Intel® In-Memory Analytics Accelerator Plugin for RocksDB* Storage Engine (Intel® IAA Plugin for RocksDB* Storage Engine)

Intel® IAA-accelerated RocksDB* Storage Engine delivers higher performance at lower cost and power.

Authors

Binuraj Ravindran
Principal Engineer
Intel Corporation

Luca Giacchino
Cloud Software Development Engineer
Intel Corporation

Kalyan Jee
Cloud Solution Architect
Intel Corporation

Nandita Narendra Babu
Cloud Software Development Engineer
Intel Corporation

As organizations and businesses make increasingly data-driven decisions, it is essential to have a high-performance and cost-effective data analytics framework to derive valuable insights from large datasets. Intel® In-Memory Analytics Accelerator (Intel® IAA), one of the Intel® Accelerator Engines in 4th Generation Intel® Xeon® Scalable processors, is designed to provide a range of benefits, including increased application performance, reduced costs, and improved power efficiency for data-analytics workloads.

This paper highlights performance improvements and cost savings that Intel IAA can provide for data analytics workloads using RocksDB* Storage Engine. RocksDB* is an embedded persistent key-value store for faster storage.

Intel IAA-accelerated RocksDB* Storage Engine achieves 103% higher throughput (operations/s) and 64% lower p99 latency compared to zstd-based software solutions for 80:20 read/write scenarios with 16KB block size and 256B value size. Compared to a zstd-based software solution to achieve the same throughput, the cost-of-compute can be reduced by 67% because compression and decompression operations can be offloaded to Intel IAA to save CPU cores.

4th Gen Intel® Xeon® Scalable Processor Accelerator Integration

The 4th Gen Intel® Xeon® Scalable processor integrates multiple accelerators, delivering enhanced performance across various established and emerging workloads. Depending on the CPU model, the number of accelerators and instances may vary, cost-effectively matching diverse workload needs at data centers.

Intel IAA is one of the integrated accelerators. It provides hardware acceleration for compression/decompression and common data analytics operations, enabling higher queries per second, memory compaction, and efficient database scan.

This paper focuses on Intel IAA and how it will deliver higher performance, lower cost, and power efficiency for data analytics workloads.

Doc. Ref: 357191

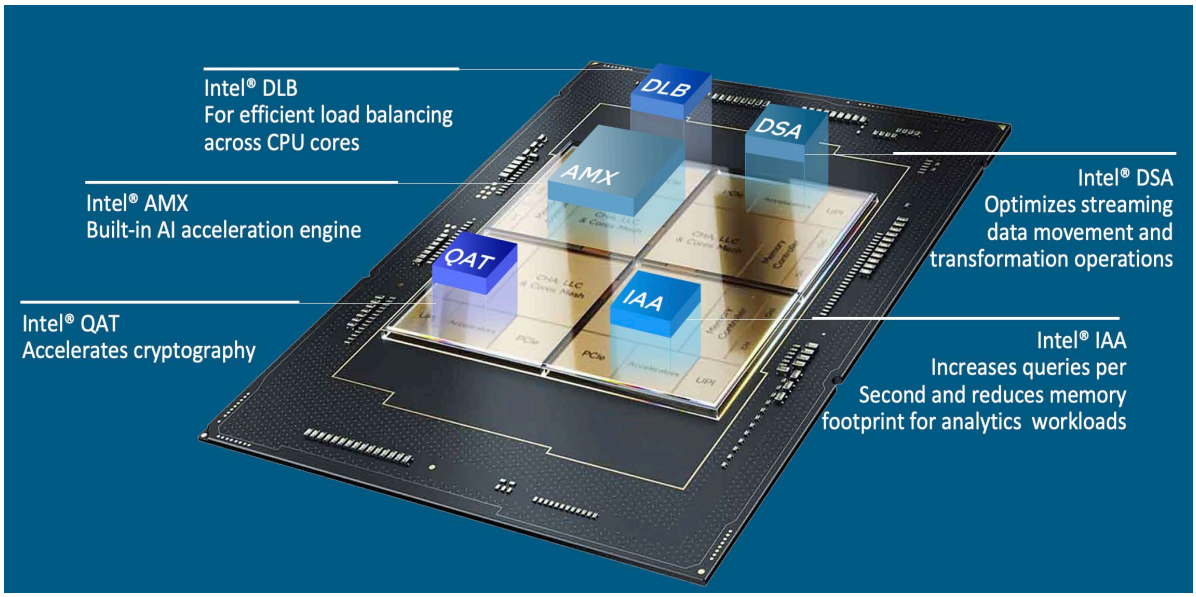

Figure 1: Intel® Accelerator Engines in 4th Gen Intel® Xeon® Scalable processor

Data Analytics and Integrated Accelerators in the 4th Gen Intel® Xeon® Scalable Processor

Data Analytics analyzes a database to find valuable information for organizational business needs. Today, data analytics represents a huge segment of the cloud landscape, which can be broadly divided into three sub-segments:

- Analytics for an In-Memory Database (IMDB) for real-time transaction processing.
- Analytics for a Large-Scale Database.
- Analytics for machine learning (image recognition, speech, text, etc.).

These databases are further classified into types of data that they store:

- **Relational Database:** Has a predefined schema for defining and manipulating data.
  - Uses structured query language (SQL).
  - Table-based and considered best for multi-row transactions.
- **Non-Relational Database (NoSQL):** Has dynamic schemas for defining and manipulating data.
  - Unstructured.
  - Best for key-value, wide-column databases, and documents

Data Analysis involves a large amount of data scanning, storage, and movement across memory tiers to achieve time-sensitive scan/query results and cost-efficient business objectives. Such outcomes require the acceleration of the following functions:

- **Performance of Data-Scanning Algorithms** (Query Processing):
  - Can be accelerated by offloading the query/scan function to a dedicated hardware accelerator.
- **Data Compactness for Data Storage**:
  - Data storage performance can be accelerated by offloading the compression /decompression function to a dedicated hardware accelerator.

As illustrated in Figure 1, the 4th Gen Intel Xeon Scalable processor technology coupled with Intel® Accelerator Engines supports the unique requirements of data analytics.

- Accelerate data-scan throughput and latency with Intel IAA.
- Accelerate data-compactness throughput and latency with Intel IAA.

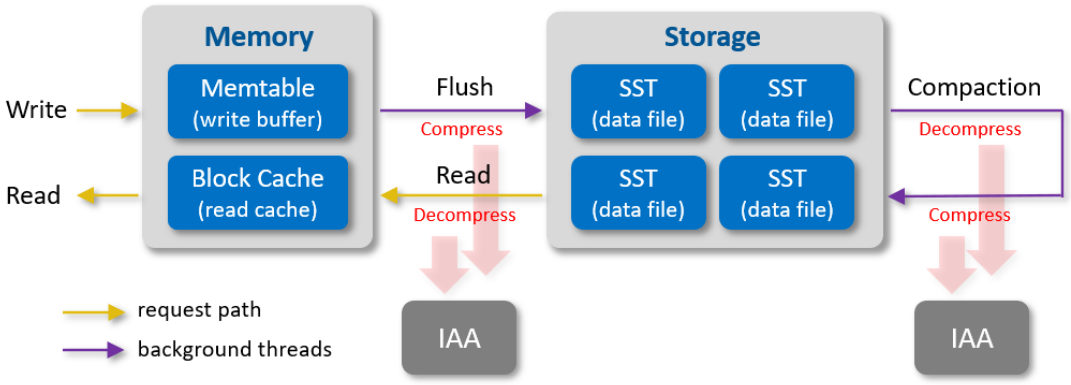

Figure 2: RocksDB Architecture*

RocksDB* Storage Engine: Introduction

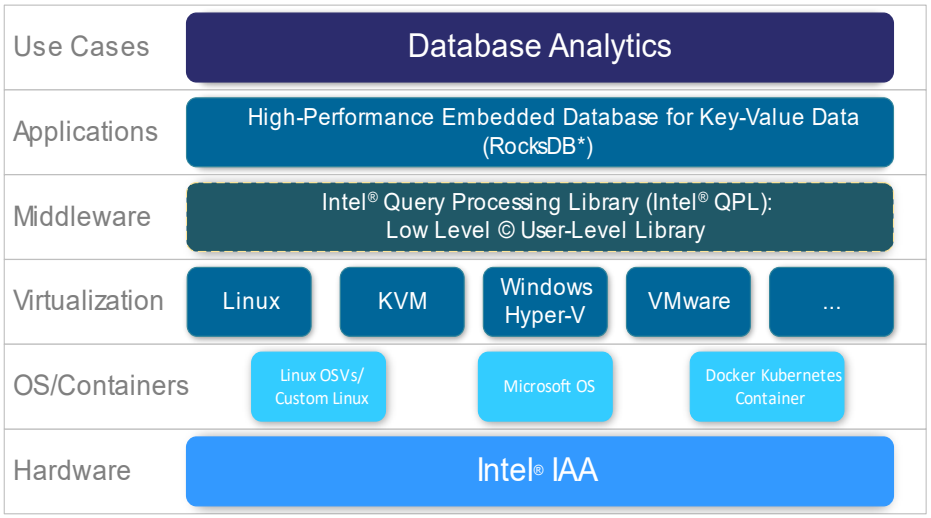

Figure 3: Intel® Query Processing Library (Intel® QPL) for Intel® IAA

RocksDB is a high-performance, open-source, embedded, key-value store. It is a widely deployed storage engine used in common database applications (e.g., MySQL, MariaDB, MongoDB, Redis), consequently improving the performance of RocksDB* benefits all applications that rely on it internally. Figure 2 captures the high-level architecture of RocksDB, including:

- Incoming writes are buffered in memory in data structures called *memtables*.
  - As memtables are filled, they are flushed to storage (e.g., SSD). RocksDB persists data in files called Sorted Sequence Tables (SST). Data can be compressed as part of this process.
- Once in storage, SST files are regularly compacted to clean up any redundant updates and execute deletions.
  - RocksDB uses a log-structured merge tree (LSM) for files in storage, dividing SSTs into levels of growing size and moving files along levels during compaction.
  - As part of the compaction process, data from source files is decompressed, compacted, and recompressed into the output files.
- For reading, RocksDB provides a block cache for frequently accessed data blocks in compressed or uncompressed form (the OS page cache can also be used).
  - In case of a cache miss, data blocks must be read from storage and decompressed to locate the desired entries.

SST files are commonly in a block-based table format. Data is thus divided into blocks of configurable size (e.g., 4kB or 16kB), and this block size is the unit of compression. Additional block types contain metadata used for searching and other features.

To enable acceleration for compression/decompression, a compressor plugin framework for RocksDB* was developed (refer to PRs 6717 and 7650)[6,7]. This aligns with the plugin framework already present in RocksDB, and it extends it to support compressor plugins.

Intel® IAA Plugin for RocksDB* Storage Engine*

The Intel IAA plugin[5] was developed for databases like RocksDB to offload compression and decompression operations to Intel IAA from the CPU.

Once RocksDB* is built with this plugin enabled, Intel IAA can be selected as a compression method just like any other integrated algorithm directly supported by RocksDB.

The Intel IAA plugin for RocksDB* uses Intel® Query Processing Library (Intel® QPL), which provides the API to interface with the Intel IAA hardware to accelerate compression/decompression operations. The Intel IAA plugin for RocksDB* Storage Engine[5] and Intel QPL[3] are available in open source. For databases other than RocksDB, developers can call Intel® Query Processing Library (Intel® QPL) directly to offload compression/decompression operations.

Intel® Query Processing Library (Intel® QPL)

The Intel QPL is the software API supporting the capabilities of Intel IAA on 4th Gen Intel Xeon Scalable processors. It supports:

- Extremely high throughput compression and decompression combined with primitive analytic functions.
- Highly optimized software fallback for other Intel CPUs.

Intel QPL primarily targets applications such as big data and in-memory analytic databases. As illustrated in Figure 3, Intel QPL sits on top of the system drivers. The applications will use Intel QPL to interface with the Intel IAA and offload analytics operations to the Intel IAA device, such as in 4th Gen Intel Xeon Scalable processors.

RocksDB* Performance Capture Methodology

Db_bench, a popular benchmarking tool distributed with RocksDB*, was used for performance measurements. db_bench includes many different parameters to customize the workload conditions and RocksDB options.

These are the db_bench settings used.

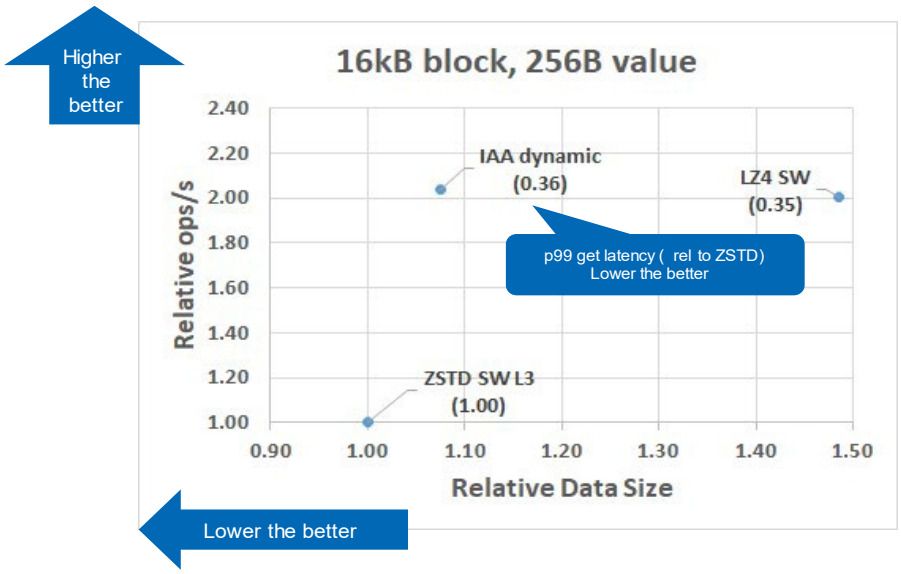

Figure 4: RocksDB—Relative Performance and Data Size Compared to ZSTD and LZ4 80:20 Read/Write Scenario.*

- **Benchmarks:** *readrandom* (read-only) and *readrandomwriterandom* (80:20 read/write ratio).
  - The database is populated using a fillseq benchmark before running workloads.
  - Read-only test exercises only decompression.
  - R/W workload exercises both compression and decompression.
- **db_bench Instances:**
  - **Read Only:**
    - Four instances.
    - Twenty-five threads per instance.
  - **Read/Write:**
    - 8 instances.
    - 10 threads per instance + background flush/compaction threads.
- **Key Size:** 16 bytes.
- **Value Size:** 32 or 256 bytes.
  - Populated using data from the Calgary corpus rather than db_bench-generated data.
  - Using the corpus required a small change to db_bench, which was contributed to RocksDB* in PR 10395.
- **Block Size:** 4kB or 16kB.
- **Data Size**: The overall data size is designed to fit the OS page cache in compressed form.
  - Eliminates dependence on storage device performance.
  - Benchmark is representative of cases where disk IO is not the bottleneck.
- **Block cache disabled:** No uncompressed data is cached.
- **Compression:** ZSTD, LZ4, Intel IAA.
  - ZSTD is a reference to compare Intel IAA performance.
  - **ZSTD:** v1.5.2.
  - LZ4 is provided to show another popular speed vs. compression ratio tradeoff.
  - **LZ4:** v1.8.3.
  - **Intel QPL for Intel IAA:** v1.1.0.

The tests were run on a sixty-core system with Intel® Hyper-Threading Technology (Intel® HT Technology) enabled. The db_bench instances were bound to one socket of a four-socket system. All four Intel IAA devices available on the socket were enabled. More information about configuring these devices can be found in *Intel® In-Memory Analytics Accelerator (Intel® IAA) User Guide*[2].

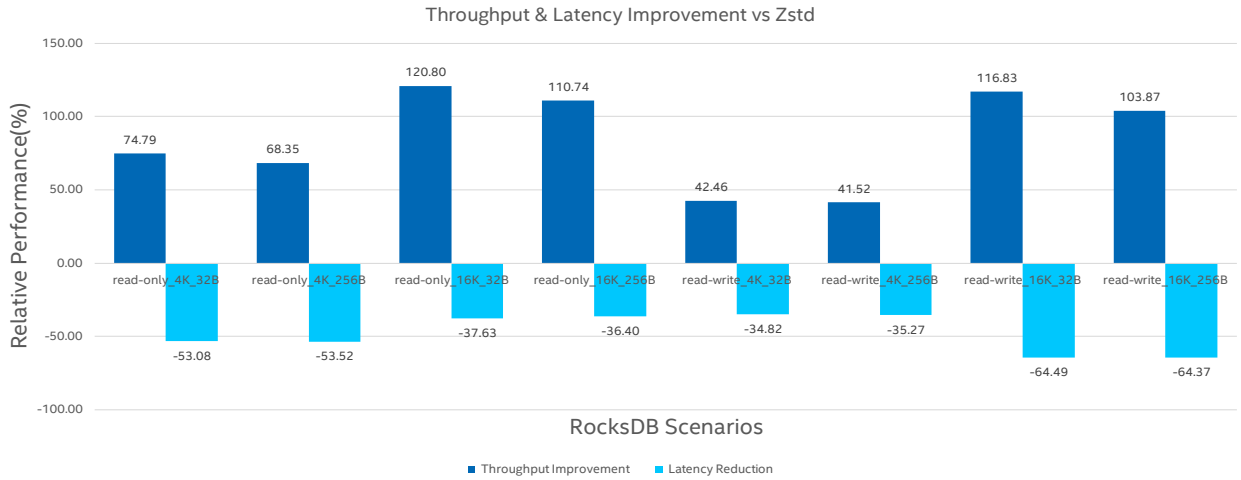

Figure 5: Throughput and Latency Improvement Across All RocksDB Scenarios*

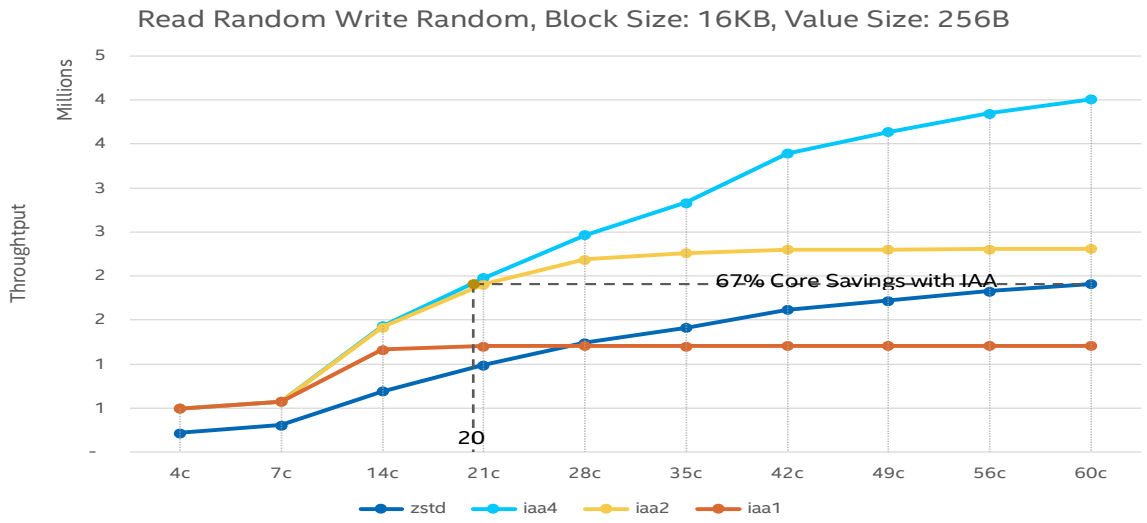

Figure 6: Throughput change with CPU core and IAA device scaling.

RocksDB* Improvement Summary

Intel IAA-accelerated RocksDB* can achieve higher throughput (operations/sec) and lower p99 get latency compared to ZSTD-based software compression across the different scenarios exercised. The performance improvement ranges for the different values, and block-size exercises follow.

- **Throughput improvement:**
  - 68–120% (read-only).
  - 41–116% (80/20 read/write).
- p99 get latency reduction:
  - 36–53% (read-only).
  - 34–64% (80/20 read/write).
- **Compressed-Data-Size Trade-Off:** 2–9%
- **CPU Savings (Same ZSTD Throughput):**
  - 58–70% (read-only).
  - 47–67% (80/20 read/write).

The relative performance of an Intel IAA vs. ZSTD depends on the access pattern, block size, and data size. Figure 4 shows the relative throughput (vertical axis) and the relative data size (horizontal axis) using ZSTD/LZ4 for a specific scenario (16Kb block size and 256-byte value size) as an example. Compared to ZSTD, the higher throughput and lower latency achievable with Intel IAA compression allow increased application performance and reduced cluster sizes. This leads to improved resource utilization and cost savings. Compared to LZ4, IAA can reduce the data size significantly (a result of the higher compression ratio of Intel IAA) to achieve the same throughput. Figure 5 summarizes the Intel IAA performance comparison against ZSTD across all the scenarios exercised.

Figure 6 plots throughput scaling across several cores with ZSTD software compression and several Intel IAA devices (one, two, and four Intel IAA devices) for 16Kb block size and 256-byte value size with an 80:20 read/write scenario. The horizontal dotted line represents the maximum throughput achieved with ZSTD. The vertical dotted line represents the number of cores that can be saved by offloading compression/decompression operations to Intel IAA while achieving the maximum ZSTD throughput.

Intel IAA can reduce the number of CPUs needed to reach the same throughput as ZSTD and at a lower latency. As in Figure 6, RocksDB with Intel IAA can achieve the same throughput as ZSTD-based RocksDB with a 67% reduction of CPU cores.

CPU core savings vary depending on the access pattern, block size, and value size. Figure 7 summarizes CPU core savings across different

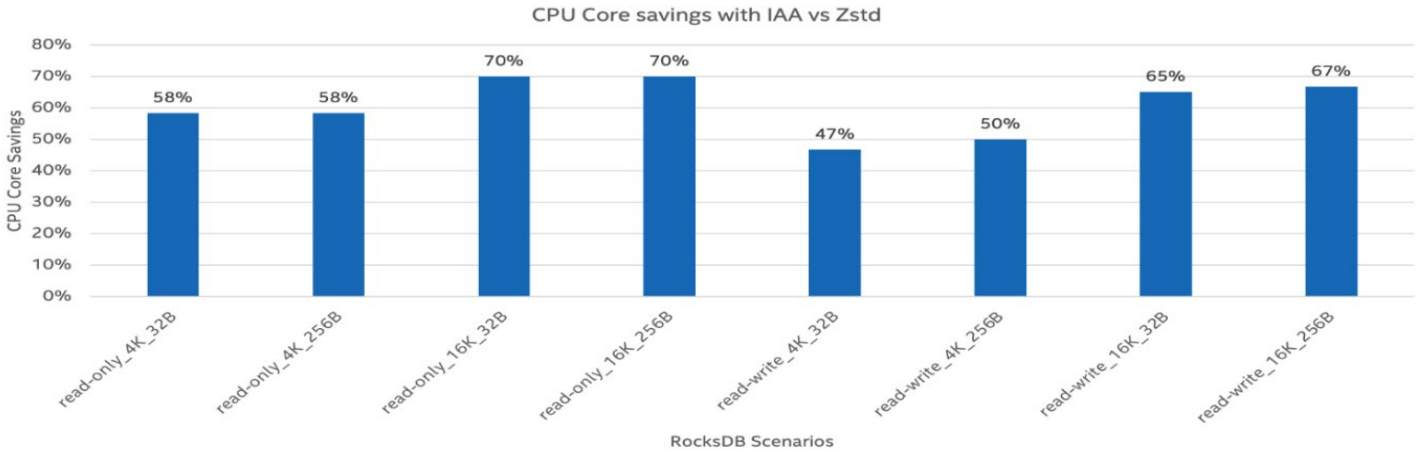

Figure 7: CPU core savings across RocksDB scenarios (higher the better)

RocksDB scenarios.

Total Cost of Ownership (TCO)

Total Cost of Ownership (TCO) is the parameter we refer to when discussing the Cost of operating a Data Center. TCO includes the cost of acquiring (Capex) a data server (CPUs, memory), its network, rack, and power infrastructure, and the cost of operating (Opex) it for four years. Figure 8 depicts various components of TCO and its breakdown[8].

As the Intel IAA accelerator is integrated, we used the term *cost-of-compute* to estimate the cost of the Intel Xeon CPU with and without an Accelerator.

Total Cost of Ownership Example

Figure 9 shows the TCO savings for a RocksDB* Storage Engine data analytics workload when the Intel IAA accelerator is used.

Using Intel IAA provides a remarkable 47%–67% savings in CPU core usage for a typical 80:20 read/write scenario. These savings are realized by offering 47%-67% more service or reducing the cost of compute of a RocksDB* server. Cost estimates are based on two assumptions:

1. CPU core savings as described in Figure 7.
2. Typical data center cost[8] as described in Figure 8.

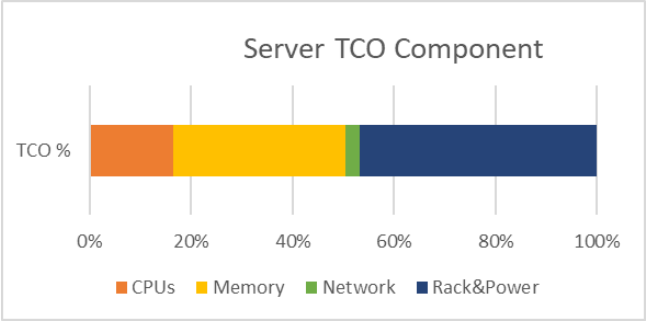

Figure 8: Typical Data Center Total Cost of Ownership (TCO) Breakdown

This cost-of-compute reduction with the IAA leads to 9% –15% overall TCO savings for a data center compared to CPUs without IAA, as illustrated in Figure 9. TCO savings vary depending on RocksDB usage. As shown in Figure 7, CPU core savings are higher for 16 KB block size than 4KB block size. This reduction in compute costs is directly reflected in TCO savings shown in Figure 9.

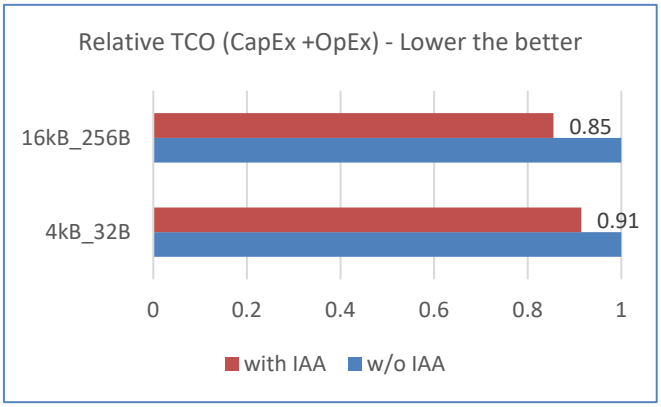

Figure 9: Relative Total Cost of Ownership (TCO) with and w/o IAA

Conclusion

Integrated accelerators are a major leap in innovation featured in 4th Gen Intel Xeon Scalable processors. Intel IAA, one of the Integrated accelerators in 4th Generation Intel Xeon Scalable processors, is designed to provide remarkably high throughput and low-latency compression/decompression operations and analytic primitive functions.

Intel IAA enhances the performance of key-value storage libraries like RocksDB, leading to increased application speed, reduced costs, and improved power efficiency.

Popular databases like Apache Cassandra, CockroachDB, MySQL (MyRocks), and Rockset can also harness the power of Intel IAA-accelerated RocksDB*. With seamless interfaces offered by Intel® QPL, these databases can effortlessly offload compression, decompression, and analytics operations to Intel IAA devices to boost application performance and improve power efficiency.

Configuration

Testing by Intel as of 3/17/2023.

- **Configuration:** 1-node, 2 sockets Next Gen Intel Xeon Scalable processor (60-cores, 4x IAA devices) pre-production platform with 512GB (16x32GB DDR5-4800MT/s) total memory, HT on, Turbo on.
- **BIOS**: EGSDCRB1.86B.9409.P15.2301131123.
- **OS:** CentOS Stream 8, Linux Kernel: 6.1.12, internal RocksDB v8.0.0 with pluggable compression support (db_bench).
- **Software Configuration:** GCC 8.5.0, ZSTD v1.5.2, p99 latency used. Performance measured on bare metal with a single-socket of a 2-socket system. Results depend on block size and database entry size. Tradeoff 2-9% in compressed data size.

References

1. Intel Corporation, "Intel® In-Memory Analytics Accelerator (Intel® IAA) Architecture Specification," 2023. [Online]. Available: https://www.intel.com/content/www/us/en/content-details/721858/intel-in-memory-analytics-accelerator-architecture-specification.html.
2. Intel Corporation, "Intel® In-Memory Analytics Accelerator (Intel® IAA) User Guide," 2023. [Online]. Available: https://cdrdv2.intel.com/v1/dl/getContent/780887.
3. Intel Corporation, "Welcome to Intel® QPL Documentation! — Intel® QPL v1.1.0 Documentation.," 2023. [Online]. Available: https://intel.github.io/qpl/.
4. Intel Corporation, "Add file-based data generation options to db_bench by missa-prime · Pull Request #10395," 2022. [Online]. Available: https://github.com/facebook/rocksdb/pull/10395.
5. Intel Corporation, "RocksDB pluggable compression," 2022. [Online]. Available: https://github.com/intel/iaa-plugin-rocksdb.
6. Intel Corporation, "Add Compressor interface #7650. GitHub/Facebook/Rocksdb," 2020. [Online]. Available: https://github.com/facebook/rocksdb/pull/7650.
7. Intel Corporation, " Add support for compressor plugins by lucagiac81 · Pull Request #6717," 2020. [Online]. Available: https://github.com/facebook/rocksdb/pull/6717.
8. TWiki @ Cern, “CostEst < Main < TWiki.” 2017. [Online]. Available: https://twiki.cern.ch/twiki/bin/view/Main/CostEst.

Notices & Disclaimers