DAOS User Group (DUG'20)

Online Compression with Intel® QAT in DAOS

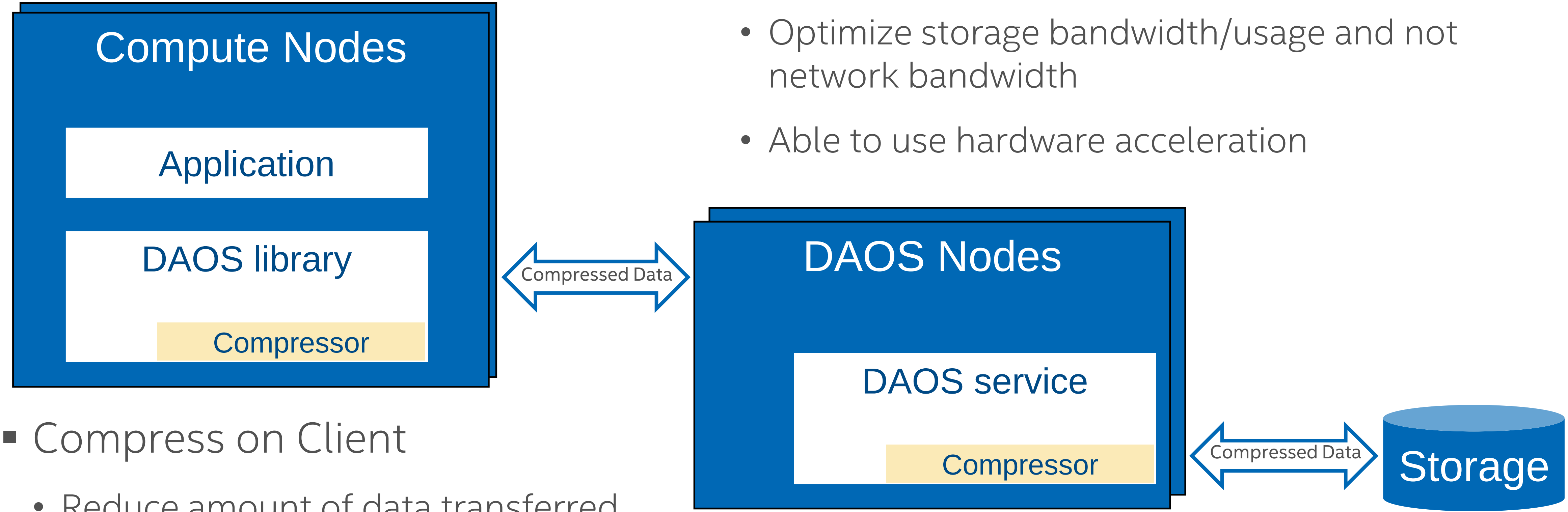
Weigang Li, Intel

Johann Lombardi, Intel

November, 2020

DAOS compression

- Compress on Server
  - Optimize storage bandwidth/usage and not network bandwidth
  - Able to use hardware acceleration



- Compress on Client
  - Reduce amount of data transferred over the network
  - Consume compute CPU cycles

Note: Only data will be compressed for now and not metadata

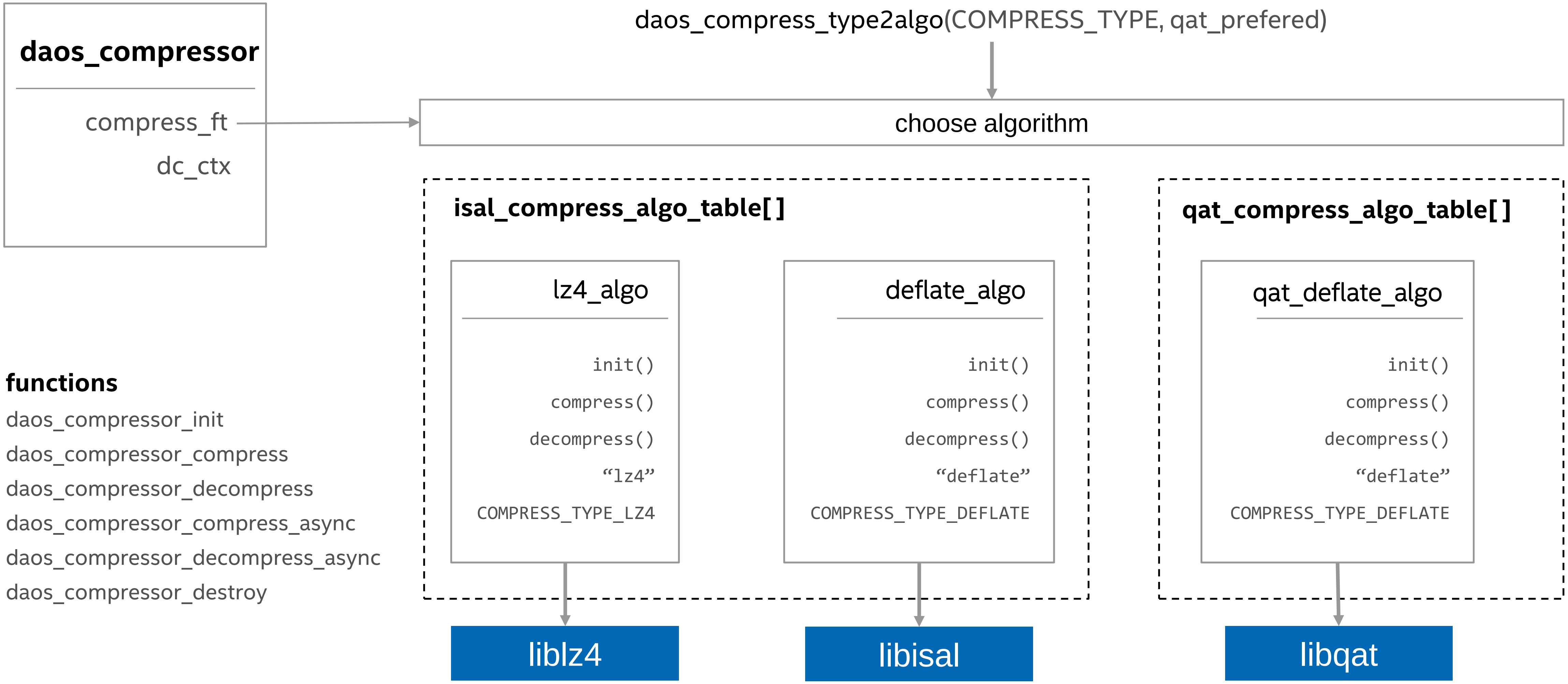
Current Status

- All the infrastructure to enable compression and encryption has been landed.
- It is set at container creation time via properties **DAOS_PROP_CO_COMPRESS** and **DAOS_PROP_CO_ENCRYPT**.

```
# daos cont get-prop --path /mnt/lustre/mycontainer --svc $SVC
Container properties for d28d9df6-5ee7-4030-b20f-
bf06c13e2226 :
label:                 container label not set
layout type:           POSIX (1)
layout version:        1
checksum type:         off
checksum chunk-size:   32768
cksum verif. on server: off
deduplication:         off
dedup threshold:       4096
redundancy factor:     rf1
redundancy level:      rack
max snapshots:         0
compression type:      off
encryption type:       off
owner:                 root@
owner-group:           root@
```

- DAOS-5605 common: add basic infrastructure for QAT #3402 (Merged)
- DAOS-5605 build: add lz4 dependency #3403 (Merged)
- DAOS-5719 compression: add framework and lz4/deflate support #3561 (Merged)
- DAOS-5719 compression: add qat support for deflate #3621 (Merged)
- QAT async support and compress_timing test (WIP)

DAOS Compressor: Abstraction layer in DAOS



daos_compressor

compress_ft

dc_ctx

daos_compress_type2algo(COMPRESS_TYPE, qat_prefered)

choose algorithm

isal_compress_algo_table[ ]

lz4_algo

```
init()
compress()
decompress()
"lz4"
COMPRESS_TYPE_LZ4
```

deflate_algo

```
init()
compress()
decompress()
"deflate"
COMPRESS_TYPE_DEFLATE
```

qat_compress_algo_table[ ]

qat_deflate_algo

```
init()
compress()
decompress()
"deflate"
COMPRESS_TYPE_DEFLATE
```

functions

daos_compressor_init

daos_compressor_compress

daos_compressor_decompress

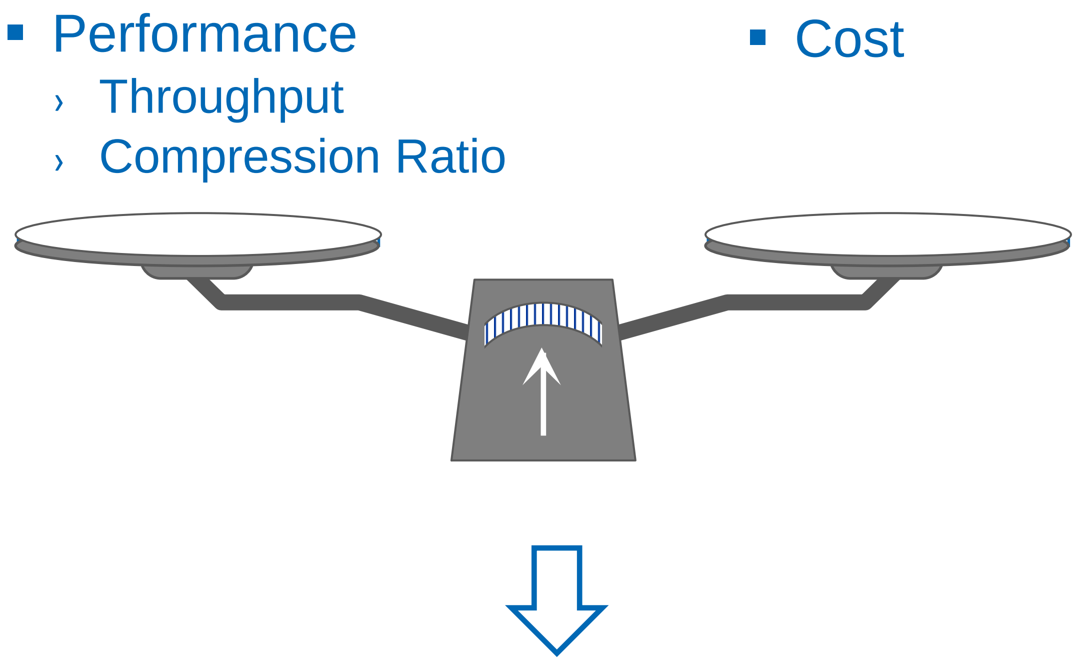
daos_compressor_compress_async

daos_compressor_decompress_async

daos_compressor_destroy

Compression algorithms

- LZ4
  - Fast, low compression ratio
- Deflate
  - Slow, high compression ratio
  - Implementation: ZLIB, GZIP
- zStandard
  - New algorithm
  - Fast, high compression ratio
  - May be added to DAOS in future

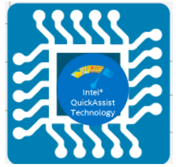


- Performance
  - Throughput
  - Compression Ratio
- Cost

Hardware acceleration

Intel® QuickAssist Technology (QAT)

- Intel® QuickAssist Technology (QAT) integrates hardware acceleration for compute intensive workloads:
  - ✓ Bulk Cryptography
  - ✓ Public Key Exchange
  - ✓ Compression
- Formfactors: Chipset, PCIE card, SoC

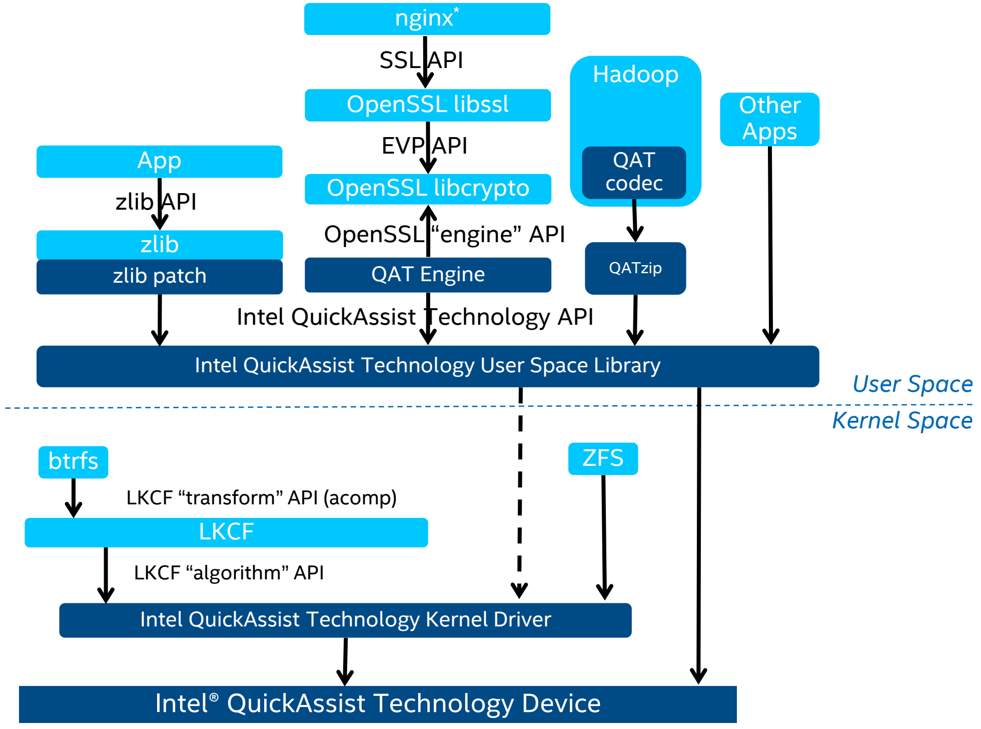


Intel® C620 Series Chipsets

Intel® QuickAssist Adapter 89xx

Rangeley, Denverton

- Integrated with many mainstream software framework



nginx*

SSL API

OpenSSL libssl

EVP API

OpenSSL libcrypto

OpenSSL "engine" API

QAT Engine

App

zlib API

zlib

zlib patch

Hadoop

QAT codec

QATzip

Other Apps

Intel QuickAssist Technology API

Intel QuickAssist Technology User Space Library

User Space

Kernel Space

btrfs

LKCF "transform" API (acomp)

LKCF

LKCF "algorithm" API

ZFS

Intel QuickAssist Technology Kernel Driver

Intel® QuickAssist Technology Device

Intel® QAT Software Stack

https://01.org/intel-quickassist-technology daos_compressor_compress

QAT Compression Sync Mode

Main Thread

callback function:

```
complete = 1;
```

qat_dc_compress:

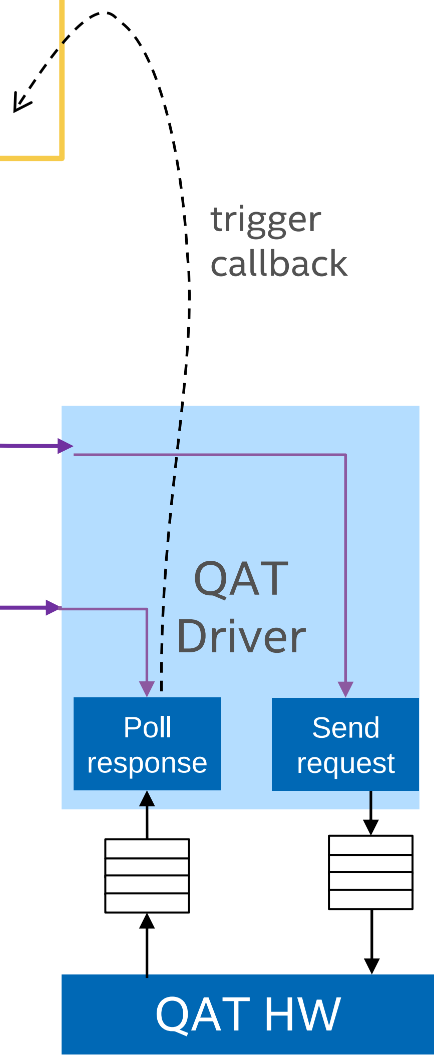
```
allocate pinned memory and copy source buffer

cpaDcCompressData2        (s_buf, d_buf, &complete)

do {
      icp_sal_DcPollInstance
} while (!complete);
```

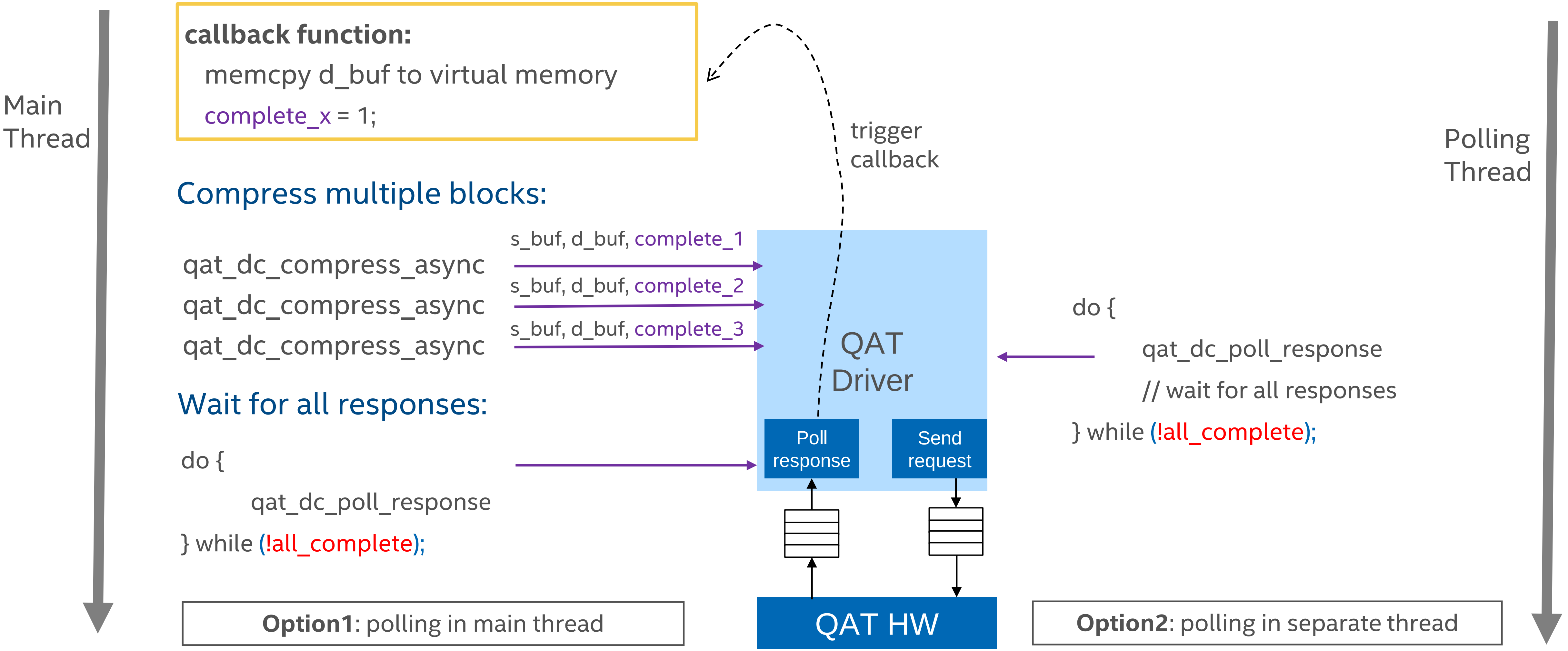
block waiting in main thread

```
copy d_buf to virtual memory
return;
```



daos_compressor_compress_async

QAT Compression Async Mode



callback function:

```
  memcpy d_buf to virtual memory
  complete_x = 1;
```

Compress multiple blocks:

```
qat_dc_compress_async
qat_dc_compress_async
qat_dc_compress_async
```

Wait for all responses:

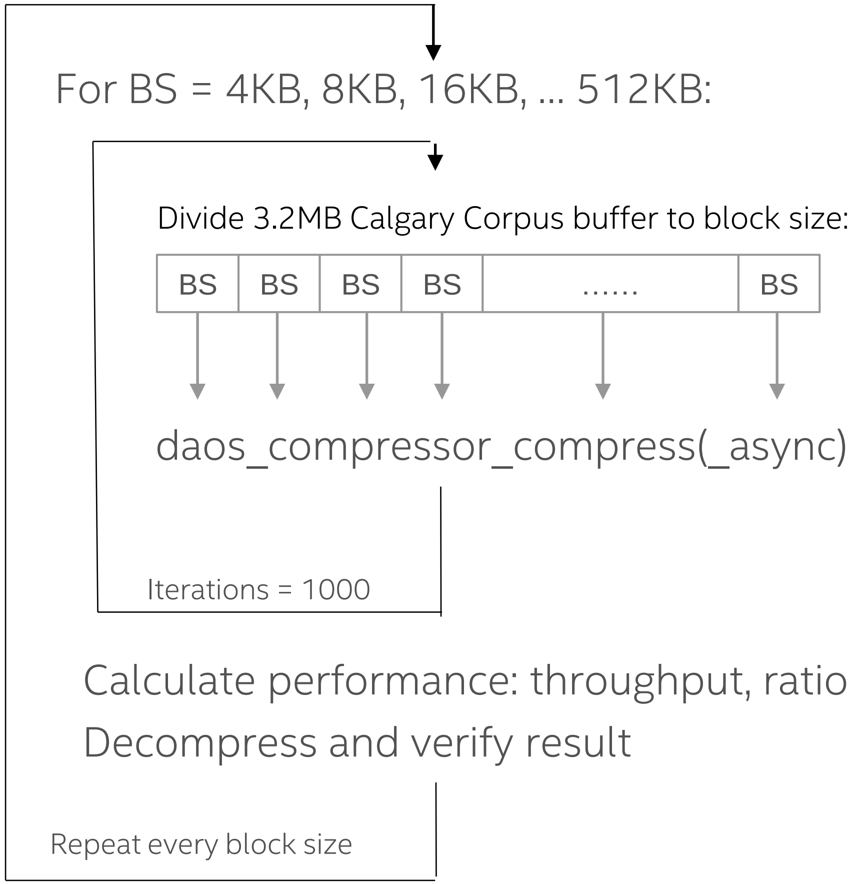
```
do {
        qat_dc_poll_response
} while (!all_complete);
```

Option1: polling in main thread

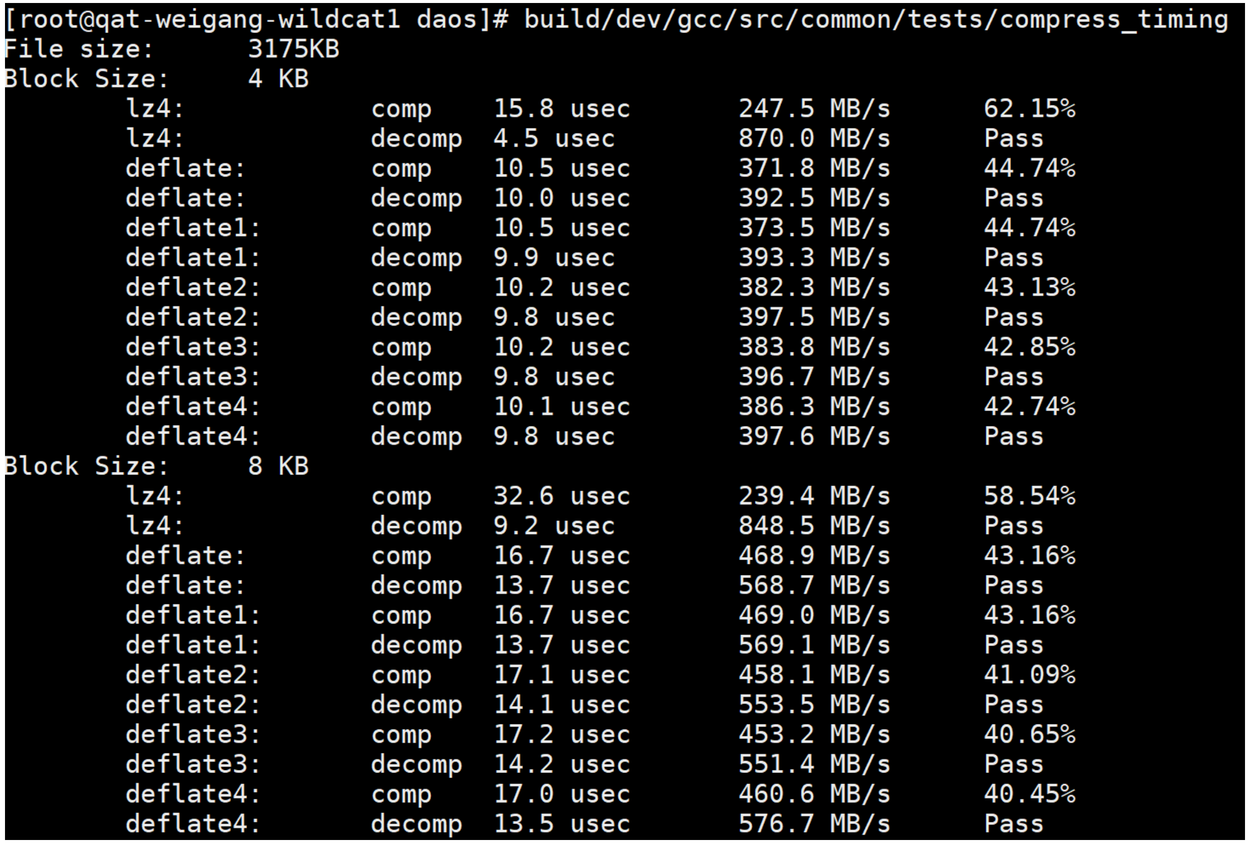
```
do {
        qat_dc_poll_response
        // wait for all responses
} while (!all_complete);
```

Option2: polling in separate thread

Compress_timing

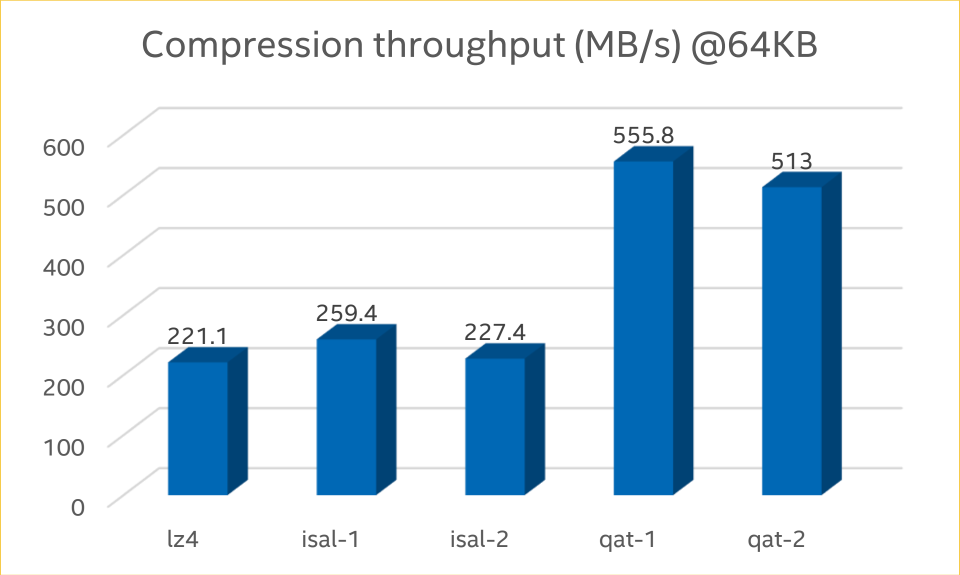

For BS = 4KB, 8KB, 16KB, … 512KB:
Divide 3.2MB Calgary Corpus buffer to block size:
BS
BS
BS
BS
……
BS
daos_compressor_compress(_async)
Iterations = 1000
Calculate performance: throughput, ratio
Decompress and verify result
Repeat every block size

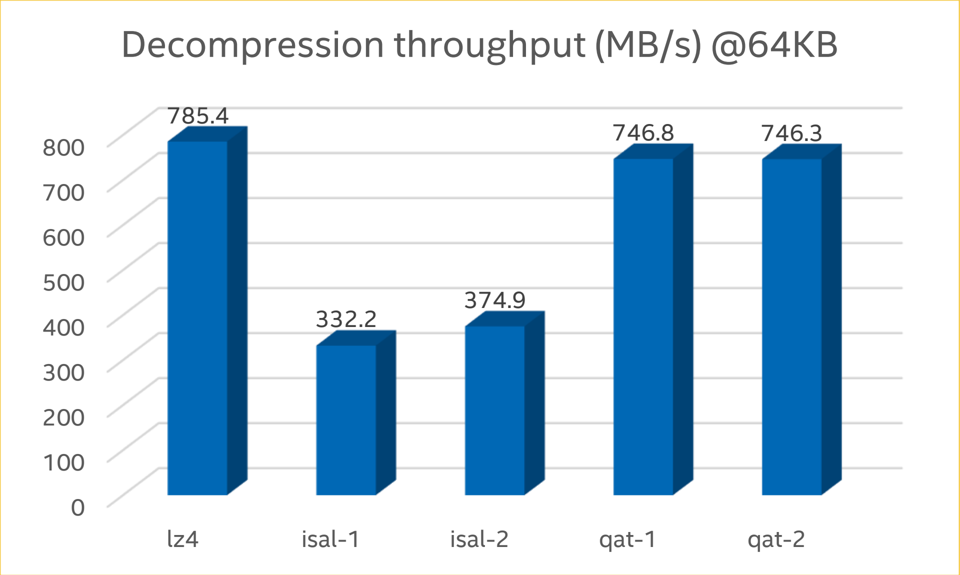
Single Thread

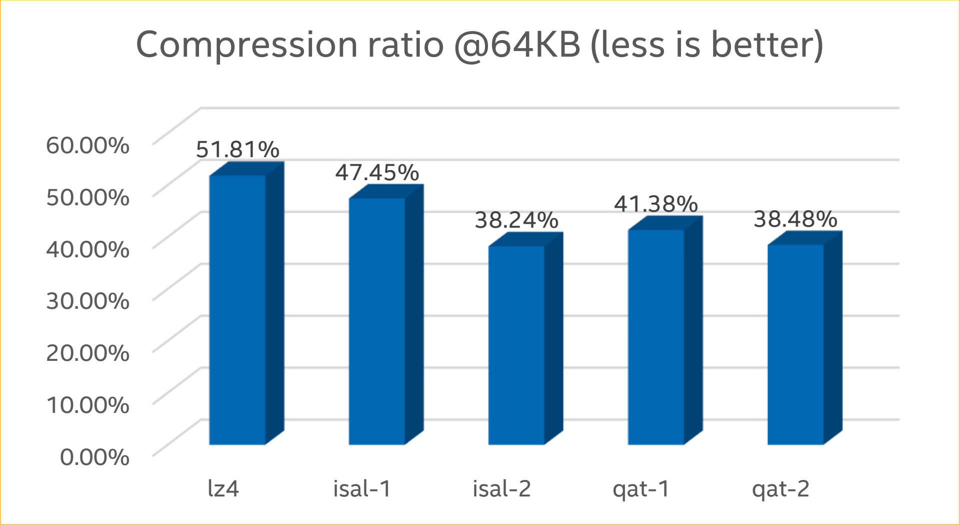


```
[root@qat-weigang-wildcat1 daos]# build/dev/gcc/src/common/tests/compress_timing
File size:      3175KB
Block Size:     4 KB
        lz4:            comp    15.8 usec       247.5 MB/s      62.15%
        lz4:            decomp  4.5 usec        870.0 MB/s      Pass
        deflate:        comp    10.5 usec       371.8 MB/s      44.74%
        deflate:        decomp  10.0 usec       392.5 MB/s      Pass
        deflate1:       comp    10.5 usec       373.5 MB/s      44.74%
        deflate1:       decomp  9.9 usec        393.3 MB/s      Pass
        deflate2:       comp    10.2 usec       382.3 MB/s      43.13%
        deflate2:       decomp  9.8 usec        397.5 MB/s      Pass
        deflate3:       comp    10.2 usec       383.8 MB/s      42.85%
        deflate3:       decomp  9.8 usec        396.7 MB/s      Pass
        deflate4:       comp    10.1 usec       386.3 MB/s      42.74%
        deflate4:       decomp  9.8 usec        397.6 MB/s      Pass
Block Size:     8 KB
        lz4:            comp    32.6 usec       239.4 MB/s      58.54%
        lz4:            decomp  9.2 usec        848.5 MB/s      Pass
        deflate:        comp    16.7 usec       468.9 MB/s      43.16%
        deflate:        decomp  13.7 usec       568.7 MB/s      Pass
        deflate1:       comp    16.7 usec       469.0 MB/s      43.16%
        deflate1:       decomp  13.7 usec       569.1 MB/s      Pass
        deflate2:       comp    17.1 usec       458.1 MB/s      41.09%
        deflate2:       decomp  14.1 usec       553.5 MB/s      Pass
        deflate3:       comp    17.2 usec       453.2 MB/s      40.65%
        deflate3:       decomp  14.2 usec       551.4 MB/s      Pass
        deflate4:       comp    17.0 usec       460.6 MB/s      40.45%
        deflate4:       decomp  13.5 usec       576.7 MB/s      Pass
```

Compress_timing results @64KB block size

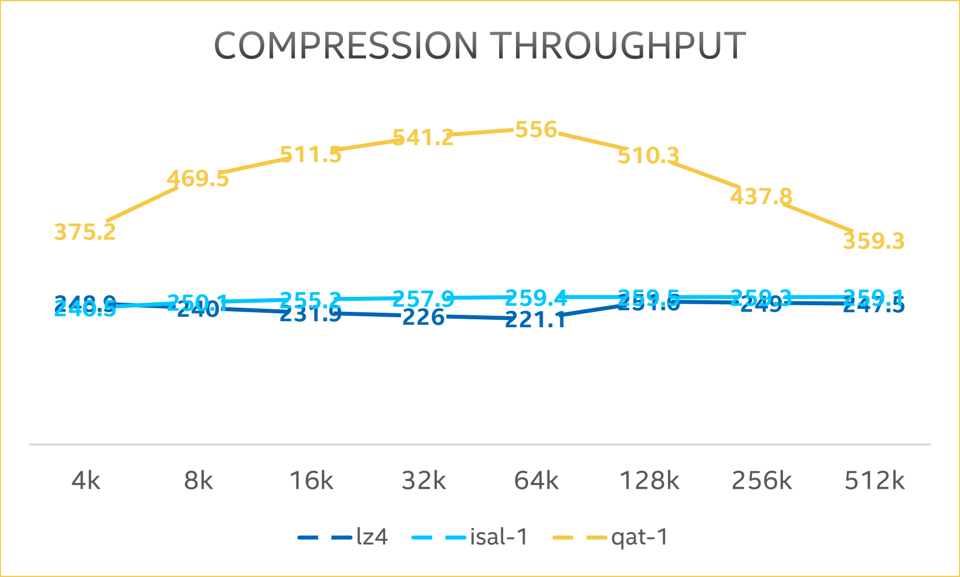


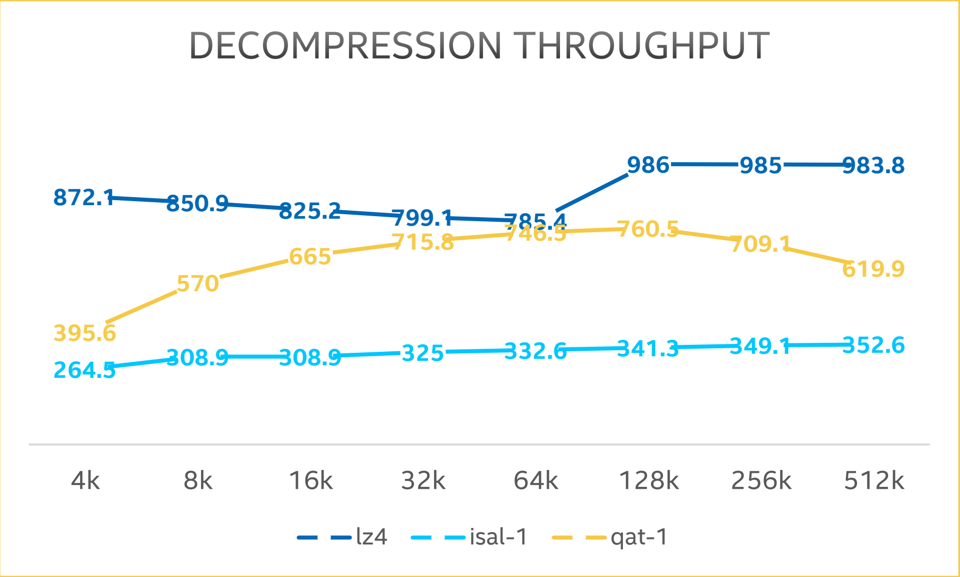


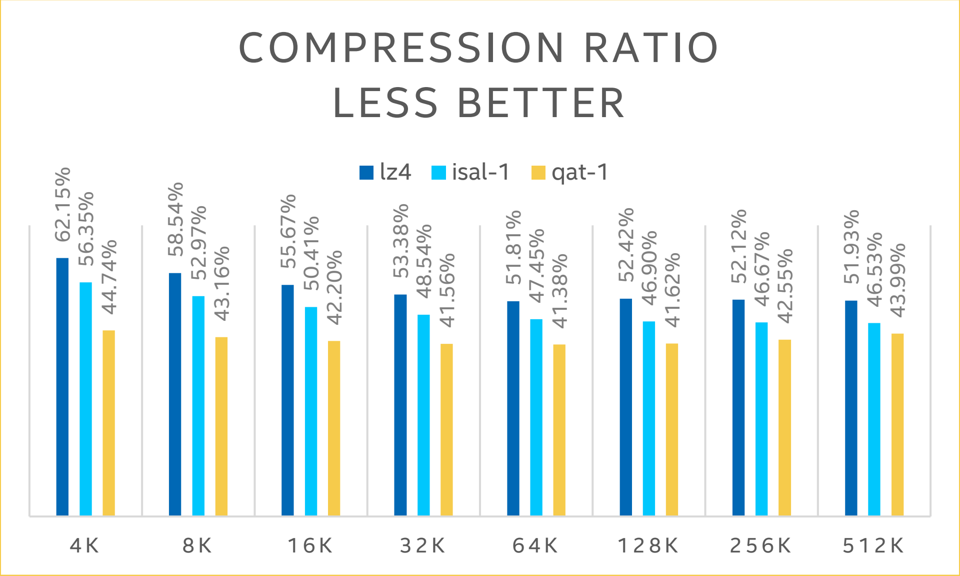


- Tested with QAT Gen-2 DH8950 plugin card
- Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz
- Ratio = compressed_size / origin_size
- Single Thread

Compress_timing results for all block sizes


COMPRESSION THROUGHPUT


DECOMPRESSION THROUGHPUT


COMPRESSION RATIO
LESS BETTER

- QAT
  - ✓Best compression throughput
  - ✓Best compression ratio
- LZ4
  - ✓Best decompression throughput

Future Work

- Implement actual extents/records compression on both client and server
- Change aggregation to support compression
- Support partial read in the middle of a compressed extent
  - Similar to end-to-end checksum support where we have to read the whole extent to validate the checksum. We will just reuse this code.
- Automatically discover QAT at build time and runtime, intelligently switch between software and hardware acceleration API
- Chaining of compression and hash in one function call

Legal Disclaimer

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

- A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

- Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

- The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

- Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

- Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725,
or go to: http://www.intel.com/design/literature.htm%20

- All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

- Intel, Intel logo, Look Inside, Intel Inside, Intel Inside logo, the Look Inside Logo, Intel Atom, and Intel Xeon are trademarks of Intel Corporation in the U.S. and other countries.

- *Other names and brands may be claimed as the property of others.

- Other vendors are listed by Intel as a convenience to Intel's general customer base, but Intel does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of these devices. This list and/or these devices may be subject to change without notice.

-

intel®