Tame Cassandra® Performance and Growth

SSDs Drive Results for Large Data Sets

Overview

When we scale a database — either locally or in the cloud — performance[1] is imperative. Without high performance, a massive-scale database is little more than an active archive.

When an entire data set is small and fits into memory (DRAM), performance is straightforward and storage system capability is less important. However, with immense data growth, a dwindling percentage of data fits into memory affordably.

Combined with the constant demand for faster and more detailed analytics, we have arrived at a data-driven crossroads: We need high performance, high capacity and affordability. Cassandra combined with enterprise SATA SSDs can help. Building with these SSDs lets us future-proof Apache Cassandra® deployments to perform soundly as active data sets grow, extending well beyond memory capacity. Cassandra's ability to support massive scaling, combined with multi-terabyte, high-IOPS enterprise SATA SSDs, lets us build high-capacity NoSQL platforms with extreme capacity, extreme agility and extreme capability.

This technical brief highlights the performance advantages measured when we compared three four-node Cassandra clusters: one built using legacy HDDs and two built using enterprise SSDs (one per node and two per node).

Note: Due to the broad range of Cassandra deployments, we tested multiple workloads. You may find some results more relevant than others for your deployment.

Fast Facts

- Each four-node enterprise SATA SSD-equipped cluster eclipsed the capability of a four-node multidrive legacy (HDD) cluster across multiple workloads and thread counts.
- The SSD configurations measured up to 31X better performance, with more consistent, lower latency.

[1] We use the terms database operations per second (OPS) and performance interchangeably in this paper.

Enterprise SSDs Meet Growing Demands

When we built Cassandra nodes with legacy HDD storage, we scaled out by adding more nodes to the cluster. We scaled up by upgrading to larger drives. Sometimes we did both.

Adding more legacy nodes was effective (to a point), but it quickly became unwieldy. We gained capacity and a bit more performance, but as we added to the clusters, they became larger and more complex, consuming more rack space and support resources.

Upgrading to larger HDDs was somewhat effective (also to a point) because we got more capacity per node and more capacity per cluster, but these upgrades give limited additional cluster performance.

With both techniques, performance was expensive and did not scale well with growth.

High-capacity, lightning-quick SSDs, such as the Micron® 5200 series are changing the design rules. With single SSD capacities measured in terabytes (TB), throughput in hundreds of megabytes per second (MB/s) and IOPS in tens of thousands;[2] high-capacity, ultra-fast SSDs enable new design opportunities and performance thresholds.

SSD Clusters: Real Results From Immense Data Sets

As you plan your next high-capacity, high-demand Cassandra cluster, SSDs can support amazing capacity and provide compelling results. Figures 1a-1c summarize our tested storage configurations.

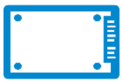


Figure 1a: SSD Configuration 1 – a single SSD per node

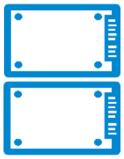


Figure 1b: SSD Configuration 2 – a pair of SSDs per node

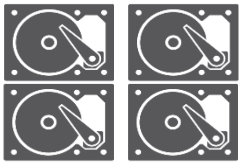


Figure 1c: Legacy Configuration – four 15K RPM HDDs per node

We used the Yahoo! Cloud Serving Benchmark (YCSB) workloads A–D and F[3] to compare three four-node Cassandra test cluster configurations:

- **SSD Configuration 1**: 1x Micron 5200 ECO (3.8TB each)
- **SSD Configuration 2**: 2x Micron 5200 ECO (3.8TB each)
- **Legacy Configuration**: 4x 15,000 RPM HDD (300GB each)

Note: Due to the broad range of Cassandra deployments, we tested multiple thread counts. See the How We Tested section for details.

With the same number of nodes and a single SSD in each node, the 1x SSD test cluster offers a 3X capacity increase over the legacy configuration (the 2x SSD test cluster offers a 6X capacity increase). We also measured significant performance improvements across all the workloads tested with each SSD test cluster, ranging from a minimum improvement of about 1.7X to a maximum improvement of about 10.7X, along with lower and more consistent latency.

[2] Capacity, MB/s and IOPS vary by SSD. This paper focuses on our 3.8 TB 5200 ECO SSD. Other SSD models and/or capacities may give different results.

[3] We did not test YCSB workload E because it is not universally supported.

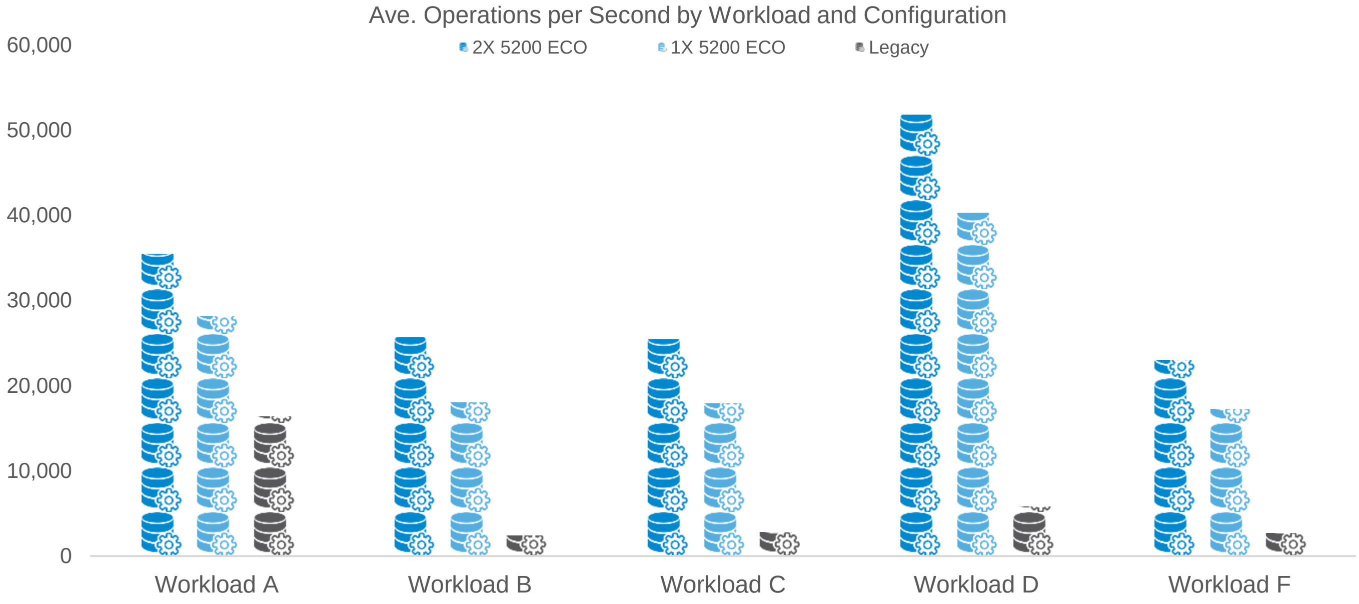


Figure 2: Relative Performance

Table 1 summarizes these results in average database operations per second. Relative (baseline) performance is shown in column 5 (2x SSD workload average performance divided by the baseline configuration's average performance with the same workload) and column 6 (a similar comparison between the 1x SSD and baseline configuration).

Configuration	2x SSD	1x SSD	Baseline	2x SSD/ Baseline	1x SSD/ Baseline
Workload A	35,483.1	28,175.8	16,378.8	2.2X	1.7X
Workload B	25,679.8	18,013.5	2,409.6	10.7X	7.5X
Workload C	25,449.9	17,939.5	2,833.7	9.0X	6.3X
Workload D	51,818.8	40,342.3	5,829.3	8.9X	6.9X
Workload F	23,007.1	17,272.8	2,690.9	8.5X	6.4X

Table 1: Workload Performance by Configuration

SSD Clusters Provide More Consistent Responses

Read Response Consistency: Since many Cassandra deployments rely heavily on fast, consistent responses, we compared the 99th percentile read response times[4] for each test cluster and workload. Figure 3 shows the 99th percentile read latency for each configuration.

Workload A

An update-heavy workload, with 50% of the total I/Os writing data. At the application level, this workload is similar to recording recent session actions.

Workload B

A read-mostly workload (95% read). At the application level, this workload is similar to adding metadata (such as tagging photographs or articles) to existing content.

Workload C

A read-only workload. At the application level, this workload is similar to reading user profiles or other static data where profiles are constructed elsewhere.

Workload D

Reading the latest entries (most recent records are the most popular). At the application level, this workload is similar to reading user status updates.

Workload F

A read/modify/write workload in which records are read, changed and written back. At the application level, this workload is similar to users reading and changing data or tracking user activity.

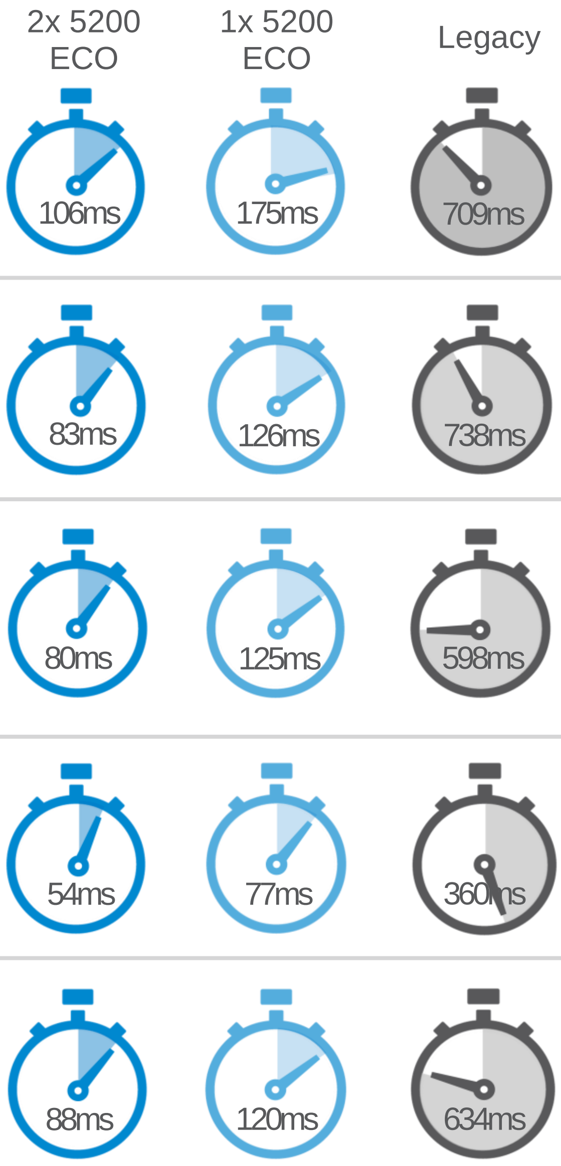
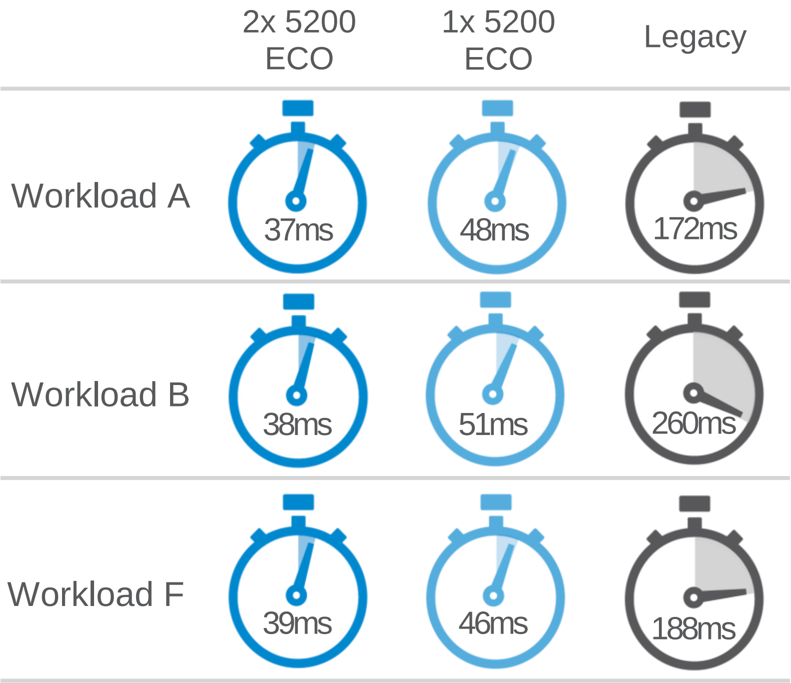


Figure 3: Read Response Consistency

[4] 99th percentile latency is the response time under which 99th of all operations complete. This is a common latency consistency measurement.

Update Response Consistency:

Some Cassandra deployments are more heavily dependent on fast and consistent inserts. We also compared the 99th percentile insert response times for workload in inserts (A, B and F). Figure 4 compares the three configurations.

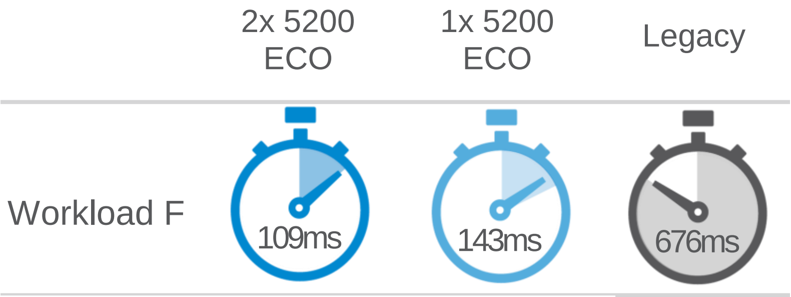


	2x 5200 ECO	1x 5200 ECO	Legacy
Workload A	37ms	48ms	172ms
Workload B	38ms	51ms	260ms
Workload F	39ms	46ms	188ms

Figure 4: Update Response Consistency

Read/Modify/Write Response Consistency:

Last, we also compared the 99th percentile read/modify/write response times for workload F (the only workload executing read/modify/write operations). Figure 5 compares the three configurations.



	2x 5200 ECO	1x 5200 ECO	Legacy
Workload F	109ms	143ms	676ms

Figure 5: Read-Modify-Write Response Consistency

The Bottom Line

High-capacity, high-performance SSDs can produce amazing results with Cassandra. Whether you are scaling your local or cloud-based Cassandra deployment for higher performance or faster, more consistent read responses, SSDs are a great option.

We tested two clusters for database performance and read responsiveness across multiple workloads. We built a legacy cluster using four 300GB 15K RPM HDDs (RAID 0) in each node and two SSD clusters, one with a single 3.8TB Micron 5200 ECO and a second with a pair of them in each node. The results were amazing.

The single SSD per node test cluster showed a significant increase in performance over all the workloads, ranging from a low of about 1.7X up to a high of 7.5X compared to the legacy configuration. The two SSD per node test cluster was still more impressive with workload increases ranging from 2.2X to 10.7X compared to the legacy configuration. We also found that the SSD-based clusters’ read responses were much faster with far greater consistency.

We expect great performance when our data set fits into memory, but immense data growth means that smaller and smaller portions of that data fit into memory affordably.

We are at a crossroads. Business demands drive us toward higher performance, and data growth drives us toward affordable capacity. When we combine these, the answer is clear: Enterprise SSDs deliver Cassandra strong results, helping tame performance demands and data growth.

Given data like the above, our customers are increasingly finding that deploying SSDs in the data center is a high-value option for better overall total cost of ownership (TCO). If you are curious how this compares to other configurations, try Micron's Move2SSD TCO Tool to estimate the savings you can see from deploying SSDs versus existing architecture.

How We Tested

Table 1 shows the tested configurations, the types of storage devices used, and the number and capacity of each as well as the number of nodes in each Cassandra test cluster. Table 2 shows the hardware and software configuration parameters used.

Component	SSD 1	SSD 2	Legacy
Platform	2U Standard Server (x4)		
CPU	Intel Gold 6142 (2.6GHz, 32 cores)		
DRAM	256GB		
Storage	Micron 5200 ECO (3.84TB) x1	Micron 5200 ECO (3.84TB) x2	15K RPM HDD (300GB) x4
OS	CentOS 7.4		
Cassandra	Datastax Community Edition 3.0.9-1		
Java	Oracle Java 8		

Table 1: Tested Configurations

Component	Configuration	Details
OS Settings for Cassandra	All	**/etc/security/limits.d/cassandra.conf:** cassandra - memlock unlimited cassandra - nofile 100000 cassandra - nproc 32768 cassandra - as unlimited **/etc/sysctl.conf:** vm.max_map_count = 131072 /etc/security/limits.d/20-nproc.conf: * - nproc 32768
OS Storage	All	File System: XFS with mount options: noatime, nodiratime, discard

Table 2: Configuration Parameters

Our test methodology approximates real-world deployments and uses for a Cassandra database. Although the test configuration is relatively small (four nodes in each cluster), Cassandra's scaling technology means these results are also relevant to larger deployments.

- Four nodes host the database.
- The replication factor for the database was set to 3 (there are three copies of the data and the cluster can sustain the loss of two data nodes and continue to function) creating a 1.5TB database.

The database is initially created by utilizing YCSB workload A’s load parameter, which generated a data set of approximately 1.5TB, far exceeding available DRAM (ensuring we measure storage system IO). The database is then backed up to a separate location on the server for quick reload of data between test runs. For each configuration under test, the database was restored from this backup, starting every test from a consistent state.

Table 3 shows the percentage of data owned by each of the four nodes.

Node	Capacity	Tokens	Percent Owned
Node01	368.81GB	256	74.1%
Node02	362.62GB	256	72.9%
Node03	389.00GB	256	78.1%
Node04	373.42GB	256	74.9%

Table 3: Data Distribution Across Nodes

Table 4 shows the testing parameters used in the tested workloads.

Parameter	Value	Description
Threads	480	Database load
Field Count	10	1K record size (standard)
Record Count	500 million	Number of database records
Operation Count	50 million	Dataset size within database

Table 4: Test Parameters

Dim_stat was used to capture statistics on the server running Datastax Cassandra. It captures IOStat, VMStat, mpstat, network load, processor load, and several other statistics. Dim_stat was configured to capture statistics on a 10-second interval.

Table 5 shows the IO profiles for tested YCSB workloads (additional details are available at YCSB Core Workloads).

Name	Type	IO Profile
A	Update heavy	50% read, 50% write
B	Read mostly	95% read, 5% write
C	Read only	100% read, 0% write
D	Read latest	95% read, 5% insert
F	Read/modify/write (R/M/W)	50% read, 50% R/M/W

Table 5: Workloads

micron.com

This technical brief is published by Micron and has not been authorized, sponsored, or otherwise approved by the Apache Software Foundation. Products are warranted only to meet Micron’s production data sheet specifications. Products, programs and specifications are subject to change without notice. Dates are estimates only. All information herein is provided on an "AS IS" basis without warranties of any kind. Micron and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners. Rev. A 2/18 CCM004-676576390-10993