PERCONA

WHITEPAPER

Testing the Value of ScaleFlux Computational Storage Drive (CSD) for MySQL

Can CSDs Deliver Compression Without Compromising Performance?

PERCONA

Overview

As an independent, unbiased, open source database expert, Percona is often asked to test and evaluate open source software. ScaleFlux approached Percona to benchmark the ScaleFlux® CSD 2000 against a similar Intel drive.

ScaleFlux claims that running MySQL on their drive enables customers to use less Flash and achieve better database performance - both in terms of Queries Per Second (QPS) and Latency. They attribute these benefits to two features: (1) their innovative integration of hardware compression/decompression directly into the drive and (2) support for Atomic Writes. They refer to the first feature as transparent compression since the application or host does not need to do anything to trigger the compression function, it happens automatically. Since compression usually adds latency and decreases performance, we were interested to test the ScaleFlux claims. Enabling Atomic Writes allows us to turn off the Double Write Buffer in MySQL (details discussed in the "Atomic Writes" section).

This white paper describes how we tested the claims that ScaleFlux makes regarding the CSD 2000 and gives information on the performance and results that Percona measured.

Percona performed benchmarks on the ScaleFlux CSD 2000 drive and a similar Intel NVMe drive. The tests were conducted on two X86 servers provided by ScaleFlux, which consisted of one database machine and one application node.

Percona and ScaleFlux agreed to perform three different tests:
- Simple Read access
- Read and Write operations
- Write only operation

The drives used in these tests were:
- ScaleFlux - CSD 2000 4TB
- Intel - P4610 3.2TB

Initial Expectations

We expected to see a performance benefit for Write/Read-Write workloads for two reasons:
- ScaleFlux's datasheet lists high mixed and write-only IOPs based on their testing
- We were able to disable the Double Write Buffer when using theScaleFlux Drive, taking advantage of its atomic write feature.

For Read workload we expected the ScaleFlux Drive to show similar performance to the Intel drive since the compression function does not improve their Random Read IOPs.

Tests and Results Summary

Tests

We ran three workloads (Read/Write, Write Only and Read Only) against each drive under a series of conditions:

Datasets	DoubleWrite Buffer	Compression Solution
Sysbench Default Data Schema with 220G and 2.3T datasets	ScaleFlux: DWB On and Off	ScaleFlux: CSD 2000 built-in transparent compression
Sysbench Modified Data Schema with 440G and 2.5T datasets	Intel: DWB On	Intel: No Compression and InnoDB compression
Sysbench Modified Data Schema with a 4.7T dataset (ScaleFlux only)		

(See Appendix 1 for the full details of the test parameters).

Results

Based on our limited testing, our results indicate that CSD 2000 seems to deliver on ScaleFlux's claims.

The results showed that MySQL performs better on the ScaleFlux drive (storing 4.7TB on 3.2TB of NAND) than on the Intel drive (storing 2.5TB on 3.2TB of NAND) in a significant number of tests.

Nominal Capacity Results

The results with the datasets up to 2.5T showed that MySQL performed better on the ScaleFlux drive in a significant number of the tests where the dataset was compressible.

The CSD 2000 produced much higher queries per second, with lower latency and lower IOwait times for Read/Write and Write Only workloads. In Read only workloads, or when the data set doesn't compress well (as with the default data schema), the Intel and Scaleflux drives performed closely to each other.

The results showed that MySQL performs better on the ScaleFlux drive (storing 4.7TB on 3.2TB of NAND) than on the Intel drive (storing 2.5TB on 3.2TB of NAND) in a significant number of tests.

In a typical MySQL operation, however, we would expect 10-30% of the traffic to be Writes, with additional Write bursts that are significantly higher. When the application starts to Write and/or has some compressibility in the data, ScaleFlux has a clear advantage. As we increased the data set size from 220G, to 440G, all the way to 2.5TB, the performance benefits of the ScaleFlux drive became more pronounced.

Further testing with higher compressibility levels could show additional gains. But, that testing was beyond the scope of this project.

Extended Capacity Results

We ran the same tests using a 4.7TB dataset to measure the performance of the ScaleFlux drive with its "Extended Capacity" feature. ScaleFlux allows users to format the drive to a larger logical capacity than it has physical capacity. This lets users take advantage of the compression function to store more data on the drive. Clearly, we could not run the 4.7TB dataset with the 3.2TB Intel drive.

The results showed that MySQL performs better on the ScaleFlux drive (storing 4.7TB on 3.2TB of NAND) than on the Intel drive (storing 2.5TB on 3.2TB of NAND) in a significant number of tests.

In the Read/Write tests, the ScaleFlux drive produced significantly higher results for most thread counts. In the Write Only tests, the results were similar for both the ScaleFlux and Intel drives. In the Read Only tests, the results were fairly equal up to 64 threads, with the Intel drive outperforming ScaleFlux for 96+ threads.

Notes on Compression Ratio

Typically, MySQL databases are made up of several fields containing a mix of random and non-random data types. This mix of data types results in the overall database having some compressibility. A range of 1.5:1 to 5:1 compression is commonly achievable.

The ScaleFlux drive has built-in compression. With the modified schema dataset we ran, when the OS reported the logical data size was 2.5T, physically it only took up 1.4T on the CSD 2000 (1.8:1 compression ratio).

We also tested table-level compression (zlib) with the Intel drive. In this case, the 2.5T of data was compressed into 1.6T (1.6:1), but performance also dropped as is shown in the charts in the Results Details section.

The CSD 2000 beat InnoDB's table-level compression in both the compression ratio achieved and overall performance. Further testing with higher compressibility levels could show additional performance gains, but that testing was beyond the scope of this project.

About the ScaleFlux CSD 2000

The ScaleFlux Computational Storage Drive CSD 2000 Series claims to provide exceptional performance, scalability and a LowerTotal Cost of Ownership (TCO) for mainstream Flash storage deployments.

Our results indicate that ScaleFlux's devices maintain a significantly higher performance in mixed workload scenarios

ScaleFlux drives include all the features you expect in an enterprise class SSD. Additionally, ScaleFlux integrates hardware accelerated compute engines into their drives - hence the term "Computational Storage Drive" instead of just "Solid State Drive". These integrated compute engines allow ScaleFlux CSDs to achieve exceptional Read/Write data speeds, consistently lower latency, and a lower total cost of ownership.

The CSD 2000 is promoted as setting a new standard for performance consistency across workloads (random Read/Write IOPs). As our test matrix covered a variety of workloads, settings and datasets, standard SSDs might perform better in some cases - we see this with smaller datasets. Our results indicate that ScaleFlux's devices maintain a significantly higher performance in mixed workload scenarios - these are scenarios where other SSDs' performance rapidly drops off. The mixed workload performance is beneficial for MySQL as a typical MySQL workload includes a mix of Reads and Writes.

ScaleFlux claims that the CSD 2000 allows users to reduce TCO by:

1. Increasing TPS per server for better overall datacenter efficiency
2. Reducing effective cost per GB with Data Compression
3. Enabling in-field upgrades to the compute functions

The CSD 2000 drive features include:

- PCIe Gen 3 X4 host interface
- Add in Card and U.2 Drive form factors
- Up to 16TB Effective Capacity with data path compression (4/8TB raw)
- Transparent GZIP Compression / Decompression Engine
- Variable Length Mapping (VLM) Flash Translation Layer
- Adjustable drive settings to optimize performance and $/GB
- Throttling to avoid overheating and comply with slot power limitations
- End-to-end data protection and ECC (Error Correction Code) on all internal memories in the data path; Integrated LDPC engine and Flash die RAID assures 10-20 UBER
- Complete data protection from unplanned power loss

What are the Benefits of ScaleFlux CSD 2000 for MySQL?

Transparent Compression/Decompression

ScaleFlux has implemented a novel feature for SSDs - an automatic data compression/decompression feature. The CSD 2000 compresses all data written to the drive, and then decompresses the data as it is read.

MySQL data can be moderately to highly compressible

Compression has been around for a long time as a way to save storage space and reduce storage cost. However, running software compression in the CPU generally reduces application performance and increases latency. Also, software-based compression isn't scalable. Software-based throughput is limited to 10s, or a few 100, MB/s and can not keep up with the 2-3GB/s throughput per drive of modern PCIe SSDs.

With the ScaleFlux CSD 2000, every drive includes a dedicated hardware engine to perform the compression and decompression algorithms. So, the throughput scales with every drive added to the system.

This compression/decompression function can actually reduce latency and improve database performance. ScaleFlux attributes these improvements to reductions in Write amplification, garbage collection, and NAND die contention. (For further information on the write amplification and background garbage collection benefits, please consult with ScaleFlux directly.)

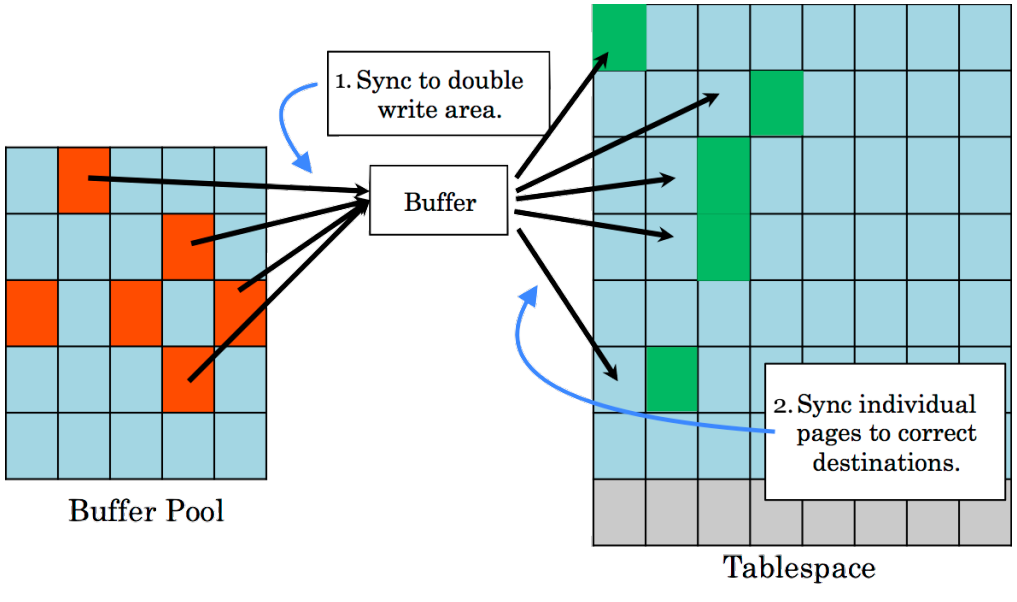
Compression/decompression can improve the effective cost of the drives. As shown in this testing, MySQL data can be moderately to highly compressible. ScaleFlux converts that compressibility into extra capacity to store data. When we included two varchar fields in the Sysbench data schema, we saw 1.8:1 compression from the CSD 2000. When discussing the benchmark results with the ScaleFlux team, they noted that their database customers report 2:1 to 5:1 compression ratios.

It is also worth noting that it is not necessary to change any code in MySQL to use this compression/decompression feature - it happens automatically.

Atomic Writes

ScaleFlux has ensured that their CSD 2000 drive can perform Atomic Writes. This is not a common feature among NVMe SSDs. In our testing we were able to disable InnoDB DoubleWrite Buffer, because the ScaleFlux drive is atomic for 4k/8k/16k/32k/64k when O_Direct is used.

Turning off DoubleWrite eliminates almost half of the fsyncs, giving higher Write performance and higher SSD endurance.



If DoubleWrite is enabled, when InnoDB Writes down dirty pages from the Buffer Pool to the final tablespace, it is going to Write first in an area called DoubleWrite. From there, it is going to sync them to the final tablespace. That is two fsync on every Write. This is a safety mechanism to protect against data loss and data corruption, which would occur if a Write was interrupted with only part of a record updated.

If the underlying storage layer supports Atomic Writes, DoubleWrite is no longer necessary. The SSD guarantees that either the entire Write operation completes, or none of it does. This prevents data corruption and data loss.

What Did Percona Test?

The test matrix was designed to identify performance differences between the ScaleFlux and Intel drives using the same schema and workload. See Appendix 1 for additional details on the test parameters.

We tested the following combinations of Workloads and MySQL settings:

	ScaleFlux NO DoubleWrite	ScaleFlux DoubleWrite	Intel DoubleWrite	Intel Compressed DoubleWrite InnoDB compression
Read Only	x	x	x	x
Write Only	x	x	x	x
Read/Write	x	x	x	x

Workloads

The first aspect of the test matrix was the workloads. We simulated the three most common scenarios using Percona Server version 8.0.19: Read/Write, Read Only and Write Only. These workloads were run across a range of thread counts (from 1 to 128) to simulate the different pressure levels you might encounter in a live deployment.

MySQL settings

The next element was the Double Write Buffer and Compression settings.

Because the ScaleFlux drive has Atomic Writes we could disable the Double Write Buffer in Percona Server. As the Intel drive is not atomic, we could not disable the Double Write Buffer in a production environment without risking data corruption.

To directly compare compression solutions, we enabled InnoDB table level compression on the Intel drive to see the performance difference. This compression uses server resources like the CPU to compress and decompress the InnoDB Pages constantly. For the ScaleFlux drive, we relied on the drive's built-in compression.

Intel showed an advantage at lower thread counts and with the smaller dataset. ScaleFlux showed an advantage with the larger dataset and at higher thread counts.

Datasets

The last variable was the dataset. We ran with the Sysbench default data schema and a modified data schema as described below.

In our baseline tests with Sysbench's default data schema, the data was either not compressible, or only slightly compressible. We ran 220G and 2.3T datasets to assess the performance when the drives were relatively empty (<20% utilized) and relatively full (>50% utilized).

To create a more realistic dataset (i.e. one that was not just random characters), we created a modified data schema. Again we ran multiple dataset sizes (440G, 2.5T, and 4.7T) to assess the performance across various fill levels of the drives. In this schema, we added two varchar fields to the database. A real world example of this is a database in which some fields are encrypted (e.g. credit card numbers) while others are not (e.g. product descriptions). Because of these extra fields and queries, the tests resulted in fewer queries per second than the traditional sysbench tests.

Results

Table 1 summarizes the results from the full matrix of tests with InnoDB compression disabled.

With the minimally compressible datasets (default schema), the results were mixed. Intel showed an advantage at lower thread counts and with the smaller dataset. ScaleFlux showed an advantage with the larger dataset and at higher thread counts.

With the more compressible dataset (modified schema), the ScaleFlux drive performed better than the Intel drive. The differences were more pronounced with the larger dataset.

Table 1: Average Performance Difference for Across 1-256 Threads (ScaleFlux - Intel)/Intel

	Read/Write	Write Only	Read Only
Sysbench Default Data (minimally compressible)			
220G	1%	-19%	-1%
2.3T	-12%	19%	-14%
Modified Data Set (1.8:1 compressible)			
440G	44%	65%	11%
2.5T	77%	104%	19%
4.7T	N/A (as only ScaleFlux can run these tests)		

*Scores indicate the % higher (lower) as calculated by: ((ScaleFlux QPS - Intel QPS) / Intel QPS)*100.*

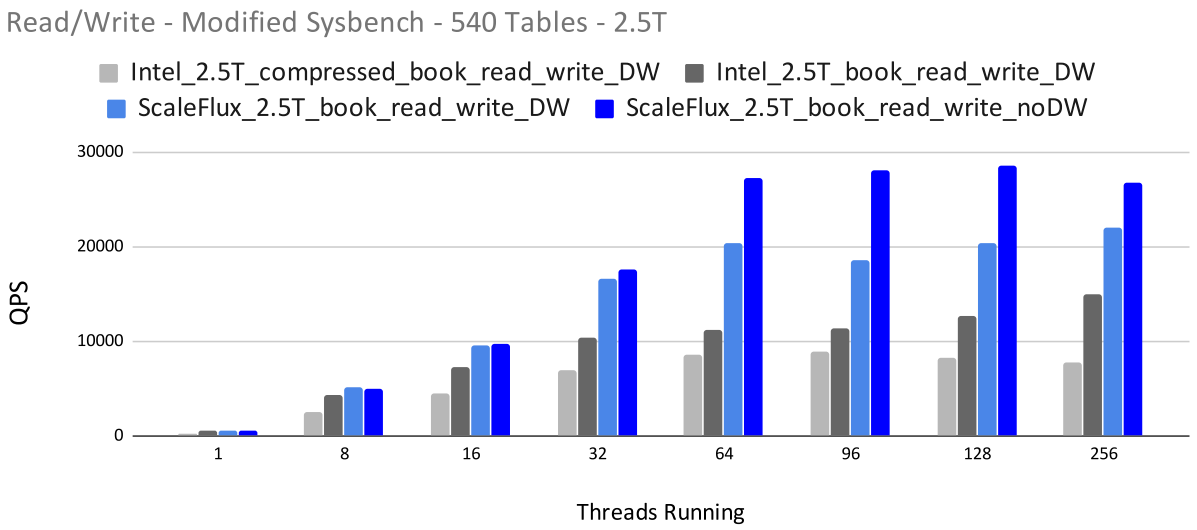
Detailed Results

Below, we discuss the detailed results of the Modified Data Schema for the 2.5T and 4.7T datasets. These two datasets represent the user experience after the drives have been deployed and filled to a moderate level.

The 2.5T dataset is 78% of the rated drive capacities of 3.2T. With the compression in the ScaleFlux drive, the 4.7T dataset filled 83% (2.6T) of the drive capacity. Most administrators avoid filling the drives much more than this to avoid performance issues.

2.5T Modified Sysbench (540 Tables)

Read/Write



Read/Write Results Commentary

After seeing the Read Only and Write Only results with the larger dataset, we expected the ScaleFlux drive to show impressive benefits in the Mixed Read/Write workload. These results met our expectations and showed that with the ScaleFlux drive running built-in compression it ramps to over 3x the performance of the Intel drive with InnoDB table level compression.

CPU Usage for Read/Write - Modified Sysbench - 540 tables - 2.5T

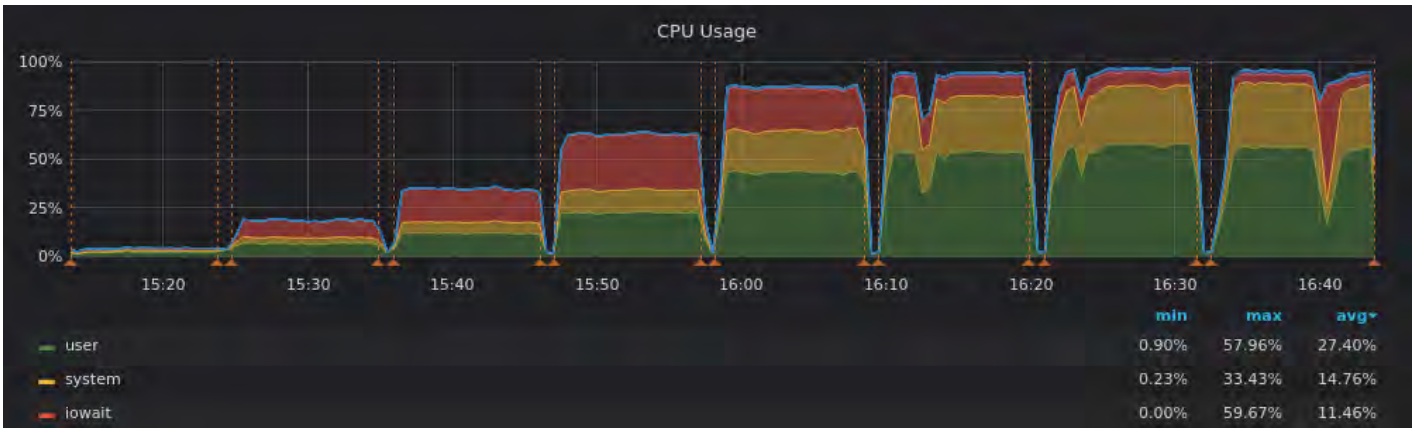
Overall, the reduced IOwait allows more of the CPU cycles to be used to work on the queries. This is shown by the larger percentage of “user” in the utilization.

In this test, we looked at how much time it takes the CPU to actively work on queries versus in a wait state. In the first table (ScaleFlux - Read/Write) the green wedges represent the percent of time the CPU is actively working (bigger is better). The red wedges represent the percent of time the CPU is waiting for I/Os to complete (smaller is better).

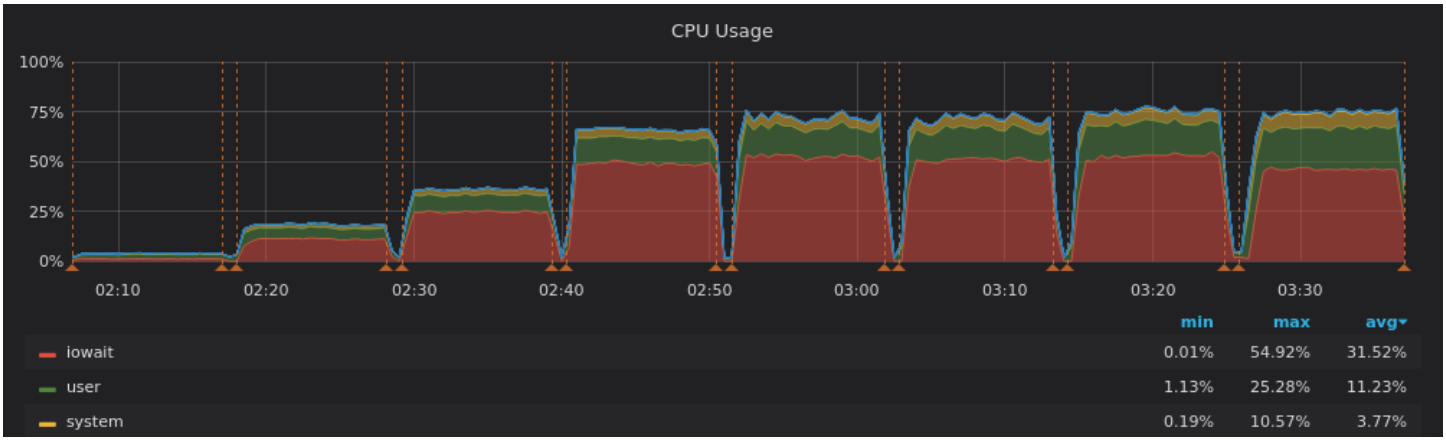
In the second table (Intel - Read/Write) the test shows much higher IOwait is with the Intel drive. This might be because it has to Write more as the InnoDB DoubleWrite is enabled.

The final table (ScaleFlux - Read/Write with DoubleWrite) shows what happened when we tested the ScaleFlux drive when DoubleWrite was enabled. The results in the third table show that the IOwait is still lower than on the Intel drive, but not that stable, as fluctuations are much higher.

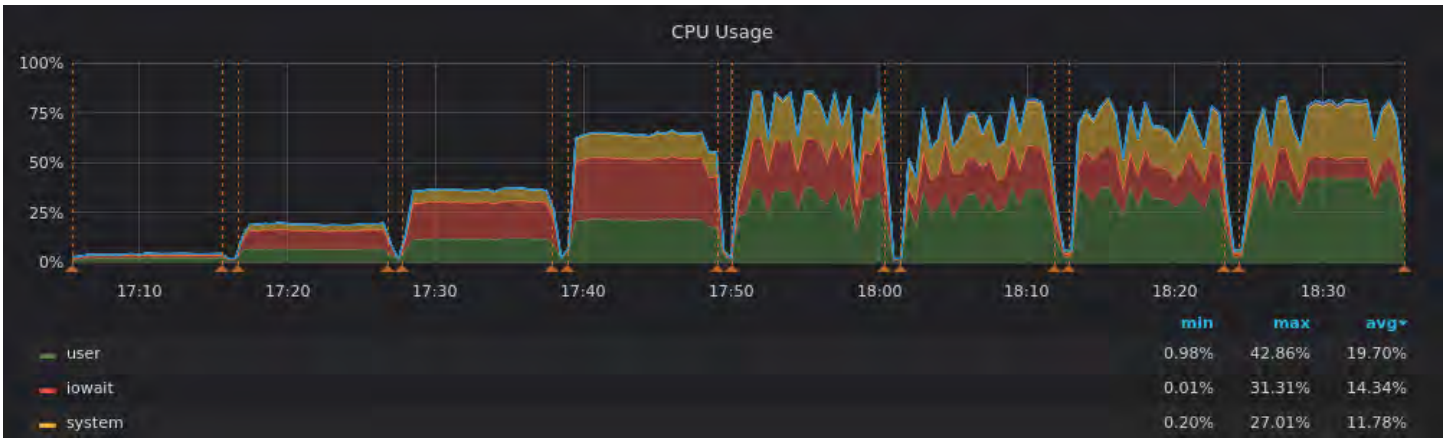
ScaleFlux - Read/Write - Modified Sysbench - 540 tables - 2.5T



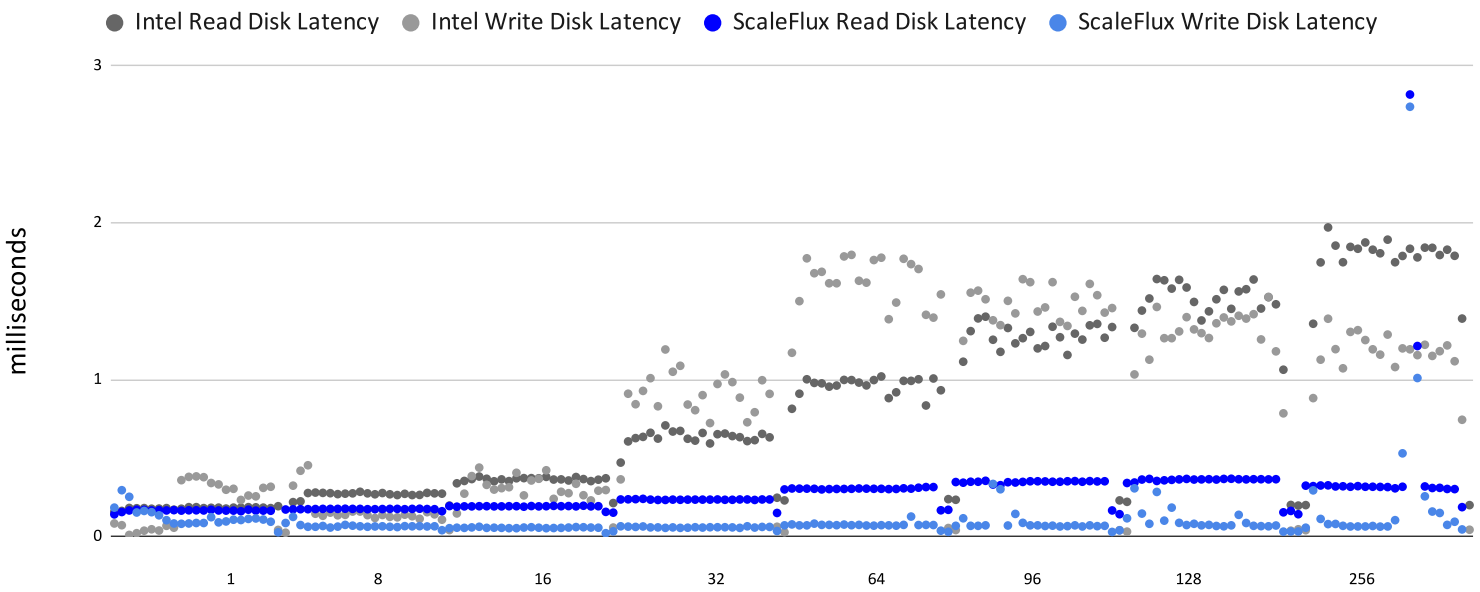
Intel - Read/Write - Modified Sysbench - 540 tables - 2.5T



ScaleFlux - Read/Write with DoubleWrite - Modified Sysbench - 540 tables - 2.5T



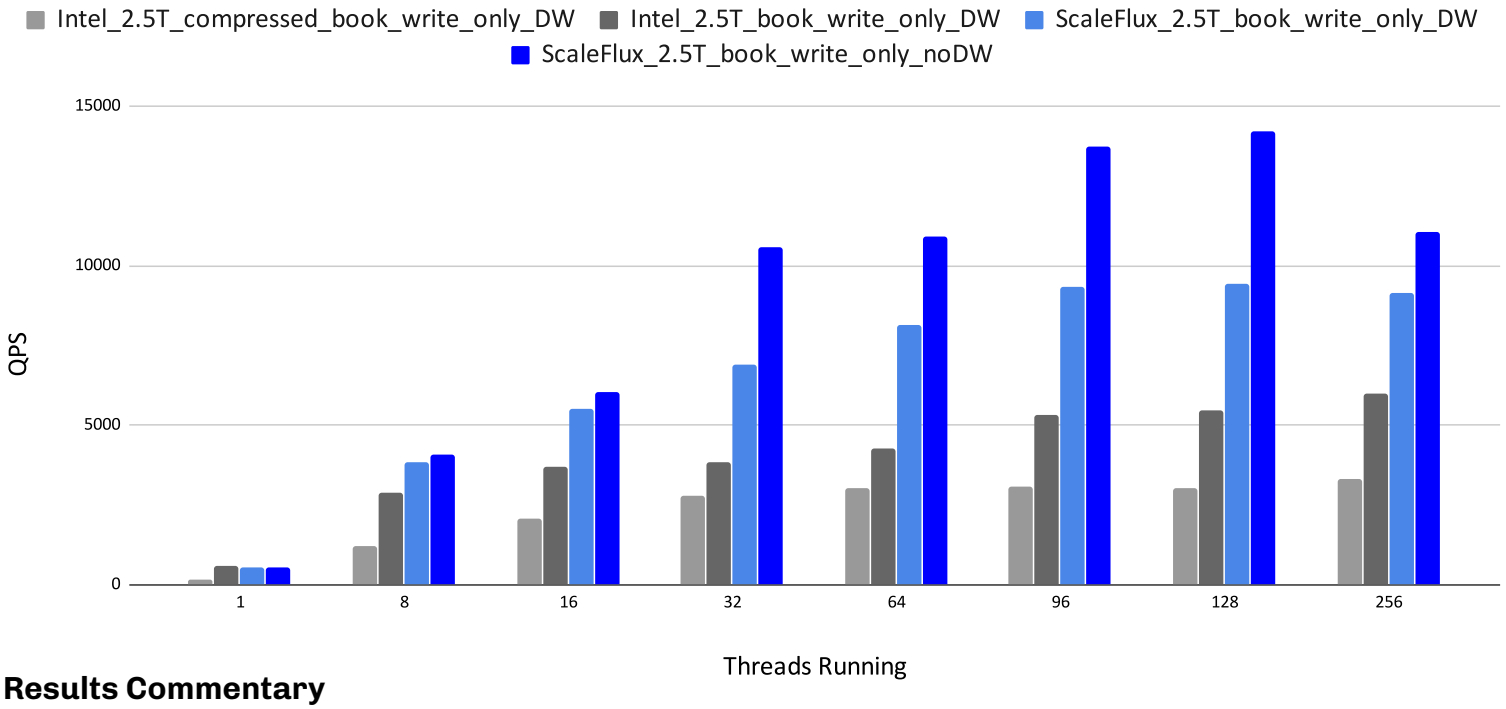
Disk Latency for Read/Write - Modified Sysbench - 540 tables - 2.5T



Results Commentary

These metrics are coming from the Operating System (OS). These are the numbers that the OS reported for disk latency. **The Scaleflux drive clearly shows a lower latency here**, which could explain why it performs better in some of the tests, and looks more stable.

Write Only - Modified Sysbench - 540 tables - 2.5T



Results Commentary

We expected to see the biggest difference between the drives on this test. With a large dataset, we would expect garbage collection to kick in and reduce Write performance. Both with and without Double Write Buffer MySQL performs far better on the ScaleFlux drive with a Write Only workload.

We also checked the performance of the Intel drive using InnoDB table level compression. Running compression on the host severely reduced performance.

How is the ScaleFlux drive able to achieve this and why is the built-in compression of the ScaleFlux drive important? This is because although the logical size of the dataset is 2.5T (around 80% of the drive capacity), due to compression it takes only up around 1.4T physically on the disk. This is less than 50% of the drive capacity, so garbage collection does not play a role on the ScaleFlux drive here.

This test confirmed that ScaleFlux built-in compression significantly outperforms CPU-based compression.

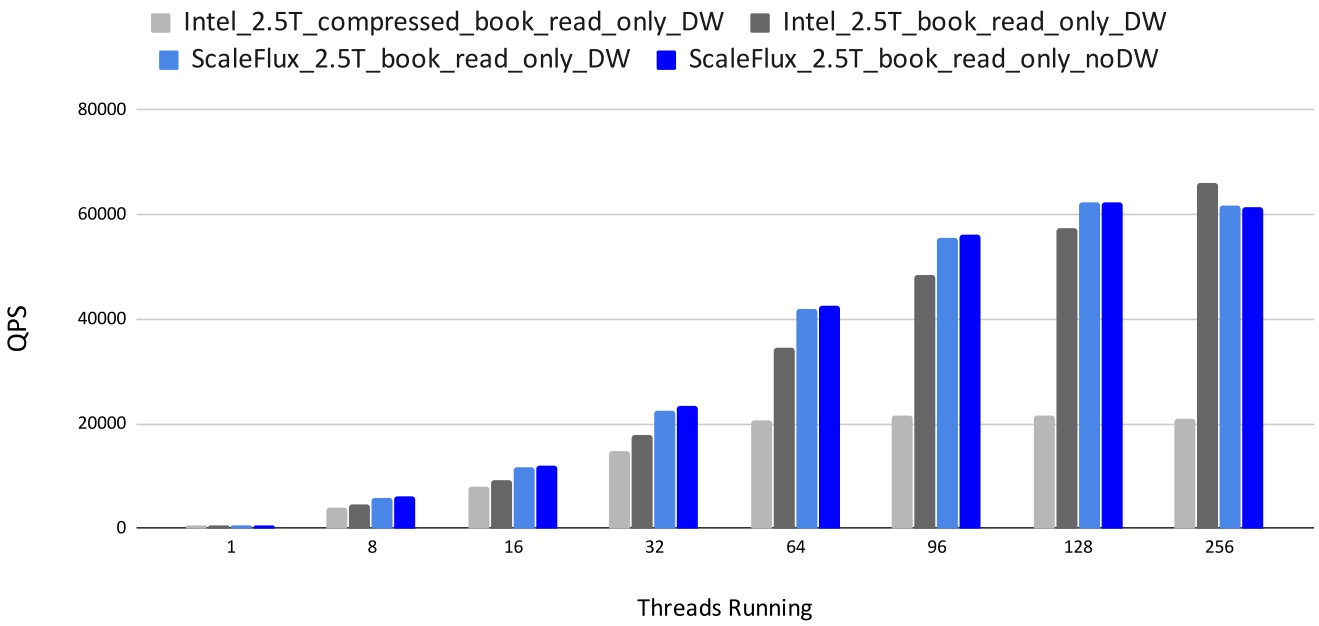
Utilizing the ScaleFlux drive in this situation gives users a significant advantage, as it can perform better even when the drive is close to the logical capacity.

Write Only Latency - Modified Sysbench - 540 tables - 2.5T - Less is better

Results Commentary

On the Write latency the drives are close to each other up until 16 threads. After that intel drive has a big jump. We then see a similar big jump on the ScaleFlux drive at 256 threads. Lower latency means faster queries and a faster application.

Read Only - Modified Sysbench - 540 tables - 2.5T



Results Commentary

With Read Only traffic MySQL performs better on the ScaleFlux drive, despite the ScaleFlux drive having built-in compression/decompression. To more clearly see the comparison between the ScaleFlux built-in compression, with software-based compression running on the CPU, we turned on MySQL table-level compression with the Intel drive.

MySQL compression consumed much of the CPU, causing the performance to plateau at just 64 threads and 20k TPS.

This test confirmed ScaleFlux's claim that the drive could **provide the space savings of compression, without sacrificing performance.**

Extended Capacity Tests - Modified Sysbench - 540 tables - 4.7T

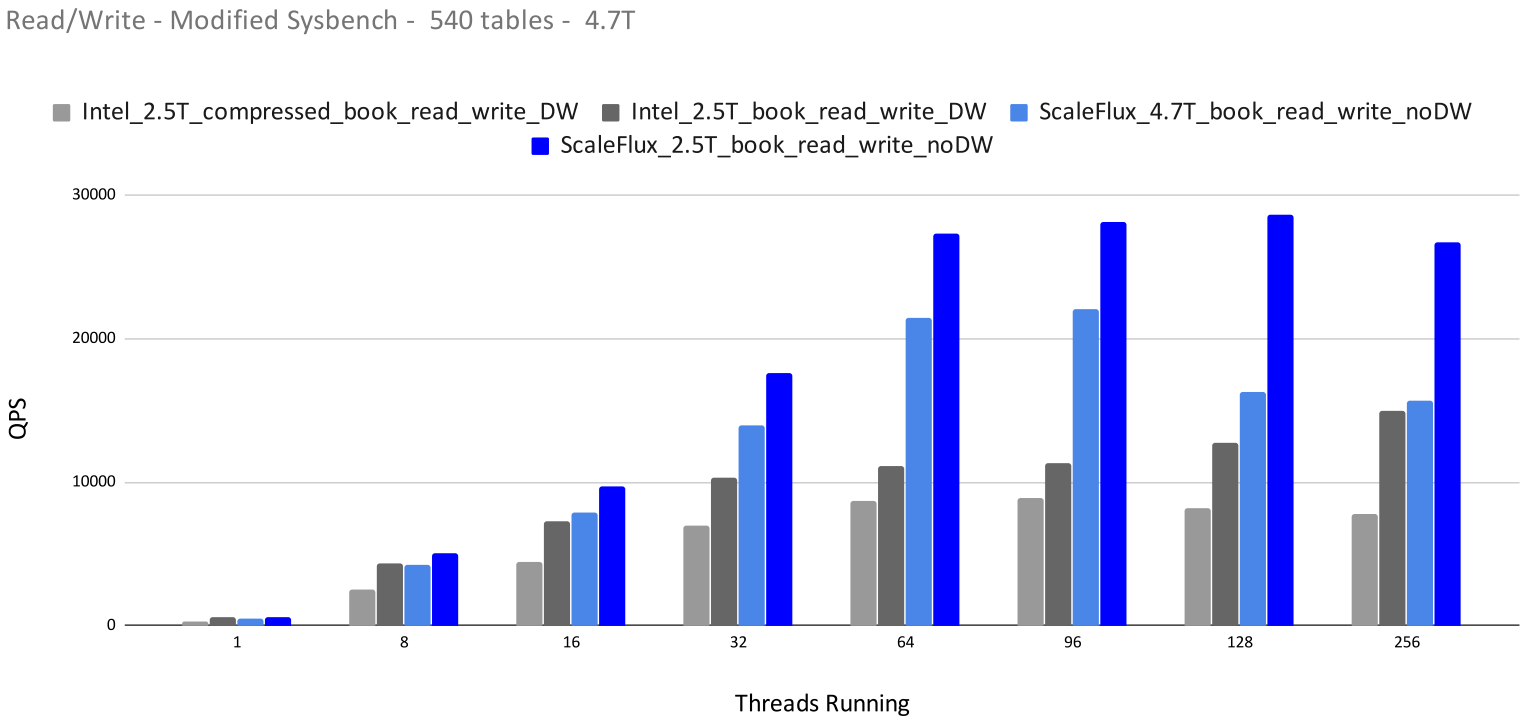
As the ScaleFlux drive has built-in compression it means we could actually Write more than 3.2T data to a 3.2T drive. If the compression ratio is 2x in theory we could Write around 6T.

To prove that we can Write more data on the drive than it has physical capacity for, we filled the ScaleFlux drive with 4.7T of data.

As before, we used 540 tables, but now each table had 20M rows. On the Intel drive this was not possible. We could have used the InnoDB table level compression, but we have already evidenced that it does not perform well.

The charts below compare the performance results of running a 4.7T dataset on the ScaleFlux drive versus running a 2.5T dataset on both the Intel drive and the ScaleFlux drive.

Read/Write



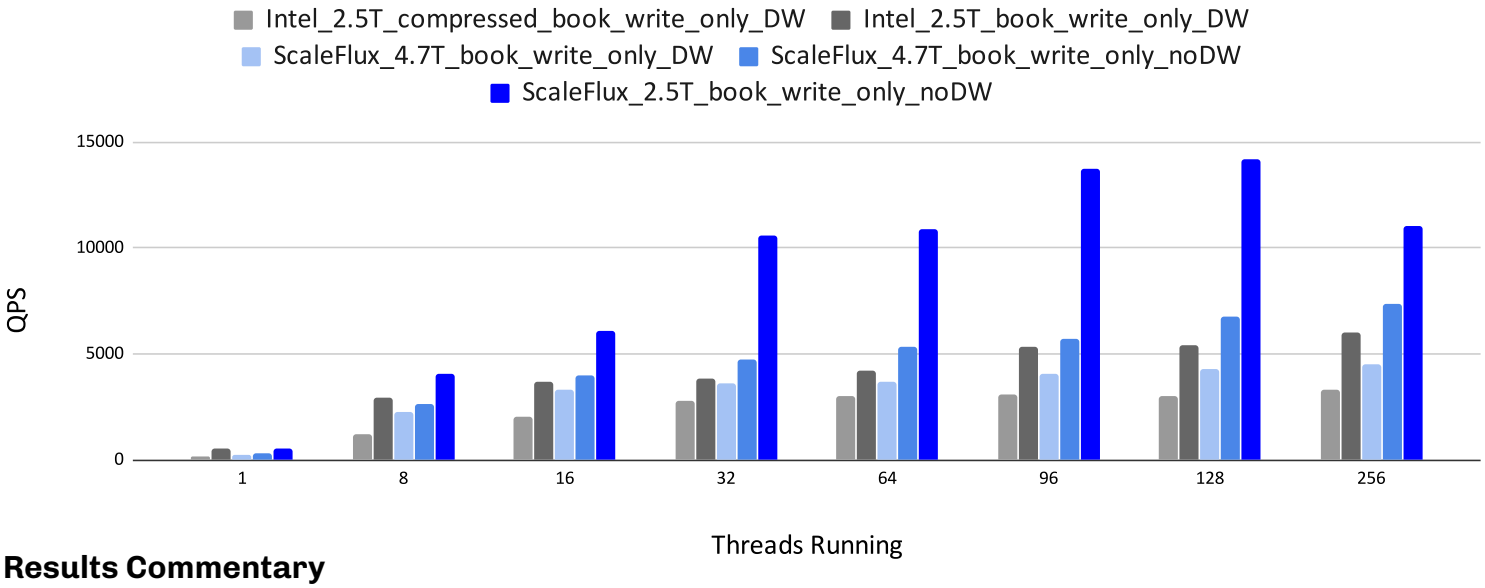
Results Commentary

With Read/Write traffic MySQL performs better on the ScaleFlux drive with 4.7T data than on Intel with 2.5T data. Using host compression significantly impairs the performance of the system using the Intel drive, producing the lowest performance of the four scenarios.

Because of built-in compression, using the same drive capacity we were able to store more data and still have better performance. This is a valuable result as typical database operations include a mix of both Reads and Writes.

Write Only

Write Only - Modified Sysbench - 540 tables - 4.7T



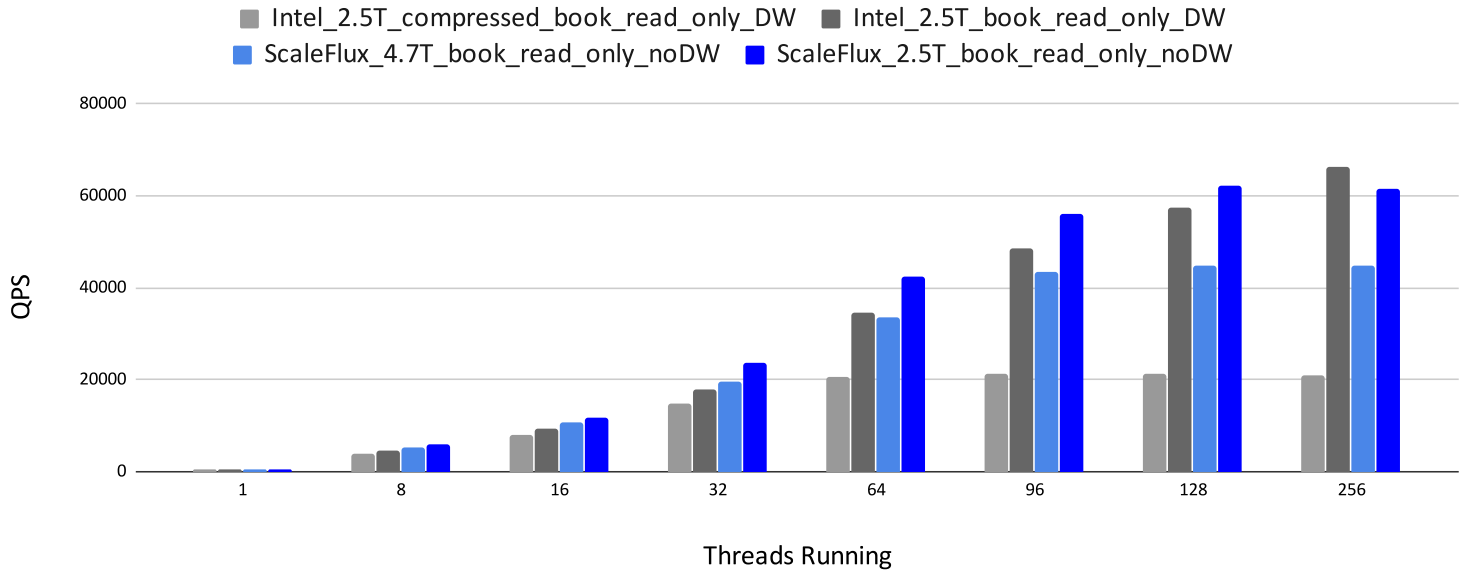
Results Commentary

The ScaleFlux drive with 4.7T data and without Double Write actually performed slightly better than Intel with 2.5T data. Even with Double Write enabled the ScaleFlux drive performs similarly until 64 threads.

This means that **on the same size drive users can store more data and still have similar, or even better, Write performance.**

Read Only

Read Only - Modified Sysbench - 540 tables - 4.7T



Results Commentary

Up until 64 threads MySQL performs similarly on Intel with 2.5T and on ScaleFlux with 4.7T. After that, Intel 2.5T has better performance. On the ScaleFlux drive, the 4.7T was actually only taking around 2.3T of physical space. ScaleFlux told us they have had customer feedback that databases with more fields often result in higher compression ratios.

Conclusion - How ScaleFlux Can Benefit Your Business

As discussed, the ScaleFlux drive offers two major features: Atomic Writes and built-in compression.

Our testing verifies ScalesFlux's claims that if a customer has a Write heavy or mixed Read/Write workload the ScaleFlux drive can improve their performance.

We were also able to verify the claim of extending the usable capacity beyond the physical capacity of the drive. Using the ScaleFlux drive, the application can perform more Writes with lower latency while taking up less storage because of the built-in transparent compression.

Based on the benchmark testing and the capabilities of the ScaleFlux CSD 2000, it is well suited to customers who are concerned with:

- Consistent performance and latency in common transactional database workloads to meet SLAs
- Reducing their overall Flash storage costs without sacrificing performance
- Extending the life of their SSD.

Further test result information can be reviewed on the Percona Blog, or contact ScaleFlux directly for more details.

Contact Us

We can provide onsite or remote Percona Consulting for current or planned projects, migrations, or emergency situations. Every engagement is unique and we work alongside you to plan and create the most effective solutions for your business.

Percona Managed Services can support and help you manage your existing database infrastructure; whether hosted on-premise, or at a co-location facility, or if you purchase services from a cloud provider or database-as-as-ervice provider.

To learn more about how Percona can help, and for pricing information, please contact us at +1-888-316-9775 (USA), +44 203 608 6727 (Europe), or email us at sales@percona.com.

Appendix 1

How Percona Tested

Scenarios tested:

- Sysbench Default Schema:
  - » Read Only Test - Default Sysbench - 220G
  - » Write Only Test - Default Sysbench - 220G
  - » Read/Write Test - Default Sysbench - 220G
  - » Read Only Test - Default Sysbench - 2.3T
  - » Write Only Test - Default Sysbench - 2.3T
- Sysbench Modified Data Schema:
  - » Read Only - Modified Sysbench - 100 Tables - 440G
  - » Write Only - Modified Sysbench - 100 Tables - 440G
  - » Read/Write - Modified Sysbench - 100 Tables - 440G
  - » Read Only - Modified Sysbench - 540 tables - 2.5T
  - » Write Only - Modified Sysbench - 540 tables - 2.5T
  - » Read/Write - Modified Sysbench - 540 tables - 2.5T
  - » Read Only Latency - Modified Sysbench - 540 tables - 2.5T - Less is better
  - » Write Only Latency - Modified Sysbench - 540 tables - 2.5T - Less is better
- In order to demonstrate the ability of the CSD 2000's compression feature to store more data, we also ran the following tests:
  - » Read Only - Modified Sysbench - 540 tables - 4.7T
  - » Write Only - Modified Sysbench - 540 tables - 4.7T
  - » Read/Write - Modified Sysbench - 540 tables - 4.7T

To achieve the above scenarios we used the same benchmark engine (sysbench) for consistency.

We ran the test with the default oltp sysbench schema and workload and used the default oltp_Read_only.lua, oltp_Read_Write.lua, oltp_Write_only.lua.

Additionally, we ran some custom tests, where we added two extra varchar fields to the sbtest schema. In these fields we inserted random lines from books downloaded from https://www.gutenberg.org/. We added additional queries to sysbench using these extra fields.

By doing this we were trying to simulate a more realistic dataset, one which contains more than just random characters. A real world example of this is a database in which some fields are encrypted (e.g. credit card numbers) while others are not (e.g. product descriptions). Because of these extra fields and queries we had fewer queries per second than the traditional sysbench tests. The tests were also more disk heavy, which was one of our goals.

We were trying to put more pressure on the disk layers. As we were using only a small buffer (8GB), Percona Server had to do a lot of the disk operations. With a bigger Buffer Pool we might have achieved better test results, but, we wanted to simulate how a system performs when the dataset can not fit in memory, and put as much pressure on the disk layer as possible.

The tests were executed against the following table configurations:

- default lua scripts - 100 tables - 10ML rows - 220G
- default lua scripts - 1000 tables - 10ML rows - 2.3T
- modified lua scripts - 100 tables - 10ML rows - 440G
- modified lua scripts - 540 tables - 10ML rows - 2.5T
- modified lua scripts - 540 tables - 20ML rows - 4.7T

Tests were running on the application node, connected to the database server with a 10G network link. Each test was run with 1-8-16-32-64-96-128-256 thRead(s) for 10 minutes for each thread set. Each iteration had at least two runs.

Each full iteration was repeated at different times in the week and on different days. These iterations happened in the months of May 2020 and June 2020.

The only setting change between the tests was: **enabled or disabled for the InnoDB Double Write Buffer.**

Application server: Supermicro; SYS-6019U-TN4RT
Database Server: Inspur; SA5212M4

MySQL Configuration

We used the following configuration options for MySQL:

```
innodb_buffer_pool_size=8G
innodb_log_file_size = 2G
max_connections=500
log_bin=OFF
slow_query_log=off
disable_log_bin
innodb_doubleWrite=ON/OFF
tmpdir = /var/lib/mysql/
innodb_adaptive_hash_index=off
innodb_flush_method=O_DIRECT
innodb_purge_thReads=32
sync_binlog=0
slow_query_log=0
max_prepared_stmt_count=4000000
```

The aim was to keep everything on default as much as possible, but we turned off binary logging, slow query logging and adaptive hash index, which would use the memory instead of the disk.

The only thing which changed between the tests was the innodb_doubleWrite.

We used ext4 filesystem on both drive with the following mount options: rw,noatime,data=Writeback

Compression Ratio

The typical compression ratio in a MySQL application ranges from around 1.5:1 to 5:1. The ScaleFlux drive has built-in compression. With the mixed dataset we ran, when the OS reported the logical data size was 2.5T, physically it only took up 1.4T on disk (1.8:1).

We also tested table level compression with the Intel drive. In this case, the 2.5T of data was compressed into 1.6T (1.6:1) using InnoDB zlib, but performance also dropped, as shown in the results presented earlier in this document.