WHITE PAPER

DataStax Enterprise on VMware vSAN™ 6.7 All-Flash for Production

DATASTAX® | vmware®

CONTENTS

1. EXECUTIVE SUMMARY

This section covers the business case, solution overview, and key results of DataStax Enterprise on VMware vSAN for production.

1.1 Business Case

Modern cloud applications—that create and leverage real-time value and run at epic scale—require a change in data management with an unprecedented transformation to the decade-old way that databases have been designed and operated. Requirements from cloud applications have pushed beyond the boundaries of the relational database management system (RDBMS) and have introduced new data management requirements to handle always-on, globally distributed, and real-time applications.

VMware vSAN™, the market leader in Hyperconverged Infrastructure (HCI), enables low cost and high performance next-generations HCI solutions. vSAN converges traditional IT infrastructure silos onto industry-standard servers and virtualizes physical infrastructures to help customers easily evolve their infrastructure without risk, improve TCO over traditional resource silos, and scale to tomorrow with support for new hardware, applications, and cloud strategies. The natively integrated VMware infrastructure combines radically simple VMware vSAN storage management, the market-leading VMware vSphere® Hypervisor, and the VMware vCenter Server® unified management solution, all on the broadest and deepest set of HCI deployment options.

DataStax Enterprise (DSE) is the always-on data management platform powered by the industry's best distribution of Apache Cassandra™—and includes search, analytics, developer tooling, and operations management, all in a unified security model. DSE makes it easy to distribute your data across data centers or cloud regions, making your applications always on, ready to scale, and able to create instant insights and experiences. Your applications are ready for anything—be it enormous growth, handling mixed workloads, or enduring catastrophic failure. With DSE's unique, fully distributed, masterless architecture, your application scales reliably and effortlessly.

VMware vSAN combines the benefits of HCI with the performance and scale of DSE. Referred to as Host Affinity[1] (introduced in vSAN 6.7), this policy offers customers additional flexibility to configure vSAN data placement and replication specific to the application that has been deployed. Host Affinity delegates replication to DSE, while maintaining data locality with DSE compute. The Host Affinity policy is available in addition to standard vSAN replication policies and intended to offer customers choice of deployment based on their criticality, uptime, and maintenance requirements.

VMware and DataStax have jointly undertaken an extensive technical validation to demonstrate vSAN as a storage platform for globally distributed cloud applications. In the second phase of this effort, DataStax and VMware jointly advocate this solution for production environments.

1 Note the Host Affinity feature requires VMware validation before the deployment. See Host Affinity for more information.

1.2 Solution Overview

This joint solution is a showcase of using VMware vSAN as a platform for deploying DSE in a vSphere environment. All storage management moves into a single software stack, thus taking advantage of the security, operational simplicity, and cost-effectiveness of vSAN in production environments. Workloads can be easily migrated from bare metal configurations to a modern, dynamic, and consolidated HCI based on vSAN. vSAN is natively integrated with vSphere, and vSAN helps to provide smarter solutions to reduce the design and operational burden of a data center.

1.3 Key Results

This technical white paper:

- Provides the solution architecture for deploying DSE 6.0 in a vSAN cluster for production.
- Measures performance when running DSE in a vSAN cluster, to the extent of the testing and cluster size described.
- Evaluates the impact of different parameter settings in the performance testing.
- Performs operation testing and validates the scalability of running DSE in vSAN.
- Identifies the steps required to ensure resiliency and availability against various failures.
- Provides best practice guidelines.

2. INTRODUCTION

This section provides the purpose, scope, and audience of this document.

2.1 Purpose

This white paper illustrates how DataStax Enterprise can be run in a vSAN and provides testing results based on parameter variations running various workloads.

2.2 Scope

This white paper covers the following testing scenarios:

- Baseline testing
- 90% write and 10% read performance testing
- 50% write and 50% read performance testing
- 10% write and 90% read performance testing
- Operation testing
- Scalability testing
- Resiliency and availability testing

2.3 Audience

This technical white paper is intended for DataStax Enterprise administrators and storage architects involved in the planning, design, or administration of DSE on vSAN for production purposes.

3. TECHNOLOGY OVERVIEW

This section provides an overview of the technologies used in this solution:

- VMware vSphere 6.7
- VMware vSAN 6.7
- DataStax Enterprise
- Samsung NVMe SSD

3.1 VMware vSphere 6.7

VMware vSphere 6.7 is the next-generation infrastructure for next-generation applications. It provides a powerful, flexible, and secure foundation for business agility that accelerates the digital transformation to cloud computing and promotes success in the digital economy.

vSphere 6.7 supports both existing and next-generation applications through its:

- Simplified customer experience for automation and management at scale
- Comprehensive built-in security for protecting data, infrastructure, and access
- Universal application platform for running any application anywhere

With vSphere 6.7, customers can run, manage, connect, and secure their applications in a common operating environment, across clouds and devices.

3.2 VMware vSAN 6.7

VMware's industry-leading HCI software stack consists of vSphere for compute virtualization, vSAN, vSphere native storage, and vCenter for virtual infrastructure management. VMware HCI is configurable and seamlessly integrates with VMware NSX™ to provide secure network virtualization and/or vRealize Suite™ for advanced hybrid cloud management capabilities. HCI can be extended to the public cloud, as VMware-powered HCI has native services with two of the top four cloud providers, Amazon Web Services (AWS) and IBM.

We are now introducing vSAN 6.7 Update 1, which makes it easy to adopt HCI with simplified operations, efficient infrastructure, and rapid support resolution. With vSAN 6.7 Update 1, customers can quickly build and integrate cloud infrastructure. vSAN's automation and intelligence keeps your infrastructure stable, secure, and minimizes maintenance disruptions. vSAN 6.7 Update 1 lowers TCO and makes your storage more efficient through automatic capacity reclamation, and it helps avoid overspending on storage by helping users size capacity needs correctly and incrementally. Finally, vSAN Support Insight reduces time to resolution while lessening customer involvement in the support process, as well as expediting self-help.

See VMware vSAN documentation for more information.

3.3 DataStax Enterprise

DataStax Enterprise, powered by the best distribution of Apache Cassandra, seamlessly integrates your code, allowing applications to utilize a breadth of techniques to produce a mobile app or online applications. DSE makes it easy to distribute your data across data centers or cloud regions, making your applications always on, ready to scale, and able to create instant insight and experiences. DSE provides the flexibility to deploy on any on-premises, cloud infrastructure, or hybrid cloud, plus the ability to use multiple operational workloads, such as analytics and search, without any operational performance degradation.

Check out more information about DataStax Enterprise at https://www.datastax.com/products/datastax-enterprise.

The following terms and parameters are introduced in this white paper:

- **DataStax Enterprise OpsCenter** is a visual management and monitoring solution for DataStax Enterprise.
- **SSTable –** A sorted string table (SSTable) is an immutable data file to which Cassandra writes memtables periodically. SSTables are stored on disk sequentially and maintained for each Cassandra table.
- **Replication factor (RF) –**The total number of replicas across the DSE cluster is referred to as the replication factor. A replication factor of 1 means that there is only one copy of each row in the cluster. If the node containing the row goes down, the row cannot be retrieved. A replication factor of 3 means three copies of each row, where each copy is on a different DSE node. All replicas are equally important, so there is no primary or master replica.
- **Consistency level (CL) –** Consistency level is a setting that defines a successful write or read by the number of cluster replicas that acknowledge the write or respond to the read request, respectively.
  - For consistency level LOCAL_ONE, a write must be sent to, and successfully acknowledged by, at least one replica node in the local data center. By default, a read repair runs in the background to make the other replicas consistent. It provides the highest availability of all the levels, if the application can tolerate a comparatively high probability of stale data being read. The replicas contacted for reads may not always have the most recent write.
  - For consistency level QUORUM, a write must be written to the commit log and memtable on a quorum of replica nodes across all data centers. A read returns the record after a quorum of replicas from all data centers has responded.
- **Compaction –**The process of consolidating SSTables, discarding tombstones, and regenerating the SSTable index. Compaction does not modify the existing SSTables (SSTables are immutable) and only creates a new SSTable from the existing ones. When a new SSTable is created, the older ones are marked for deletion. Thus, the used space is temporarily higher during compaction. The amount of space overhead due to compaction depends on the compaction strategy used. This space overhead needs to be accounted for during the planning process.

3.4 Samsung NVMe SSD

Samsung is well equipped to offer enterprise environments superb solid-state drives (SSDs) that deliver exceptional performance in multi-threaded applications, such as compute and virtualization, relational databases, and storage. These high-performing SSDs also deliver outstanding reliability for continual operation regardless of unanticipated power loss. Using their proven expertise and wealth of experience in cutting-edge SSD technology, Samsung memory solutions help data centers operate continually at the highest performance levels. Samsung has the added advantage of being the sole manufacturer of all its SSD components, ensuring end-to-end integration, quality assurance, and the utmost compatibility.

Samsung PM1725a SSD delivers:

- **Extreme performance –**The highest levels with unsurpassed random read speeds and an ultra-low latency rate using Samsung's highly innovative 3D vertical-NAND (V-NAND) flash memory and an optimized controller.
- **Outstanding reliability –** Features five DWPDs (drive writes per day) for five years, which translates to writing a total of 32TB per day during that time. This means users can write 6,400 files of 5GB-equivalent data or video every day, which represents a level of reliability that is more than sufficient for enterprise storage systems that have to perform ultrafast transmission of large amounts of data.
- **High capacities –** Depending on your storage requirements and applications, 800GB, 1.6TB, 3.2TB, and 6.4TB capacities are available.

This solution chooses the 1.6TB Samsung PM1725 SSD as the cache tier for the vSAN cluster.

See Samsung PM1725a NVMe SSD for more information.

4. CONFIGURATION

This section introduces the resources and configurations:

- Test setup
- Solution architecture
- Hardware resources
- Software resources
- VM and database configuration
- Test tool and settings

4.1 Test Setup

We created an eight-node vSphere and vSAN all-flash cluster and deployed an eight-node DSE cluster, and we also deployed DSE OpsCenter and eight DSE stress client VMs running Cassandra-stress on another hybrid vSAN cluster in the same VM network. We used a client to server ratio of 1:1 to eliminate any potential bottlenecks from the client side during the performance benchmark.

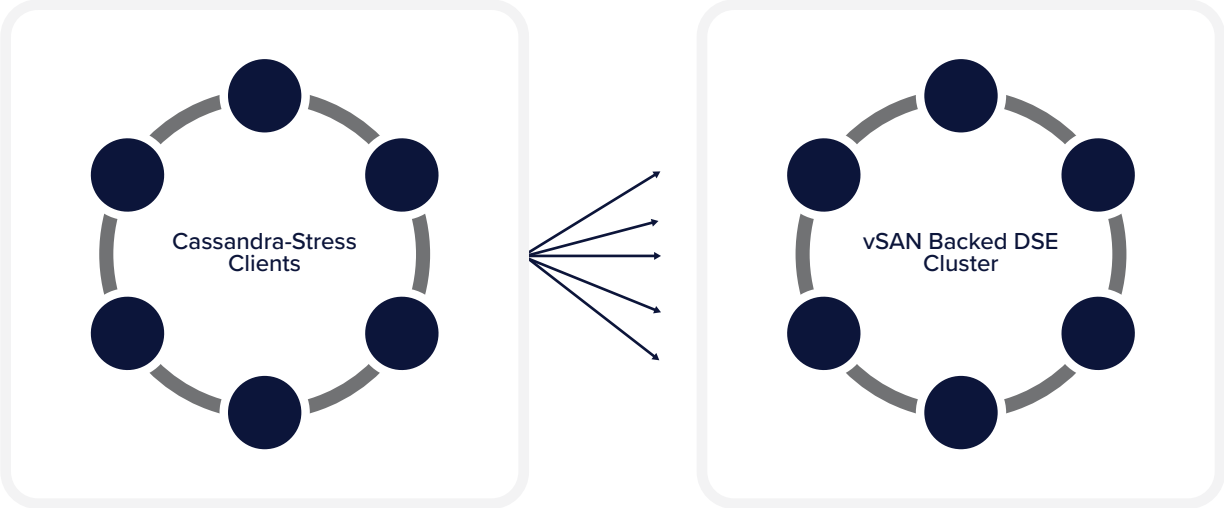

Figure 1. Solution Setup

4.2 Solution Architecture

To ensure continued data protection and availability during planned or unplanned downtime, DataStax recommends a minimum of four nodes for the vSAN cluster. For testing, an eight-node vSAN cluster was used with eight DSE nodes to validate that the cluster will function as expected for typical workloads and scale. Figure 3 shows that we deployed one DSE VM on each ESXi host.

In our solution validation, we used NVMe[2] as a cache tier and configured two disk groups per node. Each disk group had one cache NVMe and four capacity SSDs. vSAN storage policy PFTT (Primary Number of Failures to Tolerate) was set to 0 and "host affinity" is enabled. For "host affinity" to work properly, HA/DRS needs to be turned off and upgrades and patches must be carefully managed. Other parameters in this storage policy are copied from the default vSAN storage policy. We applied the "host affinity" policy to all the VMs' disks, including OS disk, data disk, and log disk. You can customize the storage policy for different DSE applications to satisfy performance, resource commitment, checksum protection, and quality of service requirements.

2 This solution architecture uses NVMe. However, it is not a requirement for all customer environments.

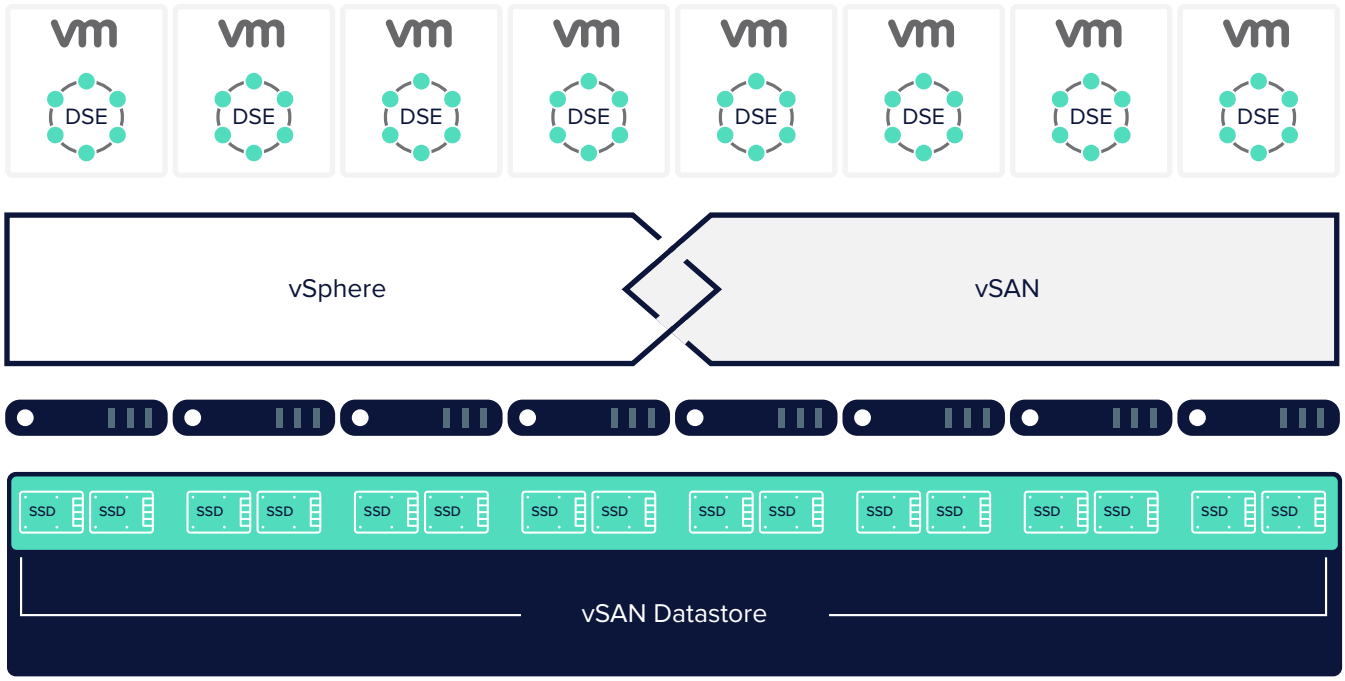

Figure 2. Solution Architecture

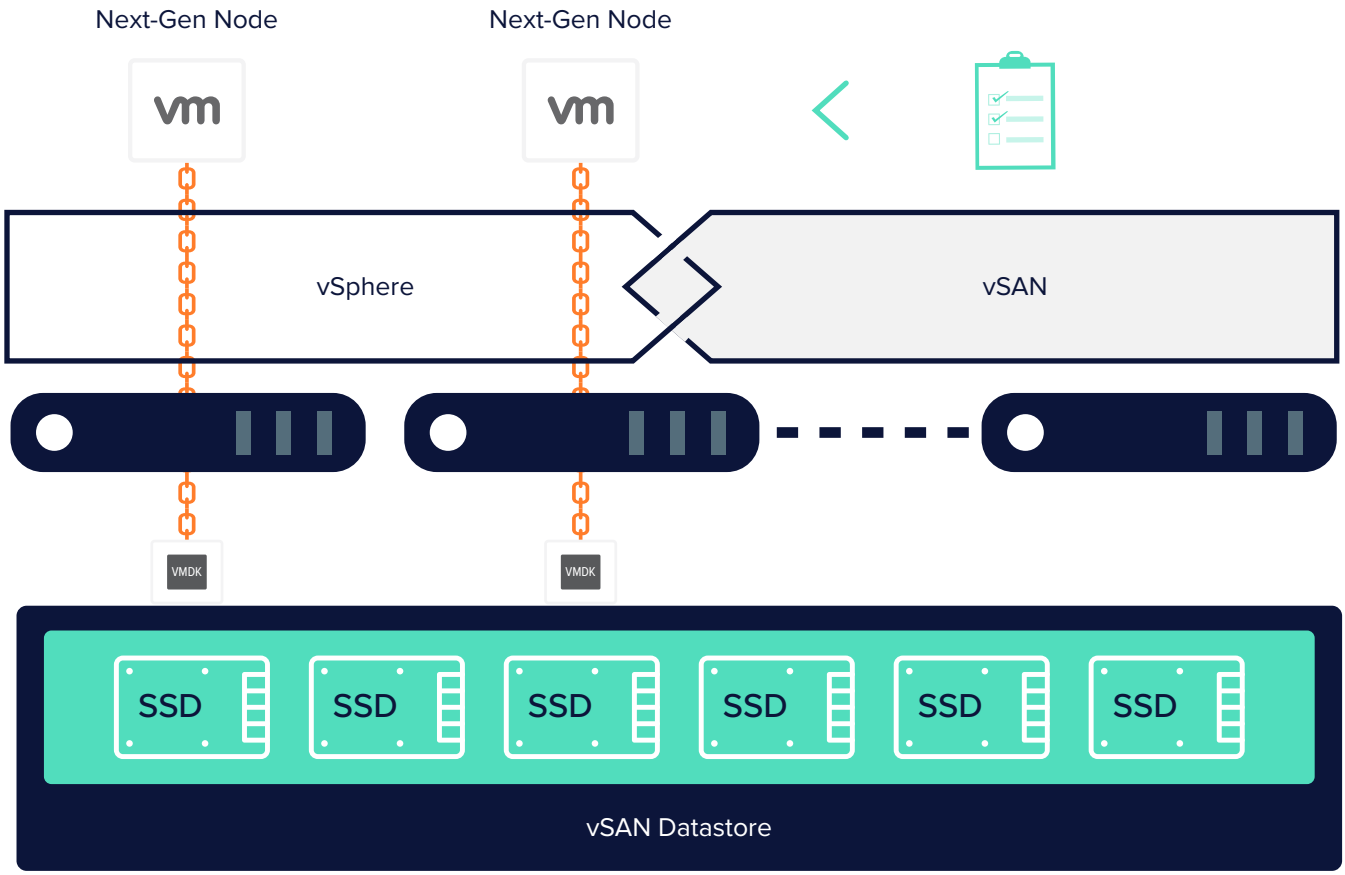

Figure 3. Host Affinity Depiction

As depicted in Figure 3, with "host affinity" policy enabled on a VM, its VMDKs reside on the same physical host as the VM. For each data block there is only copy in the vSAN layer and it's considered local to a VM.

4.3 Hardware Resources

Table 1 shows the hardware resources used in this solution. Each ESXi Server in the vSAN cluster has the following configuration.

Table 1. Hardware Resources per ESXi Server

PROPERTY	SPECIFICATION
Server	DELL R630
CPU and cores	2 sockets, 12 cores each of 2.3GHz with hyper-threading enabled
RAM	256GB
Network adapter	2 x 10Gb NIC
Storage adapter	SAS Controller Dell LSI PERC H730 Mini
Disks	Cache-layer SSD: 2 x 1.6TB Samsung Electronics Co Ltd Express Flash PM1725a AIC NVMe (controller included) Capacity-layer SSD: 8 x 800GB 2.5-inch Enterprise Performance SATA S3710

4.4 Software Resources

Table 2 shows the software resources used in this solution.

Table 2. Software Resources

SOFTWARE	VERSION	PURPOSE
VMware vCenter Server and ESXi	6.7 (vSAN 6.7 is included)	ESXi cluster to host virtual machines and provide vSAN Cluster. VMware vCenter Server provides a centralized platform for managing VMware vSphere environments.
VMware vSAN	6.7	Solution for Hyperconverged Infrastructure.
Ubuntu	16.04	Ubuntu 16.04 is used as the guest operating system of all the virtual machines.
DSE	6.0	DataStax Enterprise 6.0.
Cassandra-stress	3.10	The Cassandra-stress tool is a Java-based stress testing utility for basic benchmarking and load testing a Cassandra cluster.
ebdse	2.0.32	Ebdse is a benchmark tool developed by DataStax for benchmarking and load testing a Cassandra cluster.

4.5 VM and Database Configuration

We used the virtual machine settings as the base configuration as shown in Table 3. We configured DSE data directories on the VM data disk and the commit log directory on the log disk.

For VM sizing, the rule of thumb is that the aggregated CPU cores and memory should not exceed the physical resources to avoid contention. When calculating physical resources, we should count the physical cores before hyper-threading is taken into consideration.

Table 3. DSE VM and Database Configuration

PROPERTY	SPECIFICATION
vCPU	24
RAM	216GB
Disks	OS disk: 40GB Data disk: 1,150GB Log disk: 100GB

Table 4. Configuration of Stress Client VMs

PROPERTY	SPECIFICATION
vCPU	12
RAM	32GB
Disks	OS disk: 40GB

Best practices:

- We used a paravirtual SCSI controller and made each virtual disk use a separate controller for better performance.
- We configured a separate storage cluster on the hybrid management cluster to avoid any performance impact on the tested DSE cluster.
- We followed the recommended production setting for optimizing DSE installation on Linux.

4.6 Test Tool and Workload

Cassandra-stress: The Cassandra-stress tool is a Java-based stress testing utility for basic benchmarking and load testing a Cassandra cluster. The eight client nodes in our solution all ran Cassandra-stress. The three types of workloads used were:

- 90% write and 10% read
- 50% write and 50% read
- 10% write and 90% read

When started without a YAML file, Cassandra-stress creates a keyspace, keyspace1, tables, and standard1. These elements are automatically created the first time you run a stress test and are reused on subsequent runs. It also supports a YAML-based profile for testing reads, writes, and mixed workloads, plus the ability to specify schemas with various compaction strategies, cache settings, and types.

Writes in Cassandra are durable. All writes to a replica node are recorded both in memory and in a commit log on disk before they are acknowledged as a success. If a crash or server failure occurs before the memtables are flushed to disk, the commit log is replayed on restart to recover any lost writes.

For testing with a preloaded data set as a starting point: when we prepare the data set, we set commitlog_sync to “periodic” where writes may be acknowledged immediately, and the commit log is simply synced every 1,000 milliseconds (ms) for this stage only to ingest as fast as possible. We configured commitlog_sync in batch mode in the cassandra.yaml file for testing. DSE will not acknowledge writes until the commit log has been fsynced to the disk, this is the requirement for most real applications.

5. SOLUTION VALIDATION

In this section we present the test methodologies and results used to validate this solution.

5.1 Overview

We conducted an extensive validation to demonstrate vSAN as a platform for globally distributed cloud applications for production environments.

Our goal was to select workloads that are typical of today's modern applications; therefore, we tested a standard mixed workload without YAML file in the baseline performance testing and tested a user-defined typical IoT workload with YAML file as an example for a user to conduct testing based on their own data modeling.

We used large data sets to simulate a real case, so the data set of each node should exceed the RAM capacity, and we loaded a base data set of at least 500GB per node. In addition, we used OpsCenter to monitor the vitals of the DSE cluster continuously.

To generate the initial data set, we ran a 100% write workload on all eight clients concurrently until all data was loaded and all the compaction processes were completed. After the data was loaded, we took snapshots.

We loaded the snapshot as the preloaded data set before each workload testing. This solution included the following tests:

- **Performance testing –** to validate the cluster functions as expected for typical workloads for performance consistency and predictable latency in the production environments.
- **Operation testing –** to verify that the DSE cluster is robust and the performance impact is negligible during a normal operation. We conducted operations including:
  - Permanently removing a DSE node.
  - Adding a DSE node.
- **Scalability testing –** to validate the ability to scale throughput of the cluster in a near linear fashion by adding nodes, and to further validate that doing so did not impact the latency.
- **Resiliency and availability testing –** to verify vSAN's storage layer resiliency features combined with DSE's peer-to-peer design that enable this solution to meet performance and data availability requirements of even the most demanding applications under predictable failure scenarios, and to evaluate the impact on application performance.

5.2 Result Collection and Aggregation

- Cassandra-stress provides its results in both structured text format and as an HTML file. The throughput in operations per second (ops/sec), operation specifics in latencies, and other metrics are recorded for each test on each client instance.
- The test results are aggregated as each client instance produces its own output. The total throughput ops/sec is the sum-up of ops/sec in all clients and latencies are the average values of all client instances.
- For determining request percentiles, we selected the 50th and 99th percentile latency. The 99th percentile is the most commonly used one. We lined these three percentile points to show the latency curves. The flatter the curve is, the better the result will be. All latency charts are displayed in the same way throughout this paper.
- If there are any abnormal results, we use vSAN Performance Service to monitor each level of performance in a top-down approach of the vSAN stack.

5.3 Performance Testing

Performance Testing Procedure

Write the Base Data Set

A set of stress tests was run on each of the eight stress client VMs, wich parallelized the data generation across the cluster. The YAML file on each of the eight stress client VMs should list an increasing range of sequence numbers for the partition key (below is an example).

Node1 columnspec: - name: machine_id population: seq(1..100000000)
Node2 columnspec: - name: machine_id population: seq(100000000..200000000)
Node3 columnspec: - name: machine_id population: seq(200000000..300000000)
...
Node8 columnspec: - name: machine_id population: seq(700000000..800000000)

After the eight YAML files were configured and in place on each stress client, we ran a Cassandra-stress write test on each stress client node pointing to all of the eight DSE VMs. Then we ran nodetool status to see the amount of data per node. We repeated this process, continuing to increment the seq distributions on each stress client until 500GB per node was reached (note this may take several iterations; for example, 3 iterations).

After each node held over 500GB data, we replaced seq with uniform under the column_spec section of YAML file on all of the nodes. The purpose of this is because Cassandra-stress randomly inserts and reads in subsequent tests rather than in sequence.

Node1 columnspec: - name: machine_id population: uniform(1..300000000)
Node2 columnspec: - name: machine_id population: uniform(30000000..600000000)
Node3 columnspec: - name: machine_id population: uniform(600000000..900000000)
...
Node8 columnspec: - name: machine_id population: uniform(2100000000..2400000000)

Get Peak Performance

We started by having the eight stress client nodes with each running a 90% write and 10% read workload to get a measure of how we stress the cluster and to work toward maximizing the peak performance of the cluster, and then backing off from that to a workload we would advise customers to run in a typical scenario.

For throughput tests, each stress client node runs Cassandra-stress starting with 300 threads. You then monitor the DSE cluster to see CPU hover in the 90 to 100% range. If the CPU utilization is low, you can increase the threads; if the CPU utilization is high, you can reduce the number in 50 thread decrements. If tests are optimized for latency, a better gauge would be to run the stress test starting with 300 threads, then tune the number of threads up or down in increments of 50 until you see a 50% latency hover between 50ms and 100ms or at your required SLA. If you get 450 threads without pushing the latency to 50-100ms, you will need to add stress client VMs to generate additional load. Alternatively, for production scenarios, you should test your known schema against the required SLAs.

For either scenario you should then decrease the thread count by approximately 30% to ensure a real-world scenario. Clusters are never run at their peak to ensure that other nodes can absorb additional transactions of a set of nodes when they are down and for each node to ensure the overhead is available for management operations.

Test at Expected Load

We ran the stress tests from the eight stress client VMs and reduced the thread count on each one by 30%.

We conducted the performance testing with various parameters to evaluate the impact:

- **90% write and 10% read workload: we chose this for a typical IoT workload.**
- **50% write and 50% read workload: a generic mixed 1:1 workload to evaluate the storage performance.**
- **10% write and 90% read workload: this is to test a generic read-intensive workload.**

90% Write and 10% Read Performance Testing Results

The following Cassandra-stress commands were used on the eight stress nodes to generate the workload.

```
cassandra-stress user profile=stress.yaml ops\(insert=9,query_by_machine_id=1\) duration=1hr cl=ONE no-warmup -
mode native cql3 protocolVersion=3 -errors ignore -rate threads=350 -pop seq=1..14000000 contents=SORTED -insert
visits=fixed\(100\) -node <host1>,<host2>,<host3> -log file=90Write_10Read_10Mpop_seq1.log
hdrfile=90Write_10Read_10Mpop_seq1.hdr -graph file=90Write_10Read_10Mpop_seq1.html
title=90Write_10Read_10Mpop_seq1
```

The workload averaged 233,120 writes and 25,903 reads per second for a total of 259,023 transactions per second. The test showed a median latency of 2.62 milliseconds and a 99th percentile latency of 143.96ms. A breakout of insert and query latency is provided in Figure 5.

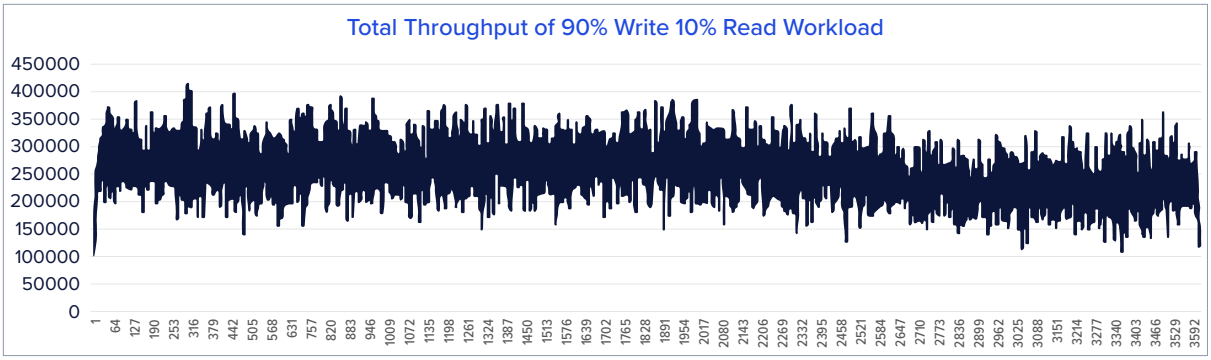

Figure 4. Total Throughput of 90% Write 10% Read Workload

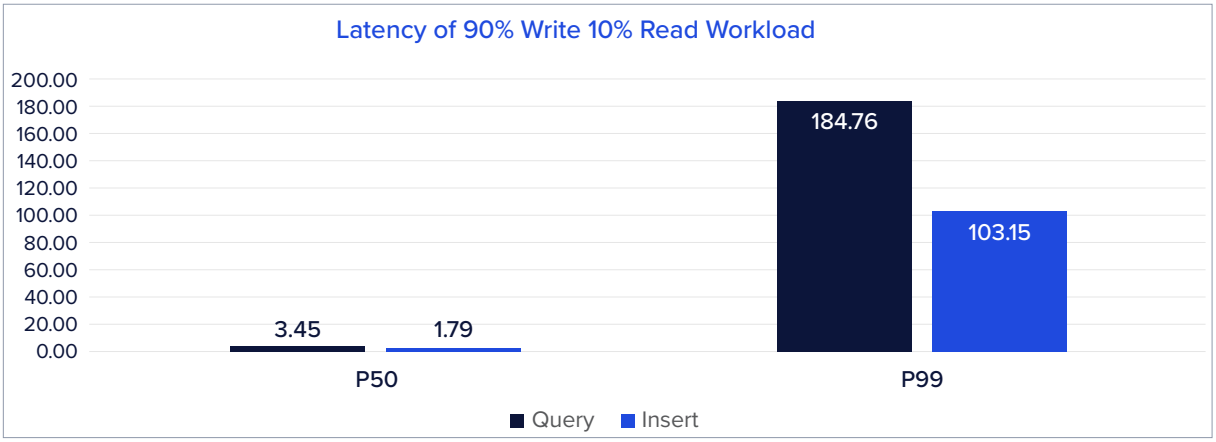

Figure 5. Latency of 90% Write 10% Read Workload

50% Write and 50% Read Performance Testing Results

Before running this test, we restored from snapshots and ran a 100% read 10-minute stress test.

cassandra-stress user profile=stress.yaml ops\(insert=5,query_by_machine_id=5\) duration=1hr cl=ONE no-warmup -mode native cql3 protocolVersion=3 -errors ignore -rate threads=350 -pop seq=1..14000000 contents=SORTED -insert visits=fixed\(100\) -node <host1>,<host2>,<host3> -log file=50Write_50Read_10Mpop_seq1.log hdrfile=50Write_50Read_10Mpop_seq1.hdr -graph file=50Write_50Read_10Mpop_seq1.html title=50Write_50Read_10Mpop_seq1

The workload averaged 113,814 writes and 113,839 reads per second for a total of 227,710 transactions per second. The test showed a median latency of 2.42 milliseconds and a 99th percentile latency of 113.04ms. A breakout of insert and query latency is provided in Figure 7.

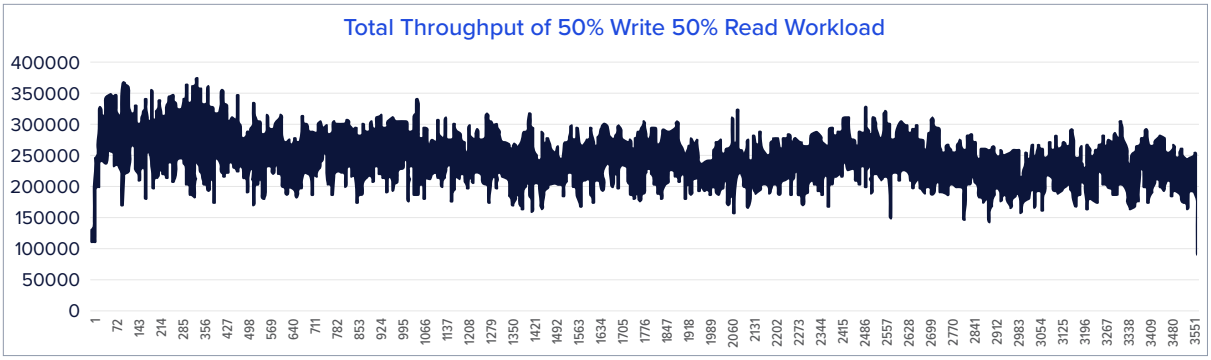

Figure 6. Total Throughput of 50% Write 50% Read Workload

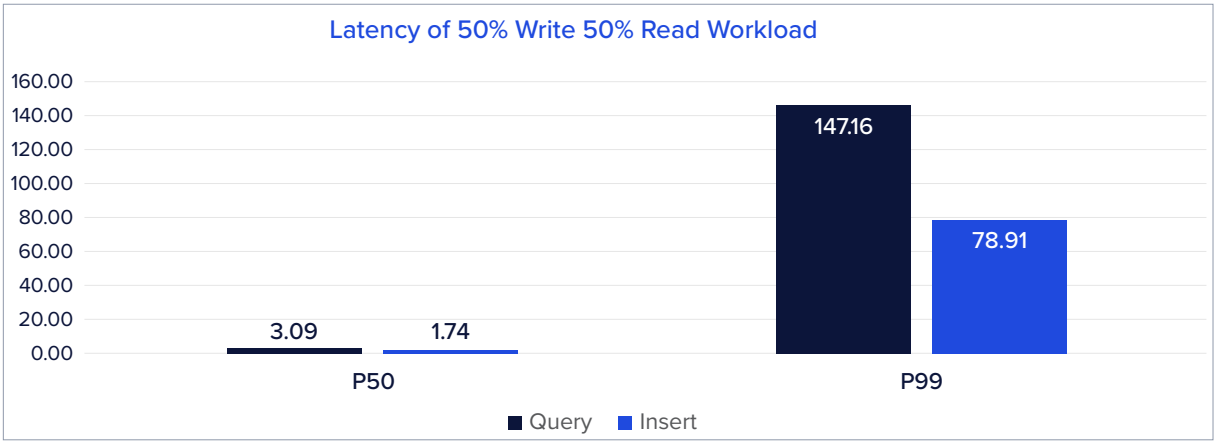

Figure 7. Latency of 50% Write 50% Read Workload

10% Write and 90% Read Performance Testing Results

Before running this test, we restored from snapshots and ran a 100% read 10-minute stress test.

cassandra-stress user profile=stress.yaml ops\(insert=1,query_ by_machine_id=9\) duration=1hr cl=ONE no-warmup -mode native cql3 protocolVersion=3 -errors ignore -rate threads=350 -pop s eq=1..14000000 contents=SORTED -insert visits=fixed\(100\) -no de <host1>,<host2>,<host3> -log file=10Write_90Read_10Mpop_seq 1.log hdrfile=10Write_90Read_10Mpop_seq1.hdr -graph file=10Wri te_90Read_10Mpop_seq1.html title=10Write_90Read_10Mpop_seq1

The workload averaged 22,923 writes and 206,387 reads per second for a total of 229,310 transactions per second. The test showed a median latency of 2.27 milliseconds and a 99th percentile latency of 89.37ms. A breakout of insert and query latency is provided in Figure 9.

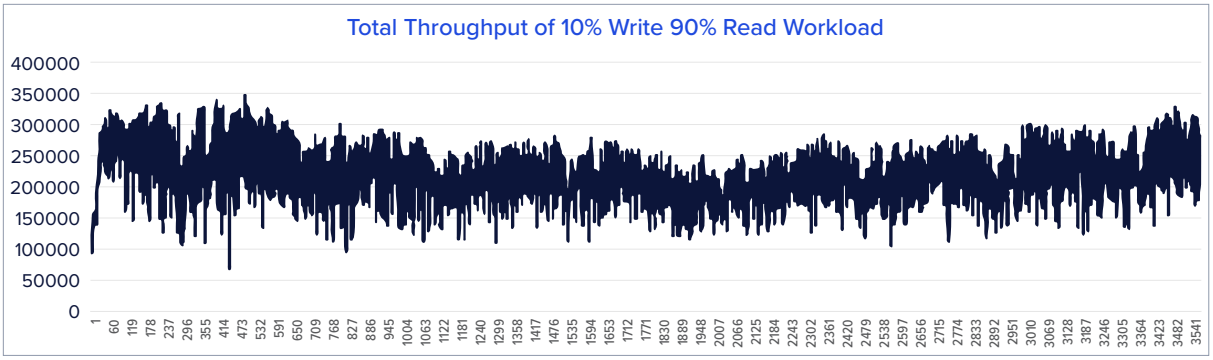

Figure 8. Total Throughput of 10% Write 90% Read Workload

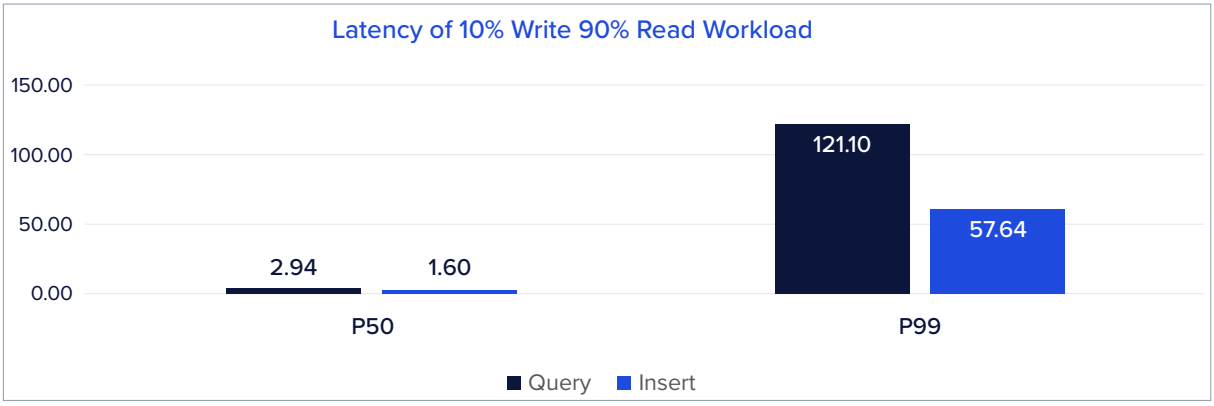

Figure 9. Latency of 10% Write 90% Read Workload

5.4 Operation Testing

Initially, the test bed was configured with eight DSE nodes as described in the performance testing sections. In this section, we evaluate the performance impact when we permanently removed a DSE node out of the cluster. During the test, we removed a node from the cluster so there were seven nodes after the removal. Figure 10 shows the performance monitoring result during this test.

Typically, after removal, the overall performance should be reduced by 1/8 in the ops/sec perspective. Our test lasted for one hour and the re-sync of the DSE cluster was still running during this one-hour time frame. We observed that the performance did eventually decrease and there was no interruption. The DSE cluster kept running normally during the node removal operation.

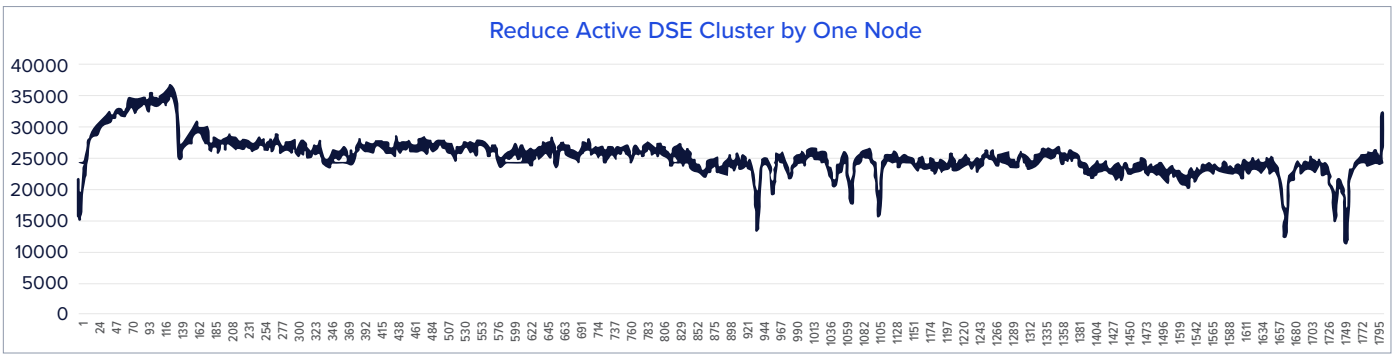

Figure 10. Permanently Remove One DSE Node

Adding a Node

After the test of removing a DSE node, we tested the impact of adding a DSE node back. Before the test started, there were seven nodes in this DSE cluster. During the test, we added a node back to the cluster so there were eight nodes after the operation. Figure 11 shows the performance monitoring result during this test.

Typically, after adding the node, the overall performance should increase by 1/7 in the ops/sec perspective. Our test lasted for one hour and the re-sync of the DSE cluster was still running during this one-hour time frame. We observed that the performance did eventually increase and there was no interruption. The DSE cluster ran normally during the node addition operation.

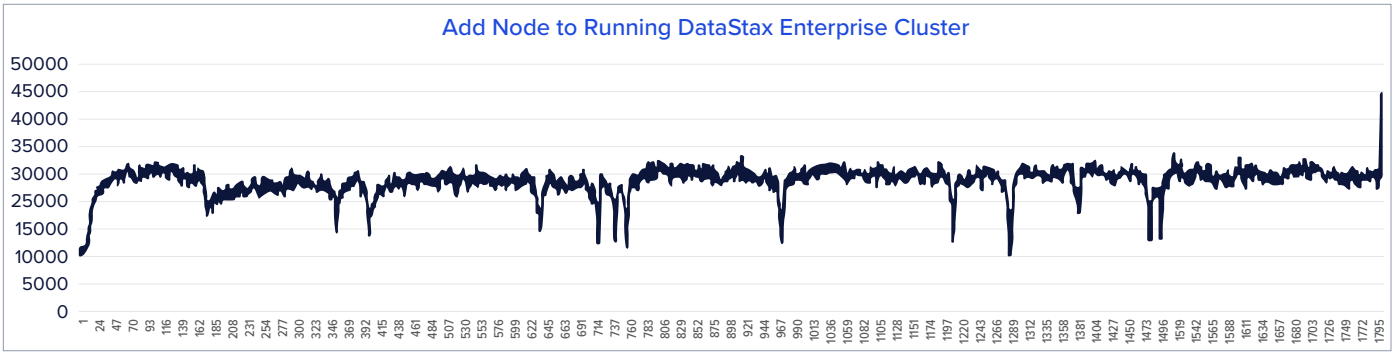

Figure 11. Add a DSE Node

5.5 Scalability Testing

For scalability testing, we tested the DSE cluster growing from 4-node to 6-node to 8-node. We had a 1:1 ratio of VM to physical server, so we also reduced the vSAN cluster's size to 4-server, 6-server, and 8-server correspondingly.

For performance testing, we also tested "90% write and 10% read", "50% write and 50% read", and "10% write and 90% read" to validate the scalability under different kinds of workloads.

We started by determining the peak performance of the 4-node DSE cluster under each workload then backed off by 25% in terms of ops/sec. This is considered closer to a real-world production scenario since in a production environment we would not usually push the servers to full.

Then after the DSE cluster scaled out to 6-node and 8-node, we validated that ops/sec can increase to 1.5 and 2 times respectively. We used the "ebdse" benchmark tool in this testing scenario since it has a deeper controllability of ops/sec rate.

90% Write and 10% Read

As described above, for a 4-node DSE cluster we started with threads=200 and got a maximum ops/sec of 64,000. Then we backed off by 25%, which was 48,000. When the DSE cluster grew to 6-node and 8-node, we also set the threads=300 and threads=400, respectively. As shown in Figure 12, the test results show that the ops/sec can grow linearly to 72,000 and 96,000 when the DSE cluster scales out.

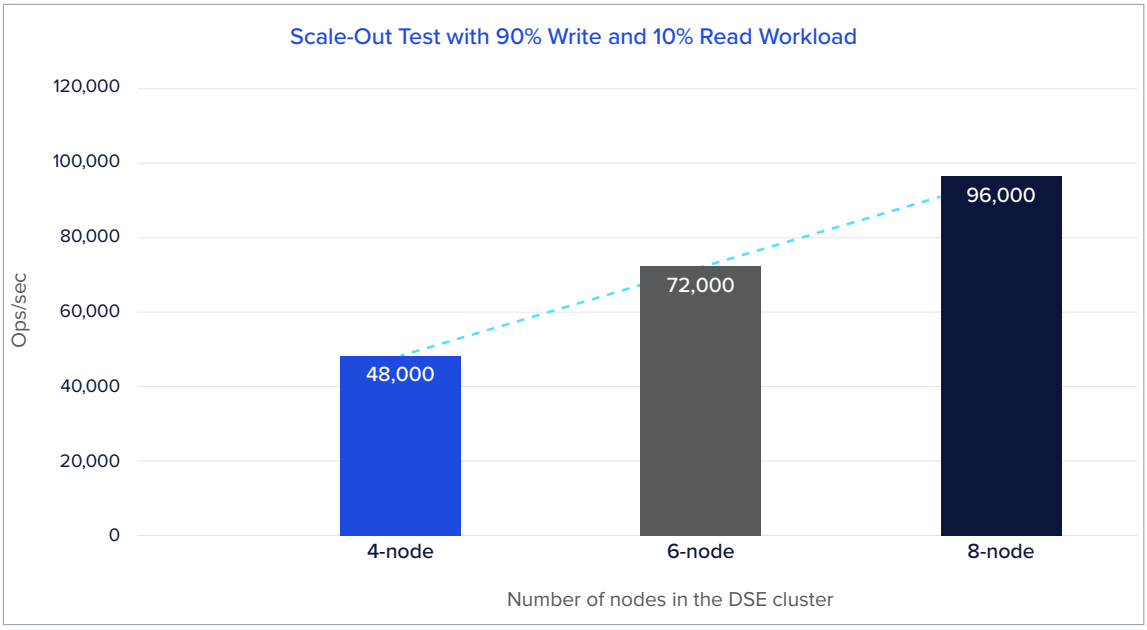

Figure 12. Scale-Out Test with 90% Write and 10% Read Workload

50% Write and 50% Read

For this workload we also used the same testing methodology as above. For a 4-node DSE cluster we started with threads=200 and got a maximum ops/sec of 9,300. Then we backed off by 25%, which was about 7,000. When the DSE cluster grew to 6-node and 8-node, we also set the threads=300 and threads=400, respectively. As shown in Figure 13, the test results show that the ops/sec can grow linearly to 10,500 and 14,000 when the DSE cluster scales out.

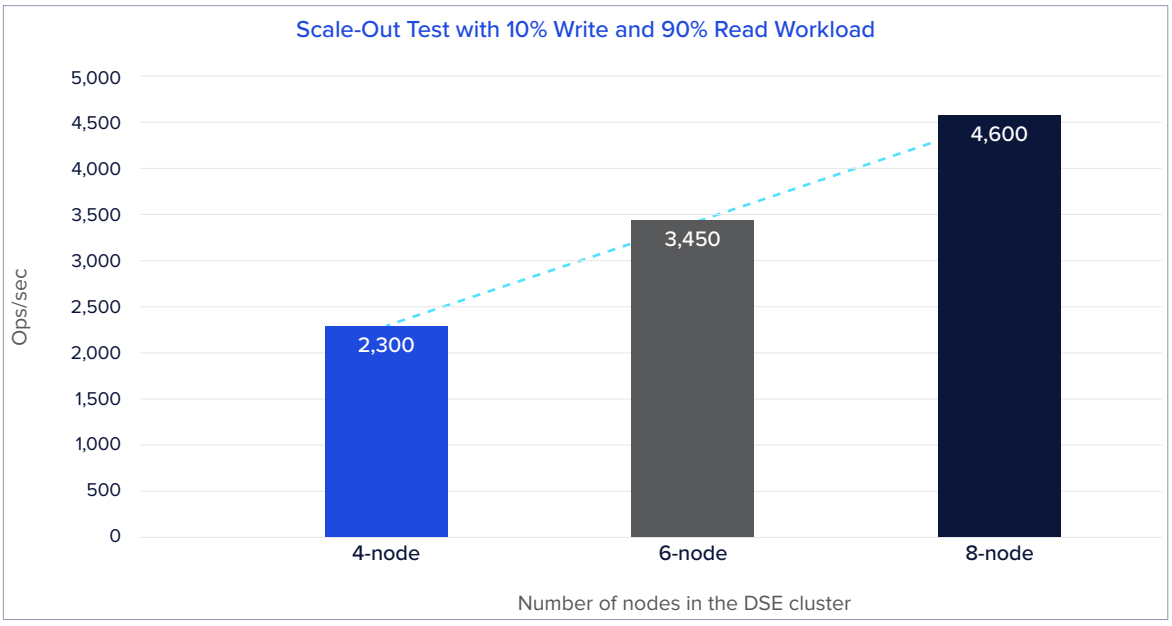

Figure 13. Scale-Out Test with 50% Write and 50% Read Workload

90% Read and 10% Write

For this workload, we also used the same testing methodology as above. For a 4-node DSE cluster we started with threads=200 and got a maximum ops/sec of 3,050. Then we backed off by 25%, which was about 2,300. When the DSE cluster grew to 6-node and 8-node, we also set the threads=300 and threads=400, respectively. As shown in Figure 14, the test results show that the ops/sec can grow linearly to 3,450 and 4,600 when the DSE cluster scales out.

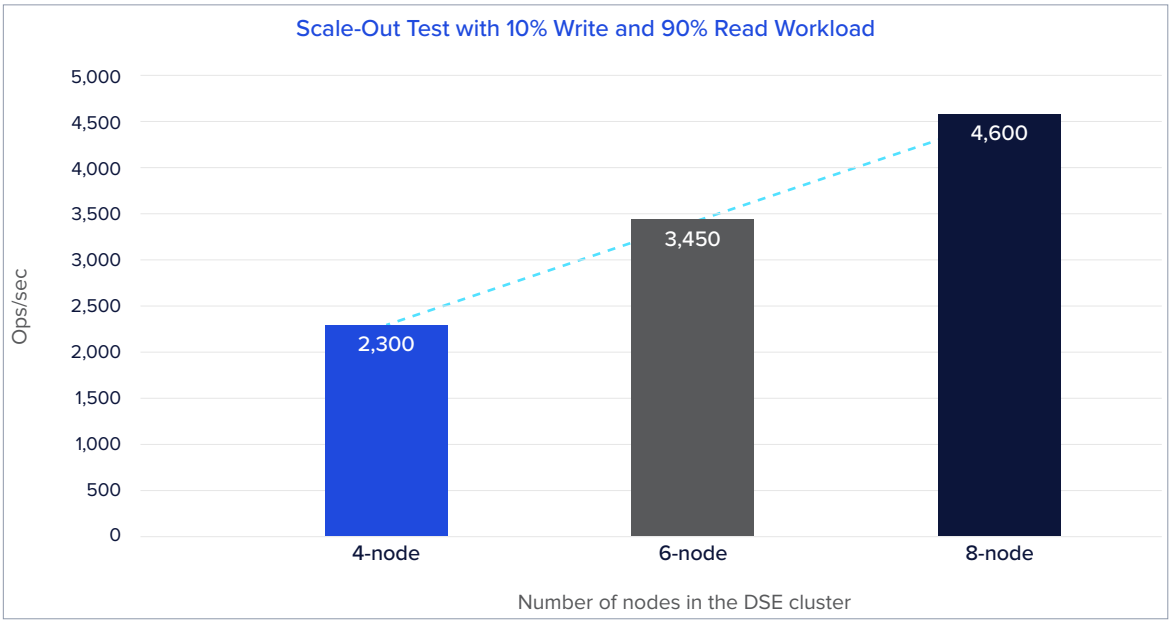

Figure 14. Scale-Out Test with 10% Write and 90% Read Workload

5.6 Resiliency and Availability

vSAN's storage layer resiliency features combined with DSE's peer-to-peer masterless design enables this solution to meet the data availability requirements of even the most demanding applications. A set of failure scenarios were created to validate data availability. In the failure testing, we used ebdse against a preloaded data set with RF=3, and we validated failures while running the performance testing workload. From the perspective of failure, we conducted three types of failure:

- A DSE VM shutdown in the DSE cluster, which will cause loss of a DSE node, but application resiliency ensures the service is not interrupted and only performance is impacted since the cluster is smaller.
- Kill a DSE application process in the DSE cluster, which will cause loss of a DSE node, but application resiliency ensures the service is not interrupted and only performance is impacted since the cluster is smaller.
- An ESXi host power-off in the vSAN cluster, which will cause loss of a DSE node, but application resiliency ensures the service is not interrupted and only performance is impacted since the cluster is smaller. Besides, since we were using the vSAN storage policy of "FTT=0" and "host affinity" enabled, no other vSAN objects would be impacted.

Notes:

- The testing was conducted using the XFS filesystem.
- Users must also mount the data and log filesystem with data=journal, all data and metadata are written to the journal before being written to disk, you can always replay interrupted I/O jobs in case of a crash. Other filesystems were not tested but the journaling requirement would remain constant.
- We used the "ebdse" as the client during the failure testing because of its ability to use advanced driver settings which is not possible in Cassandra-stress. These settings are as follows:
  - Heartbeat on at a 1-second interval
  - TCP recv timeout set to 2 seconds
  - TCP connection timeout set to 1 second
  - Keepalive enabled
  - All statements in the workload are idempotent
  - Speculative retry set within the driver using "speculative=4ms:50000" and on the table set to 3ms

VM Failure Testing

We simulated a DSE node failure by shutting down the VM. The following procedures were used:

1. We started the expected performance workload testing.
2. We shut down one of the DSE nodes from the vSphere web client console, and we did not bring it back online during the one-hour test.

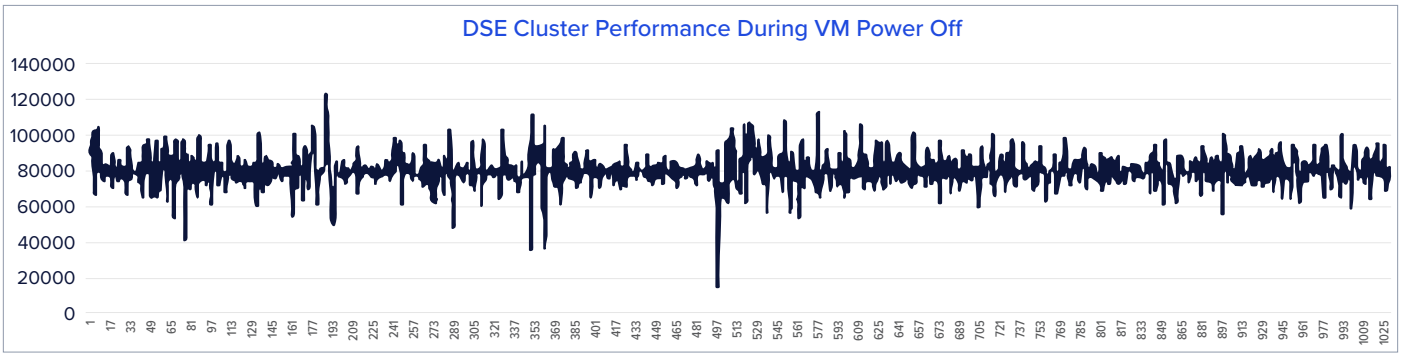

Figure 15. DSE VM Shut Down

3. We verified the expected results: DSE application service was not interrupted and the performance was not impacted since we left a 25% buffer for tolerating a node failure.

Kill the DSE Process Testing

We simulated another type of DSE node failure by killing the process. The following procedures were used:

1. We started the expected performance workload testing.
2. On one of the DSE nodes, we used the 'ps' command to identify the DSE process ID. Then we used the 'kill -9 <Process ID>' to kill a DSE process.
3. We verified the expected results: There was a transient ops/sec drop but it resumed to normal quickly. Meanwhile, DSE application service was not interrupted and the stable state performance after the failure was not impacted since we left a 25% buffer for tolerating a node failure.

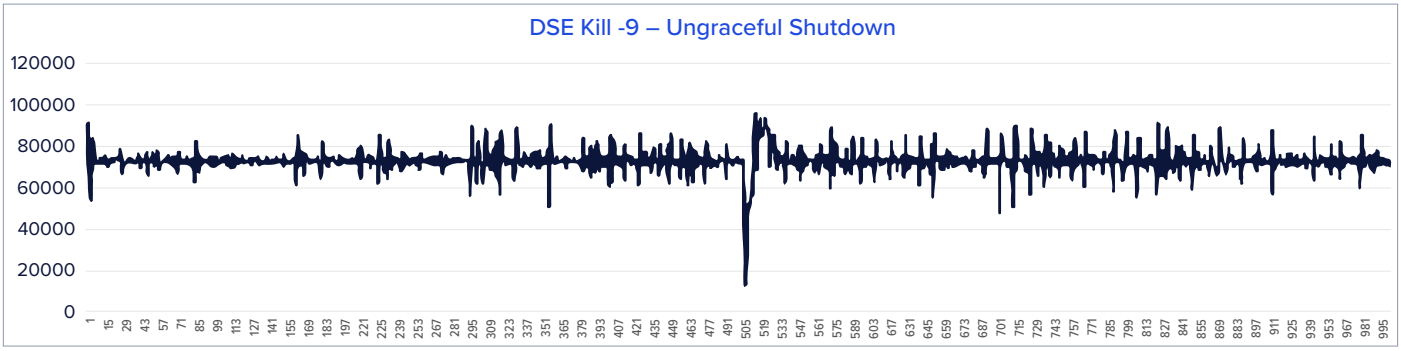

Figure 16. Kill a DSE Process on One of the DSE Nodes

Host Failure Testing

We simulated another type of failure by powering off an ESXi host. The following procedures were used:

1. We started the expected performance workload testing.
2. From the vSphere Web Client, we chose an ESXi host and clicked the "power off" button.
3. We verified the expected results: There was a transient ops/sec drop but it resumed to normal quickly. Meanwhile, DSE application service was not interrupted and the stable state performance after the failure was not impacted since we left a 25% buffer for tolerating a node failure.

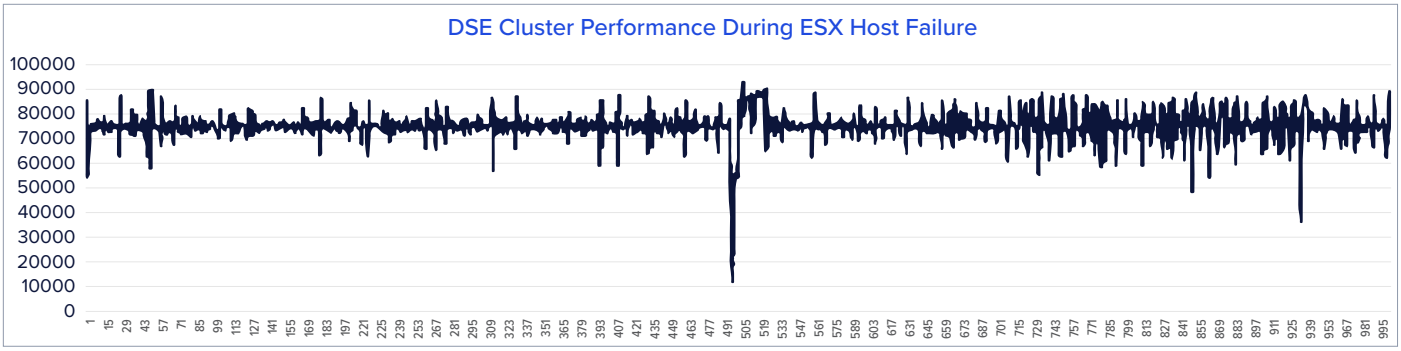

Figure 17. Host Failure Testing by Powering Off an ESXi Host

Failure Testing Summary

The failure testing results are summarized in Table 5.

Table 5. Failure Testing Results

FAILURE TYPE	TEST DESCRIPTION	RESULT
VM failure	Shutdown a VM and power on after an hour.	Service was not interrupted, performance was not impacted.
Kill the DSE process	Use the Linux 'kill' command to kill a DSE process.	There was a transient performance drop but it resumed to normal quickly. The stable state performance was not impacted after the failure.
Host failure	Power off an ESXi host during the testing.	There was a transient performance drop but it resumed to normal quickly. The stable state performance was not impacted after the failure.

6. BEST PRACTICES

This section provides the recommended best practices for this solution.

6.1 Best Practices

When configuring DSE in a vSAN cluster, consider the following best practices:

- Before deployment, plan the database size and required performance level. This will ensure the hardware is properly deployed and software settings are best suited to serve the requests. Plan the hardware resource per the VMware vSAN Design and Sizing Guide. For DSE database capacity planning, the database routinely requires disk capacity for compaction and repair operations during normal operations. For optimal performance and cluster health, DataStax recommends not filling disks to capacity, but running at 50% to 80% capacity depending on the compaction strategy and size of the compactions, so be sure to consider the extra capacity needed.
- When configuring the virtual CPU and memory for DSE VMs, choose an appropriate CPU and memory number to best suit the users' requirements. VM aggregated CPU cores and memory should not exceed the physical resources to avoid contention. Choosing an optimum JVM heap size and leaving enough memory for the OS file cache is important, follow the set the heap size for optional Java garbage collection in DataStax Enterprise guide to determine the optimum heap size.
- Use different virtual disks for the DSE data directory and the log directory. If components of DSE VMs with one data disk are not fully distributed across the vSAN datastore, customers can use multiple virtual disks for the data directory to make full use of physical disks.
- Deploy one DSE node per physical host for best performance.
- Use the normal operating system tuning parameters. Turn off the swap.
- Set the type of virtual SCSI controller to paravirtual. The maximum number of SCSI controller is four, make each virtual disk use a separate controller if the VMDK number is equal to or less than four.
- Configure a separate storage cluster on the management cluster to avoid performance impact on the tested DSE cluster.
- For production environments, after the base data set is loaded and the compaction is finished, take DSE application-level snapshots for reuse.
- Higher consistency levels affect performance, so choose an appropriate consistency level to meet application requirements.
- Tune client threads based on the specific workload. In general, generate the peak performance, and reduce thread count by 30%.
- Follow DataStax recommended production settings.

In addition to on-premises deployment, you can also run DSE on VMware Cloud. See the following network connectivity best practices for hybrid and multi-cloud deployment.

Hybrid and Multi-Cloud Best Practices

Generally, an internet-based Virtual Private Network (VPN) will meet the need for a reliable, secure connection between the on-premises and public cloud, but if it is crucial for low-latency and high-speed network connections, Direct Connect from AWS and ExpressRoute from Microsoft Azure come into play.

AWS Direct Connect (DX) is a cloud service solution that makes it easy to establish a dedicated network connection between the on-premises environment to AWS. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. See Using AWS Direct Connect with VMware Cloud on AWS.

In our demo hybrid deployment, VMware has a corporate VPN into AWS. Our SDDC in VMware Cloud on AWS is connected to that connection via DX, there are two DX private virtual interfaces (VIFs). It already has redundancy/ backup, route-based IPSEC VPN as standby is not very useful in this scenario. If a customer is trying to save cost and only has one DX private VIF, VPN as standby can be very useful for providing backup to DX private VIF.

See DataStax Enterprise on VMware Cloud for Hybrid Deployment for more details.

7. CONCLUSION

7.1 Conclusion

Overall, deploying, running, and managing DSE applications on VMware vSAN provides predictable performance and high availability. All storage management moves into a single software stack, thus taking advantage of the security, performance, scalability, operational simplicity, and cost-effectiveness of vSAN.

It is simple to expand using a scale-up or scale-out approach without incurring any downtime. With the joint efforts of VMware and DataStax, customers can deploy DSE clusters on vSAN for their modern cloud applications with ease and confidence in production environments.

8. APPENDIX

8.1 Appendix

Here is the stress.yaml file we used in the testing.

```
# Keyspace Name keyspace: iot_space
# The CQL for creating a keyspace (optional if it already exists) keyspace_definition: |
CREATE KEYSPACE iot_space WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 3}; table: iot_table
# The CQL for creating a table you wish to stress (optional if it already exists) table_definition: |
CREATE TABLE iot_space.iot_table ( station_id blob,
machine_id blob, machine_type text, sensor_value double, time bigint,
PRIMARY KEY (machine_id, time)
) WITH CLUSTERING ORDER BY (time DESC)
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'};
### Column Distribution Specifications ### columnspec:
-           name: station_id
population: seq(1..5b) # 5 Billion potential station_ids size: fixed(16)
-           name: machine_type
size: uniform(10..20) # machine_type is 10-20 chars population: uniform(1..10) # there are 10 types of machine
-           name: machine_id size: fixed(16)
population: seq(1..5b) # 5 Billion unique machines
-           name: sensor_value
population: gaussian(0..1000) # sensor_values range from 0-1000 and follow a gaussian distribution
-           name: time
cluster: fixed(100) # 100 sensor_values updates per machine population: seq(0..99)
### Batch Ratio Distribution Specifications ### insert:
partitions: fixed(1)
select:    fixed(1)/100 # Inserts will be single row batchtype: UNLOGGED
# A list of queries you wish to run against the schema #
queries: query_by_machine_id:
cql: SELECT machine_id, sensor_value, time FROM iot_table WHERE machine_id = ? and time >= 90 LIMIT 10 fields: samerow
```

9. REFERENCE

9.1 Reference

See more vSAN details and customer stories:

- vSAN
- Virtual Blocks Blog
- Customer Stories

For more information regarding DataStax and VMware, see DataStax.com/vmware.

10. ABOUT THE AUTHORS

10.1 About the Authors

Sophie Yin, Senior Solutions Architect in the Product Enablement team of the Storage and Availability Business Unit at VMware wrote the original version of this paper.

Victor Chen, Senior Solutions Engineer in the Product Enablement team of the Storage and Availability Business Unit at VMware wrote the original version of this paper.

Kathryn Erickson, Director of Strategic Partnerships at DataStax, worked with Sophie Yin and Victor Chen on this paper.

Catherine Xu, Senior Technical Writer in the Product Enablement team of the Storage and Availability Business Unit edited this paper to ensure that the contents conform to the VMware writing style.

ABOUT DATASTAX

DataStax delivers the only active everywhere hybrid cloud database built on Apache Cassandra™: DataStax Enterprise and DataStax Distribution of Apache Cassandra, a production-certified, 100% open source compatible distribution of Cassandra with expert support. The foundation for contextual, always-on, real-time, distributed applications at scale, DataStax makes it easy for enterprises to seamlessly build and deploy modern applications in hybrid cloud. DataStax also offers DataStax Managed Services, a fully managed, white-glove service with guaranteed uptime, end-to-end security, and 24x7x365 lights-out management provided by experts at handling enterprise applications at cloud scale. More than 400 of the world's leading brands like Capital One, Cisco, Comcast, Delta Airlines, eBay, Macy's, McDonald's, Safeway, Sony, and Walmart use DataStax to build modern applications that can work across any cloud. For more information, visit www.DataStax.com and follow us on Twitter @DataStax.