Driver in SPDK

Jin Yu intel

Environment setup

❑ Hardware:

NVMe SSD / remote NVMF_TGT.

❑ Software

CURRENT_DIR = $(SPDK_ROOT_DIR)

scripts:

./scripts/setup.sh

Environment setup

- ./scripts/setup.sh :
  - ✓ Driver_override
  - ✓ Hugepage
  - ✓ Status
- Example：
  - ✓ HUGEMEM=4096 \

    ./scripts/setup.sh

```
[config|reset|status|cleanup|help] - as following:
config            Default mode. Allocate hugepages and bind PCI devices.
cleanup            Remove any orphaned files that can be left in the system after SPDK application exit
reset             Rebind PCI devices back to their original drivers.
                  Also cleanup any leftover spdk files/resources.
                  Hugepage memory size will remain unchanged.
status            Print status of all SPDK-compatible devices on the system.
help              Print this help message.

The following environment variables can be specified.
HUGEMEM           Size of hugepage memory to allocate (in MB). 2048 by default.
                  For NUMA systems, the hugepages will be evenly distributed
                  between CPU nodes
NRHUGE            Number of hugepages to allocate. This variable overwrites HUGEMEM.
HUGENODE          Specific NUMA node to allocate hugepages on. To allocate
                  hugepages on multiple nodes run this script multiple times -
                  once for each node.
PCI_WHITELIST
PCI_BLACKLIST     Whitespace separated list of PCI devices (NVMe, I/OAT, VMD, Virtio).
                  Each device must be specified as a full PCI address.
                  E.g. PCI_WHITELIST="0000:01:00.0 0000:02:00.0"
                  To blacklist all PCI devices use a non-valid address.
                  E.g. PCI_WHITELIST="none"
                  If PCI_WHITELIST and PCI_BLACKLIST are empty or unset, all PCI devices
                  will be bound.
                  Each device in PCI_BLACKLIST will be ignored (driver won't be changed).
                  PCI_BLACKLIST has precedence over PCI_WHITELIST.
TARGET_USER       User that will own hugepage mountpoint directory and vfio groups.
                  By default the current user will be used.
DRIVER_OVERRIDE   Disable automatic vfio-pci/uio_pci_generic selection and forcefully
                  bind devices to the given driver.
                  E.g. DRIVER_OVERRIDE=uio_pci_generic or DRIVER_OVERRIDE=/home/public/dpdk/build/kmod/igb_uio.ko
```

Perf analysis

- Example Path:

  examples/nvme/perf/

- Usage:

./perf -q 1 -o 4096 -w randrw -M 70 \

-t 600 -c 0xf –r “trtype:PCIe

traddr: 0000:04:00.0”

```
./perf options
        [-q io depth]
        [-o io size in bytes]
        [-n number of io queues per namespace. default: 1]
        [-U number of unused io queues per controller. default: 0]
        [-w io pattern type, must be one of
                (read, write, randread, randwrite, rw, randrw)]
        [-M rwmixread (100 for reads, 0 for writes)]
        [-L enable latency tracking via sw, default: disabled]
                -L for latency summary, -LL for detailed histogram
        [-l enable latency tracking via ssd (if supported), default: disabled]
        [-t time in seconds]
        [-c core mask for I/O submission/completion.]
                (default: 1)
        [-D disable submission queue in controller memory buffer, default: enabled]
        [-H enable header digest for TCP transport, default: disabled]
        [-I enable data digest for TCP transport, default: disabled]
        [-r Transport ID for local PCIe NVMe or NVMeoF]
         Format: 'key:value [key:value] ...'
         Keys:
          trtype      Transport type (e.g. PCIe, RDMA)
          adrfam      Address family (e.g. IPv4, IPv6)
          traddr      Transport address (e.g. 0000:04:00.0 for PCIe or 192.168.100.8 for RDMA)
          trsvcid     Transport service identifier (e.g. 4420)
          subnqn      Subsystem NQN (default: nqn.2014-08.org.nvmexpress.discovery)
         Example: -r 'trtype:PCIe traddr:0000:04:00.0' for PCIe or
                  -r 'trtype:RDMA adrfam:IPv4 traddr:192.168.100.8 trsvcid:4420' for NVMeoF
        [-e metadata configuration]
         Keys:
          PRACT      Protection Information Action bit (PRACT=1 or PRACT=0)
          PRCHK      Control of Protection Information Checking (PRCHK=GUARD|REFTAG|APPTAG)
         Example: -e 'PRACT=0,PRCHK=GUARD|REFTAG|APPTAG'
                  -e 'PRACT=1,PRCHK=GUARD'
        [-k keep alive timeout period in millisecond]
        [-s DPDK huge memory size in MB.]
        [-m max completions per poll]
                (default: 0 - unlimited)
        [-i shared memory group ID]
        [-V enable VMD enumeration]
        [-G enable debug logging]
```

Perf analysis

- Initialize environment

spdk_env_opts_init(&opts);

spdk_env_init(&opts)

- Register driver:

spdk_nvme_probe(trid, cb_ctx, probe_cb, attach_cb, remove_cb)

How the NVMe works

NVMe:

- ❑ Namespace
- ❑ Queue pair

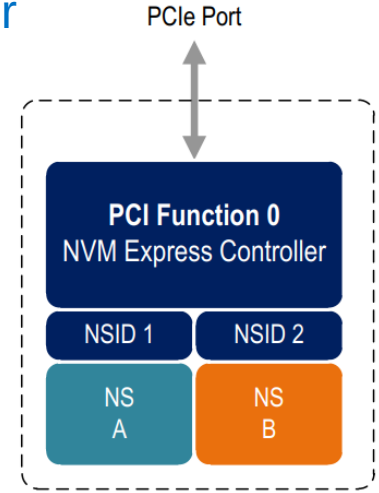
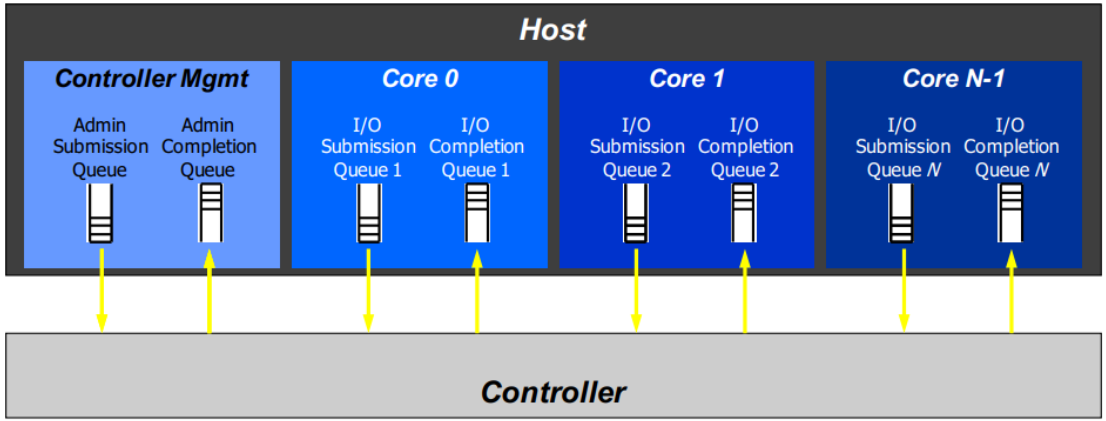


Figure 1: Queue Pair Example, 1:1 Mapping



Queue pair

❑ Functions:

spdk_nvme_ctrlr_alloc_io_qpair()

spdk_nvme_ctrlr_free_io_qpair()

spdk_nvme_qpair_process_completions（）

❑Lib:

Nvme_ctrlr.c

Nvme_qpair.c

Namespace

❑ Functions:

spdk_nvme_ns_cmd_read_with_md

spdk_nvme_ns_cmd_write_with_md

❑ Lib：

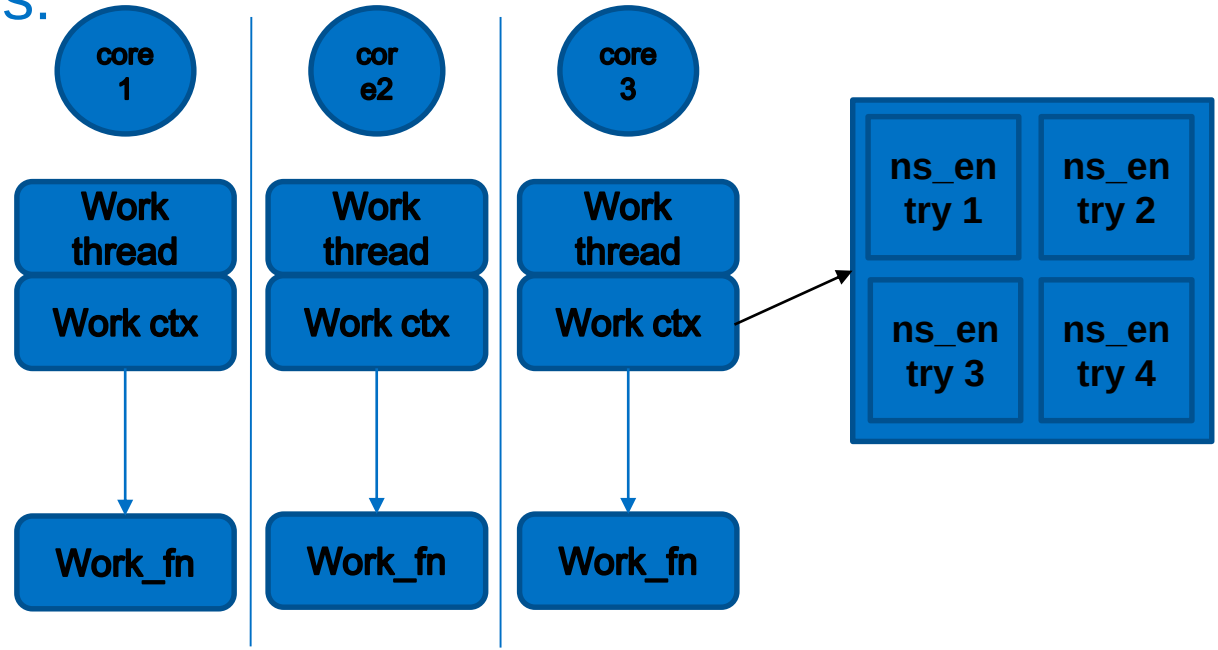
nvme_ns.c

nvme_ctrlr_cmd.c

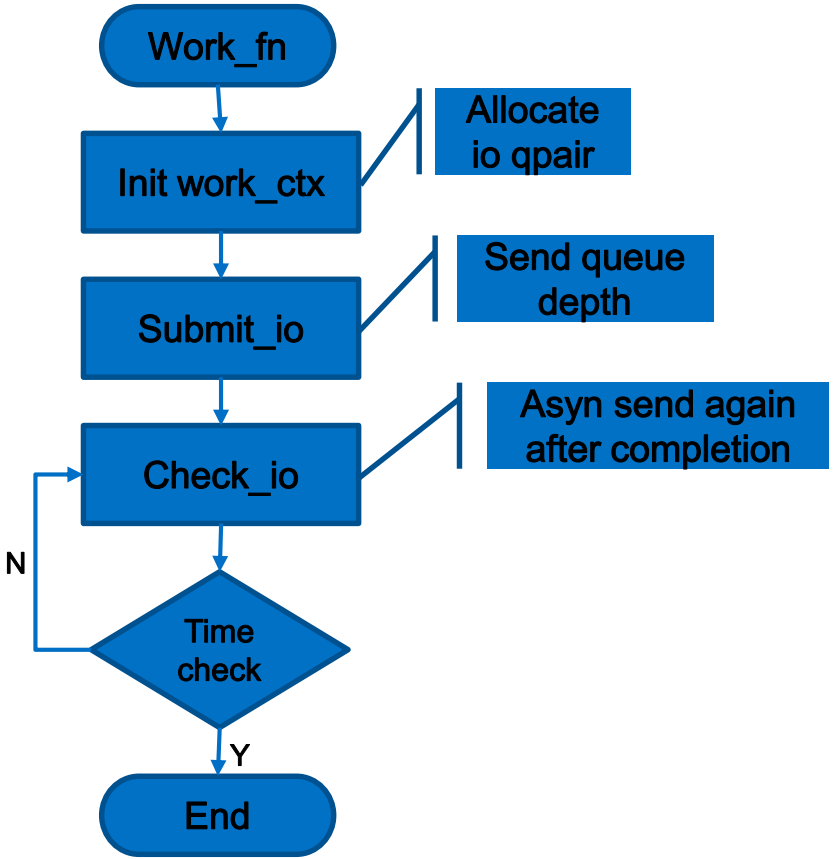
Perf analysis

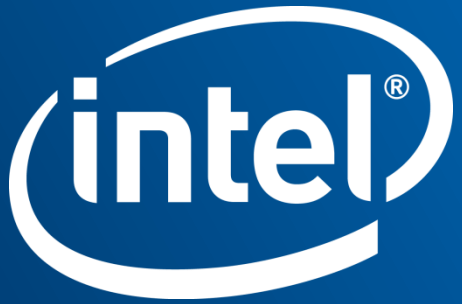
Core and NVMe SSDs:

- ❑ Parallel
- ❑ No lock
- ❑ Polling



Perf work flow




intel®

BACKUP

How to identify NVMe disks in userspace

Identify app path:

- ✓ examples/nvme/identify

Usage:

- ✓ ./identify

```
./identify [options]
options:
-r trid    remote NVMe over Fabrics target address
   Format: 'key:value [key:value] ...'
   Keys:
    trtype      Transport type (e.g. RDMA)
    adrfam      Address family (e.g. IPv4, IPv6)
    traddr      Transport address (e.g. 192.168.100.8)
    trsvcid     Transport service identifier (e.g. 4420)
    subnqn      Subsystem NQN (default: nqn.2014-08.org.nvmexpress.discovery)
   Example: -r 'trtype:RDMA adrfam:IPv4 traddr:192.168.100.8 trsvcid:4420'
-L, --logflag <flag>    enable debug log flag (all, log, nvme, opal, thread, vmd)
-i         shared memory group ID
-p         core number in decimal to run this application which started from 0
-d         DPDK huge memory size in MB
-x         print hex dump of raw data
-v         verbose (enable warnings)
-V         enumerate VMD
-H         show this usage
```

How to manage NVMe disks in userspace

NVMe_manage app path:

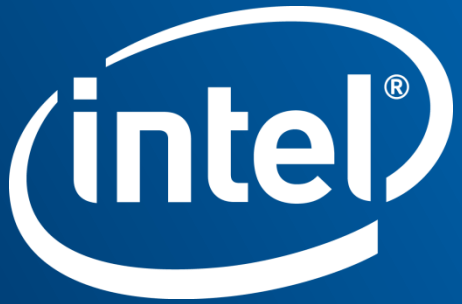
- ✓ ./examples/nvme/nvme_manage

Usage and options:

- ✓ ./nvme_manage

```
NVMe Management Options
        [1: list controllers]
        [2: create namespace]
        [3: delete namespace]
        [4: attach namespace to controller]
        [5: detach namespace from controller]
        [6: format namespace or controller]
        [7: firmware update]
        [8: opal]
        [9: quit]
```

NOTICE: Should confirm hardware support these features


intel®