; GW: 1 node

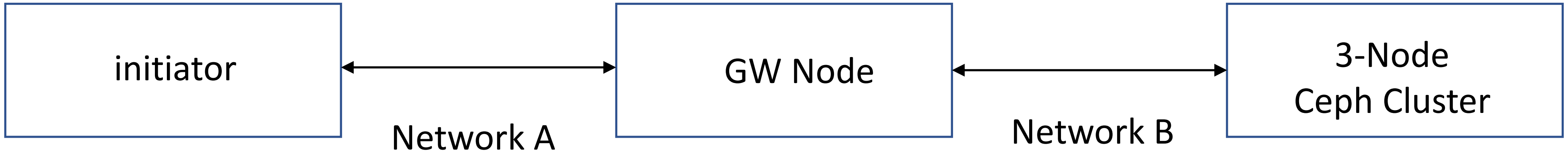
- Test setup
  - Ceph Pacific & Quincy w/ rbd_cache=FALSE
  - Block size = 16KiB, total QD=256, total volume size = 512GiB

SPDK Initiator w/ Ceph Gateway Test System

Interconnect: 100G

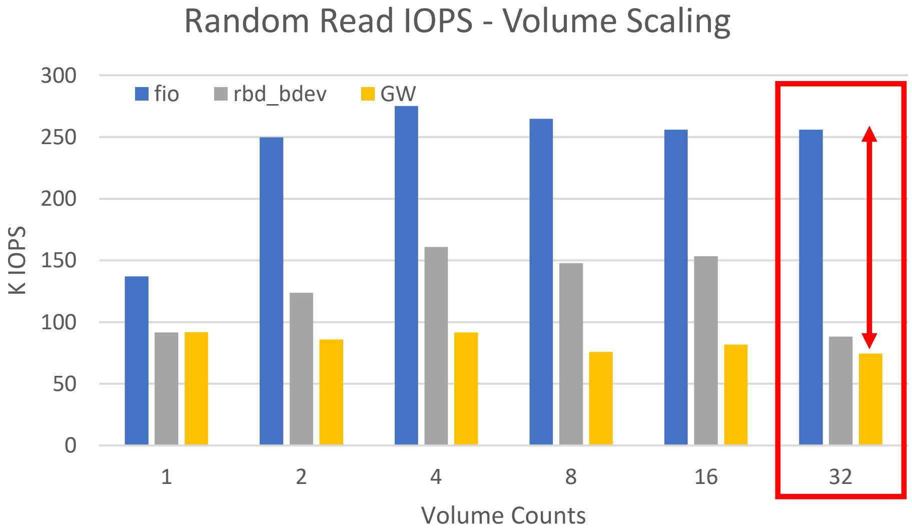


Separate network between initiator and GW node, and GW node between Ceph cluster

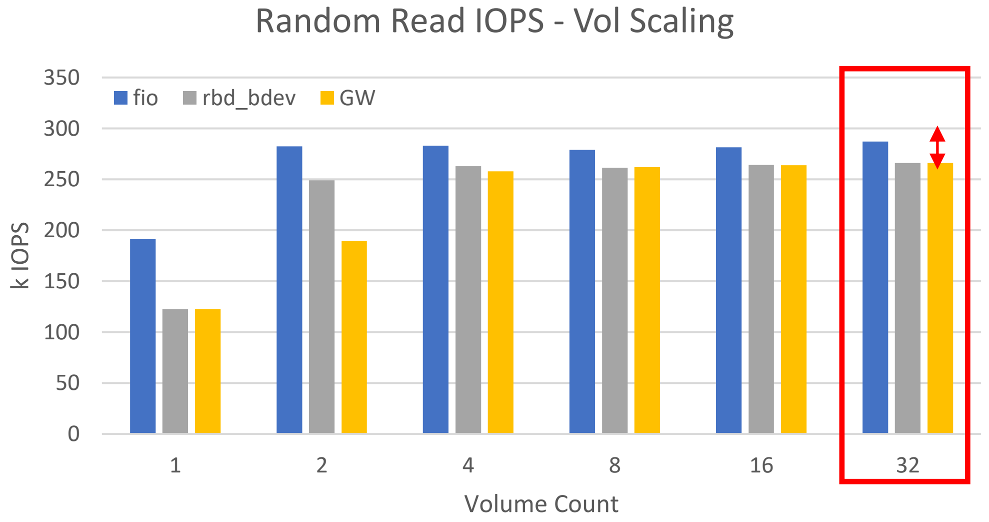


Random Read IOPS on Volume Scaling (SSD)

- Spdk bdev (bdevperf w/ rbd_bdev) and GW (bdevperf w/ nvmf) performance cannot match librbd's (left)
- In most cases, GW's performance is very close to expected performance (fio) (right)



Initial Performance



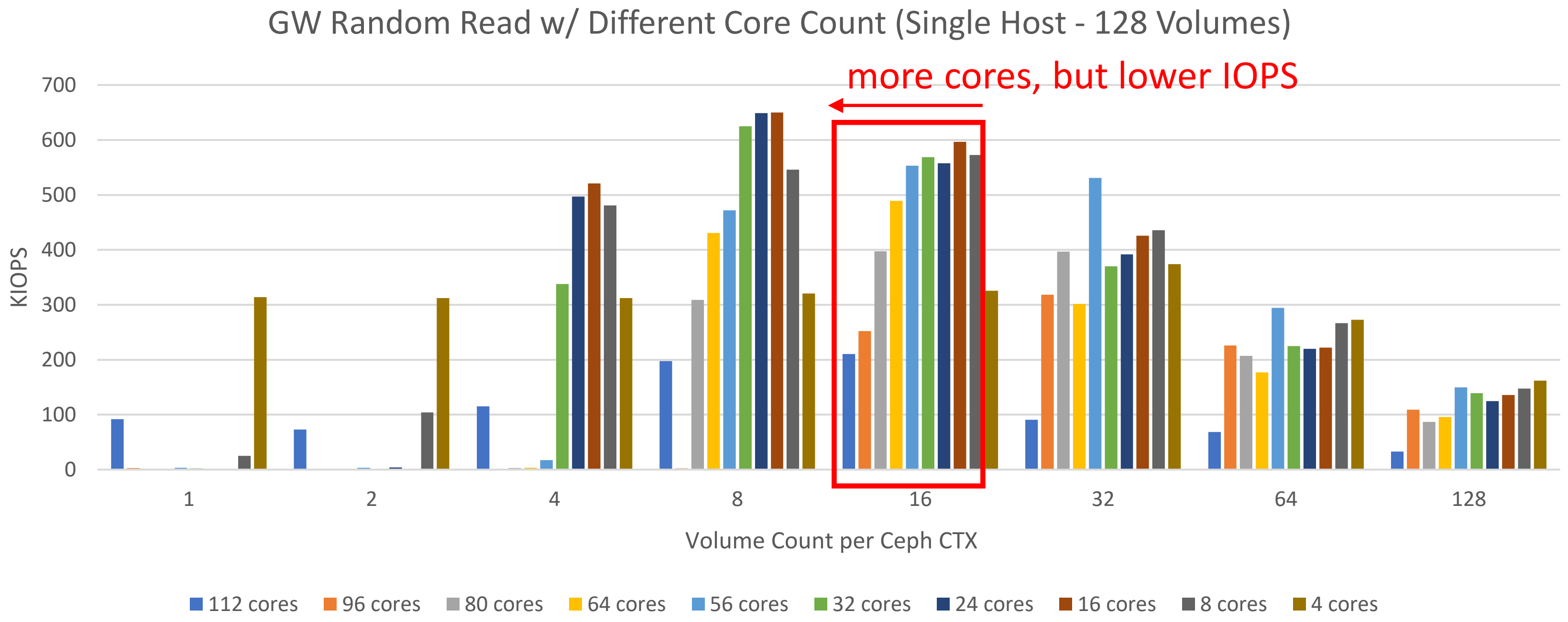
Current Performance

Keys to Performance

- NUMA affinity for cores and NIC
- rbd_bdev IO loadshare between CPU cores
  - https://review.spdk.io/gerrit/c/spdk/spdk/+/10416
  - We helped identify the importance of this patch
- Ceph client instantiations sharing between RBD volumes
  - On slide 6 , every rbd_bdev has its own Ceph client instantiations for right figure, but all rbd_bdev share one Ceph client instantiation for left figure
  - We are identifying the performance effect of sharing Ceph context (see next slide)
- Ceph version
  - We switched from Octopus to Pacific (we are now on Qunicy)
- tcmalloc instead of jmalloc

How do Core Count and Ceph CTX Count Affect IOPS?

- Performance numbers are gathered on RAMDisk based OSD (on Ceph Quincy)
- **The overall performance does not always benefit from high core count and Ceph CTX count**



Current Things To Do for Performance

- Support specified core mask for librbd when creating rbd_bdev.
    - We are going to submit a patch to enable this via bdev_rbd_create
- Find the “optimal” setup for the ratio of core count and Ceph CTX for given RBD volume count
- Understand the relationship between performance and SPDK_DEFAULT_MSG_MEMPOOL_SIZE
    - We submitted a patch to enable configurable SPDK_DEFAULT_MSG_MEMPOOL_SIZE via cmd line
        - https://review.spdk.io/gerrit/c/spdk/spdk/+/15552
- Volume scaling tests for volume count beyond 128

Thank you!

https://github.com/ceph/ceph-NVMeoF
https://pad.ceph.com/p/rbd_NVMeoF