Use DPDK to accelerate data compression in SPDK

FIONA TRAHE, INTEL
PAUL LUSE, INTEL
JIM HARRIS, INTEL
Agenda

- Overview of DPDK and its compressdev infrastructure
- Overview of SPDK and how it uses DPDK
- Creating a compression device with SPDK’s new reducelib, vbdev and compressdev
DPDK
compressdev
overview
<table>
<thead>
<tr>
<th>Category</th>
<th>Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core libraries</td>
<td>Core functions such as memory management, software rings, timers, bus/device mgmt, etc.</td>
</tr>
<tr>
<td>Packet classification</td>
<td>Software libraries for hash/exact match, LPM, ACL etc.</td>
</tr>
<tr>
<td>Accelerated SW libraries</td>
<td>Common functions such as IP fragmentation, reassembly, reordering etc.</td>
</tr>
<tr>
<td>Stats</td>
<td>Libraries for collecting and reporting statistics.</td>
</tr>
<tr>
<td>QoS</td>
<td>Libraries for QoS scheduling and metering/policing.</td>
</tr>
<tr>
<td>Packet Framework</td>
<td>Libraries for creating complex pipelines in software.</td>
</tr>
<tr>
<td>Device APIs</td>
<td>ETHDEV: PMDs for physical and virtual Ethernet devices</td>
</tr>
<tr>
<td></td>
<td>CRYPTODEV: PMDs for HW and SW crypto accelerators</td>
</tr>
<tr>
<td></td>
<td>EVENTDEV: Event-driven PMDs</td>
</tr>
<tr>
<td></td>
<td>SECURITY: Hardware acceleration APIs for security protocols</td>
</tr>
<tr>
<td></td>
<td>COMPRESSDEV: PMDs for HW and SW compression accelerators</td>
</tr>
<tr>
<td></td>
<td>BBDEV: PMDs for HW and SW wireless accelerators</td>
</tr>
</tbody>
</table>
# dpdk/compressdev key features

<table>
<thead>
<tr>
<th><strong>Asynchronous burst API</strong></th>
<th><strong>Chained mbufs</strong></th>
<th><strong>compression Algorithms</strong></th>
<th><strong>Compression Levels</strong></th>
<th><strong>Checksum</strong></th>
<th><strong>Hash Generation</strong></th>
</tr>
</thead>
</table>
| To support HW & SW acceleration | To allow compression of data greater than 64K-1. Can attach external data buffers to mbufs. | • Deflate  
  • LZS | 1: PMD default  
  1: Fastest  
  9: Best Ratio | o CRC32  
  o Adler32  
  o Combined Adler32_CR_C32 | • SHA1  
  • SHA256 |
compressdev components

Compression Application

DPDK

DPDK Compression API Components
- Device Mgmt
- Operation Mgmt
- Data-path enq/dequeue
- Stream Mgmt
- Capabilities

ISA-L
libisal.a

ZLIB
libzlib.a

QAT
QAT H/W

Octeontx

Poll Mode Drivers

Software libraries

Hardware Accelerators
Stateless compression with SGLs

• A chunk passed in or out of an operation may be comprised of one or more buffers (segments) chained together.
• Segments can be any size < 64k.
• There is no correlation between the number of segments passed in for compression and the number of segments it will decompress to.
Find out more about compressdev

- **compressdev home on dpdk.org**
  
  http://doc.dpdk.org/guides/prog_guide/compressdev.html

- **compressdev poll mode drivers**
  
  https://doc.dpdk.org/guides/compressdevs/index.html

- **Deflate your data with DPDK – presentation from DPDK Dublin summit**
  
SPDK overview
What is SPDK?

Storage Performance Development Kit

Available via spdk.io

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
SPDK Community

Main Web Presence

Email Discussions

Weekly Calls
Multiple Annual Meetups

Real Time Chat w/
Development Community

Backlog and
Ideas for Things to Do

Code Reviews & Repo

Continuous Integration

http://SPDK.IO

GerritHub

Jenkins
SPDK High Level Architecture*

*simplyfied to just the modules relevant to this presentation

Storage Protocols
- NVMe-oF* Target
  - RDMA
  - TCP
  - NVMe

Storage Services
- Block Device Abstraction (bdev)
  - Virtual Modules
    - DPDK Encryption
    - DPDK Compression
    - reduceLib
    - PMDK
  - Base Modules

Drivers
- NVMe Driver

Env Abstraction

DPDK Provides:
- Mem allocation
- PCI enumeration
- Threads
- Lock-Free Rings
- Etc...

Released
In Progress

DPDK

PMDK
reduceLib
Creating a compression device in SPDK
The Compression Vbdev module

**Application**

<SPDK Defined Block API>

**Block Device Layer**

- VBDEV
  - libReduce
  - PMDK
- VBDEV
  - Compression
- BDEV
  - BDEV

**Drivers**

**Persistent Memory**

metadata device
metadata operations

data device
compress/decompress operations

**DPDK**

- compressdev API
- ISA-L PMD
- QAT PMD

**QAT HW**

NVMe device
I/O operations
The Crypto Vbdev Module

Application
<SPDK Defined Block API>

Block Device Layer
- VBDEV
- VBDEV
- BDEV

<Driver Specific API>

Drivers

I/O

Crypto Operations

DPDK
- CryptoDev API
- AESNI PMD
- QAT PMD

QAT HW

Crypto Operations
Find out more

• SPDK home
  https://spdk.io/

• The secret to customizing SPDK: (all about Virtual BDEVs)
  https://www.snia.org/sites/default/files/SDC/2018/presentations/SSS_NVM_PM_NVDIMM/Luse_P_Trahe_F_Virtual_BDEVs_The_Secret_to_Customizing_SPDK.pdf

• PMDK - persistent memory development kit
  https://ci.spdk.io/download/events/2018-summit/day1_10_LusePMDKSPDK.pdf

Get involved in the development

• compressdev vbdev (WIP)
  https://review.gerrithub.io/c/spdk/spdk/+/429395

• libreduce design (WIP)
  https://review.gerrithub.io/c/spdk/spdk/+/430385