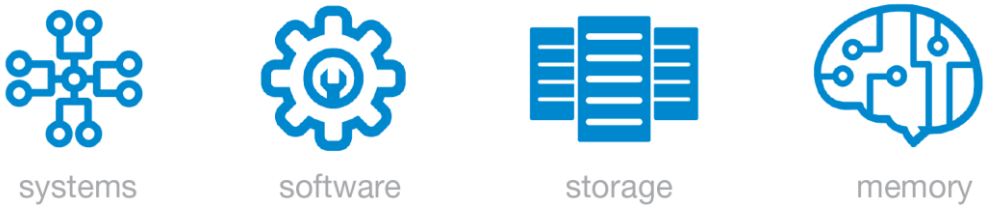
A **Micron** Reference Architecture

Micron® Accelerated All-Flash vSAN™ 6.7 Update 1 on Dell® R740xd Solution

Reference Architecture

Collin Murphy, Senior Storage Solutions Engineer
Doug Rollins, Principal Technical Marketing Engineer



Contents

Executive Summary

This document describes the optimized configuration of an all-flash VMware® vSAN™ platform combining two classes of SATA SSDs in its cache and capacity tiers, Dell® R740xd server platforms and 25 GbE networking. The combination of SATA SSDs with standard servers provides an optimal balance of performance and cost. Similar to an AF-8 configuration, this VMware vSAN 6.7 Update 1 (U1) all-flash reference design enables:

- **Faster deployment:** The configuration has been pre-validated and is thoroughly documented to enable faster deployment.
- **Balanced design:** The right combination of cache and capacity SSDs, DRAM, processors and networking ensures subsystems are balanced and performance is matched to optimize results versus solution cost, rather than highest-performance density per unit. See our Micron Accelerated Solutions site for designs that focus on density.
- **Broad deployment:** Complete tuning and performance characterization across multiple IO profiles for broad deployment across multiple workloads

This document also details the hardware and software building blocks of the reference architecture (RA) as well as the measurement techniques used to characterize the RA's performance and its composition, including the vSphere® and network switch configurations, vSAN tuning parameters, Micron® reference nodes and Micron SSD configurations.

The configuration in this RA ensures easy integration and operation with vSAN 6.7 U1, offering predictably high performance that is easy to deploy and manage. It is a pragmatic blueprint for administrators, solution architects and IT planners who need to build and tailor a high-performance vSAN infrastructure that scales for IO-intensive workloads.

This reference architecture focuses on SATA SSDs. Find other reference architectures that optimize for performance, cost and/or density at the Micron Accelerated Solutions site. Note that the performance shown in this document was measured using the components also specified in this document. Different component combinations may yield different results.

Why Micron for This Solution

SSDs and DRAM can represent up to 80 percent of the value of today's advanced server/storage solutions. Micron is a leading designer, manufacturer and supplier of advanced storage and memory technologies with extensive in-house software, application, workload and system design experience.

Micron's silicon-to-systems approach provides unique value in our reference architectures, ensuring these core elements are engineered to perform in highly demanding applications like vSAN and are holistically balanced at the platform level. This reference architecture solution leverages decades of technical expertise as well as direct, engineer-to-engineer collaboration with industry leaders.

Micron Reference Architectures

Micron reference architectures are optimized, pre-engineered, enterprise-leading solution templates co-developed between Micron and industry-leading hardware and software companies.

Designed and tested at Micron's Storage Solutions Center, they provide end users, system builders, independent software vendors (ISVs) and OEMs with a proven template to build next-generation solutions with reduced time investment and risk.

Solution Overview

A vSAN storage cluster is built from multiple vSAN-enabled vSphere nodes for scalability, fault-tolerance and performance. Each node is based on commodity hardware and utilizes VMware's ESXi™ hypervisor to:

- Store and retrieve data
- Replicate (and/or deduplicate) data
- Monitor and report on cluster health
- Redistribute data dynamically (rebalance)
- Ensure data integrity (scrubbing)
- Detect and recover from faults and failures

Enabling vSAN on a vSphere cluster creates a single vSAN data store. When virtual machines (VMs) are created, virtual machine disks (VMDKs) can be carved out from the vSAN data store. When a VMDK is created, the designated vSAN storage policy and underlying algorithms handle any fault tolerance logic instead of the host. When the host writes to its VMDK, vSAN handles all necessary operations such as data duplication, erasure coding, checksum and placement based on the selected storage policy.

Storage policies can be applied to the entire data store, a VM or a VMDK. Using storage policies allows a user to determine whether to add more performance, capacity or availability to an object. Numerous storage policies can be used on the same data store, enabling high-performance VMDKs to be created for things such as database log files and high-capacity/availability disk groups for critical data files.

Figure 1 shows the logical layers of the vSAN stack, from the hosts to the vSAN data store.

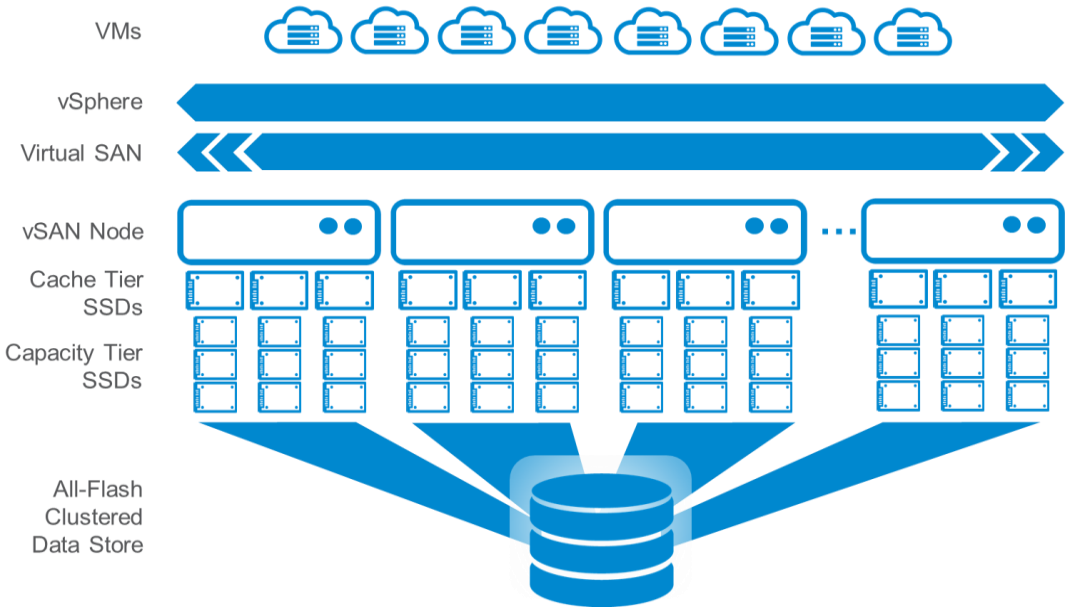


Figure 1: vSAN Architecture

Client VMs write to vSAN VMDKs while the vSAN algorithms determine how data is distributed across physical disks, depending on the storage policy for that VMDK. Below are some storage policy options; settings used in tested configurations are shown in Table 5.

- **Primary levels of failures to tolerate (FTT)**: Specifies how many copies of data can be lost while still retaining full data integrity. By default, this value is set to 1, meaning there are two copies of every piece of data, as well as potentially a witness object to make quorum in the case of an evenly split cluster. See Table 5 for the FTT settings used for each tested configuration.

- **Failure tolerance method (FTM)**: Specifies the method of fault tolerance, which is 1) RAID-1 (Mirroring) or 2) RAID-5/6 (Erasure coding). RAID-1 (Mirroring) creates duplicate copies of data in the amount of 1 + FTT. RAID-5/6 (Erasure coding) stripes data over three or four blocks, as well as one or two parity blocks, for RAID-5 and RAID-6 respectively. Selecting FTT = 1 means the object will behave similar to RAID-5; selecting FTT = 2 means the object will be similar to RAID6. The default is RAID-1 (Mirroring).
- **Object space reservation (OSR)**: Specifies the percentage of the object that will be reserved (thick-provisioned) upon creation. The default value is 0%.
- **Disable object checksum**: If Yes is selected, the checksum operation is not performed. This reduces data integrity but can increase performance (when performance is more important than data integrity). The default value is No.
- **Number of disk stripes per object (DSPO)**: The number of objects over which a single piece of data is striped. This applies to the capacity tier only (not the cache tier). The default value is 1 but can be set as high as 12. Note that vSAN objects are automatically split into 255GB chunks but are not guaranteed to reside on different physical disks. Increasing the number of disk stripes guarantees that they reside on different disks on the host if possible.

Design Overview

This section describes the configuration of each component of the reference architecture and how they are connected.

Hardware

Mellanox® SN2410 25 GbE Switches

Micron Components

3X Micron 5200 MAX SATA SSDs (960 GB)

9X Micron 5200 ECO SATA SSDs (3840 GB)

12X Micron 2666 MHz RAM (32 GB)

1X Micron Enterprise SATA SSDs (OS drive)

Other Components

Dell® R740xd 2-socket rack mount server

2X Intel Xeon Gold 6142 16-core 2.60GHz CPUs

vSphere 6.7 U1 (10244745)

ESXi 6.7.0 (10764712)

vSAN 6.7 U1

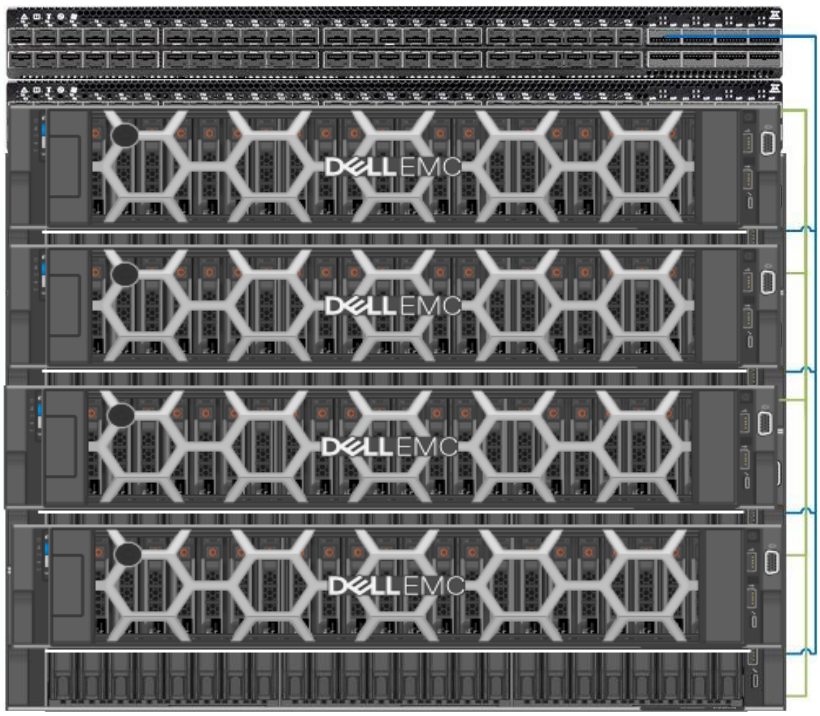


Figure 2: vSAN RA Hardware

Software

VMware's vSAN is an industry-leading hyper-converged infrastructure (HCI) solution, combining traditional virtualization with multi-host software-defined storage. vSAN is a technology that is part of VMware's vSphere environment, coupled with the ESXi type-1 hypervisor.

Micron chose vSAN 6.7 U1 for this solution because of its benefits to a broad range of customers, applications and workloads. According to VMWare's Virtual Blocks blog (October 16, 2018), vSAN 6.7 U1 adds[1]:

- **Streamlined deployment**: vSAN 6.7 U1 includes simplified day one and two operations by streamlining the deployment process, improving lifecycle management, reducing disruptions during maintenance operations, and improving capacity reporting. These updates help administrators more quickly and easily deploy and extend infrastructure while minimizing disruptions and keeping the environment up to date.
- **Simplified operations (cluster quick-start)**: A new "quick-start" guided cluster creation wizard gives administrators a streamlined mechanism for deploying vSAN and non-vSAN clusters. An easy-to-use, step-by-step configuration wizard makes creating a production-ready vSAN cluster effortless.
- **Driver and firmware updates using Update Manager:** Updated in vSAN 6.7 U1, all ESXi, driver and firmware update functions previously handled by the Configuration Assist workflow have been moved to vSphere Update Manager.
- **Decommissioning and maintenance mode safeguards:** Since each vSAN host in a cluster contributes to the cluster storage capacity, putting a host into maintenance mode takes on an additional set of tasks when compared to a traditional architecture.

In accordance with VMware documentation, a vSAN cluster is created by installing ESXi on at least three nodes (four or more is recommended) and enabling vSAN via a license in the vSphere client.

vSAN uses a two-tier storage architecture where all write operations are sent to the cache tier and are subsequently de-staged to the capacity tier over time. Up to 600GB of cache tier storage can be utilized per disk group, with up to five disk groups per host.

vSAN can operate in two modes:

- **Hybrid:** SSDs for the caching tier and rotating media for the capacity tier
- **All-flash:** SSDs for both cache and capacity tiers

With a hybrid configuration, the cache tier is used as both a read and write cache, keeping hot data in the cache to improve hybrid design performance. In this configuration, 70% of the cache tier capacity is dedicated to the read cache and the remaining 30% is dedicated to the write buffer.

In an all-flash configuration, 100% of the cache tier is used for the write buffer, with no read cache.

[1] See https://blogs.vmware.com/virtualblocks/2018/10/16/whats-new-in-vsan-6-7-update-1/ for additional details on vSAN 6.7 U1 features.

Micron Components

This RA employs Micron's 5200 MAX and 5200 ECO enterprise SATA SSDs (MAX for the cache tier and ECO for the capacity tier) to optimize cache and capacity performance versus overall cost for general vSAN use cases. This solution also utilizes 384GB Micron 2666 MHz DRAM (32GB x 12).

SSD	Use	Random Read	Random Write	Read Throughput	Endurance (TBW)
5200 MAX	Cache Tier	95,000 IOPS	75,000 IOPS	540 MB/s	8.8 PB
5200 ECO	Capacity Tier	95,000 IOPS	17,000 IOPS	540 MB/s	7.7 PB

Table 1: Micron SSDs

See www.micron.com for additional details and specifications on these and other Micron SSD products.

Server Platforms

This solution utilizes Dell R740xd servers (2U, dual-socket) based on the Intel® Purley platform. Each server is configured with two Intel Gold 6142 processors, (with 16 cores at 2.60 GHz per processor). These processors align with VMware's AF-8 minimum requirements (VMWare's nomenclature for a large-sized all-flash configuration).

Switch

vSAN utilizes commodity Ethernet networking hardware. This solution uses two Mellanox SN2410 switches for all cluster-related traffic. The switches are interconnected with a single QSFP+ cable between them. The Spanning Tree protocol is enabled to avoid loops in the network. All ports are configured in general mode, with VLANs 100–115 allowed. Each server is connected via a QSFP+ quad-port breakout cable.

In accordance with VMware's best practices, vSAN should have at least three separate logical networks that are all segregated using different VLANs and subnets on the same switches. The three networks in this RA, and their respective VLANs, are as follows:

- Management/VM network: VLAN 100, subnet 172.16.17.X/16
- vMotion: VLAN 103, subnet 192.168.1.X/24
- vSAN: VLAN 104, subnet 192.168.2.X/24

While using different subnets or VLANs alone would suffice, adding both ensures that each network has its own separate broadcast domain, even if an interface is configured with either a duplicate VLAN or IP address. To ensure availability, one port from each server is connected to each of the two switches, and the interfaces are configured in an active/passive mode.

Tip: Networking

Use different subnets and VLANs to ensure each network has its own separate broadcast domain (even if an interface is configured with an incorrect VLAN or IP address).

Connect each node to both switches to ensure availability.

Following VMware's best practices for vSAN networking, this study utilizes a Virtual Distributed Switch with network I/O control enabled. Using network I/O control, vSAN is given "high" shares, as well as 18.75 Gb/s reservation, which is the highest allowed. This ensures that vSAN is not limited by the networking.

Network Interface Cards

Each server has a single dual-port BCM57414 25 GbE network interface card (NIC), with one port of each NIC connected to one of each of the switches to ensure high availability in case one of the two switches is lost. vSAN is active on one link and in standby on the other while management and vMotion are active on the opposite link. This ensures that vSAN gets full utilization of one of the links and is uninterrupted by any external traffic.

Hardware Components

The tables below summarize the hardware components used in this RA. If other components are substituted, results may vary from those described.

Node Components	
Dell R740xd 2-socket rack mount server	2X Intel Xeon Gold 6142 16-core 2.60 GHz CPUs
Micron 384GB 2666 MHz DRAM (32GB x 12)	1X 1920GB Micron Enterprise SATA SSD (OS boot drive)
3X Micron 960GB SATA SSDs (5200 MAX)	1X Broadcom® Limited BCM57414 NetXtreme®-E 25 GbE NIC
9X Micron 3.84TB SATA SSDs (5200 ECO)	1X Dell HBA330

Table 2: Components

Network Infrastructure

Network Components	
2X Mellanox SN2410 25 GbE switches	2X Mellanox QSFP+ Copper Breakout Cables

Table 3: Network

Software

Software Components	
Server BIOS version 1.6.12	Disk Format version 7
vCenter Server Appliance 6.7.0.20000 build 10244745	HBA driver 17.00.01.00
ESXi build 10764712	HBA firmware 16.00.04.00
vSAN 6.7 U1	5200 MAX firmware D1MU404
	5200 ECO firmware D1MU404

Table 4: Software

Planning Considerations

Part of planning any configuration is determining what hardware to use. Configuring a system with the most expensive hardware might mean overspending, whereas selecting the cheapest hardware possible may not meet performance requirements.

This study targeted a configuration based on VMware's AF-8 specifications, which aims to provide up to 80K IOPS per node. An AF-8 configuration typically calls for at least 12TB of raw storage capacity per node, dual processors with at least 12 cores per processor, 384GB of memory, two disk groups per node with six capacity drives, and a 10 GbE networking minimum. This configuration utilizes three disk groups per node with three capacity drives per disk group, resulting in three cache drives and nine capacity drives per node.

For further information on AF-8 requirements, see VMware's vSAN Hardware Quick Reference Guide.

It is important to note that performance can be increased in many ways, but they all come with added cost. Using a processor with a higher clock speed would potentially add performance, but it could add thousands of dollars to the configuration. Adding more disk groups would also add significant performance, but again, it would require additional cache drives, which would add significant cost to the solution. Furthermore, adding faster networking — like 40 GbE, 100 GbE or Infiniband — would potentially yield better performance, but the additional required hardware would add significant cost to the solution. The solution chosen for this RA is moderately sized for good performance at a balanced price point based on testing of vSAN in Micron labs.

Measuring Performance

Test Methodology

Benchmarking virtualization can be a challenge because of the many different system components that can be tested. However, this RA aims to primarily analyze vSAN's storage component and its ability to deliver many transactions at a low latency. As a result, this study focuses on using synthetic benchmarking to gauge storage performance.

The benchmark tool used for this study was HCIBench. HCIBench is primarily a wrapper around Oracle's Vdbench with extended functionality to deploy and configure VMs, run vSAN Observer, and aggregate data. HCIBench also provides an ergonomic web interface to run tests.

In this case, HCIBench was deployed as a VM template. A separate vSAN cluster was set up for all infrastructure services, such as for HCIBench, DNS, routing, and so on. The HCIBench Open Virtualization Format (OVF) template was deployed to this cluster, and a VM was created from the template. An additional virtual network was created on a separate VLAN (111), and the HCIBench VM's virtual NIC was assigned to this network to ensure that it could not send unwarranted traffic.

vSAN offers multiple options to define a storage policy. To understand how each of these affect performance, four test configurations were chosen:

Configuration	FT Method	FTT	Checksum	Deduplication + Compression
Baseline	RAID-1 (Mirroring)	1	No	No
Performance	RAID-1 (Mirroring)	1	Yes	No
Balanced	RAID-5/6 (Erasure Coding)	1	Yes	No
Density	RAID-5/6 (Erasure Coding)	1	Yes	Yes

Table 5. Storage Policies

For each configuration, five different workload profiles were run, all generating 4K random read/write mixtures. Since read and write performance differ drastically, a sweep was run across different read%/write% mixtures of 0/100, 30/70, 50/50, 70/30, and 100/0. This allows inferring approximate performance based on the deployment's specific read/write mixture goals.

Furthermore, two working set sizes were used to show both the difference in performance when the working set fits 100% in the cache tier and when the working set is too large to fit fully in cache. In this document, we describe the tests where the working set fits in the cache tier as a cache test and where the working set is spread across both cache and capacity tiers as a capacity test.

Tip: Data Set Size

When the data set fits entirely in the cache tier, minimal de-staging to the capacity tier is observed. Testing two data sets — one that fits in the cache tier and one that is much larger — shows how the capacity tier can affect overall performance.

To ensure that all storage is properly utilized, it is important to distribute worker threads evenly amongst all nodes and all drives. To do this, each test created four VMs on each node. Each VM had eight 100GB VMDKs. The capacity test utilized 100% of the total capacity of the VMDK while the cache test utilized only 10% of the VMDK.

Upon deployment, each configuration was initialized (or preconditioned) with HCIBench using a 128K sequential write test that ran sufficiently long to ensure that the entire data set was written over twice. This ensures that the VMDKs have readable data, instead of simply all zeros. This is particularly important to ensure that the checksum is always calculated on non-zero data. A checksum is meaningless when the data is all zeros. Additionally, OSR was set to 100% for all tests, except for the density profile. Stripe width remained at the default value of 1 as per the vSAN policy described earlier. This ensured that data was spread physically across the entire usable space of each disk.

When benchmarking storage utilities, it is important to ensure consistent and repeatable data. This means ensuring that every test is run the same way, under the same conditions. Many things should be considered to ensure repeatable results. Each test must start in the same state, which is why the "clear read/write cache before each testing" option was selected in HCIBench. Before starting to measure performance in each test, steady state performance was reached. Steady state is found by running a test, monitoring performance and determining when variance is minimal. For all tests conducted in this paper, the time to reach steady state was approximately two hours (called ramp-up time or duration). After ramp-up, performance data was captured over a long enough period to ensure a good average was collected while not collecting too long since many runs need to be conducted. For this testing, the data capture period was one hour.

Table 6 shows the HCIBench parameters used for all cache and capacity tests. We also selected four threads per VMDK based on experimentation (four threads seemed to be optimal, where IOPS was near its highest value while keeping latency at an acceptable level).

HCIBench Test Parameters	Cache	Capacity
Threads Per VMDK:	4	
Test Duration:	1 hour	
Rampup Duration:	2 hours	
% Read:	0/30/50/70/100%	
% Random:	100%	
Working Set Size:	10%	100%
Disk Initialization:	128K sequential	
Clear Cache Before Testing:	Yes	

Table 6: HCIBench Test Parameters

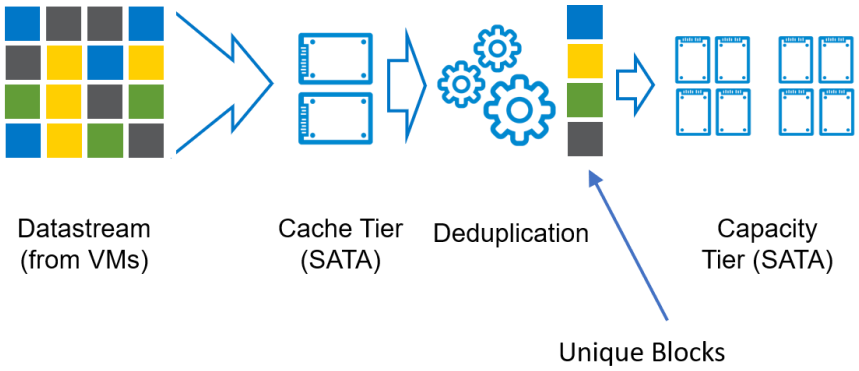


Figure 3: Deduplication

Storage Policies

Depending on the storage policy chosen, vSAN duplicates blocks of data over multiple hosts differently. For RAID-1 (Mirroring), vSAN writes two copies of data to two different hosts and a third block to another separate host as a witness to break quorum in the case of a split cluster. The traffic of the witness object is negligible, so roughly 2:1 writes at the vSAN level were recorded as compared to what the VMs think they are writing.

When moving to RAID-5/6 (Erasure Coding) with FTT of 1, writes occur in a 3 + 1 format, meaning a single block of data is split into three chunks, each written to different hosts while the fourth host gets a parity value computed from the original block. The parity can help recreate a missing block of data in the case of a node failure. This means that vSAN will write four smaller blocks of data for every one block (striped across three smaller blocks) the VMs attempt to write. This is important to consider when studying performance differences between different storage policies. RAID-5/6 writes less data to the physical devices, but because the CPU must work harder to perform the parity calculations, its performance is typically lower.

Deduplication and Compression Testing

vSAN performs deduplication and compression in what is called near-line while de-staging from cache to capacity. During de-staging, each 4K block is hashed. If that hash matches another block's hash in the capacity tier, it skips the write entirely and writes a pointer to the previously written block. If the block's hash does not match, it tries to compress the block. If the block can be compressed to less than 2K, it will be written as a compressed block. If the block cannot be compressed to less than 2K. If not, it is written as the original uncompressed raw 4K block.

If a data set is incompressible or minimally compressible, enabling deduplication and compression is not likely to offer a significant capacity benefit and may potentially reduce performance. Figure 3 shows vSAN's deduplication.

Testing deduplication and compression is slightly different from testing other profiles. Deduplication and compression offers no benefit if the data is not compressible. For this reason, the data set must be compressible instead of purely random.

HCIBench utilizes Vdbench as its load-generating tool, which supports options for duplicable and compressible data sets. While HCIBench itself does not give options to configure deduplication and compression options, it is easy to directly modify the Vdbench parameter files to do so. Appendix A details the modifications to the parameter files used in this reference architecture. The settings used resulted in approximately a 3.23X deduplication and compression ratio.

To ensure relevant results, OSR was set to 0% for the density profile. Otherwise, the deduplication and compression factor is unmeasurable by vSAN because it reserves 100% of the raw capacity, regardless of how much gets utilized.

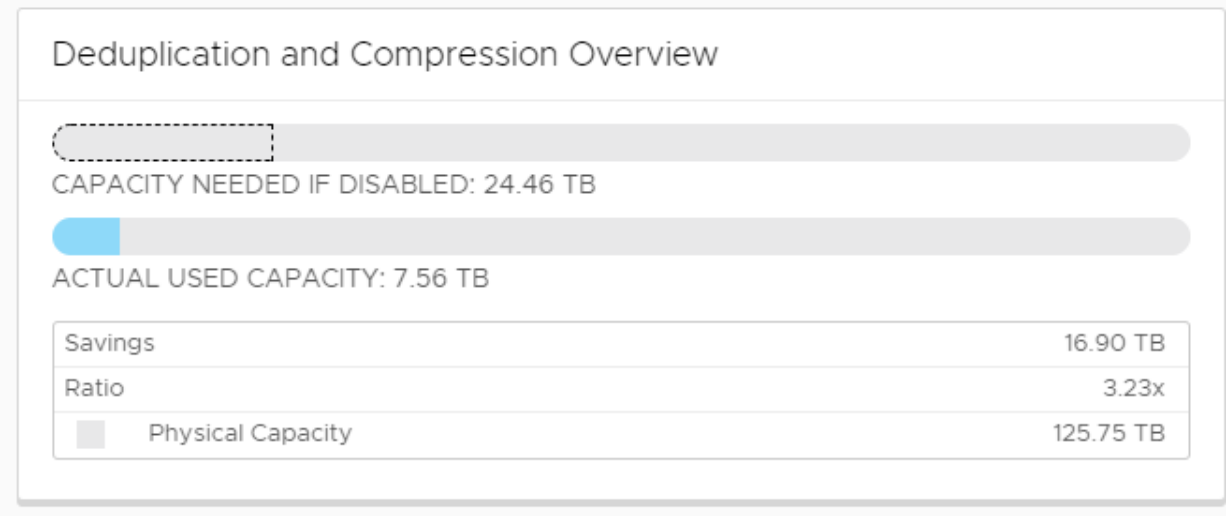


Figure 4. Deduplication and Compression Ratio

Baseline Testing

To get a set of baseline performance data, a run was executed with a storage policy consisting of RAID-1, checksum disabled, and FTT of 1. This removed the overhead from CPU-intensive policies, such as RAID-5/6, checksum, and deduplication and compression. This is the test by which we gauged each policy's reduction in performance.

This policy is not recommended for most uses because disabling the checksum may result in an undetectable bit error. However, this policy allows us to see just how much performance is lost by enabling the checksum and other additional features.

Except for the density profile, each test was run with an OSR of 100% to ensure writing to the intended total amount of disk space. Furthermore, all tests started with an initialization of random data by running a 128K sequential write test.

Test Results and Analysis

Test Configurations

Each FTM has tradeoffs. The performance configuration offers better performance but requires twice the capacity that the data set occupies. The density configuration improves upon this, requiring an additional 33% more space than the data set occupies but at a (potential) performance penalty.

Table 7 shows the additional raw storage needed for each option. Note that when enabling deduplication and compression, capacity can be further extended but it is highly dependent on data compressibility. The table below shows the capacity multiplier for each FTM and FTT.

FTM	FTT	Raid Level	Data Copies	Capacity Multiplier
RAID-1 (Mirroring)	1	RAID-1	2	2
RAID-1 (Mirroring)	2	RAID-1	3	3
RAID-5/6 (Erasure Coding)	1	RAID-5	3 + 1p	1.33
RAID-5/6 (Erasure Coding)	2	RAID-6	4 + 2p	1.5

Table 7: Additional Storage (by Option)

Performance Results: Baseline

To get a comparison point, we start with a baseline run. The following graphs show the average IOPS and latency this configuration can deliver with the baseline storage profile across each read/write mix.

Note that all test graphs show IOPS on the primary axis (left) and latency on the secondary axis (right), where the bars show IOPS and the lines show latency.

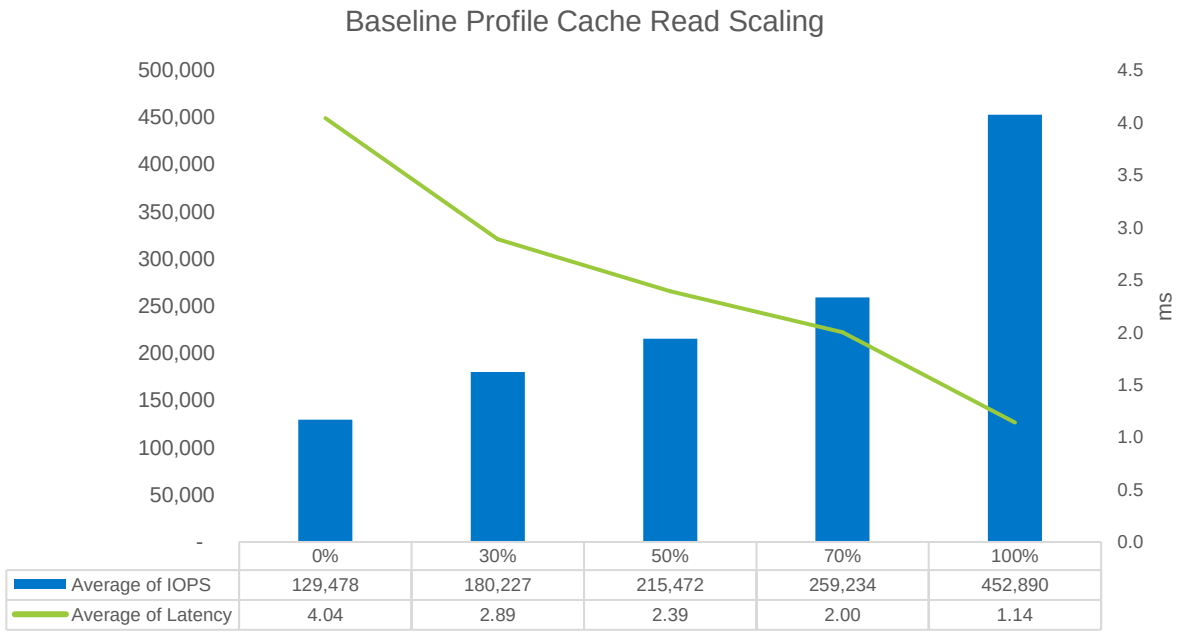
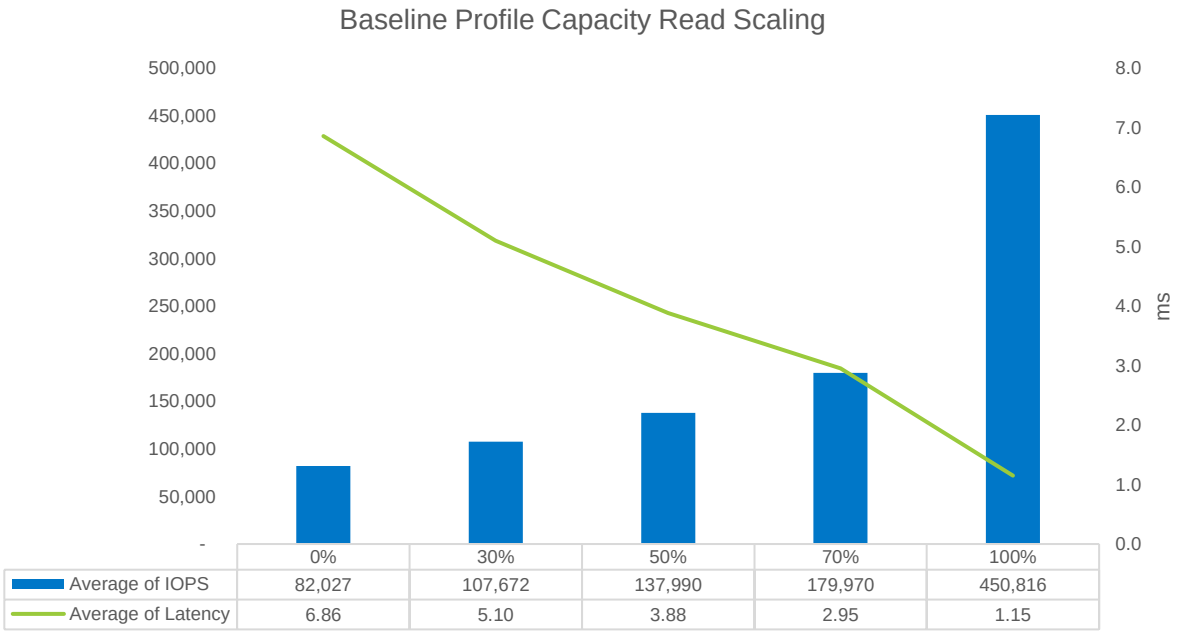


Figure 5. Baseline Cache Test

Figure 5 shows the IOPS and latency for the baseline for each read percentage mixture. Performing a pure write test produces 129K IOPS at an average latency of 4.04ms. As more reads are added, performance begins to increase, resulting in higher IOPS and lower latency. At 100% read, IOPS are over 452K at 1.14ms latency. This means each node can deliver more than 113K IOPS, which is 141% more than what vSAN claims an AF-8 configuration should consistently be able to serve (at 80K IOPS). See the vSAN Hardware Quick Reference Guide for additional details.



	0%	30%	50%	70%	100%
Average of IOPS	82,027	107,672	137,990	179,970	450,816
Average of Latency	6.86	5.10	3.88	2.95	1.15

Figure 6. Baseline Capacity Test

Figure 6 shows the IOPS and latency for the baseline capacity test. The same trend is followed as with the cache test, but with slightly lower performance, especially for the write workloads. As the reads approach 100% of the workload, the difference in performance becomes smaller because all de-staged reads come from the capacity tier in an all-flash configuration.

At 100% reads, the capacity test shows much less of a performance difference from the cache test. Read caching (in memory) is a large contributor to this observation because vSAN dedicates a small amount of memory in each host for caching some data. The smaller the working set size used, the more apparent this feature will be, resulting in higher read performance.

Performance Results – Cache Test

The first comparison is with a working set size that fits 100% in cache (cache test). This test eliminates most de-staging actions and increases performance for the mixed tests because the cache tier performs much better than the capacity tier.

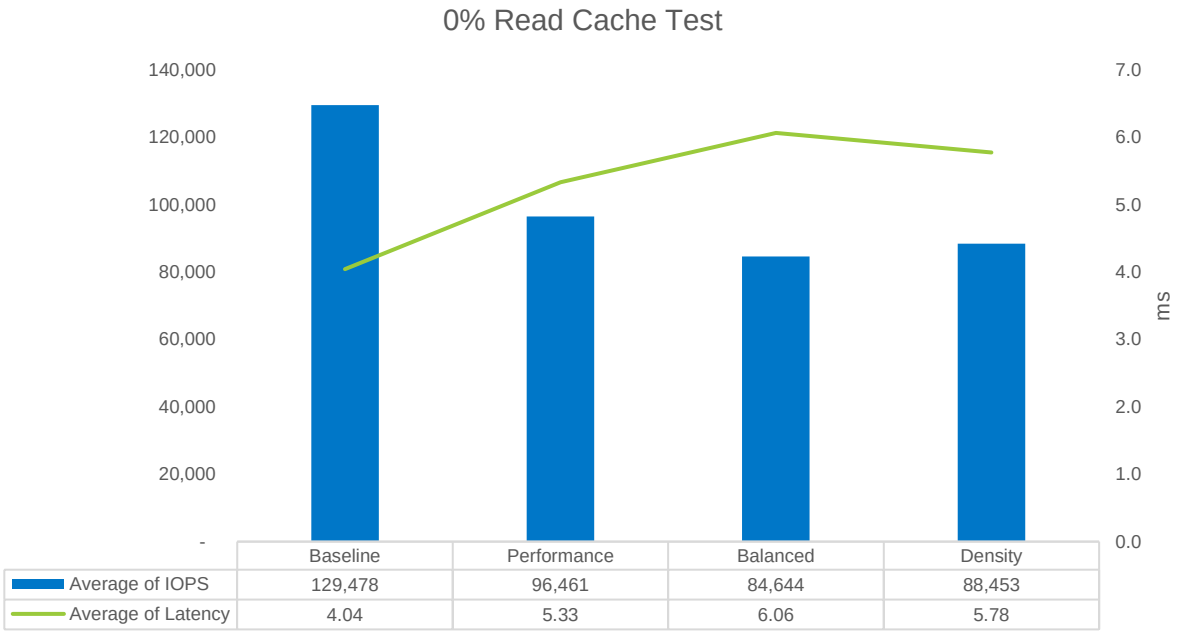


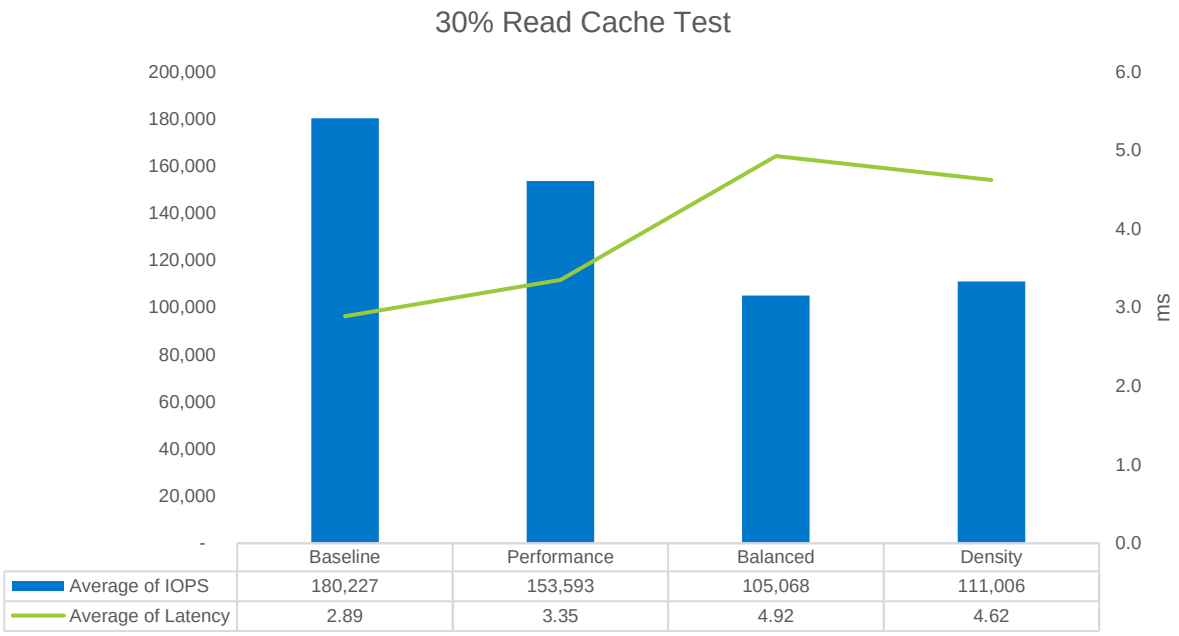
Figure 7. 0% Read Cache Test

Figure 7 shows how the performance changes for a pure write test on each storage profile. As expected, enabling checksum adds some overhead during write operations because computing the checksum requires additional CPU cycles for each write operation.

For this test, enabling checksum reduced IOPS by roughly 25% from the baseline and added approximately 32% latency.

The balanced profile, which utilizes RAID-5/6, shows a small performance reduction from the performance profile at 12% lower IOPS and 14% higher latency. Write performance is typically expected to suffer with enabling RAID-5/6 because parity calculations are performed.

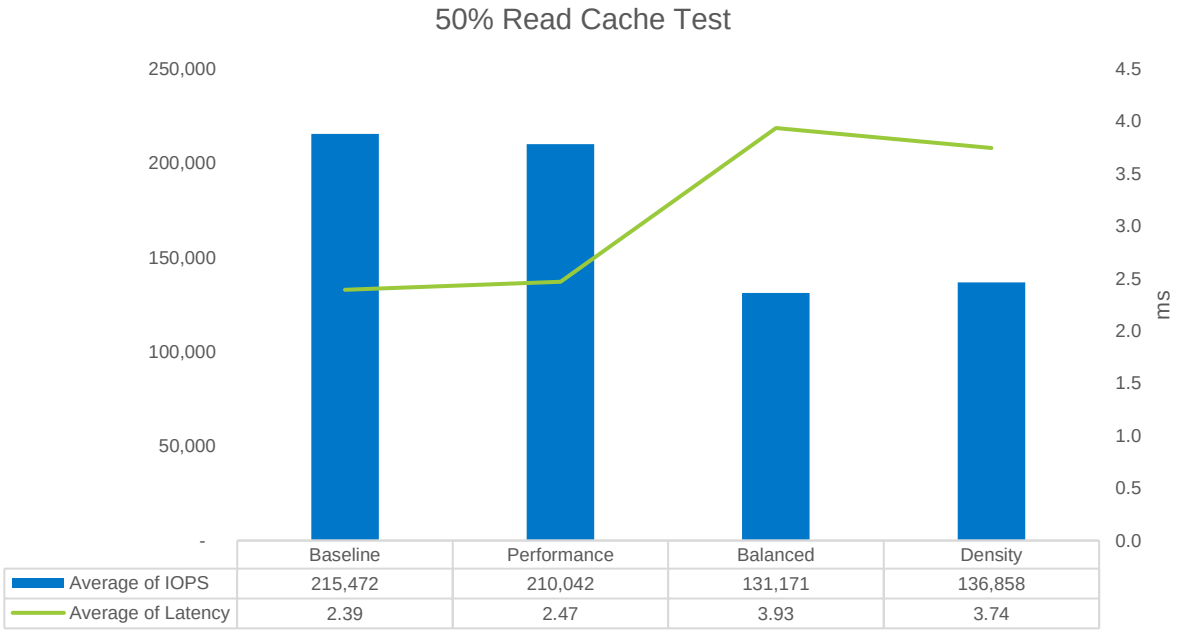
The density profile produces about 5% higher IOPS and 5% lower latency than the balanced profile. Typically, enabling deduplication and compression adds some CPU overhead. However, in this case, the CPU requirement had a small effect on the overall latency, with the disk write latency being a larger contributor to the overall latency. Because the deduplication and compression ratio was significant, very little de-staging occurred, and thus more writes can be absorbed by the cache tier.



	Baseline	Performance	Balanced	Density
Average of IOPS	180,227	153,593	105,068	111,006
Average of Latency	2.89	3.35	4.92	4.62

Figure 8. 30% Read Cache Test

Adding 30% writes showed an increase in performance on all profiles as expected. The performance profile showed a 15% reduction in IOPS and a 16% increase in latency. Switching to RAID-5/6 in the balanced profile showed another large decrease, with 32% lower IOPS and 47% higher latency. Lastly, enabling deduplication and compression showed a small improvement in performance at 6% higher IOPS and 6% lower latency.

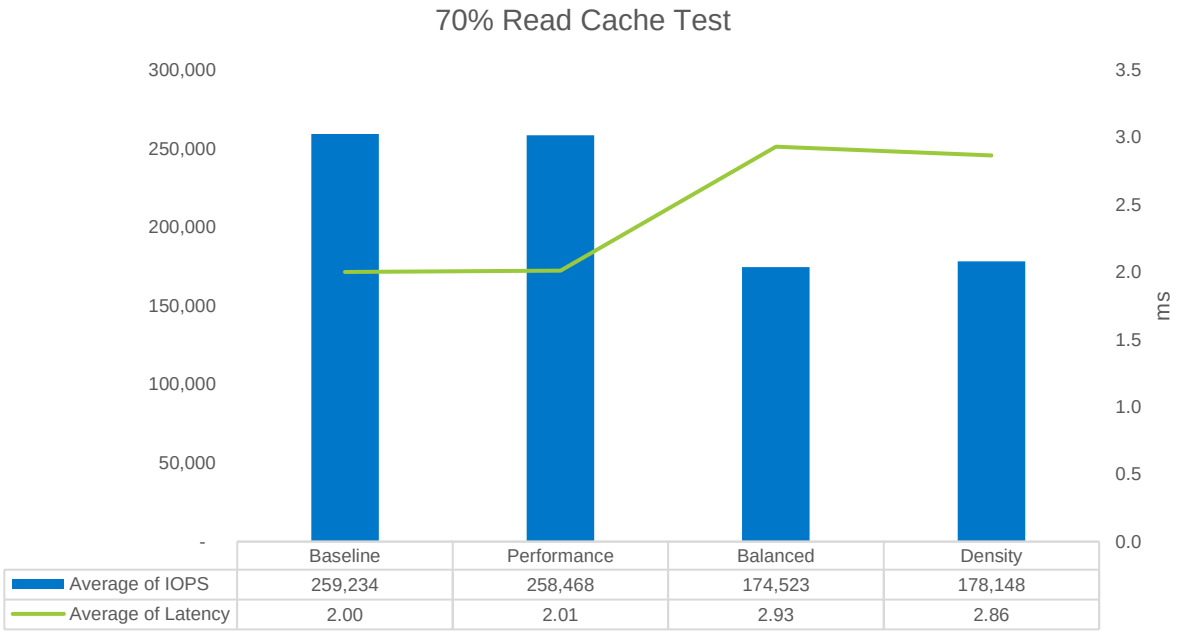


	Baseline	Performance	Balanced	Density
Average of IOPS	215,472	210,042	131,171	136,858
Average of Latency	2.39	2.47	3.93	3.74

Figure 9. 50% Read Cache Test

At 50% writes, performance was again higher for all profiles, and the difference between each profile became more apparent. The performance profile showed 3% less IOPS and 3% higher latency than the

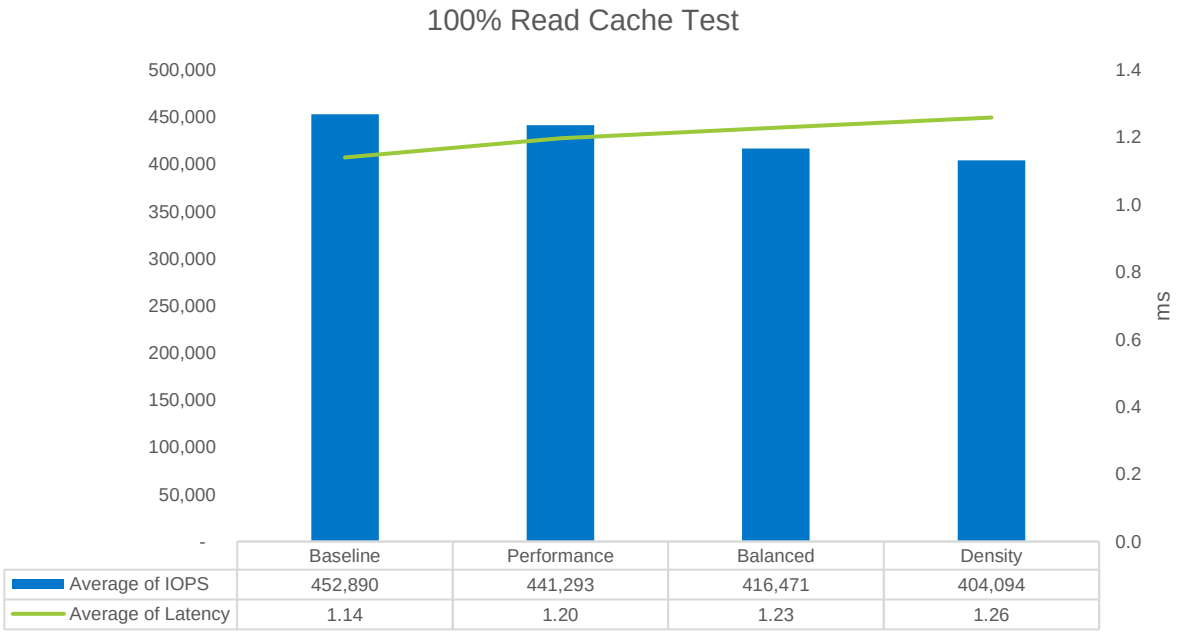
baseline. The balanced profile showed a large performance reduction, with 38% lower IOPS and 59% increase in latency. The density profile gained some performance back with 4% higher IOPS and 5% lower latency. At that point, RAID-5/6 had a big effect on performance, but deduplication and compression did not.



	Baseline	Performance	Balanced	Density
Average of IOPS	259,234	258,468	174,523	178,148
Average of Latency	2.00	2.01	2.93	2.86

Figure 10. 70% Read Cache Test

At 70% reads, we continued to see IOPS increase and latency decrease. The performance profile reduced IOPS less than 1% from the baseline and increased latency less than 1% as well. The balanced profile reduced IOPS by 32% and increased latency by 46%. Enabling deduplication and compression again showed a small performance increase, increasing IOPS 2% and lowering latency by 2%.



	Baseline	Performance	Balanced	Density
Average of IOPS	452,890	441,293	416,471	404,094
Average of Latency	1.14	1.20	1.23	1.26

Figure 11. 100% Read Cache Test

At 100% reads, the highest performance for each profile was experienced. The baseline showed up to 452K IOPS at 1.14ms latency, which is almost 2 GB/s throughput. Enabling checksum for the performance profile only reduced IOPS by 3% and increased latency by 5%, inferring that checksumming had a small effect on reads. Changing to RAID-5/6 from mirroring (balanced) further reduced IOPS by 6% and increased latency by 3%. Enabling deduplication and compression (density) also had minimal impact, reducing IOPS by 3% and increasing latency by 2%.

This first test shows that if you have a relatively small working set size (fitting entirely or mostly in the cache tier), there is very little downside to utilizing deduplication and compression if you are utilizing RAID-5/6, especially if the workload is mostly writes. If increased usable capacity is the primary goal, the density profile can potentially provide much more usable capacity than the raw capacity, thanks mostly to deduplication and compression. It is important to note that not all workloads are the same. Some are more compressible, and some are less compressible. If the workload is highly compressible, using deduplication and compression is strongly recommended. If the data set is minimally compressible, using deduplication and compression may not be ideal because it may result in a small performance penalty with little or no capacity benefit.

Performance Results – Capacity Test

The second comparison looks at performance differences when the working set does not all fit into the cache tier. In this study, the total working set size per node was approximately 3.2TB, with each node able to use 600GB per disk group for cache. Therefore, only about 56% (maximum) of user data can reside in the cache tier while the rest must be held in the capacity tier. Consequently, considerable destaging operations is required for write-intensive workloads, which reduces performance.

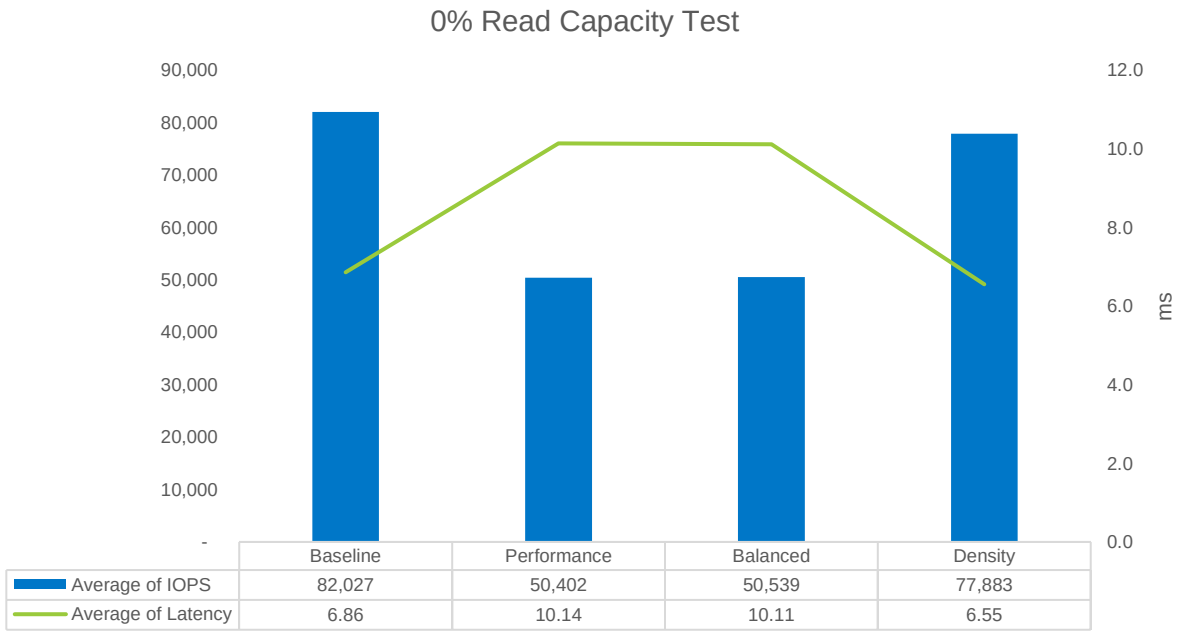


Figure 12. 0% Read Capacity Test

The capacity test showed lower performance across the board. This makes sense with the significant amount of destaging that occurred as along with the smaller percent of the data set that was being cached in memory.

Enabling checksumming drastically reduced performance, with a 39% reduction in IOPS and 48% increase in latency. The balanced profile showed almost identical IOPS and latency. As with the cache

test, enabling deduplication and compression resulted in higher performance because fewer bits were being destaged, resulting in a 54% increase in IOPS and 35% reduction in latency.

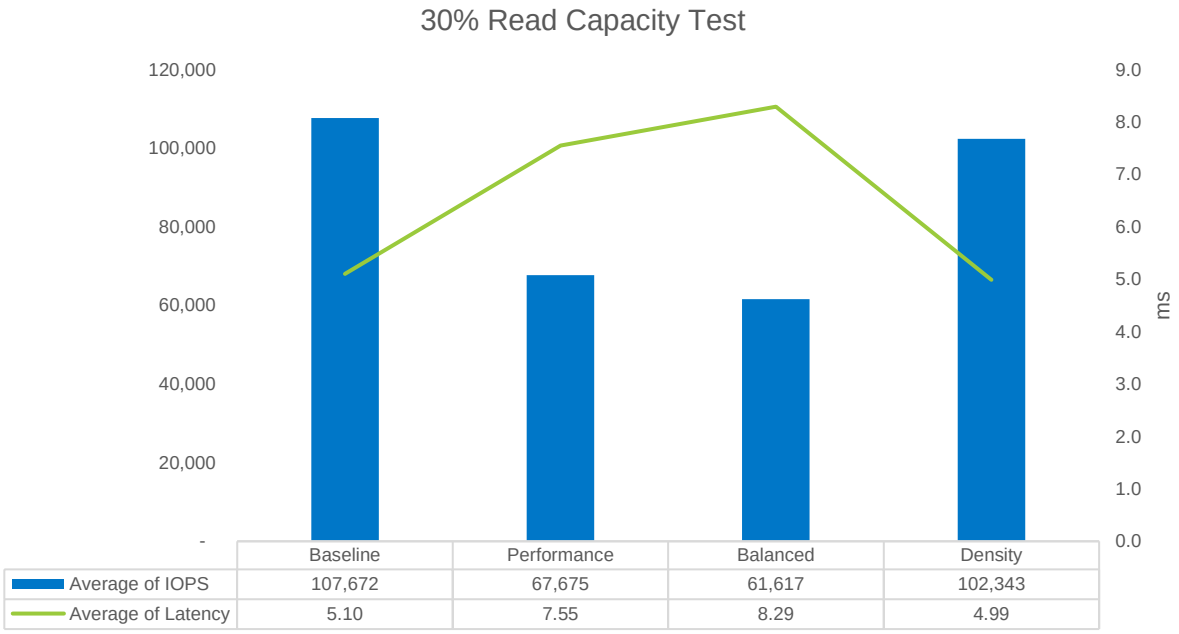


Figure 13. 30% Read Capacity Test

30% reads showed a similar trend to 0%, but with higher performance. Enabling checksum (performance) reduces IOPS by 41% and increases latency by 48%. Switching to RAID-5/6 (balanced) shows a small reduction in performance, reducing IOPS by 9% and increasing latency by 10%. Enabling deduplication and compression again showed a large performance boost, with 66% higher IOPS and 40% lower latency.

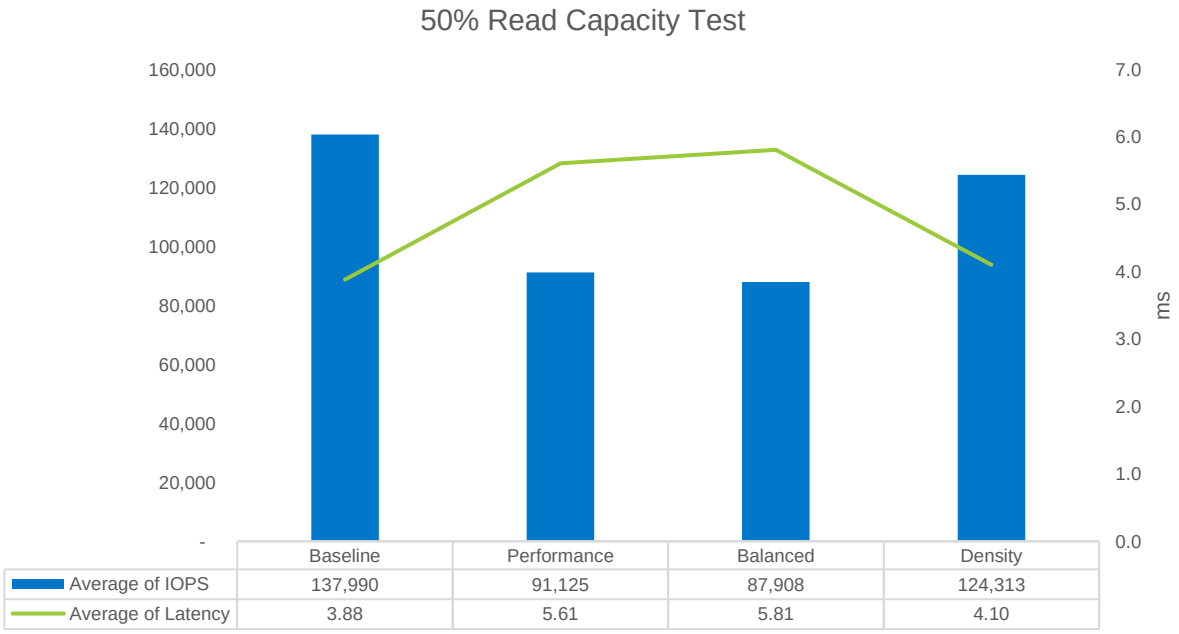
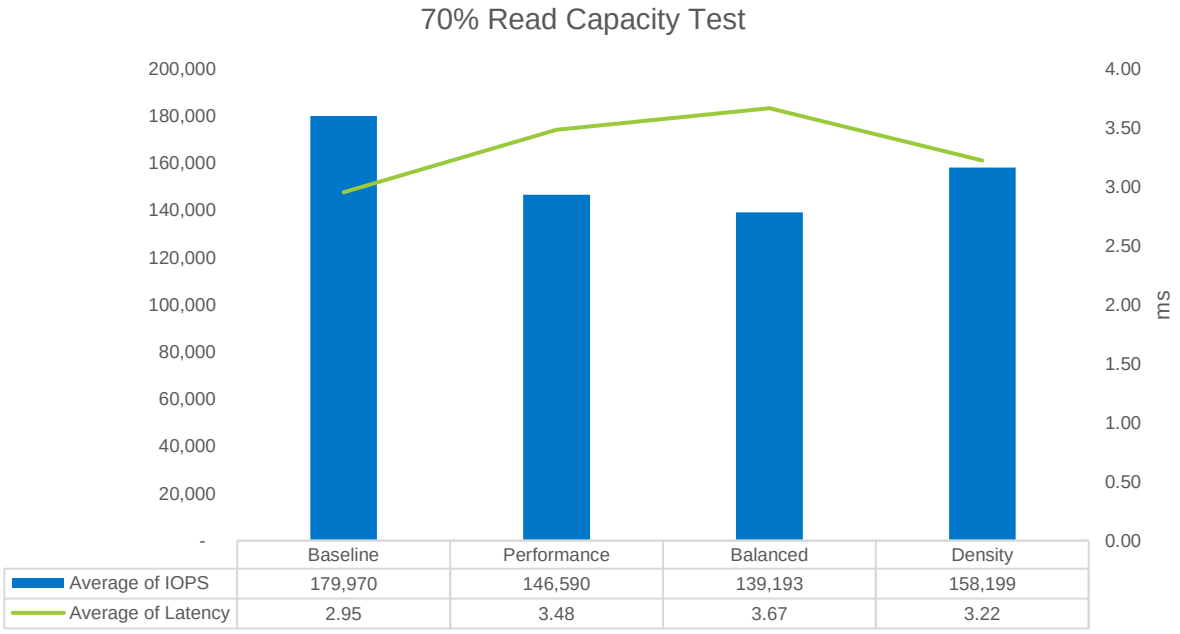


Figure 14. 50% Read Capacity Test

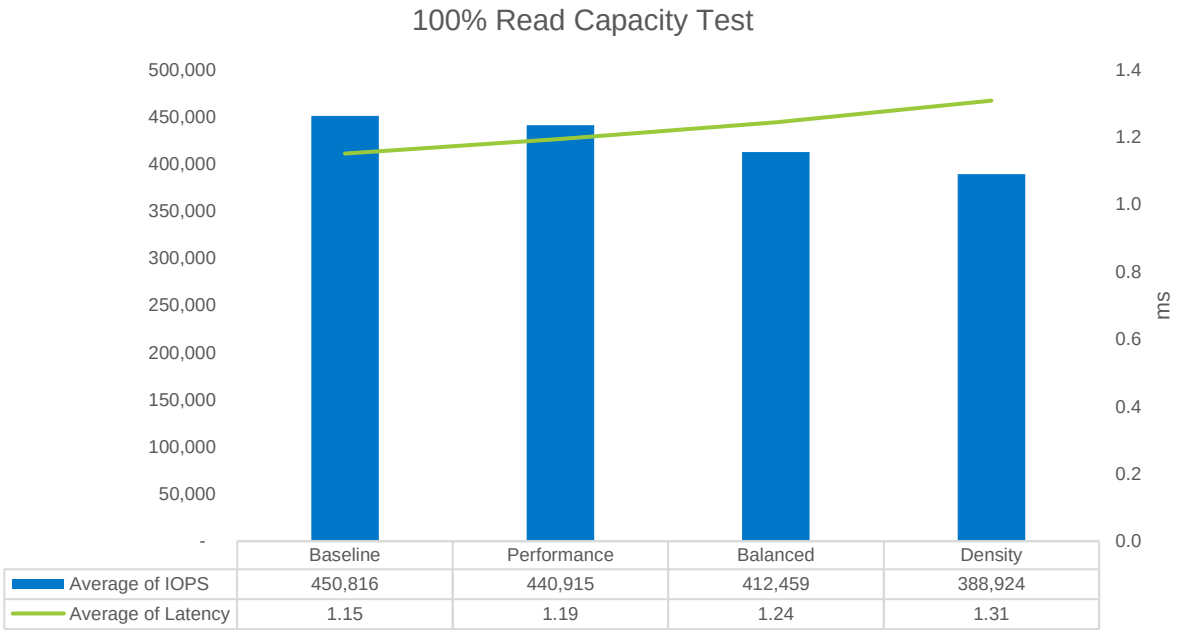
50% reads also followed the same trend, but with even higher performance. The performance profile resulted in a 44% reduction in IOPS and 45% increase in latency. Balanced showed less than 4% difference from performance. Density again increased IOPS by 41% and decreased latency by 31%.



	Baseline	Performance	Balanced	Density
Average of IOPS	179,970	146,590	139,193	158,199
Average of Latency	2.95	3.48	3.67	3.22

Figure 15. 70% Read Capacity Test

At 70% reads, we see the trend continuing. Enabling checksum reduced IOPS by 19% and increased latency by 18%. Switching to RAID-5/6 further reduced IOPS by 5% and increased latency by 5%. Lastly, enabling deduplication and compression increased IOPS 14% and decreased latency 12%.



	Baseline	Performance	Balanced	Density
Average of IOPS	450,816	440,915	412,459	388,924
Average of Latency	1.15	1.19	1.24	1.31

Figure 16. 100% Read Capacity Test

100% reads showed a slightly different trend, which more closely aligns with what is typically expected. Enabling checksum had a small performance penalty, reducing IOPS by 2% and increasing latency by 3%. Switching to RAID-5/6 further reduced IOPS by 6% and increased latency by 4%. Enabling deduplication and compression had a small negative effect on performance, which resulted in a 6% reduction in IOPS and increase in latency.

Summary

The two test cases showed that the performance of this vSAN cluster is strongly dependent on the working set size and read/write ratio. If the working set fits mostly in the cache tier, much higher performance results compared to when only a small portion fits in the cache tier — especially if checksumming will be enabled. Normally, enabling deduplication and compression reduces performance significantly, but with these SATA drives and a large percentage of writes on a working set size that does not fit in the cache tier, performance improved, along with the obvious space savings benefit. If the working set size is large and consists of a large percentage of writes, and if performance is a main goal, enabling deduplication and compression may be beneficial.

Appendix A: vSAN Configuration Details

Tuning Parameters

vSAN's default tunings are set up to be safe for all users. When doing heavy write tests, a disk group can quickly run out of memory and run into memory congestion, causing a decrease in performance. To overcome this, Micron followed VMware's performance documents listed below to alter these three advanced configuration parameters. The documents referenced for the tunings are as follows:

- https://storagehub.vmware.com/#!/vmware-vsan/vsan-6-6-performance-improvements
- https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2150012

Table 8 shows the default value and the value this configuration used.

Tunings		
Parameter	Default	Tuned
/LSOM/blPLOGCacheLines	128K	512K
/LSOM/blPLOGLsnCacheLines	4K	32K
/LSOM/blLLOGCacheLines	128	32K

Table 8: vSAN Tuning

Note that even with these performance tunings implemented, occasionally vSAN experiences various forms of congestion. Congestion appears to occur more often during runs with a high write percentage.

Vdbench Parameter File

Below is a sample Vdbench parameter file for a 0% read test against eight VMDKs with a run time of one hour and a warmup (ramp) time of two hours. This parameter file is used for testing deduplication and compression, using a deduplication ratio of 10 with 4K units, and a compression ratio of 10. This resulted in an initial compression ratio of 3.23x, which, after accounting for using RAID-5/6, puts the total ratio of usable capacity to raw capacity at 2.43. The highlighted section denotes the modifications made to the Vdbench parameter file that was generated by HCIBench.

```
*Auto Generated Vdbench Parameter File
*8 raw disk, 100% random, 0% read
*SD:    Storage Definition
*WD:    Workload Definition
*RD:    Run Definition
debug=86
data_errors=10000
dedupratio=10
dedupunit=4k
compratio=10
sd=sd1,lun=/dev/sda,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd2,lun=/dev/sdb,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd3,lun=/dev/sdc,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd4,lun=/dev/sdd,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd5,lun=/dev/sde,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd6,lun=/dev/sdf,openflags=o_direct,hitarea=0,range=(0,100),threads=4
sd=sd7,lun=/dev/sdg,openflags=o_direct,hitarea=0,range=(0,100),threads=4
```

```
sd=sd8,lun=/dev/sdh,openflags=o_direct,hitarea=0,range=(0,100),threads=4
wd=wd1,sd=(sd1,sd2,sd3,sd4,sd5,sd6,sd7,sd8),xfersize=4k,rdpct=0,seekpct=100
rd=run1,wd=wd1,iorate=max,elapsed=3600,warmup=7200,interval=30
```

Switch Configuration (Sample Subset)

Below is a collection of sample sections of one of the switch configurations. The "…" denotes an irrelevant missing piece between sections of the configuration file.

```
…
##
## Interface Split configuration
##
  interface ethernet 1/49 module-type qsfp-split-4 force
  interface ethernet 1/51 module-type qsfp-split-4 force

##
## Interface Ethernet configuration
##
…
  interface ethernet 1/51/1 switchport mode trunk
  interface ethernet 1/51/2 switchport mode trunk
  interface ethernet 1/51/3 switchport mode trunk
  interface ethernet 1/51/4 switchport mode trunk
…
##
## VLAN configuration
##
  vlan 100-104
  vlan 110-115
  interface ethernet 1/49/1 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/1 switchport trunk allowed-vlan add 100-104
  interface ethernet 1/49/1 switchport trunk allowed-vlan add 110-115
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 100-104
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 100-104
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 110-115
  interface ethernet 1/49/3 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/3 switchport trunk allowed-vlan add 100-104
  interface ethernet 1/49/3 switchport trunk allowed-vlan add 110-115
  interface ethernet 1/49/4 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/4 switchport trunk allowed-vlan add 100-104
  interface ethernet 1/49/4 switchport trunk allowed-vlan add 110-115
```

Appendix B: Monitoring Performance and Measurement Tools

- **HCIBench:** HCIBench is developed by VMware and is a wrapper around many individual tools, such as vSAN Observer, Vdbench and Ruby vSphere® Console (RVC). HCIBench allows you to create VMs and configure them, run Vdbench files against each VM, and run vSAN Observer and aggregate the data at the end of the run into a single results file.
- **vSAN Observer:** vSAN observer is built in to the vCenter Server® Appliance™ (VCSA) and can be enabled via the RVC. HCIBench starts an observer instance with each test and stores it alongside the test results files.
- **Vdbench:** Vdbench is a synthetic benchmarking tool developed by Oracle. It allows you to create workloads for a set of disks on a host and specify parameters such as run time, warmup, read percentage and random percentage.
- **RVC:** RVC is built in to the vSphere Center Appliance as an administration tool. With RVC, you can complete many of the tasks that can be done through the web GUI and more, such as start a vSAN Observer run.
- **vSphere Performance Monitoring:** vSphere now has many performance metrics built right into the VCSA, including front-end and back-end IOPS and latency

Appendix C: Bill of Materials

Component	Qty per Node	Part Number	Description
Server	1	R740xd	Dell R740xd Server
CPU	2	BX806736142	6142 Gold 16 core 2.60GHz
Memory	12	MTA36ASF4G72PZ	Micron 32GB DDR4-2666MHz RDIMM ECC
Boot Drive	1	MTFDDAK480TCB-1AR1ZABYY	Micron 5100 PRO SATA 480GB SSD
Cache SSD	3	MTFDDAK960TDN-1AT16A	Micron 5200 MAX SATA 960GB SSD
Capacity SSD	9	MTFDDAK3T8TDC-1AT1ZABYY	Micron 5200 ECO SATA 3840GB SSD
Networking (NIC)	1	406-BBKX	Broadcom BCM57414 NetXtreme-E 25GbE

Table 9: Bill of Materials