DPDK expands into Storage domain

FIONA TRAHE, DAREK STOJACZYK
SEPTEMBER 2019
agenda

- What is SPDK?
- cryptodev
- compressdev
- memory management
- PCI access
- vhost
- Wrap-up
What is SPDK?

Storage Performance Development Kit

Open Source Software
- Optimized for latest generation CPUs and SSDs
- Software building blocks (BSD licensed)
- Designed to extract maximum performance from non-volatile media

Scalable and Efficient Software Ingredients
- User space, lockless, polled-mode components
- Up to millions of IOPS per core
- Minimize average and tail latencies

Available via spdk.io
SPDK Community

- Email Discussions
- Weekly Calls
- Multiple Annual Meetups
- Code Reviews & Repo
- Real Time Chat w/ Development Community
- Backlog and Ideas for Things to Do
- Main Web Presence
- Continuous Integration

http://SPDK.IO
SPDK HIGH-LEVEL ARCHITECTURE*

* stripped to only the modules relevant to this presentation

Storage Protocols
- NVMeOF Target
- Block Device Abstraction Layer (bdev)
- Virtual bdevs
- RDMA
- TCP

Storage Services
- Encapsulation
- Compression
- Virtual bdevs

Drivers
- NVMe PCI driver
- PCI
- mem mgmt.

SPDK

DPDK

vhost
agenda

- What is SPDK?
- cryptodev
- compressdev
- memory management
- PCI access
- vhost
- Wrap-up
CRYPTO BDEV MODULE

Application
<SPDK Defined Block API>

Block Device Layer
VBDEV
VBDEV
BDEV

<Driver Specific API>

Drivers

DPDK
CryptoDev API
AESNI PMD
QAT PMD
QAT HW

Crypto Operations
CRYPTO BDEV MODULE

Application
<SPDK Defined Block API>

Block Device Layer
| VBDEV | Crypto |
| VBDEV |       |
| BDEV  | BDEV  |

<Driver Specific API>

Drivers

DPDK
CrptoDev API
AESNI PMD
QAT PMD

QAT HW

I/O

1
CRYPTO BDEV MODULE

Application
<SPDK Defined Block API>

Block Device Layer
VBDEV
VBDEV
BDEV

Drivers

<Driver Specific API>

I/O

1

2

I/O

Crypto Operations

DPDK

CryptoDev API

AESNI
PMD

QAT
PMD

QAT
HW

I/O

Crypto Operations
CRYPTO BDEV MODULE

Application
<SPDK Defined Block API>

Block Device Layer
VBDEV
VBDEV
BDEV

Drivers

I/O

1

Crypto Operations

2

3

I/O

<Driver Specific API>

DPDK
CryptoDev API
AESNI PMD
QAT PMD
QAT HW

I/O
Cryptodev integration

• Uses cipher-only. No auth, no cipher-auth chaining.
• Cipher algorithm: AES-CBC
• AES-XTS – an algorithm commonly used for disk encryption has been implemented in cryptodev. On SPDK backlog to be integrated.
• Uses QAT PMD for hardware encryption and aesni-mb PMD for software encryption
agenda

- What is SPDK?
- cryptodev
- compressdev
- memory management
- PCI access
- vhost
- Wrap-up
COMPRESSION BDEV MODULE

Application
<SPDK Defined Block API>

Block Device Layer

- VBDEV
- VBDEV
- BDEV

<Driver Specific API>

Drivers

Persistent Memory

DPDK

compressdev API

ISA-L PMD

QAT PMD

QAT HW

I/O
VBDEV
VBDEV
BDEV

libReduce PMDK
Compression
BDEV

metadata device
metadata operations
NVMe device
I/O operations
**Compression Bdev Module**

**Block Device Layer**
- VBDEV
- VBDEV
- BDEV

**Application**
- <Defined Block API>
- I/O

**Drivers**
- libReduce
- PMDK
- Compression
- BDEV

**Persistent Memory**

**DPDK**
- compressdev API
- ISA-L PMD
- QAT PMD

**QAT HW**

**I/O**
- VBDEV
- VBDEV
- BDEV
COMPRESSION BDEV MODULE

Application

<Defined Block API>

I/O

Block Device Layer

1

VBDEV

libReduce

PMDK

2

Compression

Block Device Layer

VBDEV

BDEV

<Driver Specific API>

Drivers

Persistent Memory

Metadata

I/O

VBDEV

BDEV

DPDK

compressdev API

ISA-L PMD

QAT PMD

QAT HW

metadata device

metadata operations

NVMe device

I/O operations

Metadata
Compression BDEV Module

Application

Block Device Layer

Persistent Memory

DPDK

1. <Defined Block API>

2. Compression

3. <Driver Specific API>

I/O

VBDEV

libReduce

PMDK

BDEV

Metadata

Compression Operations

I/O Operations

ISA-L

PMD

QAT

PMD

NVMe device

compressdev API

Drivers

Metadata Operations

QAT HW
**Compression Bdev Module**

1. Application
   - SPDK Defined Block API
   - I/O

2. Block Device Layer
   - VBDEV
   - Compression
   - BDEV

3. DPDK
   - ISA-L PMD
   - QAT PMD
   - compressdev API

4. Drivers
   - <Driver Specific API>

Metadata

Compressed Data

Persistent Memory

Compression Operations
Compressdev integration

- More complicated than crypto
- Output data size is different to input data size and unpredictable.
- However one advantage over network compression use-cases is that the decompressed data size is known
- libReduce figures out which LBAs in the application i/f map to which LBAs on the backing device
- It uses persistent memory (PMDK) to store meta-data.
- QAT and ISAL PMDs are used for compression & decompression
What is SPDK?

- cryptodev
- compressdev
- memory management
- PCI access
- vhost

Wrap-up
Memory management

• spdk_malloc()

• spdk_mem REGISTER()
Memory management

• `spdk_malloc()` → calls `rte_malloc()`

• `spdk_mem_register()`
Memory management

- `spdk_malloc()` → calls `rte_malloc()`
- `spdk_mem_register()` → calls `rte_vfio_dma_map()`
Memory management

- `spdk_malloc()` → calls `rte_malloc()`
- `spdk_mem_register()` → calls `rte_vfio_dma_map()`

... also calls ibverbs APIs
Memory management

- spdk_malloc() → calls rte_malloc()
- spdk_mem_register() → calls rte_vfio_dma_map()

... also calls ibverbs APIs

... also shares the memory with connected vhost devices
agenda

- What is SPDK?
- cryptodev
- compressdev
- memory management
- PCI access
- vhost
- Wrap-up
PCI access

- spdk_pci_device_attach()

- spdk_pci_device_detach()
PCI access

- spdk_pci_device_attach() → can call rte_eal_hotplug_add() returns spdk_pci_device *

- spdk_pci_device_detach()
PCI access

- `spdk_pci_device_attach()` → can call `rte_eal_hotplug_add()` and returns `spdk_pci_device *`

- `spdk_pci_device_detach()` → calls `rte_eal_hotplug_remove()`
PCI access

- `spdk_pci_device_attach()` → can call `rte_eal_hotplug_add()` returns `spdk_pci_device` *

- `spdk_pci_device_detach()` → calls `rte_eal_hotplug_remove()`

- `rte_dev_event_callback_register()`
PCI access

- `spdk_pci_device_attach()`  →  can call `rte_eal_hotplug_add()`
  returns `spdk_pci_device` *

- `spdk_pci_device_detach()`  →  calls `rte_eal_hotplug_remove()`

- `rte_dev_event_callback_register()`

  →  sets `spdk_pci_device->removed = 1`
agenda

- What is SPDK?
- cryptodev
- compressdev
- memory management
- PCI access
- vhost
- Wrap-up
Vhost

- DPDK’s library for creating and polling vhost devices
- Originally created for vhost-net
- Implements mostly device-agnostic vhost-user protocol
- SPDK uses it for storage
rte_vhostExtern_callback_register()

- Hooks a function to be called on each vhost message.
- Allows overriding default rte_vhost message handling
What is SPDK?

- cryptodev
- compressdev
- memory management
- PCI access
- vhost

Wrap-up
Questions?
Fast-track to relevant SPDK urls

- The SPDK project is here: spdk.io
- SPDK codebase: github.com/spdk/spdk
- Docs describing crypto & compression bdevs and compression design
  spdk.io/doc/bdev.html
  spdk.io/doc/reduce.html