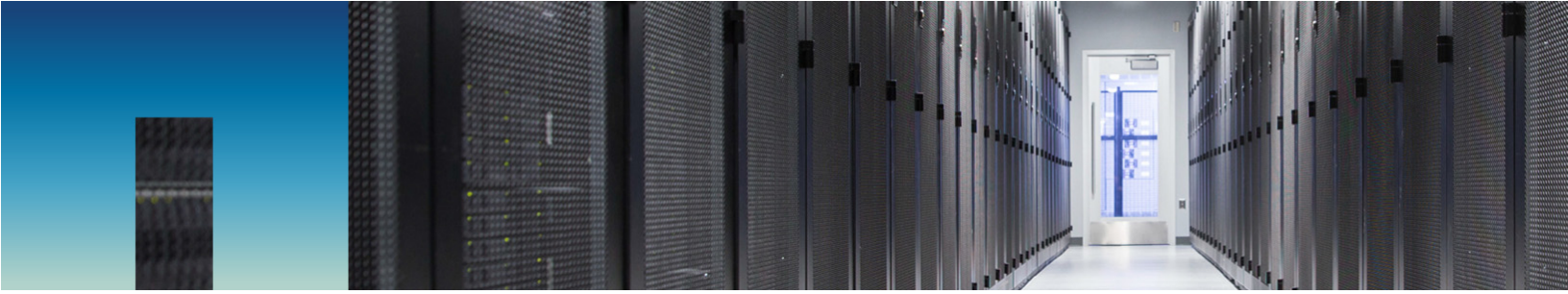


Technical Report

Levyx Performance Characterization

NetApp Baseline Establishment

Younes Ben Brahim, NetApp
May 2018 | TR-4692

In partnership with

HELIUM XENON™

Abstract

This report summarizes Levyx's evaluation of running Helium and Xenon on a NetApp® EF-Series flash storage array.

NetApp®

TABLE OF CONTENTS

1 Executive Summary 4

1.1 Apache Spark 4

1.2 NetApp 4

1.3 Helium 4

1.4 Xenon 5

1.5 Evaluation Summary 5

2 Testbed 5

3 Baseline Measurements 8

3.1 FIO Configuration 8

4 Helium Baseline 9

5 Helium Performance and Scaling 11

5.1 Cores 11

5.2 Solid State Drives 12

6 YCSB Testing 13

7 Spark-Xenon Baseline 15

7.1 Test Specifications 15

Appendix 18

Scripts, Software Versions and YCSB Configurations 18

Where to Find Additional Information 20

Version History 20

LIST OF TABLES

Table 1) EF560 server nodes specifications 6

Table 2) System specifications 6

Table 3) Multipath configuration and device state Levyx1 7

Table 4) Multipath configuration and device state Levyx2 7

Table 5) Multipath configuration and device state Levyx3 7

Table 6) Multipath configuration and device state Levyx4 7

Table 7) EF650 partitions 8

Table 8) 10Gbit network interface specifications 8

Table 9) Spark-Xenon parameters 8

Table 10) FIO Baseline measurement results, single node 8

Table 11) FIO Baseline measurement results, aggregate, two nodes 8

© 2018 NetApp, Inc. All rights reserved.

Table 12) FIO Baseline measurement results, aggregate, four nodes 9
Table 13) Optimal helium configurations 9
Table 14) Helium configuration....9
Table 15) Small key-value (16b, 8b), 56 thread results 10
Table 16) Large key-value (16b, 1024b), 56 thread results....10
Table 17) YCSB average latency 14
Table 18) YCSB total runtime....14
Table 19) Spark-Xenon TPCH test results 15
Table 20) Spark-Xenon iterative join test results....16

LIST OF FIGURES

Figure 1) Apache Spark driver and worker programs....4
Figure 2) EF560 hardware configuration 6
Figure 3) Performance trend from scaling core count, basic Helium configuration....11
Figure 4) Performance trend from scaling core count, adjusted Helium configuration 12
Figure 5) Performance results from scaling the quantity of SSDs 13
Figure 6) YCSB net performance comparison results 14
Figure 7) Iterative join architecture....17

 © 2018 NetApp, Inc. All rights reserved.

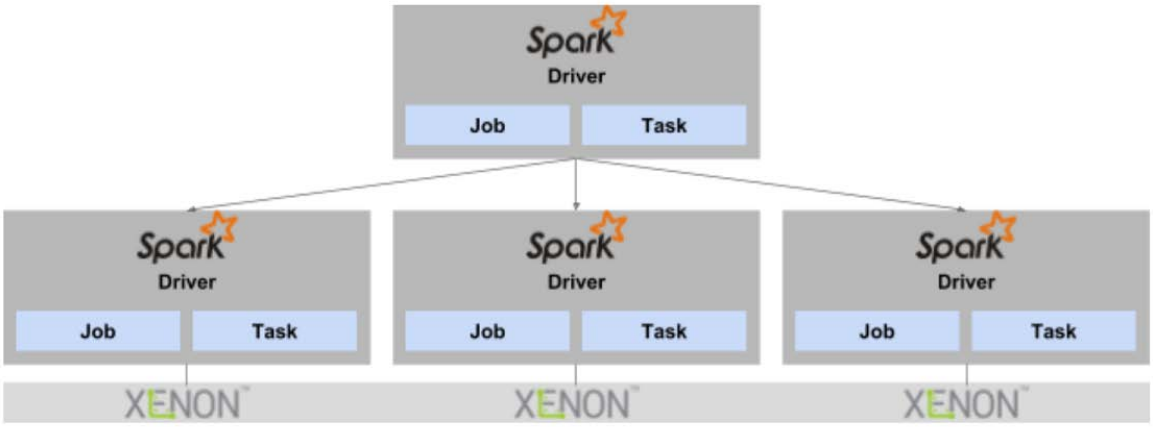
1 Executive Summary

This report summarizes Levyx's evaluation of running Helium and Xenon on a NetApp EF-Series flash storage array.

1.1 Apache Spark

An Apache Spark program has a driver program and worker programs. Worker programs run on cluster nodes or in local threads, and datasets are distributed across workers as shown in Figure 1.

Figure 1) Apache Spark driver and worker programs. (Graphic supplied by Levyx.)



1.2 NetApp

NetApp all-flash arrays offer dramatic response rates for enterprise applications, as well as high performance, efficiency, and availability, along with modular scalability. Key benefits of using NetApp all-flash arrays include enterprise reliability, advanced monitoring and diagnostics with proactive repair, and flexible cost-effective cloud backup and recovery. Refer to the NetApp All-Flash Array datasheet for details. Use cases include data analytics, databases, and dedicated workloads.

1.3 Helium

Levyx provides a software-only, high-performance key-value storage engine known as Helium™ that enables big data applications to interact seamlessly with enterprise solid state drives (SSDs). Levyx's software exposes the higher-layer applications to the memory subsystem with a direct path for data to move between flash (the SSDs) and the applications.

Helium is a key-value store designed and optimized for many core systems and SSDs, and it has an extremely small memory footprint. The design of Helium's core data structures and indexing algorithm enables Helium to scale based on threads. The memory footprint of Helium's sorted indexer is 12 bytes per object, resulting in the ability to index billions of objects on a commodity machine. Helium is designed for SSDs and is therefore optimized for SSD access in terms of both performance and reduction of read/write amplification.

One of Helium's design goals is the "scale-in before scale-out" approach, which involves enabling applications that use denser and well-balanced systems before requiring additional equipment and resources to process workloads. An article demonstrating Levyx's scale-in approach is available on the Google Cloud Platform blog.

© 2018 NetApp, Inc. All rights reserved.

1.4 Xenon

Xenon™ is a low-latency, scalable data analytics server designed to manage the retrieving, processing, and indexing of very large datasets. These datasets are collections of billions of objects, spread across a tightly coupled cluster of servers, each with multiterabyte persistent storage capabilities. Xenon is a distributed database system that has the following functional capabilities:

1. Core SQL functionality: filter, projection, selection, sort, join, group by, and aggregates on structured data such as schema-based tables.
2. Support for random lookup and neighborhood search by using an index rather than scan and filter.
3. Tight integration with the Apache Spark system for ease of deployment and use

   Note: Xenon remains fully functional in native mode, or when serving as an off-load layer for other big platforms and systems.
4. Scaling depending on the number of cores in the cluster and using SSD or other high-bandwidth, low-latency persistent storage as the storage fabric for datasets.

1.5 Evaluation Summary

Levyx's optimizations for big data hardware, combined with NetApp's enhanced-performing flash storage, allow Helium to outperform Facebook's RocksDB by a factor of:

- 8.0X in throughput for key-value insertions
- 10.8X in throughput for key-value retrievals
- 12.2X reduction in average latency
- 10.3X reduction in total runtime

It allows Spark-Xenon to outperform Apache Spark by a factor of up to:

- 5X reduction in average query runtime on select TPC-H benchmarks

2 Testbed

Levyx tested NetApp EF-Series flash array through an InfiniBand connection to four Dell PowerEdge T630 host servers, each equipped with:

- 2 Intel E5-2699 v4 CPUs
- 256GB RAM, 2 NVMe SSDs
- 10GbE network

Each host system consisted of 56 logical cores. Figure 2 shows the EF560 hardware configuration.

© 2018 NetApp, Inc. All rights reserved.

Figure 2) EF560 hardware configuration. (Graphic supplied by Levyx.)

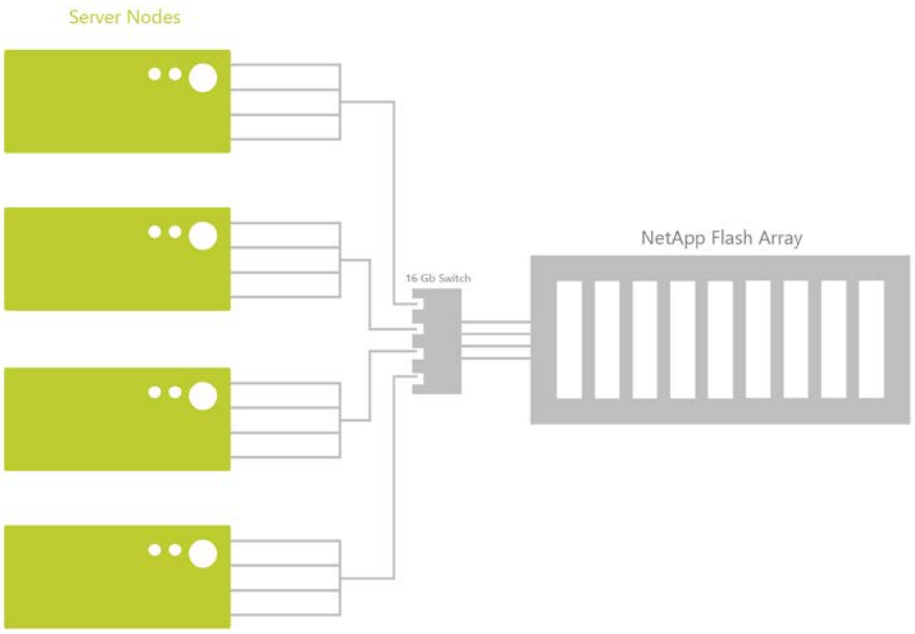


Table 1) EF560 server nodes specifications.

Node Name	IP
Levyx1	10.250.253.145
Levyx2	10.250.253.146
Levyx3	10.250.253.147
Levyx4	10.250.253.148

Table 2) System specifications.

Parameter	Value
Architecture	x86_64
Byte order	Little Endian
CPUs	56
Threads per core	2
Cores per socket	14
Sockets	2
NUMA nodes	2
Memory	188GB
OS	Linux 3.10.0-327 CentOS 7.2

© 2018 NetApp, Inc. All rights reserved.

Parameter	Value
SSDs	Based on all-flash SAN
NVMe driver version	1.0

Table 3) Multipath configuration and device state, Levyx1.

Device ID	600GB Partition	Usage	900GB Partition	Usage
360080e5000431900000183258cbc2d5	p1	HDFS	p2	
360080e50004309b8000027a658cbc16a	a1	-	a2	SP/XE
360080e5000431900000183458cbc2f8	p1	-	p2	SP/XE
360080e50004309b8000027a758cbc187	p1	-	p2	SP/XE

Table 4) Multipath configuration and device state, Levyx2.

Device ID	600GB Partition	Usage	900GB Partition	Usage
360080e5000431900000183258cbc2d5	p1	-	p2	SP/XE
360080e50004309b8000027a658cbc16a	a1	HDFS	a2	
360080e5000431900000183458cbc2f8	p1	-	p2	SP/XE
360080e50004309b8000027a758cbc187	a1	-	p2	SP/XE

Table 5) Multipath configuration and device state, Levyx3.

Device ID	600GB Partition	Usage	900 B Partition	Usage
360080e5000431900000183258cbc2d5	p1	-	p2	SP/XE
360080e50004309b8000027a658cbc16a	a1	-	a2	SP/XE
360080e5000431900000183458cbc2f8	p1	HDFS	p2	
360080e50004309b8000027a758cbc187	p1	-	p2	SP/XE

Table 6) Multipath configuration and device state, Levyx4.

Device ID	600GB Partition	Usage	900GB Partition	Usage
360080e5000431900000183258cbc2d5	p1	-	p2	SP/XE
360080e50004309b8000027a658cbc16a	a1	-	a2	SP/XE
360080e5000431900000183458cbc2f8	p1	-	p2	SP/XE
360080e50004309b8000027a758cbc187	p1	HDFS	p2	

© 2018 NetApp, Inc. All rights reserved.

Table 7) EF650 partitions.

Partition	Size	Usage
/mnt/hdfs	2.4TB	Placeholder for input raw data and HDFS shared media
/mnt/spark	3.6TB	Used for shuffle data while running native Spark

Table 8) 10Gbit network interface specifications.

Hostname	NIC	IP
Levyx1	em1	192.168.3.1
Levyx2	eth2	192.168.3.2
Levyx3	eth2	192.168.3.3
Levyx4	eth2	192.168.3.4

Table 9) Spark-Xenon parameters.

Parameter	Value
Spark version	1.6.0
Xenon version	3.2.0
Spark-Xenon version	1.1.0
Spark driver memory (native Spark)	128GB
Spark driver memory (Spark & Spark-Xenon)	32GB

3 Baseline Measurements

This section outlines the baseline measurements used in the evaluation.

3.1 FIO Measurement

The SSDs underwent preconditioning using fio version 2.2.8. The preconditioning FIO script is based on an optimal FIO configuration for 4K values such as `asyncIO`, `high queue depth`, and `O_DIRECT`.

The baseline measurements were similarly collected using optimal fio configurations, which provide performance ceiling numbers.

Table 10) FIO baseline measurement results, single node.

FIO Baseline	Sequential I/O (1M values)	Random I/O (4K values)
Write (o_direct, libaio)	6.8GB/sec, 6.8K IOPS	136K IOPS
Read (o_direct, libaio)	12.4GB/sec, 12.4K IOPS	526K IOPS

Table 11) FI measurement results, aggregate, two nodes.

FIO Baseline	Sequential I/O (1M values)	Random I/O (4K values)
Write (o_direct, libaio)	6.7GB/sec, 6.7K IOPS	137K IOPS

© 2018 NetApp, Inc. All rights reserved.

FIO Baseline	Sequential I/O (1M values)	Random I/O (4K values)
Read (o_direct, libaio)	12.1GB/sec, 12.1K IOPS	650K IOPS

Table 12) FIO Baseline measurement results, aggregate, four nodes.

FIO Baseline	Sequential I/O (1M values)	Random I/O (4K values)
Write (o_direct, libaio)	7.1GB/sec, 7.1K IOPS	137K IOPS
Read (o_direct, libaio)	12.5GB/sec, 12.5K IOPS	661K IOPS

4 Helium Baseline

The Helium version 2.11.0 results are based on the optimal throughput values from an internal matrix run of predefined values for SSD counts, workload operations, key-value sizes, and number of threads.

Table 13 shows the optimal Helium configurations. Table 14 shows the Helium configuration chosen for these tests. Although carefully chosen Helium configurations might yield better results, a fixed configuration was used for the purposes of the testing matrix.

Table 13) Optimal Helium configurations

Parameter	Value
Key-value sizes	(K:16B V:8B), (K:16B V:1024B)
Threads	56
Operations	Insertion, sequential lookup, random lookup

Table 14) Helium configuration

Parameter	Value
Fanout	128
Write buffer	1GB
Read cache	1GB
Helium datastore	Stripe across all SSDs
Operations	1 million per thread

The fanout configuration parameter exploits the amount of parallelism on an SSD through multiple parallel streams. The default Helium fanout is 32. Given the availability of the four devices for use as the Helium datastore, fanout is set to 128. The write buffer is a staging area for an application thread to write into. Worker threads flush this write buffer to SSDs at maximum possible rates limited only by CPU and I/O bandwidth. The write buffer and read cache of Helium were both set to 1GB.

Note: Helium does not rely on the operating system (OS) page cache, and therefore the memory consumption is deterministic and can be controlled by using the read cache parameter. These and other Helium configuration parameters are documented in the man page.

The generic Helium command is:

```
helium --perf -T <threads> --<operation type> -k <key size> -v <value
size> -o <operations per thread> he://.//dev/dm-2,/dev/dm-3,/dev/dm-
4,/dev/dm-5
```


© 2018 NetApp, Inc. All rights reserved.

Results

The results of two data points at 56 threads are provided, both small key value and large key value. The rationale is:

- Small key-value sizes are CPU bound. This stresses the core indexing algorithm of any key-value store and further tests its scalability; that is, the performance increase with threads.
- Large key-value sizes are I/O bound. This tests the efficiency of the key-value store to use the I/O device optimally.

The range of key-value sizes used here is representative of-low latency applications in domains such as financial stock trading and social media in the form of twitter messages, status update, and so on.

Small Key-Value Objects

Workloads were completed in this category with both 16-byte key and 8-byte value objects.

Table 15) Small key-value (16b, 8b), 56 thread results.

Operation Percentage	Ops/sec (millions)	Latency (in usec)		
100% PUT	8.8	26	145	8,506
100% SEQGET	8.0	164	383	2,076
100% RANDGET	0.8	458	662	1,178

Summary

For ingests and sequential lookups (100% GET workloads), Helium's operations per second is in the 5-10 million range. For 100% GET workloads, Helium uses the cache effectively, resulting in 8 million lookups per second.

To put things in context, only 10 insertion operations out of 100,000 take more than 26 microseconds to complete, and only 1 insertion operation out of 100,000 takes more than 164 microseconds to complete, making Helium an ideal platform for latency-sensitive applications, such as Internet of Things (IoT) and financial technologies.

Although a purely random workload is not common, it does present the worst-case scenario with regard to performance. The high read amplification in a random workload can be attributed to Helium's unit of I/O (also known as Helium page), which is 4K. Each Helium 4K page uses only 24 bytes of data (16B key plus 8B value). Due to the nature of random lookups, there is no reuse of the Helium page data, resulting in high read amplification.

Large Key-Value Objects

Workloads were completed in this category with 16-byte key and 1024-byte value objects.

Table 16) Large key-value (16b, 1024b), 56 thread results.

Operation Percentage	Ops/sec (millions)	Latency (in µsec)		
100% Put	3.6	17	555	27,384
100% SeqGet	0.8	377	568	1,872
100% RandGet	0.2	511	1,061	2,455

© 2018 NetApp, Inc. All rights reserved.

Summary

Helium is I/O bound for large object sizes, and most of the CPU time is spent on system and I/O wait. The read amplification for 100% sequential GET operations is close to 2.0. This value can be attributed to the SSD object size, which includes the metadata, the key, and the value, which might exceed 4KB and possibly span across two Helium page boundaries.

5 Helium Performance and Scaling

Helium performance benchmarks were taken while ramping up the number of threads used during testing.

5.1 Cores

Because the efficiency of scaling with cores lies in the key-value internals and indexing algorithm, small key-value objects of 100 bytes were used.

Figure 3 shows the throughput for inserts, sequential lookups, and deletes, with threads ranging from 1 through 52.

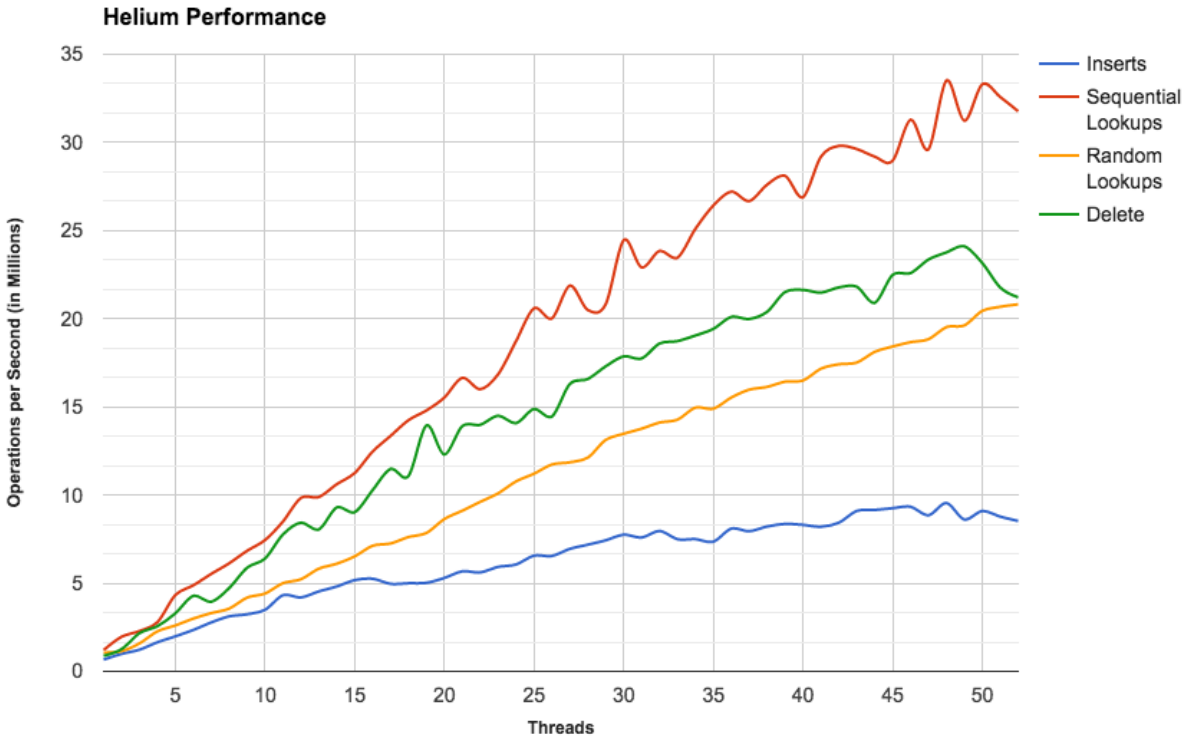
Figure 3) Performance trend from scaling core count, basic Helium configuration.

Using basic Helium configurations, throughput for sequential and random lookups reaches a zenith with 8 threads, given that the `read_cache` is filled early in the run (8 threads * (16-byte key size + 100-byte value size + 12-bytes metadata) * 1 million operations.

However, with minor adjustments to the Helium configuration, performance might improve almost 10fold.

© 2018 NetApp, Inc. All rights reserved.

Figure 4) Performance trend from scaling core count, adjusted Helium configuration.



5.2 Solid State Drives

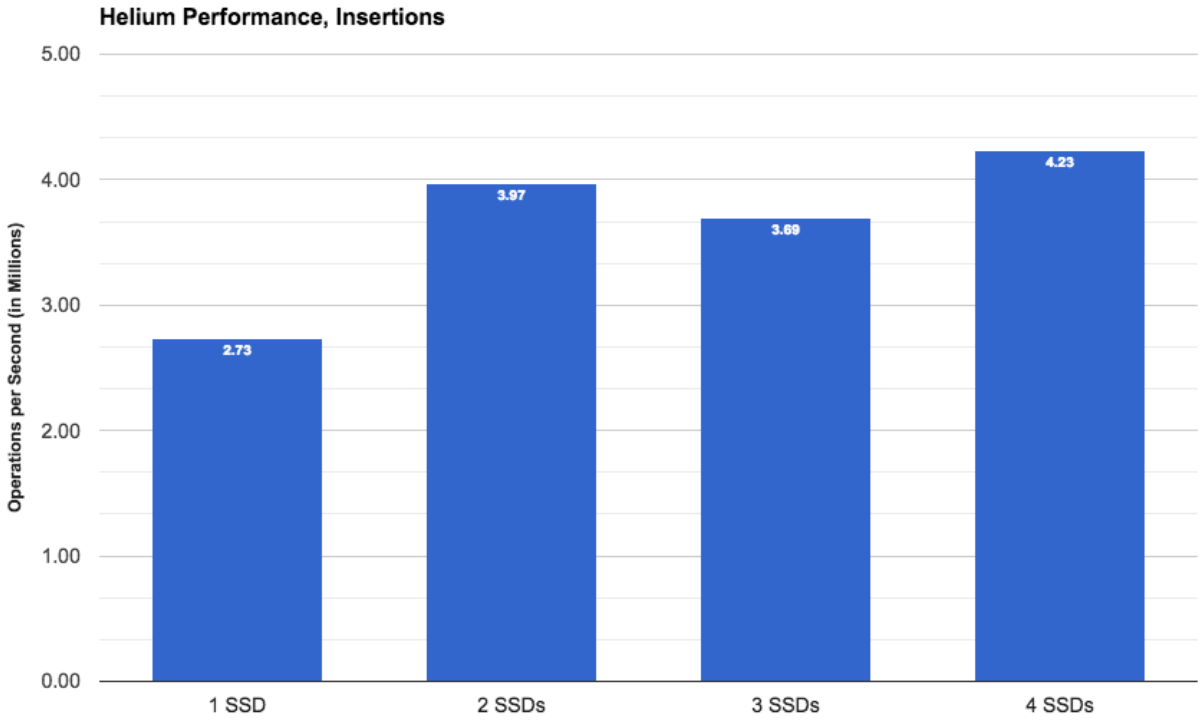
For scaling with SSDs, large key-value objects of 4KB were chosen to show the results of using one, two, three, or four SSDs. The large key-value size was chosen to emphasize the I/O scaling. The throughput results are as reported from the Helium Perf[1] program.

Figure 5 shows the throughput for 4KB object inserts as the number of SSDs used increased.

[1] The Helium Perf program is a Levyx utility available to customers for quickly benchmarking Helium performance at various read/write I/O performance levels and object sizes.

© 2018 NetApp, Inc. All rights reserved.

Figure 5) Performance results from increasing the number of SSDs.



6 YCSB Testing

Yahoo Cloud Serving Benchmark (YCSB) is a standard benchmark for testing the performance of key-value (NoSQL) datastores. YCSB version 0.1.4 was used to benchmark Helium and RocksDB. This version of YCSB provides a mapkeeper-based interface that supports write wrappers for third-party key-value stores. Levyx has created wrappers for both Helium and RocksDB. The RocksDB version used is 3.10.1. Running the latest version of RocksDB with the earlier release of YCSB results in significantly worse performance. The version of RocksDB used in this report is based on a configuration previously suggested by the authors of RocksDB. RocksDB has two sets of results, with and without write Ahead Log (WAL), which serializes memtable operations to a persistent medium as a log file.

The YCSB benchmarking works in two phases, load phase and run. During the load phase, keys are loaded or inserted; during the run phase, a read/write workload mix occurs. The default read/write workload is 80% reads and 20% writes. The YCSB configuration used is a modified configuration with an increased number of keys, configuration of 10 million operations per count, and the default of 100 threads. The Helium configuration and RocksDB configuration are specified in the Appendix. The default key-value size for YCSB is a 4KB object.

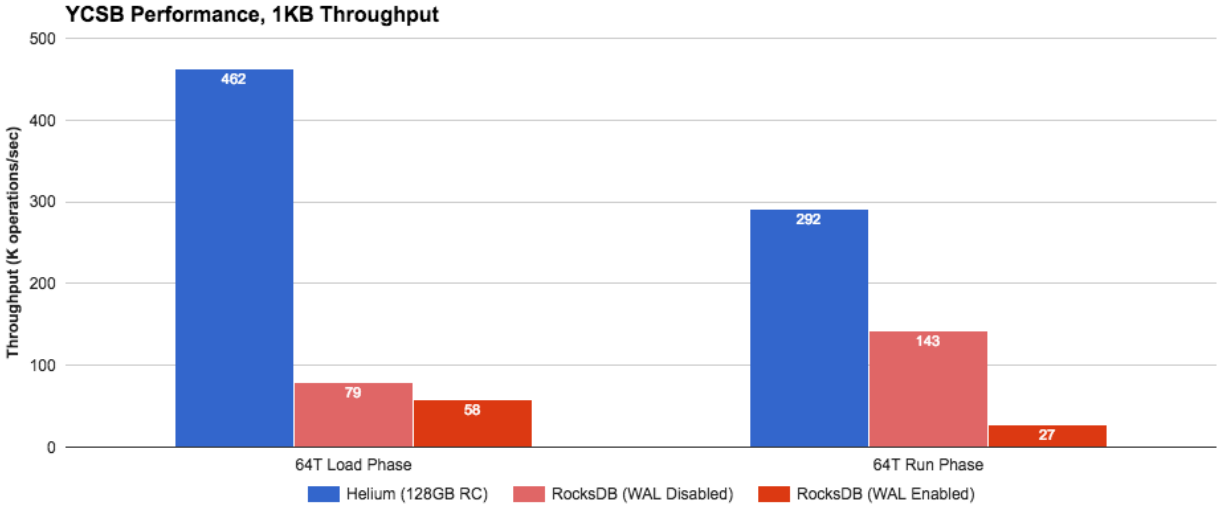
Throughput

Figure 6 compares throughput for Helium and Rocks DB with and without WAL.

Single SSD: directly using SSD for Helium versus XFS on single SSD for RocksDB.

© 2018 NetApp, Inc. All rights reserved.

Figure 6) YCSB net performance comparison results.



Latency

Table 17 shows the average latency results.

Table 17) YCSB average latency.

Program	Load Phase Latency (µsec)	Run Phase Latency (µsec)
Helium	110	167
RocksDB	810	328
RocksDB (WAL enabled)	1,103	2,277

Runtime

Table 18 shows the total runtime results.

Table 18) YCSB total runtime.

Program	Load Phase Runtime (sec)	Run Phase Runtime (sec)
Helium	692,627	3,427,932
RocksDB	4,069,304	6,996,811
RocksDB (WAL enabled)	5,561,485	36,710,173

Summary

Helium performance is orders of magnitude (~5x) improved over RocksDB during ingestion. This can be attributed to the lockless architecture, which scales with the number of threads. Helium has better numbers for performance, run-time, latency, and host write-amplification when compared to RocksDB.

© 2018 NetApp, Inc. All rights reserved.

As verified in testing of other SSDs in denser systems, the performance difference between Helium and RocksDB grows as the number of SSDs is increased, meaning that better scaling occurs with additional devices.

7 Spark-Xenon Baseline

Xenon is Levyx's distributed dataset and query accelerator engine. It both uses and extends beyond Helium to provide a flash SSD-based persistent memory computational platform. It is ideal for running I/O-intensive analytics on large datasets. Xenon is certified by Hortonworks for use with its distribution of Apache Spark/Hadoop. The goal of this set of benchmarks is to illustrate the performance improvements that can be gained from using Xenon with Apache Spark versus Apache Spark by itself.

7.1 Test Specifications

A combination of industry-standard TPCH (highlighting database query performance) benchmarks and Levyx-provided iterative join benchmark (highlighting graph database-like update operations) is used to contrast the improvements in Apache Spark execution time when Xenon is used. In addition to the tests performed for this report, comparisons using the STAC-A3 benchmark for financial algorithm back testing are available at this link: https://stacresearch.com/news/2018/02/13/LEVX171002.The following section describes the two sets of the TPCH and iterative join tests.

TPCH Queries

The TPCH Benchmark™ (TPC-H) is a decision support benchmark. It consists of a suite of business-oriented ad hoc queries and concurrent data modifications. The queries and the data populating the database were chosen to have broad industry-wide relevance. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and answer critical business questions. For more information, refer to http://www.tpc.org/tpch/.

In this test, a performance comparison between Apache Spark running with and without the Levyx-Xenon offload engine executes three different queries from the TPCH standard benchmark. Normally, datasets for Hadoop-Spark clusters are typically stored or archived as serialized HDFS files (128K block size). They are stored locally to the node and replicated to others. This test does not measure replication effects. It is a load and execute query comparison of Spark-HDFS versus Spark-Xenon designed to illustrate the benefits of using Xenon's Persistent DataFrame for caching active datasets that would not otherwise fit into Spark's memory.

Test Flow Description

1. Load TPCH tables into Xenon persisted resilient distributed datasets (RDDs) from Hadoop distributed file system (HDFS).
2. Run a sequence of queries (q3, q6, and q10) once with Xenon and once without Xenon.
3. Compare the results.

Results

Table 19) Spark-Xenon TPCH test results

Test	Query	Input Size	Nodes	Spark Execution Time (sec)	Spark-Xenon Execution Time (sec)
TPCH	q3	300GB	1	1,886	553
TPCH	q6	300GB	1	1,444	285

© 2018 NetApp, Inc. All rights reserved.

Test	Query	Input Size	Nodes	Spark Execution Time (sec)	Spark-Xenon Execution Time (sec)
TPCH	q3	1000GB	4	1,570	1,400
TPCH	q6	1000GB	4	1,049	287
TPCH	q10	1000GB	4	1,545	1,480

Fast and Shareable Storage

Apache Spark becomes inefficient when the size of datasets becomes too large and they cannot fit into memory. This causes spillovers onto disk that make the jobs run slowly and, in extreme cases, Spark runs out of memory and crashes.

EF650 provides ample space for the working sets and does not have the size constraints of memory. Along with Xenon storage, this combination might help prevent spillovers and crashes.

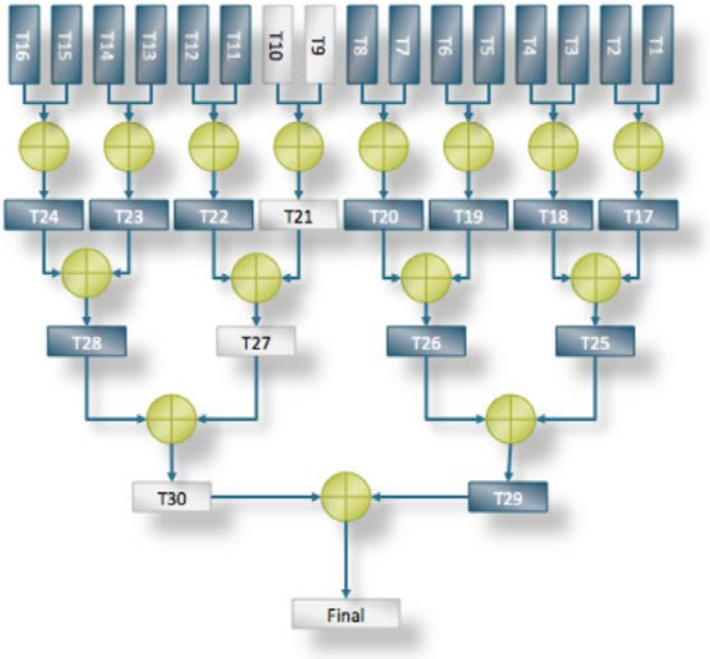
1. Create a computation tree containing a series of joins (minimum depth of tree is 4, which requires 7 joins). Figure 7 shows a tree that requires joining 16 tables, T1 to T16, where the total number of joins is 15.
2. Input random RDDs to the computation tree (T1 to T16), equivalent to database tables, which contain 10,000,000 (TBU) random rows of data with a specified schema and minimum size of 100 bytes per row (TBU).
3. In each iteration, change one of the input RDDs (tables) to a new set of data, which recalculates the output of the computation tree. This emulates algorithms where part of the input data changes in each iteration; for example, many graph algorithms show this behavior.
4. Repeat step 3 ten times.

Table 20) Spark-Xenon iterative join test results.

Test	Total Iterations	Input Size	Nodes	Spark Execution Time (sec)	Spark-Xenon Execution Time (sec)
IterativeJoin	6	-	4	15,764	11,228

© 2018 NetApp, Inc. All rights reserved.

Figure 7) Iterative join architecture. (Graphic provided by Levyx.)



Analytical with Bypass External Network

Xenon cluster servers can share a single volume and are therefore able to short-circuit operations that would usually require shuffling the data among different servers. Instead of moving the data through a network, different cluster members can access the data directly. Xenon has all the required shared access control mechanisms to handle that direct access, which improves the performance of queries that require data shuffling, such as sort, join, and group by. To test this, the same volume is given to all servers and rerun with the same queries as test four; the performance is then compared.

Xenon has a feature known as ByPass, which can improve the shuffled data exchange process by a factor of 5.6X (from 56Gb internal network bandwidth to 10Gb ethernet bandwidth).

Note: The TeraSort benchmark is an excellent tool for highlighting the value of Xenon ByPass. As faster back ends are used, these numbers scale up accordingly.

Apache Spark requires most of the data on each node in a cluster to be moved to other nodes during shuffle operations:

$$\text{Amount of data exchange} = \frac{\#\text{ nodes} - 1}{\#\text{ nodes}} \times \text{Total amount of data}$$

As the number of nodes increases in a cluster, more data must be moved around. In large clusters, most of the data should be moved at the shuffle stage. For sort operations, on average 40% of total sort process execution time is allocated to the shuffle read process.

© 2018 NetApp, Inc. All rights reserved.

Xenon ByPass is able to speed up the shuffle phase of TeraSort by more than a factor of 5X in a system with an InfiniBand back end, when compared to one using a 10Gb Ethernet. The upper limit for improving the TeraSort[2] performance is 5.6 x 40% = 2.24X[3].

Appendix

Scripts, Software Versions, and YCSB Configurations

FIO preconditioning script

1. `blkdiscard`
2. Sequential fill (2x)

```
fio --filename=$DEVICE --name=${NAME} --bs=1M --nrfiles=1 --direct=1 --
rw=write --iodepth=256 --ioengine=libaio
```

3. Random write (25% of device capacity)

```
fio --name=$NAME --filename=$DEVICE --ioengine=libaio --randrepeat=0 --
iodepth=256 --direct=1 --invalidate=1 --verify=0 --verify_fatal=0 --
numjobs=$NUMJ --rw=randwrite --blocksize=4k --size=25% --group_reporting
```

FIO baseline script example (timed random run—write)

```
fio  --time_based \
                  --runtime=300 \
                  --name=$NAME \
             --filename=$DEVICE \
                  --ioengine=libaio \
                  --randrepeat=0 \
                  --iodepth=256 \
                  --direct=1 \
                  --invalidate=1 \
                  --verify=0 \
                  --verify_fatal=0 \
                  --numjobs=88 \
                  --rw=randwrite \
                  --blocksize=4k \
                  --group_reporting
```

[2] Unfortunately, Levyx did not have enough time to run Terasort with the Xenon ByPass feature.
[3] Assuming no overlap between other operations and the shuffle read phase. In scenarios where overlap does occur, performance degrades subject to the amount of overlap.

© 2018 NetApp, Inc. All rights reserved.

Helium Version

$ helium -–version

Helium 2.11.0 [93c7b0f]

Copyright (C) 2013-2017, Levyx, Inc.

YCSB Parameters: Workload (Large)

```
[config]
recordcount=10000000
operationcount=10000000
workload=com.yahoo.ycsb.workloads.CoreWorkload
readallfields=true
readproportion=0.8
updateproportion=0.2
scanproportion=0
insertproportion=0
requestdistribution=zipfian
fieldlength=4000
fieldcount=1
threadcount=100
insertorder=ordered
```

YCSB Parameters: Helium

```
[config]
write_cache 1073741824
read_cache 1073741824
fanout 128
```

YCSB Parameters: RocksDB

```
max_write_buffer_number=3;
target_file_size_base=6710884; // 64 MB
max_background_compactions=4;
level0_file_num_compaction_trigger=8;
level0_slowdown_writes_trigger=17;
level0_stop_writes_trigger=24;
num_levels=4;
max_bytes_for_level_base=536870912; // 512 MB
max_bytes_for_level_multiplier=8;
```


© 2018 NetApp, Inc. All rights reserved.

Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

1. NetApp All-Flash Array datasheet
2. Google's cloud blog post from August 2015 about Helium:

   "Multimillion operations per second on a single Google cloud instance" http://googlecloudplatform.blogspot.com/2015/07/Multi-million-operations-per-second-on-a-single-Google-Cloud-Platform-instance.html
3. Micron 9100 NVMe SSD 9100 HHHL U.2 PCIe NVMe SSD Datasheet
4. YCSB Paper: www.cs.duke.edu/courses/fall13/cps296.4/838-CloudPapers/ycsb.pdf
5. RocksDB Write Ahead Log File Format wiki: https://github.com/facebook/rocksdb/wiki/Write-Ahead-Log-File-Format
6. TPC Overview: www.tpc.org/tpch/
7. STAC-A3 benchmark for financial algorithm backtesting https://stacresearch.com/news/2018/02/13/LEVX171002

Version History

Version	Date	Document Version History
Version 1.0	May 2018	Initial Release

© 2018 NetApp, Inc. All rights reserved.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2018 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.