linaro Connect 2025

A Journey of Performance Analysis in Storage Systems:

Taking Ceph SPDK NVMe-oF Gateway as an Example

Agenda

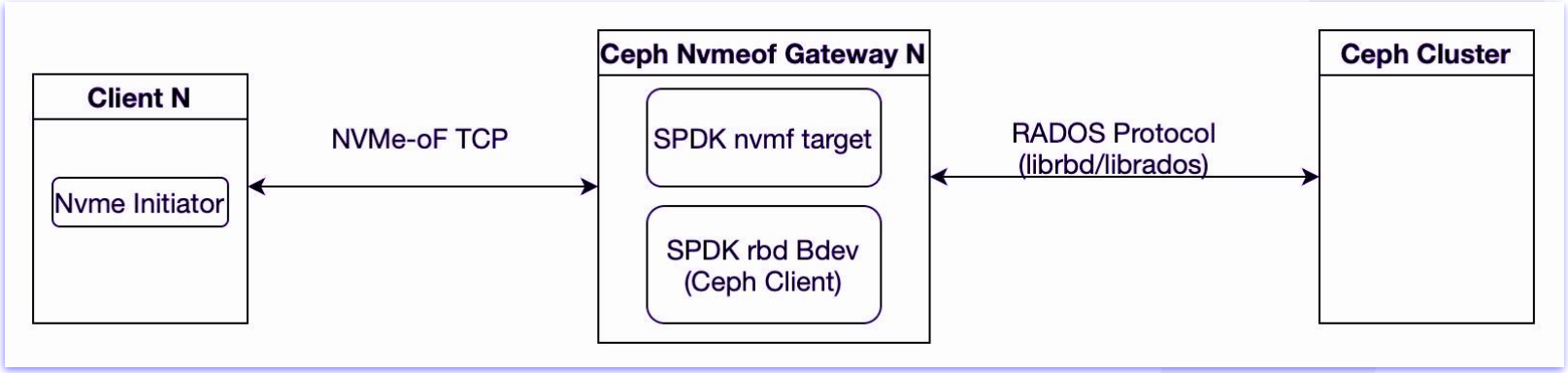
- Ceph SPDK NVMe-oF Gateway Introduction
- Ceph SPDK NVMe-oF Gateway Arm Upstream Support
- Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

Ceph SPDK NVMe-oF Gateway Introduction

Ceph SPDK NVMe-oF Gateway Introduction

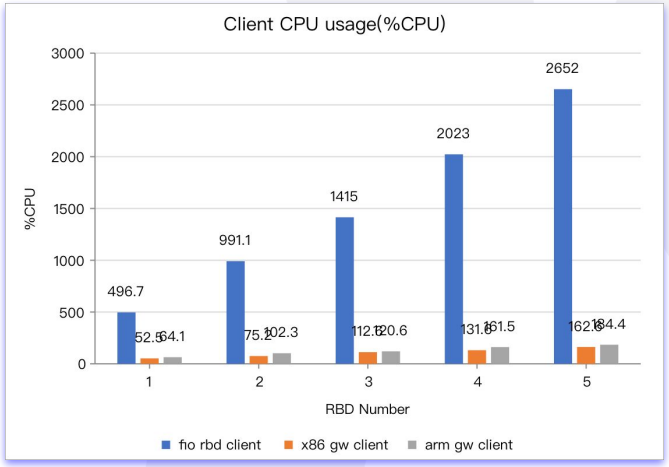
linaro Connect

- What is Ceph NVMe-oF Gateway?
  - The NVMe-oF Gateway presents an NVMe-oF target that exports RADOS Block Device (RBD) images as NVMe namespaces.
  - The NVMe-oF protocol allows clients (initiators) to send NVMe commands to storage devices (targets) over a TCP/IP network, like iSCSI.
  - It is an iSCSI SAN similar network disk solution



Ceph SPDK NVMe-oF Gateway Introduction

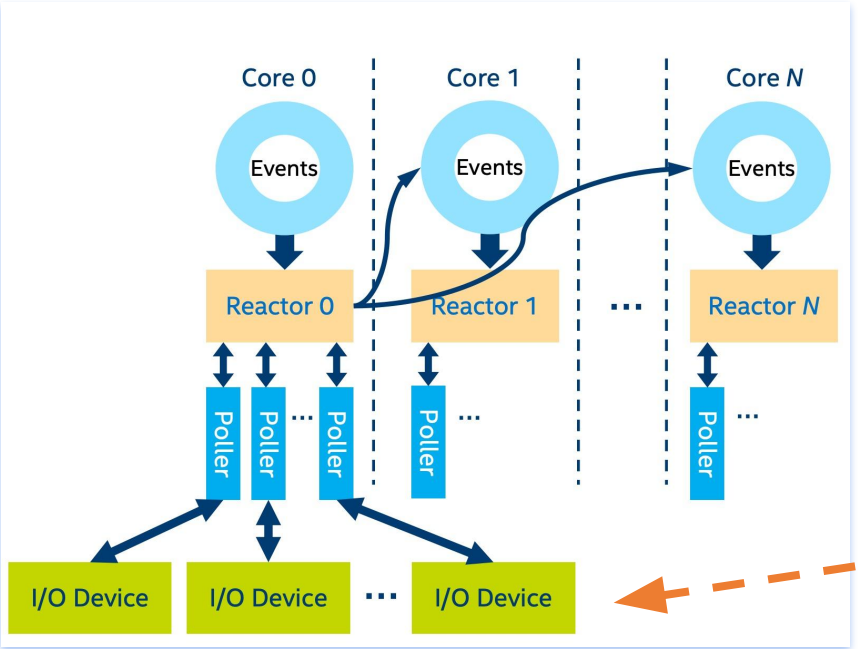
- Why is Ceph NVMe-oF Gateway?
  - Replace Ceph-iSCSI, NVMe-oF is a modern, multiple IO queues, storage protocol.
  - Enabling clients without native Ceph client support to access Ceph block storage.
  - Very low client %CPU, move Ceph client to the Gateway node.
- Use case
  - Possibly all the SAN and NVMe-oF use cases, requires client support nvme initiator
  - For no ceph client support environment
  - VMware Virtual Machine, bare metal boot volumes, Virtual Desktop
  - PDUs and hardware accelerators
  - Databases and Analytics, AI etc.
- References
  - https://docs.ceph.com/en/latest/rbd/nvmeof-overview/
  - https://www.spiceworks.com/tech/networking/articles/storage-area-network/
  - https://www.starwindsoftware.com/blog/what-is-nvme-of-nvme-over-fabrics/

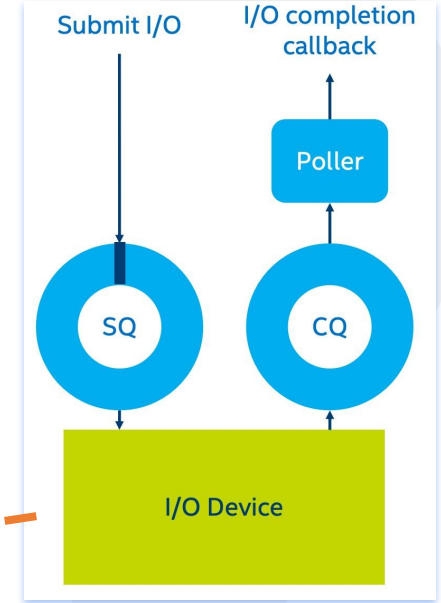


Ceph NVMe-oF Gateway Introduction

SPDK overview

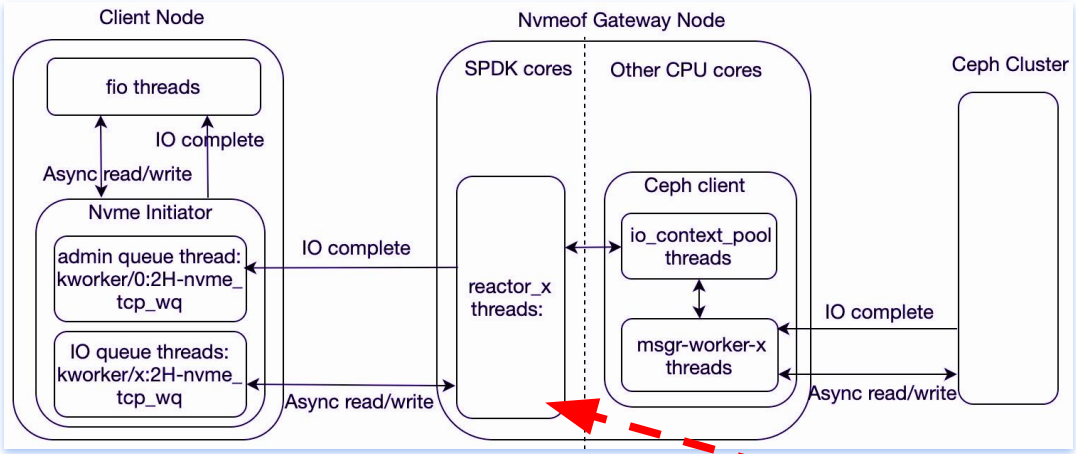
- References:
  - https://spdk.io/doc/overview.html
  - https://www.cnblogs.com/yi-mu-xi/p/12821103.html





Ceph NVMe-oF Gateway Introduction

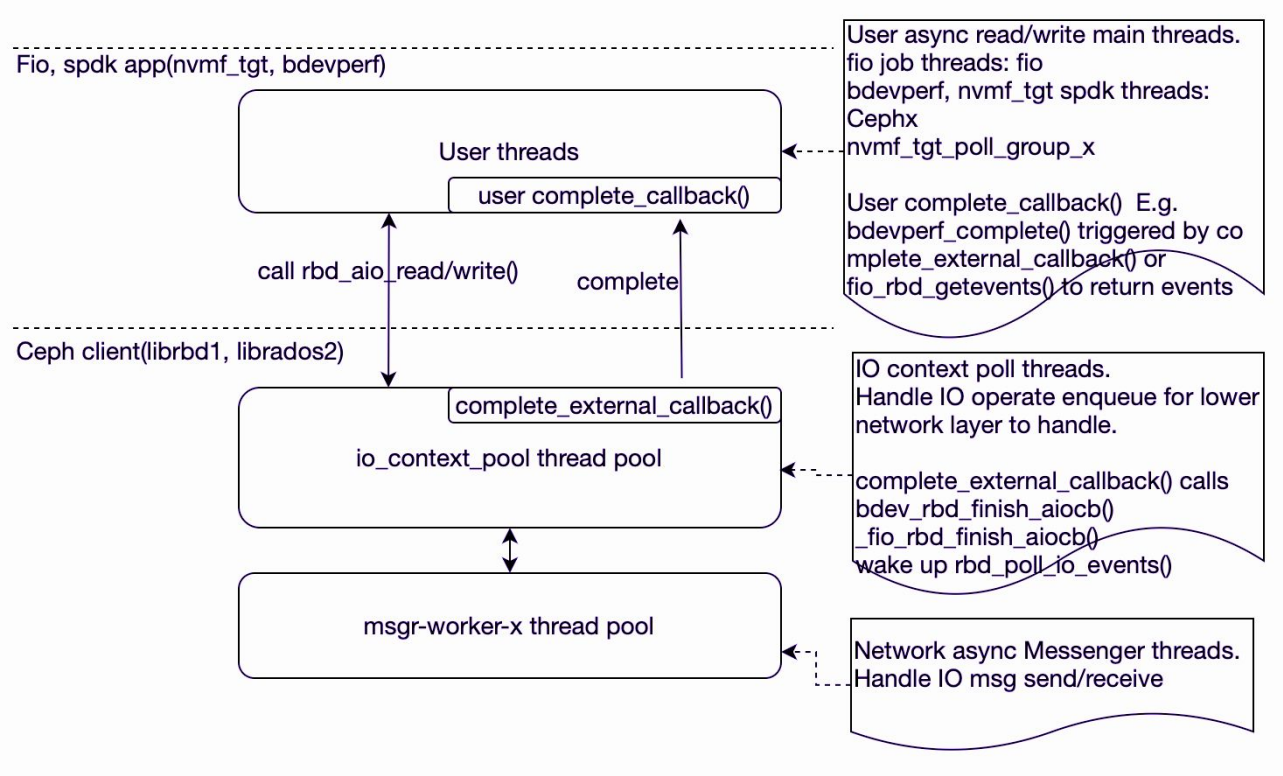
- The whole thread model




```
reactor_run()
{
        while (1) {
                // run the reactor event ring functoin()
                event_queue_run_batch(reactor);

                // poll thread for each thread nvmf tgt poll group_xxx
                spdk_thread_poll() {
                        // run the thread msg ring function()
                        // e.g. bdevperf_complete()
                        msg_queue_run_batch(thread, max_msgs);

                        // poll all the pollers of the thread,
                        // e.g. nvmf_poll_group_poll
                        thread_execute_poller()
                }
        }
}
```

Ceph NVMe-oF Gateway Introduction

- Gateway node thread model



Ceph SPDK NVMe-oF Gateway Arm Upstream Support

Ceph NVMe-oF Gateway Arm Upstream Support

linaro Connect

- Enablement – Arm64 building
  - Ticket: https://github.com/ceph/ceph-nvmeof/issues/781
  - PRs
    - Fix x86 hardcode: https://github.com/ceph/ceph-nvmeof/pull/782
    - Add Arm build: https://github.com/ceph/ceph-nvmeof/pull/783
  - Arm Build Note: https://github.com/ceph/ceph-nvmeof/blob/devel/README.md

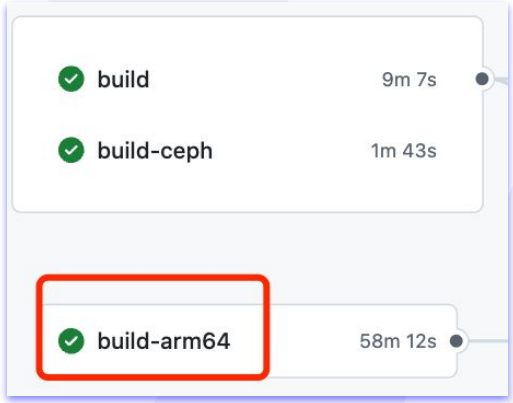
NOTE: For Arm64 build, the default SPDK building SoC is `generic`. To build SPDK for other SoC you need to override the default values of `SPDK_TARGET_ARCH` and `SPDK_MAKEFLAGS`. To know which values to set for all the supported Arm64 SoCs see the socs and implementer_xxx parts. E.g. for kunpeng920 SoC:

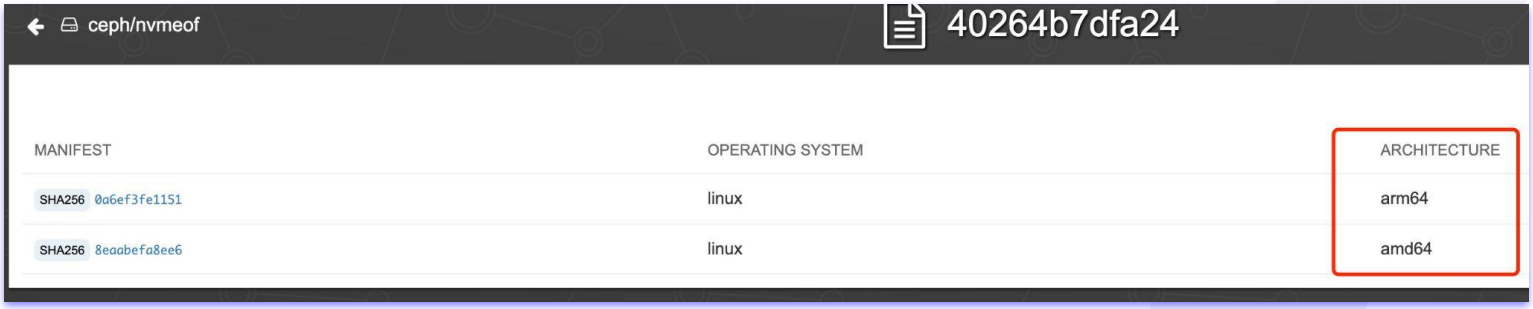
```
make build SPDK_TARGET_ARCH="armv8.2-a+crypto" \
    SPDK_MAKEFLAGS="DPDKBUILD_FLAGS=-Dplatform=kunpeng920"
```

Ceph NVMe-oF Gateway Arm Upstream Support

linaro Connect

- CI – Add build-arm64 job, multi-platform build
  - Ticket: https://github.com/ceph/ceph-nvmeof/issues/805
  - PR: https://github.com/ceph/ceph-nvmeof/pull/806
    - Migrate docker-compose from v1 to v2.
    - Support qemu multi-platform container image build.
    - Add build-arm64 job run per PR and daily.
    - Build and publish nvmeof, nvmeof-cli images to quay.io/ceph/.





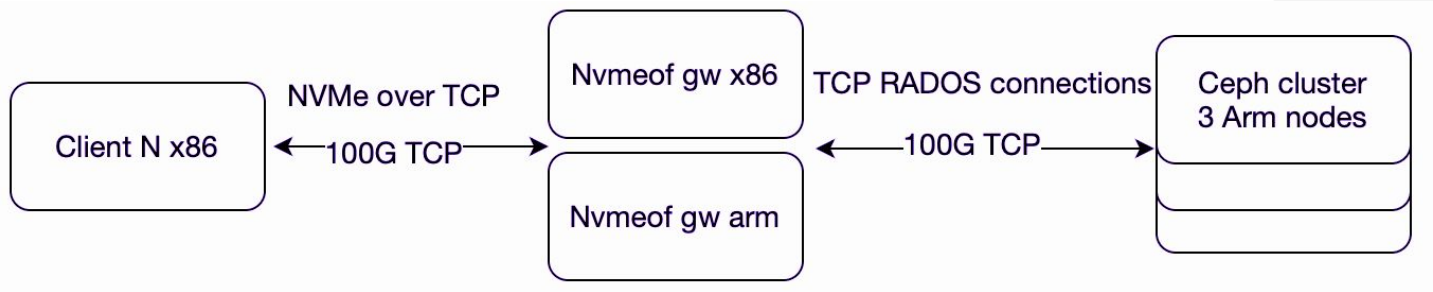
Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Testbed

- Hardware
  - Arm CPU: Kunpeng 920, 2.6GHz, 96 cpus, 4 numa nodes
  - X86 (Gateway) CPU: Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 112 cpus, 2 numa nodes
  - x86-2 (Client) CPU: Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz, 96 cpus, 2 numa nodes
  - Disk: 3 x ES3000 V6 NVMe SSD 3.2T per Arm server
  - Network: 1x MLNX ConnectX-5 100Gb IB，1x1G tcp
- Software
  - OS: openEuler 22.03 LTS SP3，kernel 5.10.0-192.0.0.105.oe2203sp3
  - **Ceph**: main-nvmeof require revert commit "nvmeof gw monitor: disable by default"
  - SPDK: 24.05 , nvmeof: master-1.3.2, fio: fio-3.29



Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

- FIO rbd config

```
(.venv) [root@client1 spdktest]# cat fio_test-rbd.conf
[global]
#stonewall
description="Run ${RW} ${BS} rbd test"
bs=${BS}
ioengine=rbd
clientname=admin
pool=nvmeof
thread=1
group_reporting=1
direct=1
verify=0
norandommap=1
time_based=1
ramp_time=10s
runtime=1m
iodepth=${IODEPTH}
rw=${RW}
numa_cpu_nodes=0


[test-job1]
rbdname=fio_test_image1
```

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

- FIO nvmeof config

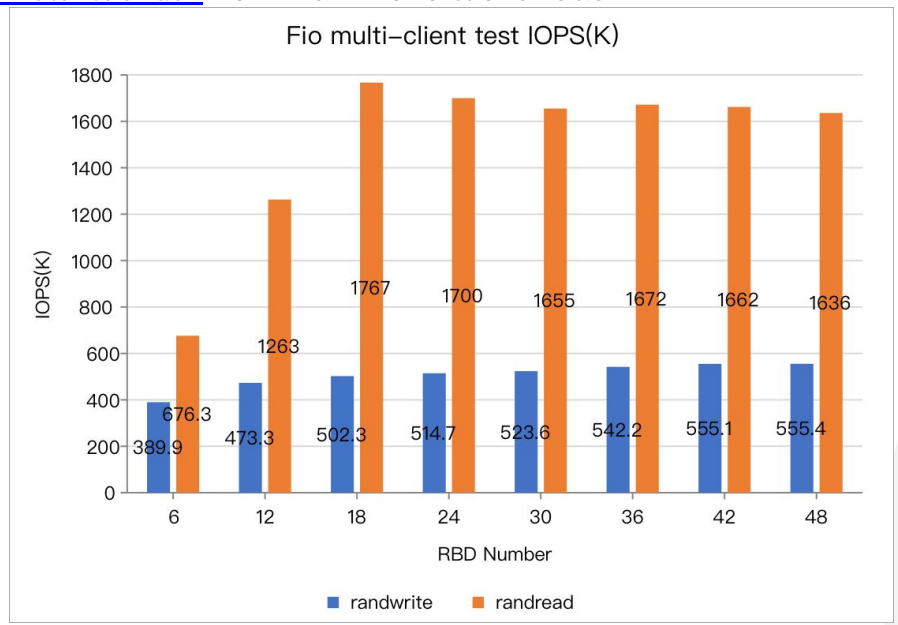
```
(.venv) [root@client1 spdktest]# cat fio_test-nvmeof.conf
[global]
#stonewall
description="Run ${RW} ${BS} NVMe ssd test"
bs=${BS}
#ioengine=libaio
ioengine=io_uring
thread=1
group_reporting=1
direct=1
verify=0
norandommap=1
time_based=1
ramp_time=10s
runtime=1m
iodepth=${IODEPTH}
rw=${RW}
numa_cpu_nodes=0

[test-job1]
#filename=/dev/nvme2n1
filename=/dev/nvme2n2
```

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Case 1: fio rbd multi-client test (48 rbds, numjob=1, client_num=6)

- Test randread/write 4K IOPS capability for backend ceph cluster
- **./fio –server&** # run in the clients nodes
- ./run-n-fio-rbd-test_multi-client.sh 48 # run in one client node



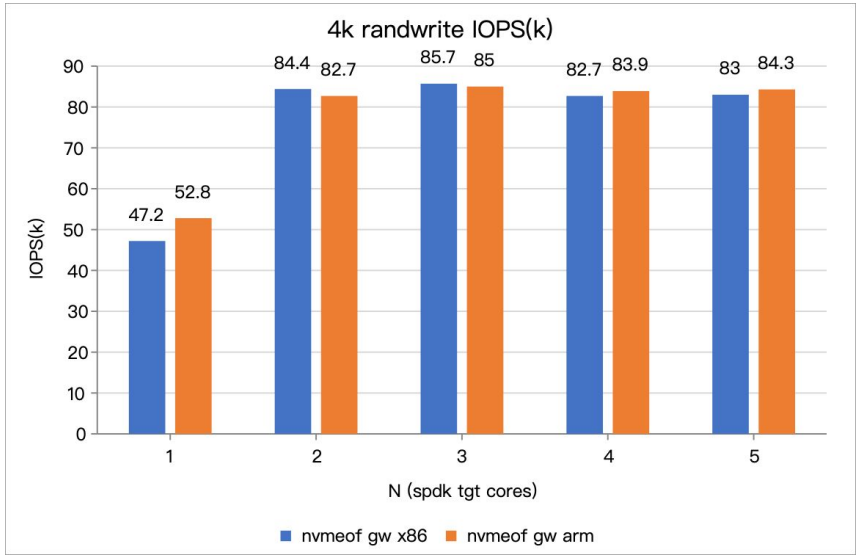
Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

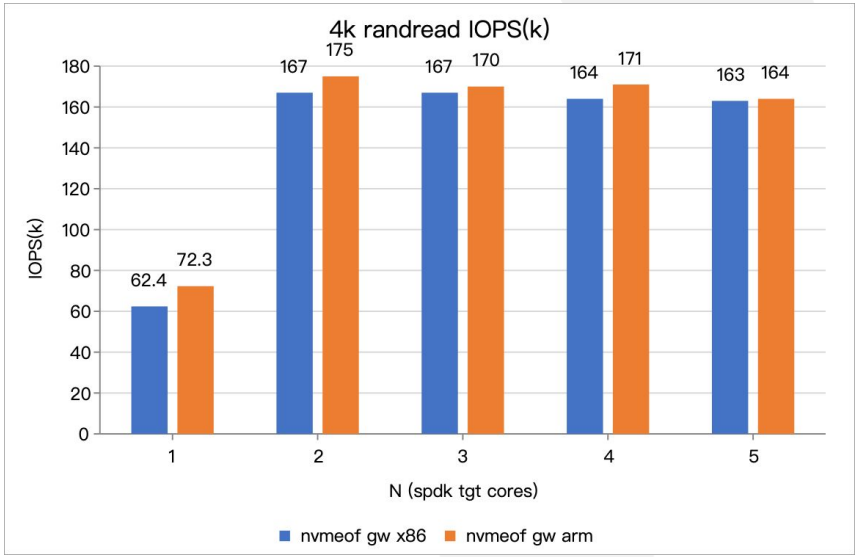
linaro Connect

Case 2: single rbd and scale spdk cores test (numjobs=8)

BS=4k RW=randwrite|randread IODEPTH=128 fio ~/spdktest/fio_test-nvmeof.conf --numjobs=8

- Test randread/write 4K IOPS capability for single core and single rbd
- ~50k IOPS per core for randwrite, ~60-70k IOPS for randread



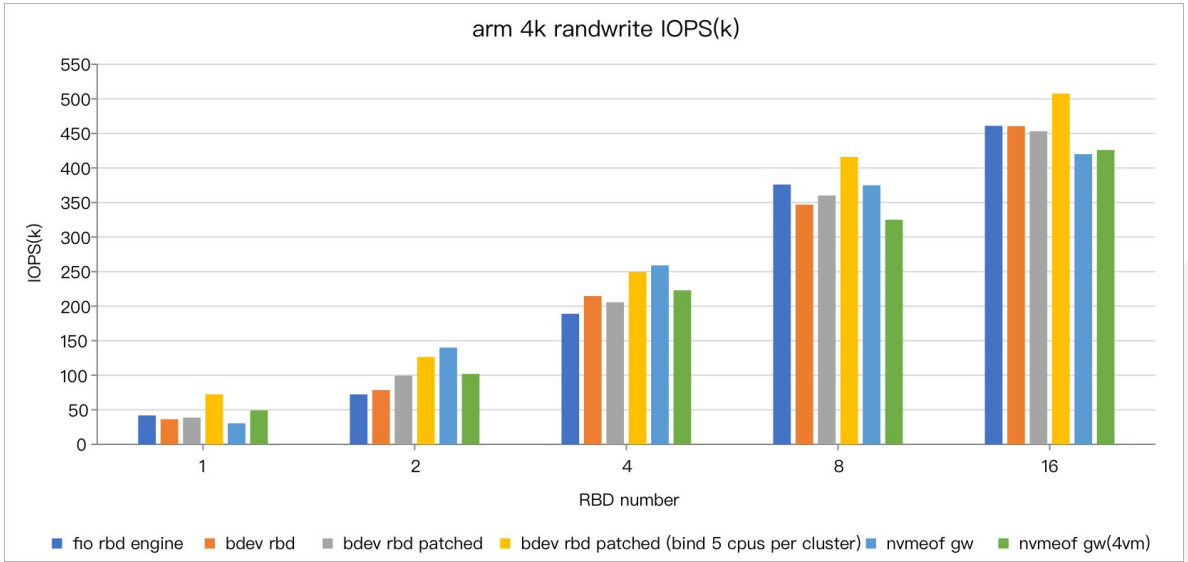


Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Case 3: multi-rbd fio, bdev, gw test (numjob=1, 16 spdk cores, bdevs_per_cluster = 1, iommu.passthrough=1)

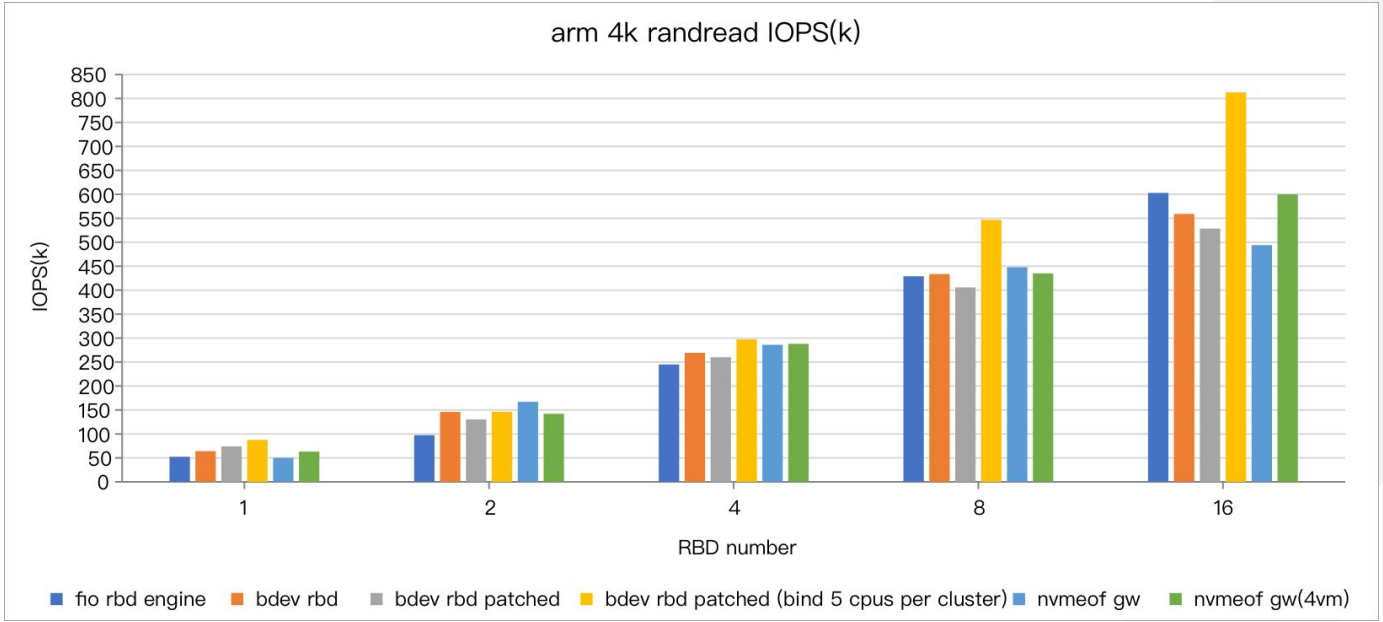
- Test randread/write 4K IOPS for fio rbd, bdev rbd, gateway
- bdevperf -q 128 -o 4096 -w randwrite -t 60 -m [0-15] -z # run in one terminal
- ./run-n-bdev-rbd-test1.sh N , ./run-n-fio-nvmeof-test1.sh N, ./run-n-fio-rbd-test1.sh N,
- ./run-n-bdev-rbd-test2.sh N 16-95 # each ceph client threads binding to 5 cpu cores



Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Case 3: multi-rbd fio, bdev, gw test (numjob=1, 16 spdk cores, bdevs_per_cluster = 1, iommu.passthrough=1)

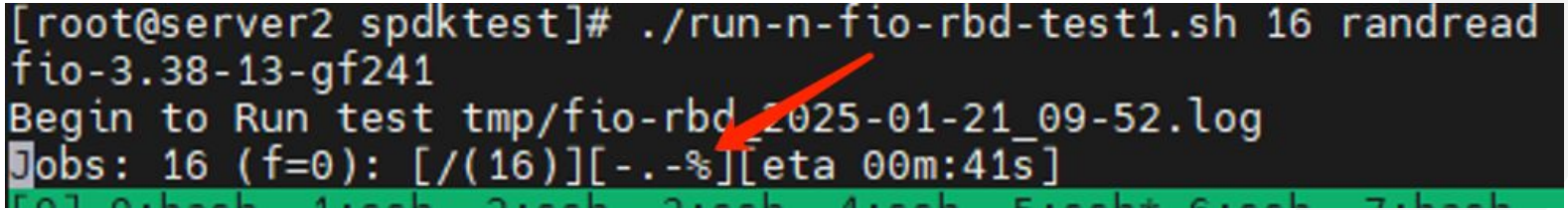


Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Issue 1: Fio rbd engine can only run small number of jobs

- Upstream tracking issue: https://github.com/axboe/fio/issues/1778
- When running more jobs it will get stuck and no any fio test is running
- **Workaround**: running in a container environment without this issue
- podman run -it --pids-limit 4096 --name bdevtest centos-stream9

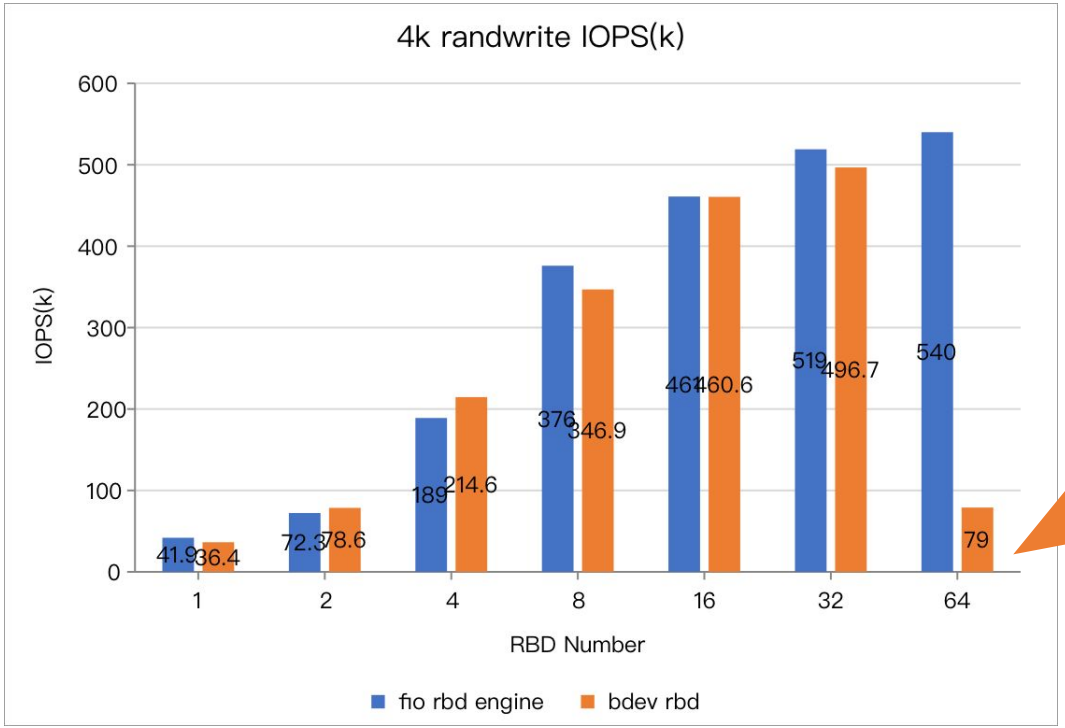


Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Issue 2: 4k randwrite/randread IOPS drops rapidly as the RBD number grows large

- Upstream tracking issue: https://github.com/spdk/spdk/issues/3547



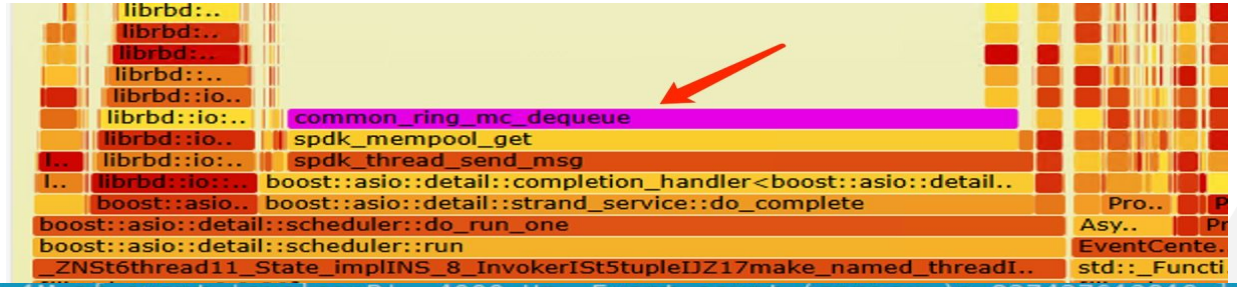
Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Issue 2: 4k randwrite/randread IOPS drops rapidly as the RBD number grows large

Bottleneck analysis

- The perf top and flame graph show that **common_ring_mc_dequeue**() calls spend too much time.
- io_context_pool threads are non-spdk threads and non-spdk thread lacks of [local caching]
- Through [DPDK programmer's guide] knows that lacking of local cache, **rte_mempool_get**() performance will suffer when multiple cores accessing.




```
Samples: 1M of event 'cycles:P', 4000 Hz, Event count (approx.): 827427613819 lost: 0/0 drop: 0/0
Overhead  Shared Object                 Symbol
  68.16%  bdevperf                      [.] common_ring_mc_dequeue
   5.14%  bdevperf                      [.] spdk_ring_dequeue
   2.17%  bdevperf                      [.] spdk_thread_poll
   1.91%  [kernel]                      [k] finish_task_switch.isra.0
   1.86%  bdevperf                      [.] reactor_run
   1.70%  bdevperf                      [.] thread_poll
   1.25%  bdevperf                      [.] msg_queue_run_batch
   0.88%  libceph-common.so.2           [.] __aarch64_ldadd8_acq_rel
   0.66%  libc.so.6                     [.] _int_free
   0.63%  [kernel]                      [k] handle_softirqs
```

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

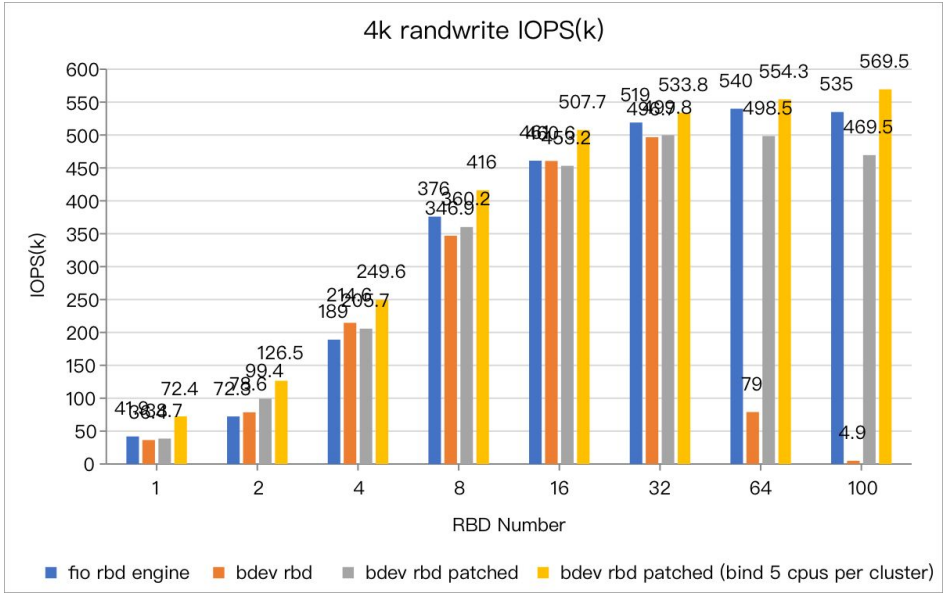
Issue 2: 4k randwrite/randread IOPS drops rapidly as the RBD number grows large

Possible fix ways

1. Revert commit “62210eff55bd bdev/rbd: Remove epoll based group polling mechanism.”

2. Use **calloc**() instead of **rte_mempool_get**() mempool

- https://review.spdk.io/c/spdk/spdk/+/25733 (reviewing)



Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 1: Arm and x86 hybrid deployment
  - **ceph config set mgr mgr/cephadm/use_repo_digest false # use image by name not digest**

```
# Deploy Ceph Cluster, mix arches x86 and arm64
wget -c https://download.ceph.com/rpm-reef/el9/noarch/cephadm
chmod +x cephadm
cephadm --image quay.io/xin3liang0/ceph:main-nvmeof  bootstrap --mon-ip 192.168.0.86
cephadm shell
ceph config set mgr mgr/cephadm/use_repo_digest false
ceph config set global container_image quay.io/xin3liang0/ceph:main-nvmeof
cephadm shell
ceph orch host add server5
ceph orch host add server1
# osds_per_device: 8 # Run more osd daemons to utilize a nvme disk
ceph orch apply -i osd_spec.yml
```

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 2: SSD rbd preconditioning
  - **Empty** rbd read IOPS is fake
  - Sequential write 2X+ size of rbd to get SSD steady state.
    - RW=write BS=128k IODEPTH=128 fio ./[fio_test-rbd.conf|fio_test-nvmeof.conf] --numjobs=1 # run 1 min
  - Reference
    - SSD Preconditioning: https://ci.spdk.io/download/performance-reports/SPDK_nvme_bdev_perf_report_2405.pdf

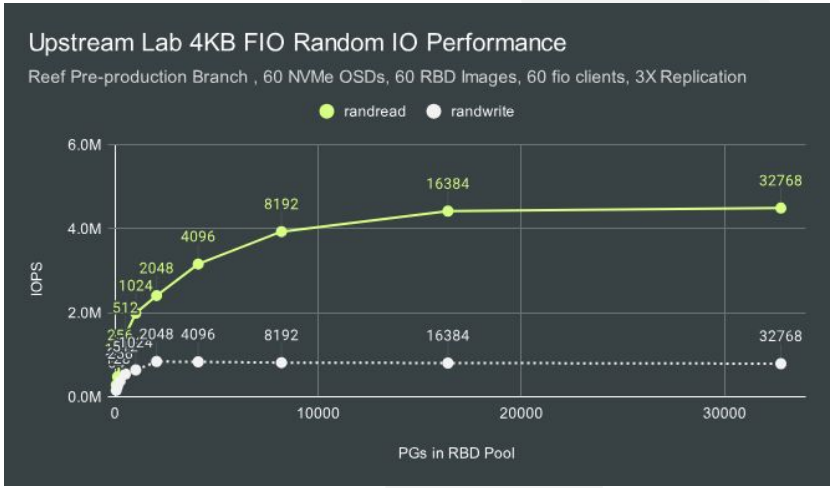
```
[ceph: root@server6 ~]# rbd -p nvmeof disk-usage
NAME                 PROVISIONED  USED
fio_test_image1          20 GiB  20 GiB
fio_test_image2          20 GiB     0 B
rbd1                     20 GiB  20 GiB
rbd2                     20 GiB  20 GiB
<TOTAL>                  80 GiB  60 GiB
[ceph: root@server6 ~]#
```

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 3: Increase Ceph pool pg_num for performance test
  - **Pg_num** impacts performance too much
  - Default pg_autoscale_mode scales pg_num according pool space utilization
  - Setting large **pg_num** at the pool creation beginning for performance test
    - pool=nvmeof; ceph osd pool create $pool --size 1 --pg_num 16384 --pgp_num 16384 --bulk
    - ceph osd pool set $pool target_size_ratio 0.9
    - ceph osd pool set $pool pg_autoscale_mode off
    - rbd pool init $pool
  - References
    - Ceph: A Journey to 1 TiB/s
    - PG Calc — Ceph Documentation
    - Placement Groups — Ceph Documentation



Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 4: Nvmeof-gateway should build release type
  - Upstream tacking card: https://github.com/ceph/ceph-nvmeof/issues/938
  - SPDK_CONFIGURE_ARGS="--with-rbd --disable-tests --disable-unit-tests --disable-examples **--enable-debug**"
  - Debug build type performance is pool.

- Tip 5: Build LSE atomic instructions for Armv8.2+ machines
  - **Check LSE**: perf top see __aarch64_xxx_xxx_rel, e.g. __aarch64_ldadd8_acq_rel
  - **Check LSE**: objdump and egrep -i 'cas|casp|swp|ldadd|stadd|ldclr|stclr|ldeor|steor|ldset|stset|ldsmax|stsmax|ldsmin|stsmin|ldumax|stumax|ldumin|stumin'
  - **Check load exclusives and store exclusives**: objdump and egrep -i 'ldxr|ldaxr|stxr|stlxr'
  - Refer to: https://learn.arm.com/learning-paths/servers-and-cloud-computing/lse/example/

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 6: Set passthrough iommu
  - iommu overhead is non-negligible
  - Disable or passthrough it
    - kernel cmdline: **iommu.passthrough**=1(arm64), iommu=pt(Intel)

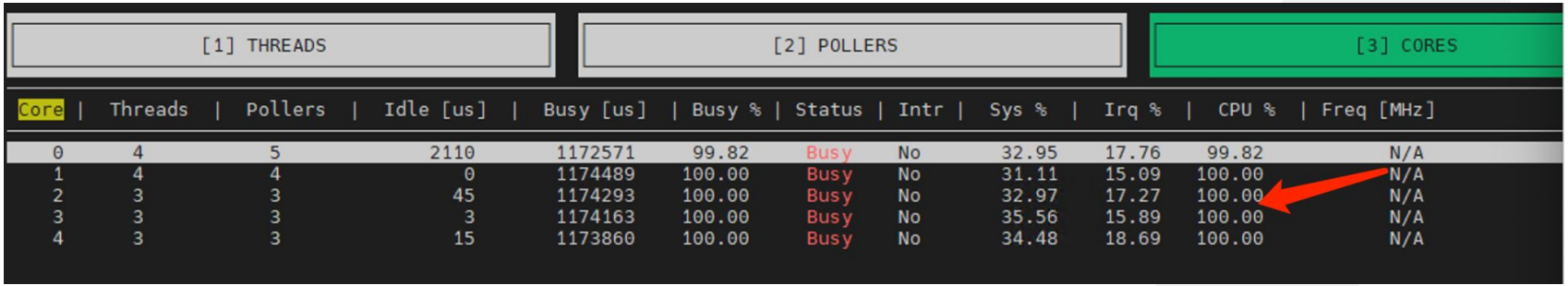
- Tip 7: Set nvmeof controller IO queue number >= SPDK CPU cores number (but not too larger than)
  - ceph-nvmeof.conf: "**max_io_qpairs_per_ctrlr**": 7
  - To utilize all the spdk cores and get best performance.

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 8: Scale spdk cores
  - Add spdk cores when spdk_top cpu% is full
  - ceph-nvmeof.conf: **tgt_cmd_extra_args** = “-m 0xFF” #default is 0xF
  - Might need to increase iobuf pool
    - RPC=/usr/libexec/spdk/scripts/rpc.py
    - bdevperf -q 128 -o 4096 -w randwrite -t 60 -m [0-15] -z --wait-for-rpc
    - $RPC iobuf_set_options --small-pool-count 32767 --large-pool-count 8191 && $RPC framework_start_init



Core	Threads	Pollers	Idle [us]	Busy [us]	Busy %	Status	Intr	Sys %	Irq %	CPU %	Freq [MHz]
0	4	5	2110	1172571	99.82	Busy	No	32.95	17.76	99.82	N/A
1	4	4	0	1174489	100.00	Busy	No	31.11	15.09	100.00	N/A
2	3	3	45	1174293	100.00	Busy	No	32.97	17.27	100.00	N/A
3	3	3	3	1174163	100.00	Busy	No	35.56	15.89	100.00	N/A
4	3	3	15	1173860	100.00	Busy	No	34.48	18.69	100.00	N/A

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

Deployment, build, test, tuning tips

- Tip 9: Tune Ceph client thread number
  - ceph-nvmeof.conf: : **bdevs_per_cluster** = 1 as fio rbd engine
    - https://github.com/ceph/ceph-nvmeof/issues/939
  - ceph.conf
    - **ms_async_op_threads**=3 is default, msgr-worker-x thread number
    - **librados_thread_count**=2 is default, io_context_pool thread number
    - Tune if top cpu% is near 100%

- Tip 10: Bind Ceph client threads
  - Binding cpu could get better performance
  - ceph-nvmeof.conf: **librbd_core_mask** = 0xFFFF0000 #avoid using spdk cores
  - bdevperf test script run-n-bdev-rbd-test2.sh
    - $RPC bdev_rbd_register_cluster $cluster --core-mask "$cpu_list"
    - cpus_per_cluster=5

Ceph SPDK NVMe-oF Gateway Performance Test and Analysis

linaro Connect

- Summary
  - Arm and x86 can hybrid deploy
  - Ceph main requires revert commit "nvmeof gw monitor: disable by default".
  - Ceph nvmeof gateway requires rebuilding to non-debug release.
  - Build LSE atomic instructions for Arm64.
  - RBD Preconditioning: sequential write 2X+ size of rbd to get SSD steady state.
  - Add spdk cores when spdk_top cpu% is full
  - Bind and tune Ceph client, spdk tgt threads, **bdevs_per_cluster**=1.
  - DPDK mempool is not working well for many non-spdk threads.
  - Useful Links
    - https://spdk.io/cn/articles/
    - https://spdk.io/doc/
    - https://www.youtube.com/@storageperformancedevelopm301


linaro
Connect
2025
Thank You!