Inspire the N

End-to-End Data Protection with SPDK

04/16/2019

Shuhei Matsumoto

SPDK Core Maintainer

IT Platform Products Management Division

Hitachi, Ltd.

Contents

1. Introduction

End-to-End Data Protection

- Data corruption can occur anywhere and silent data corruption must be avoided.
- Data Integrity Field (DIF) provides a standardized end-to-end data protection mechanism that spans transport and protocol boundaries.

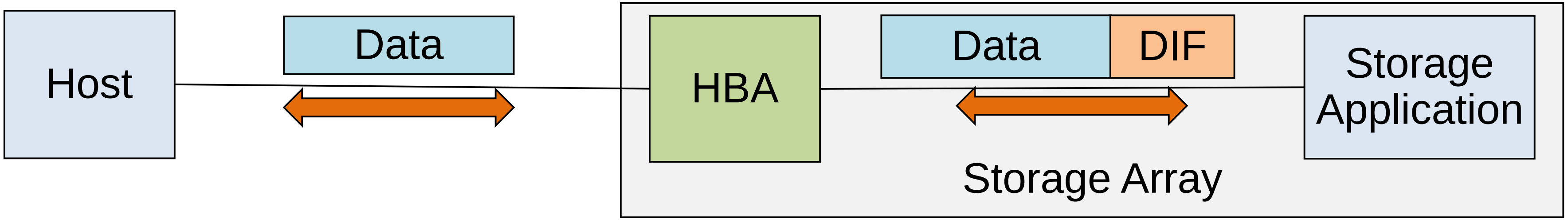
- 8 byte DIF is associated with each data block.
  - Guard (GRD) tag contains a CRC of the data block.
  - Application (APP) tag is up to application.
  - Reference (REF) tag normally contains the lower 4 bytes of the associated Logical Block Address.

Data Block	Metadata	GRD	APP	REF

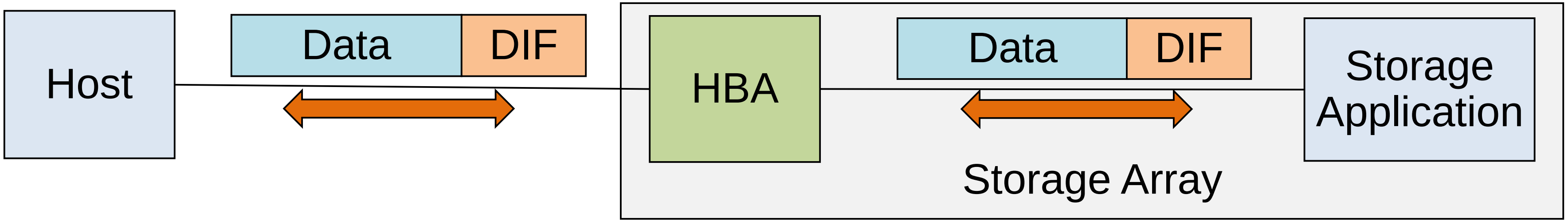
- There are many variables about DIF, metadata format, DIF settings, and DIF check types.

End-to-End Data Protection with Storage Arrays

- Dealing with data corruption is important for storage arrays and storage arrays have used DIF extensively.
- For some hosts that are not aware of DIF, iSCSI/FC HBAs have inserted/stripped DIF for write/read I/O.

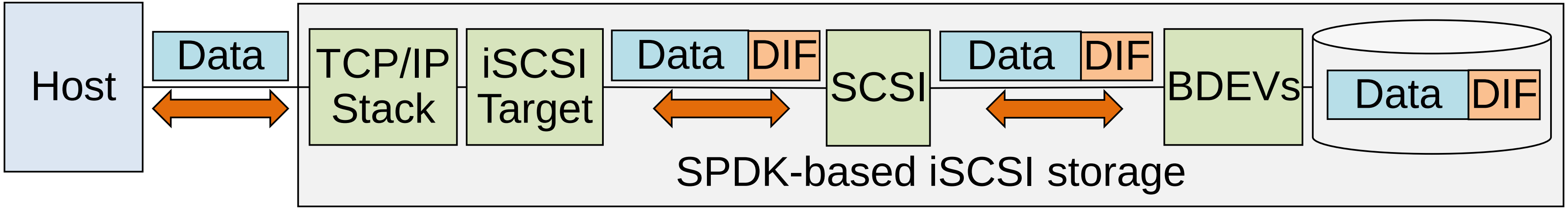


- For some hosts that are aware of DIF, iSCSI/FC HBAs pass data with DIF for read/write I/O.

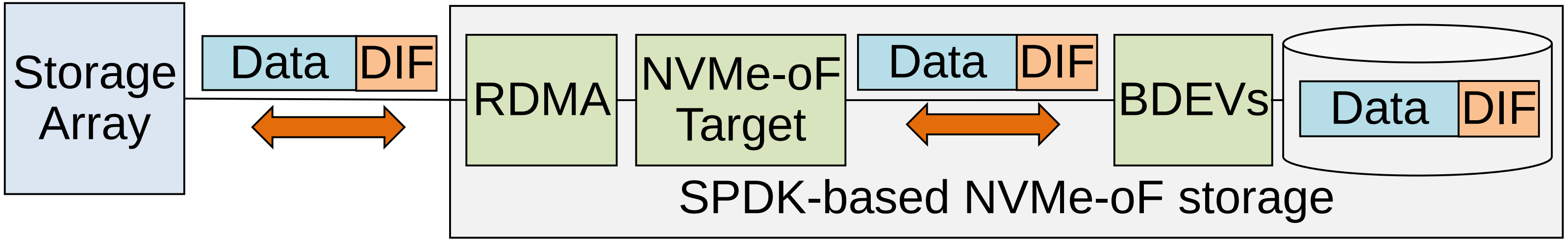


Use Cases of End-to-End Data Protection with SPDK

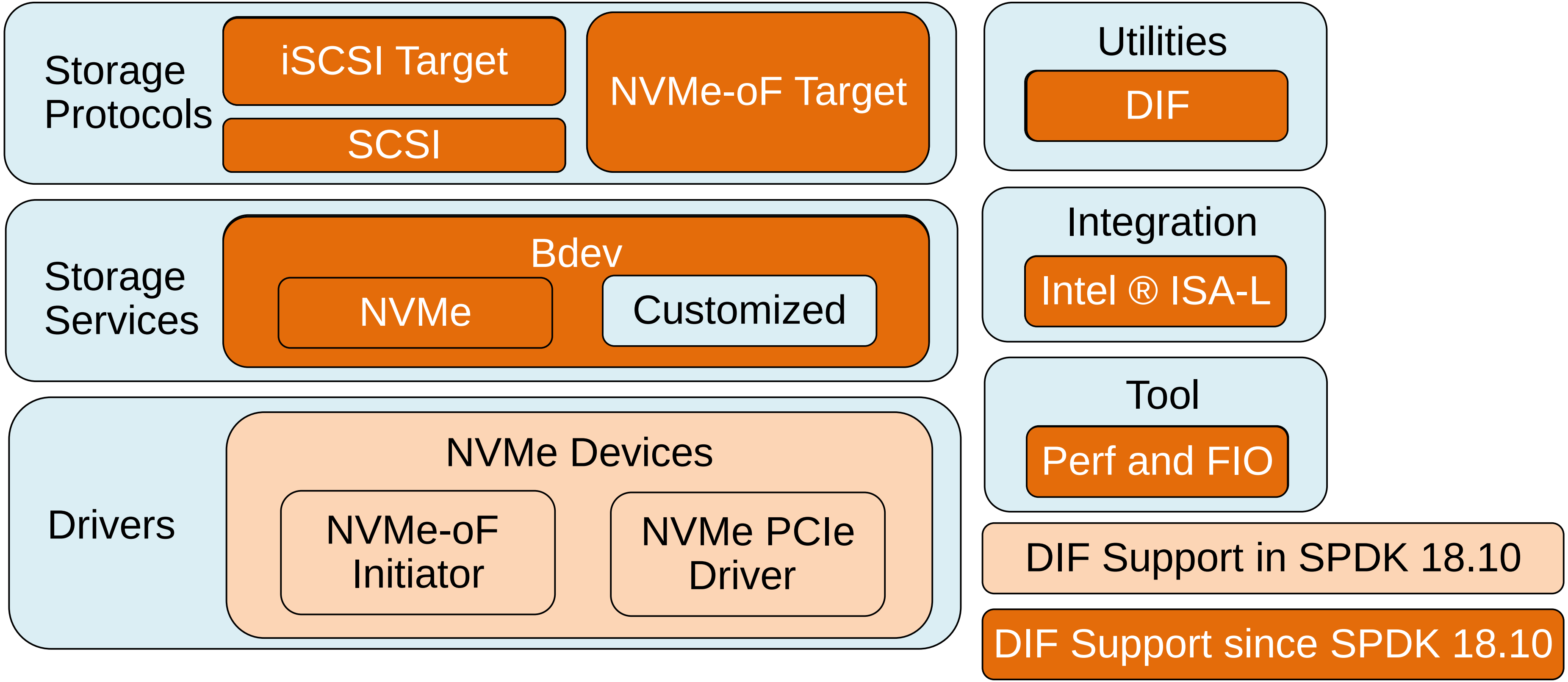
- It will be beneficial for storage applications if SPDK support DIF with compelling performance and efficiency without specialized hardware.
- DIF insert/strip feature in SPDK iSCSI target.



- DIF passthrough feature in SPDK NVMe-oF target



End-to-End Data Protection with SPDK 19.04



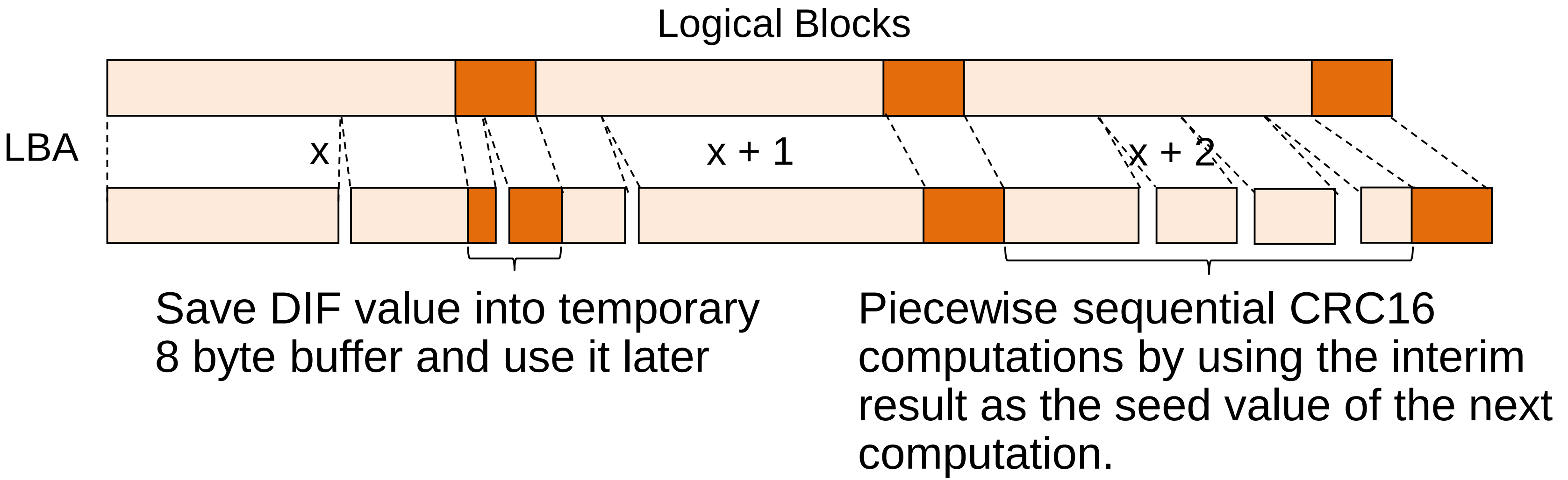
End-to-End Data Protection with SPDK 19.04 (Continued)

Component	Update
SPDK DIF Library	Provide routines to simplify integration of DIF operations into other SPDK libraries.
Intel ® Intelligent Storage Acceleration Library (ISA-L)	ISA-L is integrated with SPDK.
SPDK Bdev Layer	Expose DIF setting to upper layers.
SPDK NVMe Bdev Module	Find the location of DIF errors.
SPDK SCSI Layer	Return DIF context for SCSI read/write commands.
SPDK iSCSI Target	Support DIF insert and strip feature.
SPDK NVMe-oF Target	Support DIF passthrough feature.
SPDK NVMe Perf Tool and FIO Plugin	Support all DIF and DIX features.

2. DIF Support in SPDK

Byte Alignment and Granularity of Logical Blocks

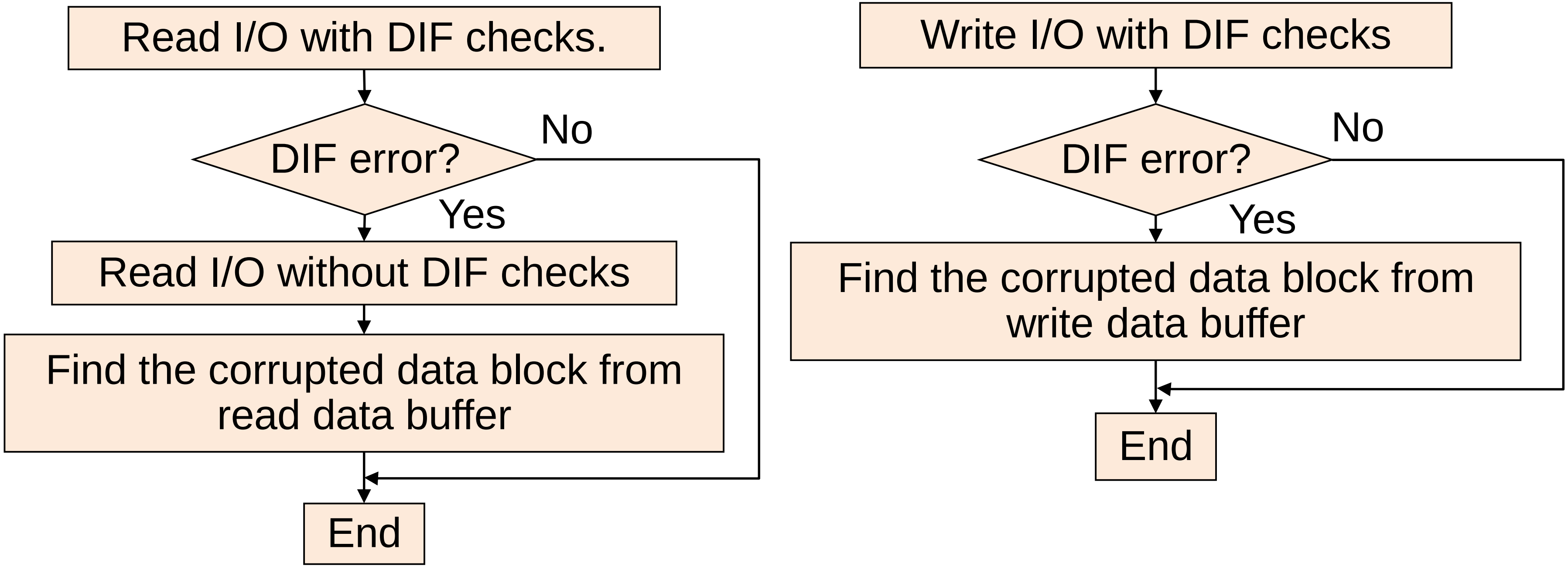
- SPDK DIF library supports byte alignment and granularity for data payload.



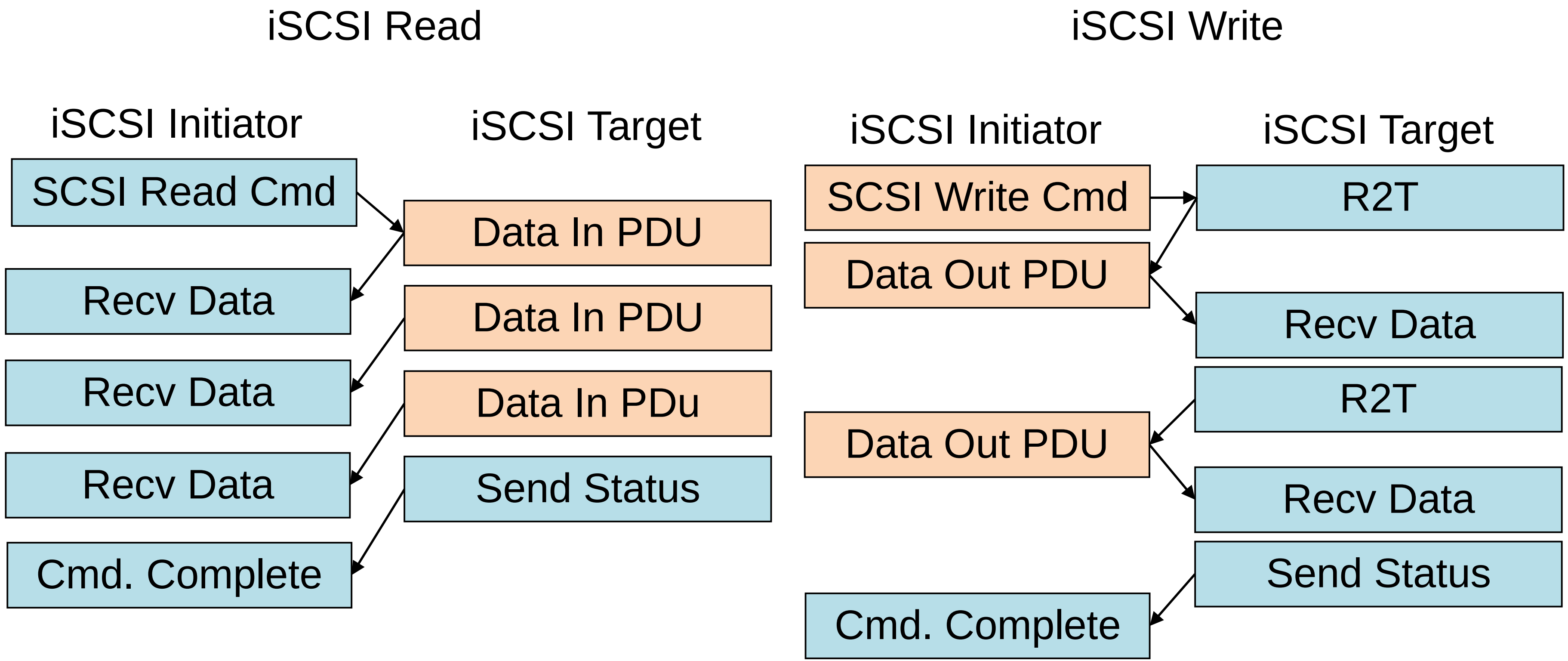
- To ensure quality by unit tests, the following are used:
  - Fault Injection which can inject bit flip error into any field and offset.
  - Use cyclic values to set in the test data buffer.

Find the Data Block at which DIF Error was Detected

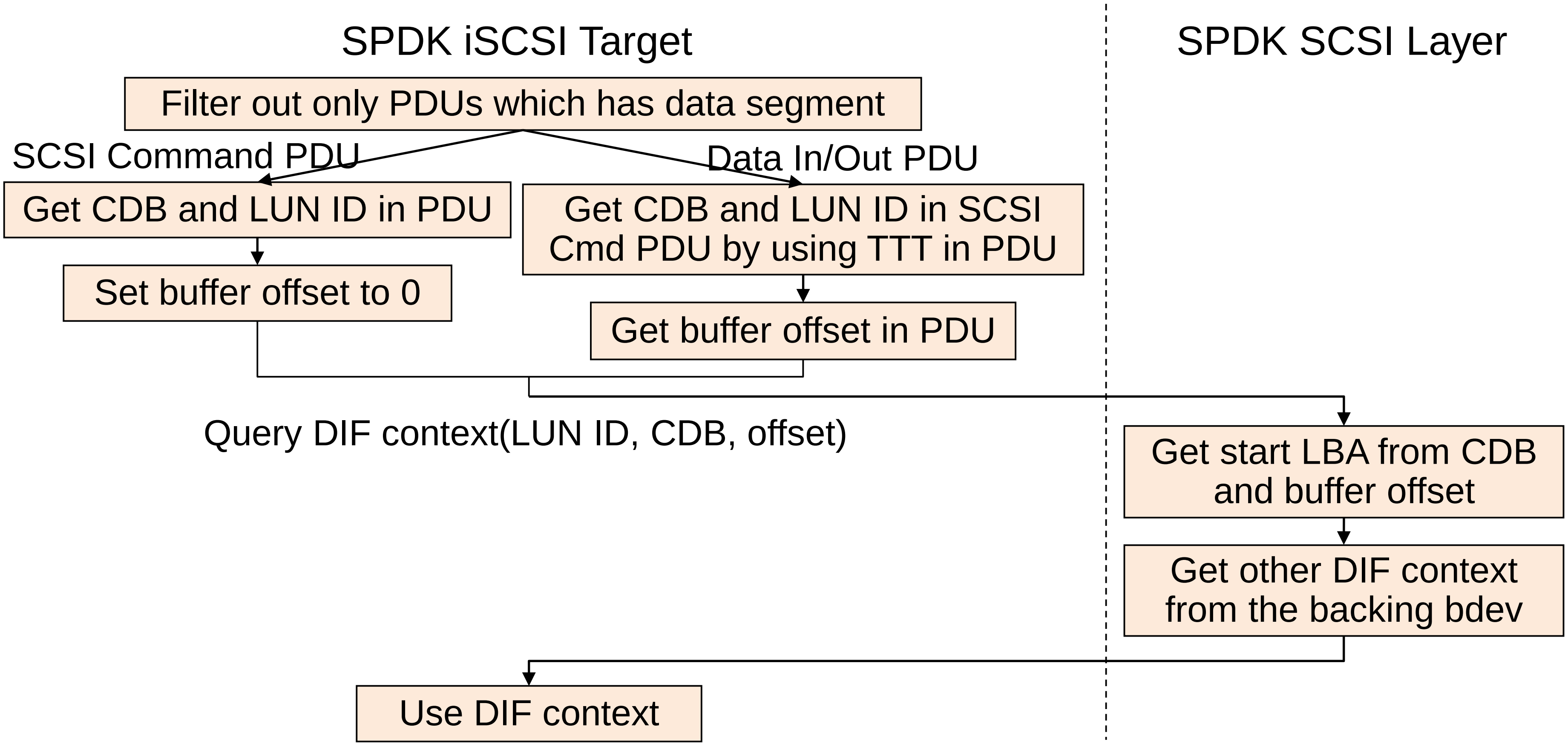
- SPDK NVMe bdev modules leaves DIF checks to NVMe controller but NVMe controller doesn’t report the location of DIF error. Hence SPDK NVMe bdev module finds the location of DIF error instead.



iSCSI Read and Write Sequence

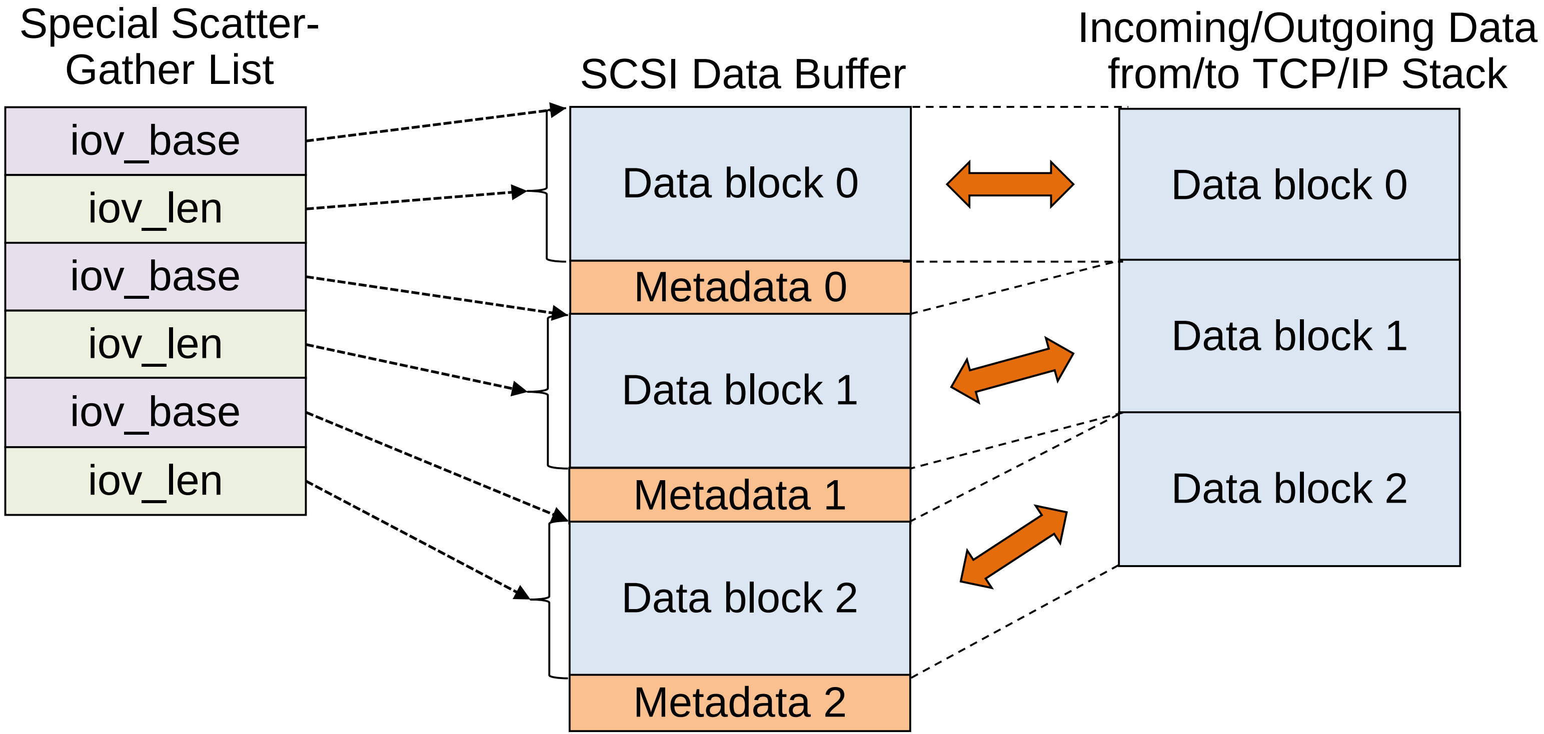


Query DIF Context to Insert/Strip DIF to/from iSCSI PDU

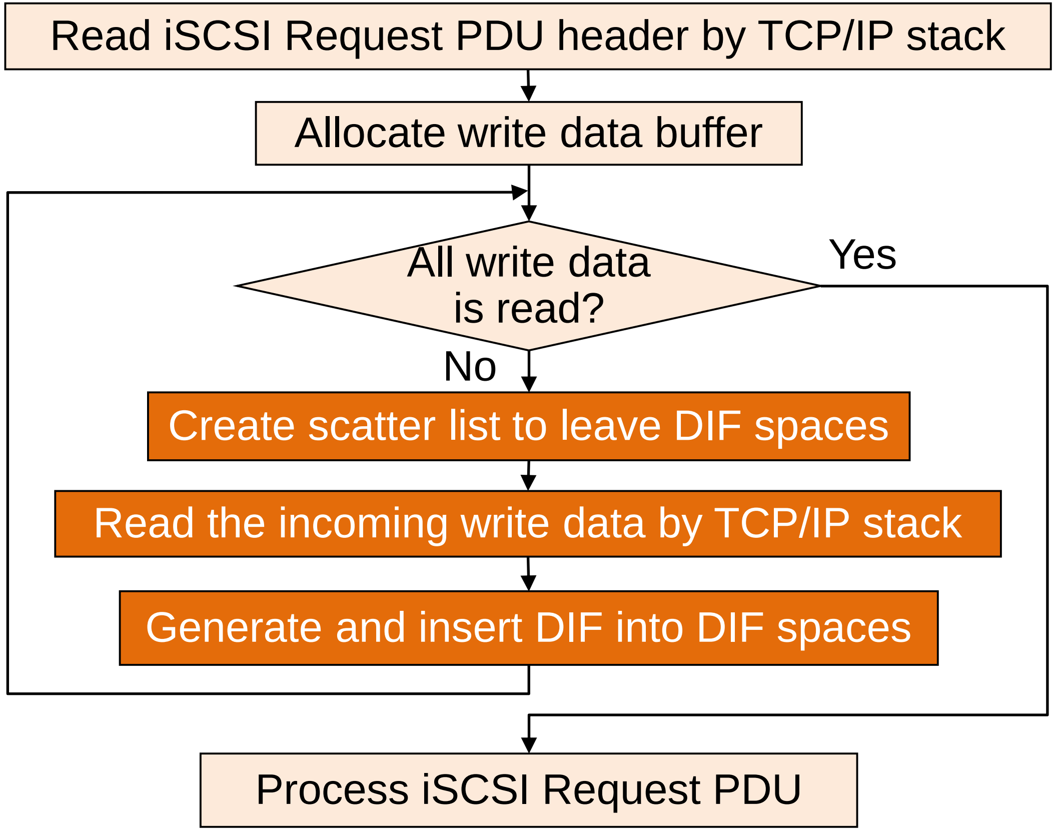


Scatter-Gather List to insert or strip DIF

- SPDK iSCSI target avoids extra data copy and any bounce buffer by using the special scatter-gather list.

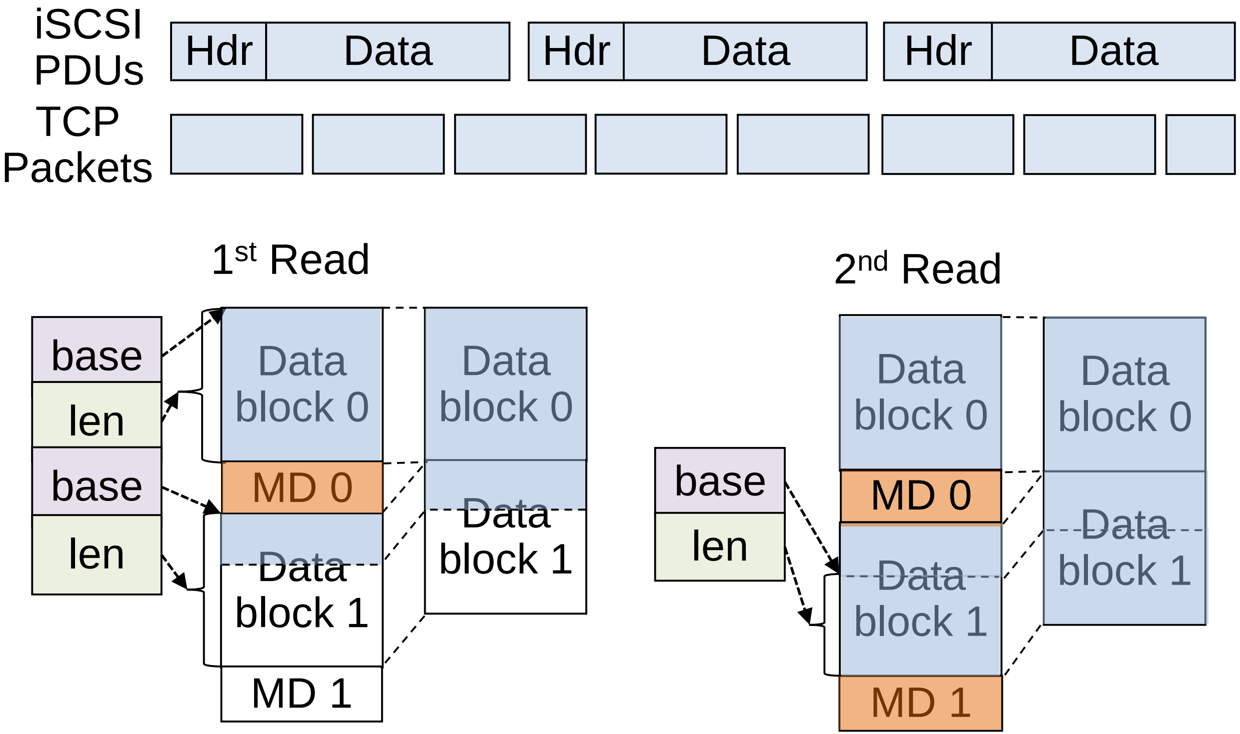


Read the Incoming Write Data with DIF Insertion

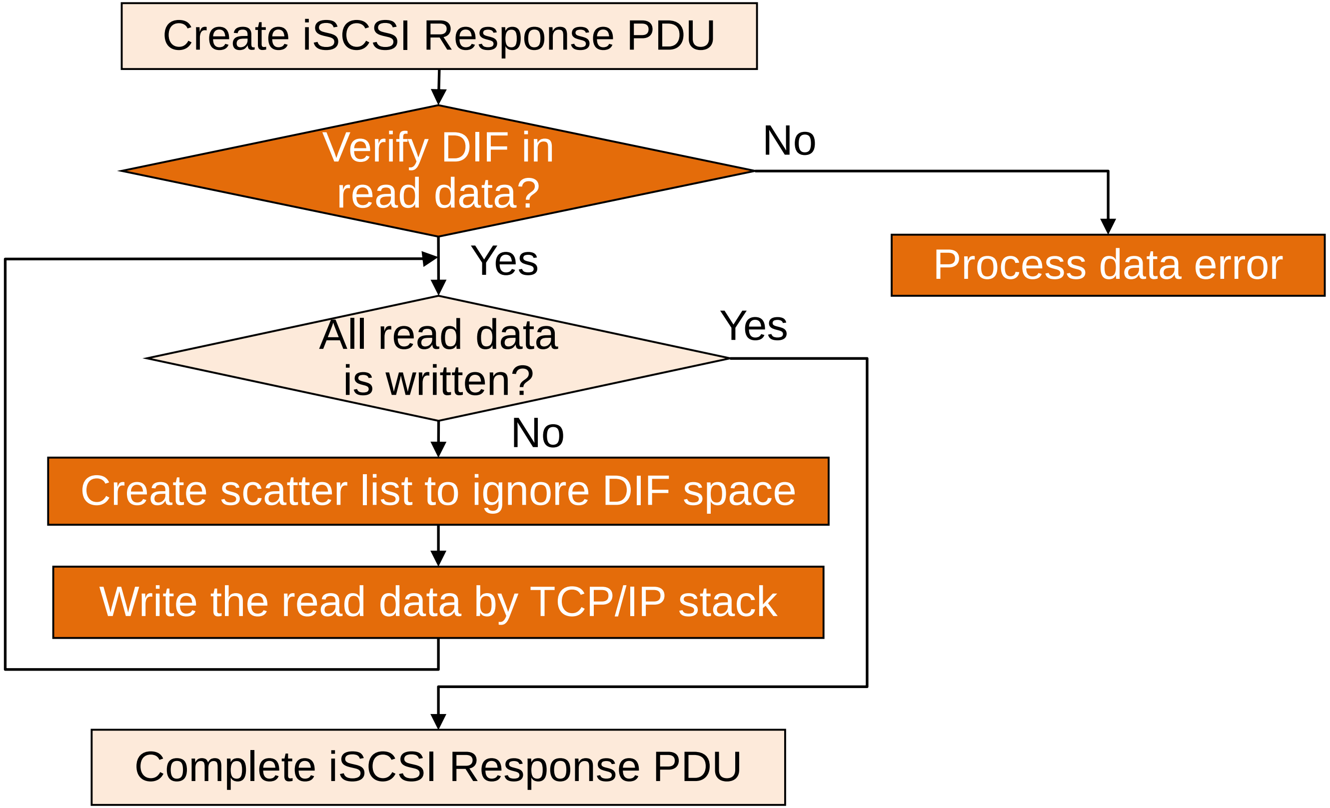


Process the Split Incoming Write Data with DIF Insertion

- Incoming write data may be split into multiple TCP packets.
- iSCSI target adjusts scatter-list before every read.
- If the newly read starts from the unaligned offset, DIF insertion steps back to the aligned offset and generates and inserts DIF to full data blocks.



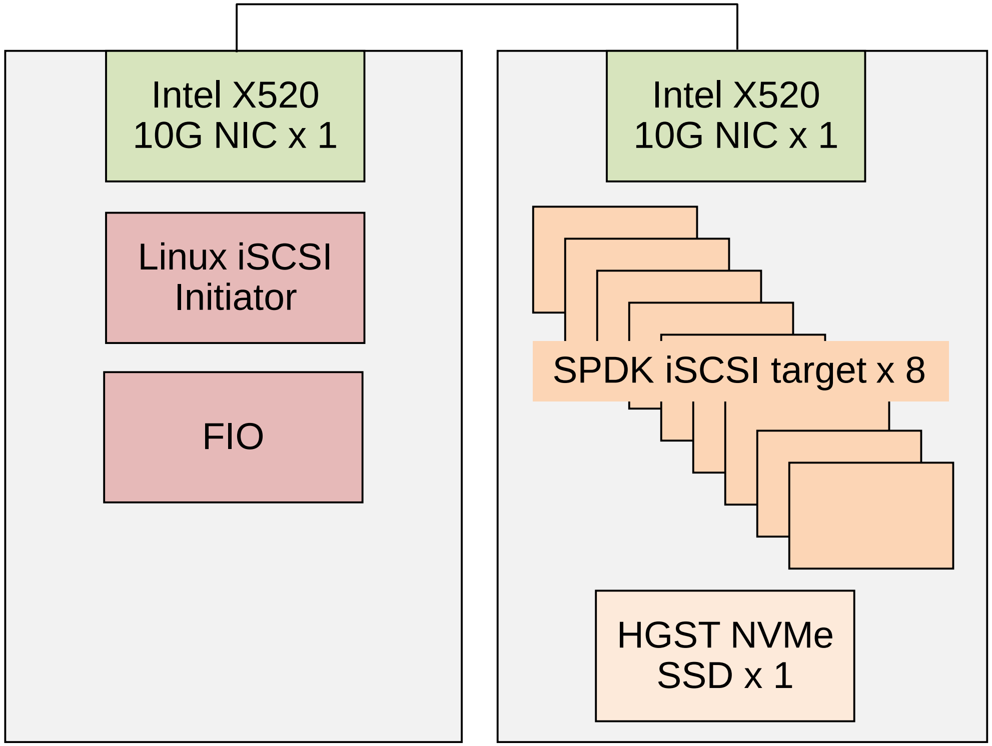
Write the Outgoing Read Data with DIF Strip



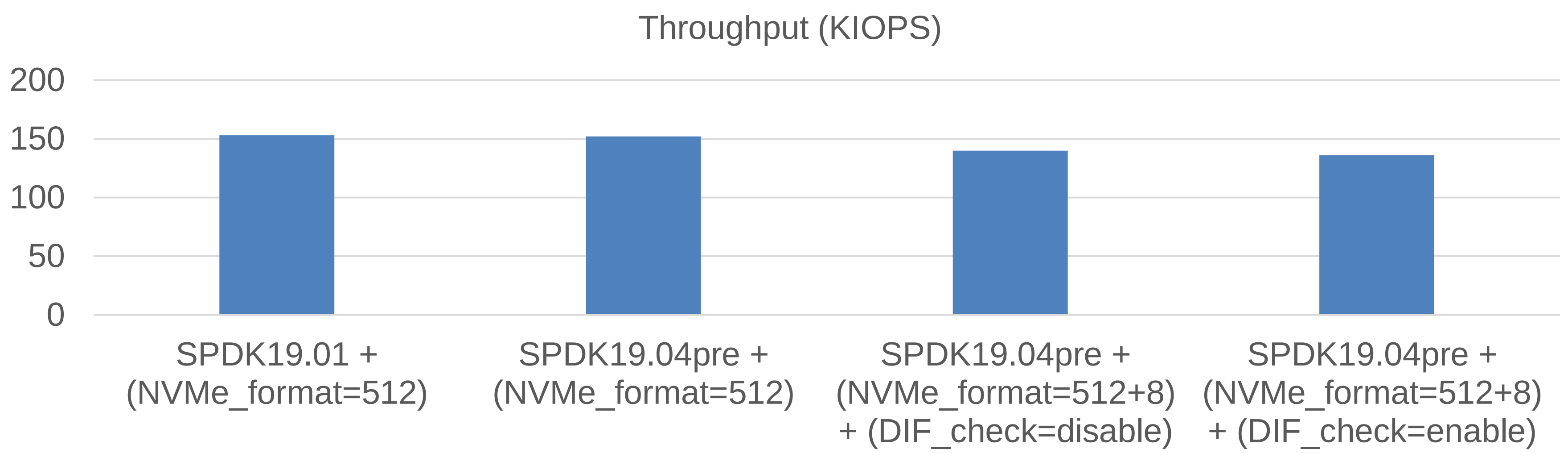
3. Performance Evaluation

System Configuration

- Use two Xeon processor servers
- SPDK iSCSI target used
  - a single CPU core.
  - Linux kernel TCP/IP stack.
  - a single NVMe SSD which supports SGL.
- iSCSI initiator used
  - Linux kernel iSCSI initiator.
- Use FIO 3.3 and create 8 iSCSI sessions on a single 10G link.
- Both 10G NICs set MTU to 9000.

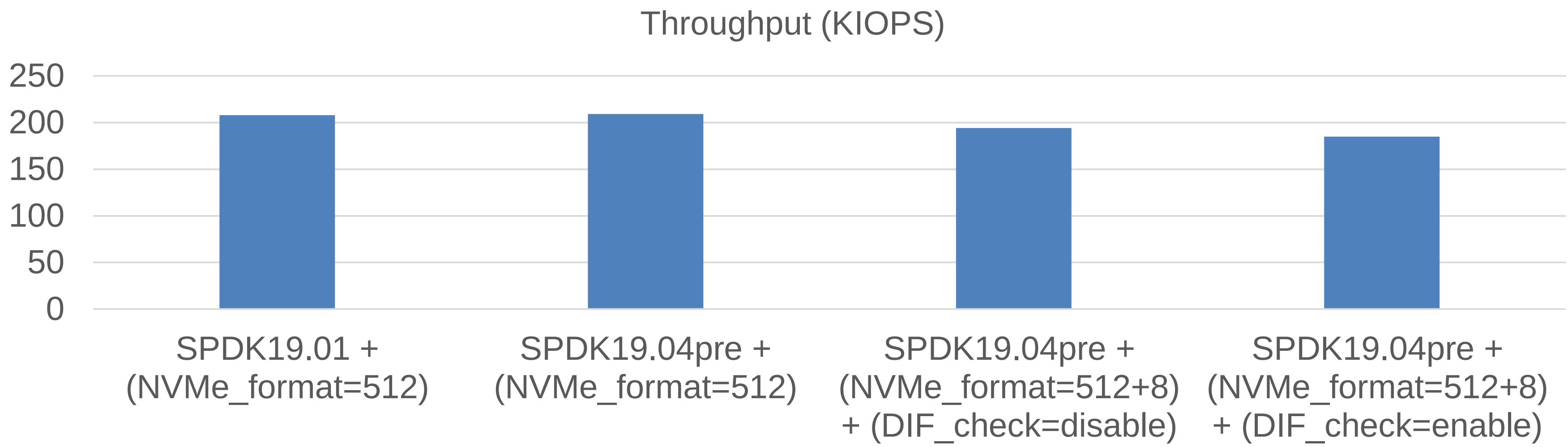


4KB Random Write



Configuration	Throughput (KIOPS)	Overhead
SPDK 19.01 + (NVMe format = 512)	153	-
SPDK 19.04 pre + (NVMe format = 512)	152	0%
SPDK 19.04 pre + (NVMe format = 512 + 8) + (DIF check = disable)	140	9.2%
SPDK 19.04 pre + (NVMe format = 512 + 8) + (DIF check = enable)	136	12.5%

4KB Random Read



Configuration	Throughput (KIOPS)	Overhead
SPDK 19.01 + (NVMe format = 512)	208	-
SPDK 19.04 pre + (NVMe format = 512)	209	0%
SPDK 19.04 pre + (NVMe format = 512 + 8) + (DIF check = disable)	194	7.2%
SPDK 19.04 pre + (NVMe format = 512 + 8) + (DIF check = enable)	185	12.4%

4. Summary and Next Steps

Summary and Next Steps

Summary

- SPDK iSCSI target provides DIF insert and strip feature without specialized hardware in SPDK 19.04.
- Performance evaluation showed that the overhead were a little more than 10% both for 4KB random read and write.

Next Steps

- Performance Evaluation with Vector Packet Processing (VPP)
- Support DIF in bdev modules (e.g. crypto, compress, and RAID)
- Support DIX (separate metadata) in bdev modules
- Support DIF insert and strip feature in NVMe-TCP target.

Inspire the N

END

End-to-End Data Protection with SPDK

04/16/2019

Shuhei Matsumoto

SPDK Core Maintainer

IT Platform Products Management Division

Hitachi, Ltd.

Inspire the N