A **Micron** Reference Architecture

Micron® Accelerated All-Flash VMware vSAN™ 6.6 on HPE ProLiant DL380 Gen10 Server Solution

Reference Architecture

systems software storage memory

Contents

Executive Summary

Data-intensive businesses that thrive in today's environment move quickly, and data platforms must move quickly with them. Technologies such as SSDs and advanced DRAM, in conjunction with standard servers, multicore processors and state-of-the-art virtualization like VMWare vSAN™, are chasing application lethargy out of the data center.

This reference architecture (RA) provides deployment and testing details for one of the most compelling configurations: The Micron Accelerated VMware vSAN all-flash, high-performance server solution.

Similar to the standard AF-6 all-flash VMware vSAN ReadyNode™ definition, this design combines high write-performance, high-endurance enterprise SATA SSDs in the cache tier and read-centric, capacity-focused enterprise SATA SSDs in the capacity tier with advanced Micron® DRAM and HPE ProLiant DL380 Gen10 rackmount servers with 25 GbE networking. This design leverages a mix of SATA SSDs with different performance and endurance characteristics to drive overall value at a more approachable price point based on our testing.

Optimized and engineered for VMware vSAN 6.6, this reference design enables:

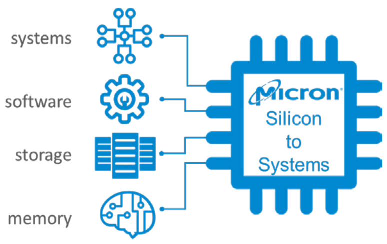
- Faster time to deployment: Lab-tested by Micron experts in vSAN and thoroughly documented so you can deploy more quickly with greater confidence
- Balanced design: The right combination of cache and capacity SSDs, DRAM, processors and networking
- Confident deployment: Micron's in-house vSAN and workload expertise means you can build and deploy the platform with confidence

The configuration in this RA ensures easy integration and operation with vSAN 6.6, offering predictably high performance that is easy to deploy and manage.

The Purpose of This Document

This document describes deploying a performance- and price-balanced all-flash vSAN-enabled VMware vSphere® cluster using a combination of Micron® enterprise SATA SSDs and HPE DL380 standard servers in a Micron reference design. It details the hardware and software building blocks and measurement techniques used to characterize performance, as well as overall design including the vSphere configuration, network switch configurations, vSAN tuning parameters, and configuration of the Micron reference nodes and Micron SSDs.

The purpose of this document is to provide a pragmatic blueprint for administrators, solution architects and IT planners who need to build and tailor a high-performance storage infrastructure that scales for I/O-intensive workloads.



Micron Reference Architectures

Micron reference architectures (RAs) are optimized, pre-engineered enterprise solution templates for platforms that are developed between Micron and industry leading enterprise providers like HPE.

Designed and tested at Micron's Storage Solution Center, these proven templates help build next-generation solutions with reduced time investment and risk.

Solution Overview

A vSAN storage cluster is built from a number of vSAN-enabled vSphere nodes for scalability, fault-tolerance, and performance. Each node is based on commodity servers and components 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 datastore. When virtual machines (VMs) are created, virtual disks (VMDKs) can be carved out from the vSAN datastore. Upon creation of a VMDK, the host does not need to handle any kind of fault tolerance logic, as it is all handled by the vSAN storage policy applied to that object and vSAN's underlying algorithms. When a host writes to its VMDK, vSAN handles all necessary operations such as data duplication, erasure coding, and checksum and placement based on the selected storage policy.

Storage policies can be applied to the entire datastore, individual VMs, or each VMDK. Using storage policies allows a vSAN administrator to determine whether to add more performance, capacity, or availability to an object. Numerous storage policies can be used on the same datastore, enabling creation of high-performance VMDKs 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 down to the vSAN datastore.

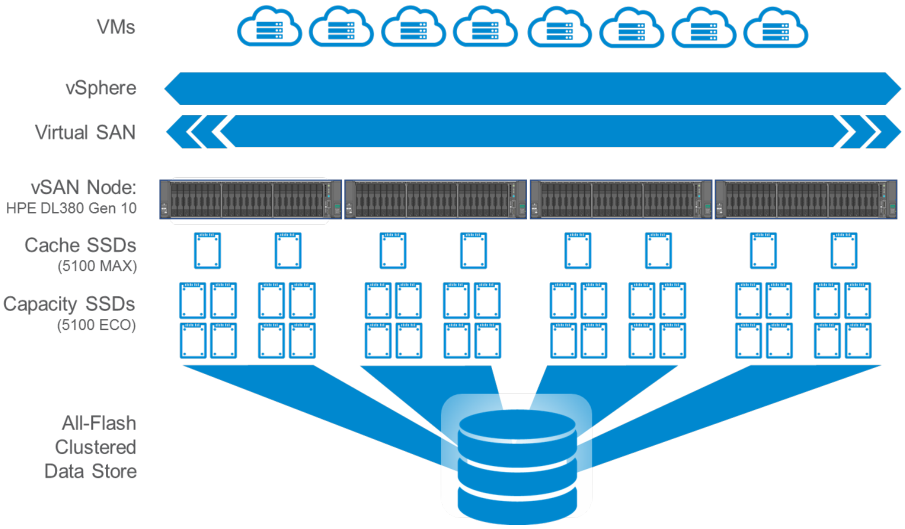


Figure 1: vSAN Architecture

Why Micron for this Solution

Storage (SSDs and DRAM) can represent up to 70% 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 RAs, ensuring these core elements are engineered to perform in highly demanding applications like vSAN and are holistically balanced at the platform level. This RA leverages decades of technical expertise as well as direct, engineer-to-engineer collaboration between Micron and leading enterprise platform and software providers.

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 or VM. Below are some of the options that make up a storage policy.

- **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 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.
- **Failure tolerance method (FTM)**: The method of fault tolerance: 1) RAID-1 (Mirroring) or 2) RAID-5/6 (Erasure coding). Choosing RAID-1 (Mirroring) created 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 1 or 2 parity blocks, for RAID-5 and RAID-6 respectively. Selecting FTT=1 means the object will behave similar to RAID-5, whereas FTT=2 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**, the checksum operation is not performed. This reduces data integrity but can increase performance (in the case 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, and 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 they reside on different disks on the host, if possible.

Design Overview

This section describes the configuration of each component shown below and how they are connected.

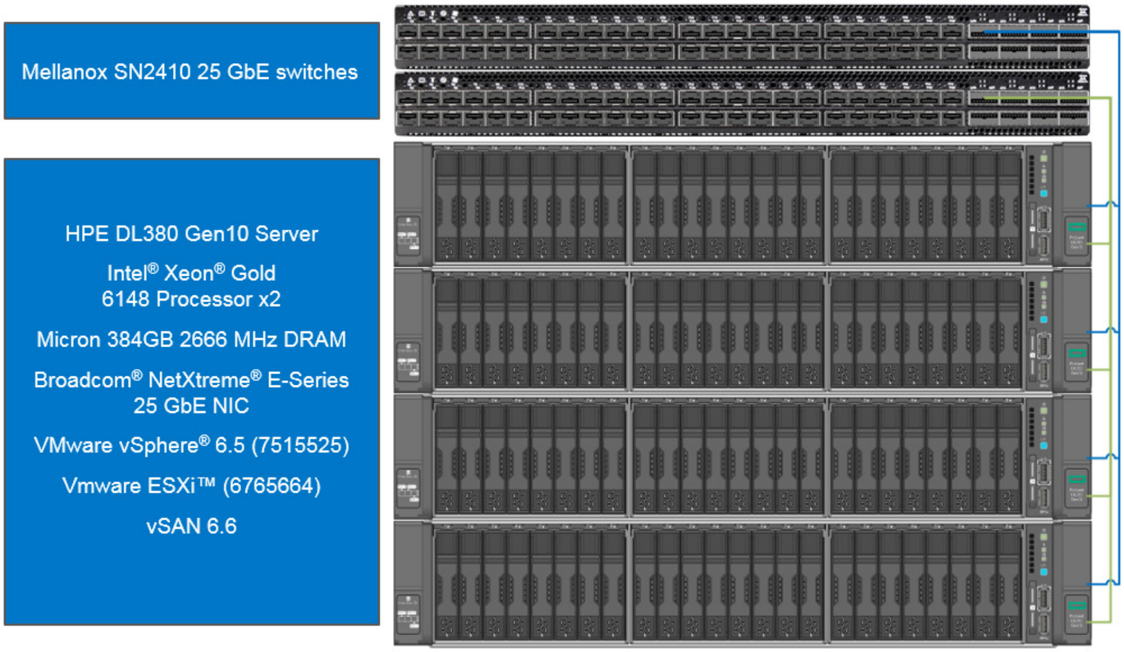

Mellanox SN2410 25 GbE switches
HPE DL380 Gen10 Server
Intel® Xeon® Gold 6148 Processor x2
Micron 384GB 2666 MHz DRAM
Broadcom® NetXtreme® E-Series 25 GbE NIC
VMware vSphere® 6.5 (7515525)
Vmware ESXi™ (6765664)
vSAN 6.6

Figure 2: vSAN RA Hardware

Software

VMware's vSAN is a hyper-converged infrastructure (HCI) solution, combining VMWare ESXi virtualization with multihost software-defined storage. vSAN is a technology that is part of VMware's vSphere suite of advanced services.

We chose vSAN 6.6 for this solution because of the substantial benefits it brings to a broad range of customers, applications and workloads, unlocking the potential of SSDs through:

- **Broad-use and flash-optimized IOPS**: According to VMware's website, vSAN is widely used for business-critical applications, remote and branch offices, virtual desktop infrastructure and disaster recovery. Compared with prior versions, vSAN 6.6 optimizations deliver up to 50% more IOPS than previously possible, deployed for over 50% less than the cost of competing hybrid hyper-converged solutions.
- **Deduplication and compression**: Software-based deduplication and compression optimizes all-flash storage capacity, providing as much as 7X data reduction[1].
- **Data protection (erasure coding)**: Increases usable storage capacity by up to 100% while keeping data resiliency unchanged.

According to VMware documentation, a vSAN cluster is created by installing ESXi on at least three nodes (four or more is recommended) and enabling the vSAN service via a license in the vSphere Client.

vSAN uses a two-tier storage architecture, where all write operations are sent to a cache tier and are subsequently de-staged to a 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:

1) **Hybrid**: SSDs in the caching tier and rotating media in the capacity tier
2) **All-Flash**: SSDs in both cache and capacity tiers

In a hybrid configuration, the cache tier is used as both a read and write cache, keeping hot data in the cache to improve 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.

Micron Components

This RA employs Micron's 5100 MAX and 5100 ECO enterprise SATA SSDs (MAX in the cache tier and ECO in the capacity tier). This solution also utilizes Micron DRAM (the details of which are not discussed further in this document).

SSD	Capacity	Design	Use	Random Read (IOPS)	Random Write (IOPS)	Read Throughput	Endurance (TBW)
5100 MAX	960GB	Write-intensive	Cache Tier	93,000	74,000	540 MB/s	8.8 PB
5100 ECO	1920GB	Read-intensive	Capacity Tier	93,000	24,000	540 MB/s	6.4 PB

Table 1: Micron SSDs

See www.micron.com for additional details, specifications and datasheets for these and other Micron SSDs.

Server Platforms

This RA utilizes HPE Proliant DL380 Gen10 servers. These Intel-based 2U dual-socket servers are configured with two Intel® Xeon® Gold 6148 processors, each with 20 cores at 2.40 GHz. These processors align with VMware's AF-6 minimum requirements (their nomenclature for a medium-sized all-flash configuration).

[1] Assumes deployment enables 7X data reduction; actual data reduction is dependent on several external factors.

Solution Network

Switches

vSAN utilizes commodity Ethernet networking hardware. This RA uses two Mellanox SN2410 switches for all cluster-related traffic. Both switches are interconnected with a single QSFP+ cable. Spanning Tree 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.

vSAN requires at least three separate logical networks, which are all segregated using different VLANs and subnets on the same switches. The three networks, and their respective VLANs, are as follows:

Role	VLAN ID	Network Subnet	Description
Management/ VM Network	100	172.16.17.x/16	This network hosts all client-server traffic between the VMs and the data center, as well as all VMware/vSAN management traffic for monitoring and management.
vMotion	101	192.168.1.x/24	This network supports movement of VMs from one node to another within the cluster.
vSAN	102	192.168.2.x/24	This network supports transfer of all storage data associated with the vSAN shared storage pool between the various nodes of the cluster.

Table 2: Network Configuration

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 the wrong 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.

Network Interface Cards

Each server has a single dual-port Broadcom BCM57414 NetExtreme-E 25 GbE NIC, with one port of the NIC connected to one of each of the switches to ensure high availability in the case of losing one of the two switches. vSAN is active on one link and standby on the other, whereas management and vMotion are active on the opposite link. This ensures that vSAN gets full utilization of one of the links and is not interrupted by any external traffic.

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.

Solution Design—Hardware

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

Hardware Components

Node Components
HPE ProLiant DL380 Gen10 2-socket rack mount server
2X Intel Xeon Gold 6148 20-core 2.40GHz CPUs
Micron 384GB 2666 MHz DRAM (32GB x 12)
8X Micron 5100 ECO SATA SSDs, 1920GB
2X Micron 5100 MAX SATA SSDs, 960GB

Node Components
1X Broadcom NetExtreme-E 25GbE SFP+ NIC (BCM57414)
1X Micron 5100 PRO (OS Drive)
HPE Smart Array P816i-a SR Gen 10
1x Mellanox ConnectX-4 Dual-port 25GbE SFP+ NIC
(MT 27630)

Table 3: Node Hardware

Network Infrastructure

Network Components
2X Mellanox SN2410 25GbE switches

Network Components
Mellanox QSFP+ copper breakout cables

Table 4: Network Hardware

Solution Design—Software

Software Components
vCenter Server Appliance 6.5.0.14000 build 7515524
ESXi build 6765664
vSAN 6.6
Disk format version 6

Network Components
HBA driver 1.0.0.1060-1OEM.650.0.0.4598673 – HBA firmware 1.04 – 5100 firmware HPG6

Table 5: 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 lowest cost hardware option may not meet your performance requirements.

This RA targets a configuration based on VMware's AF-6 specifications, which aims to provide up to 50K IOPS per node. An AF-6 configuration typically calls for at least 8TB of raw storage capacity per node, dual processors with at least 12 cores per processor, 256GB of memory, two disk groups per node with 8 capacity drives, and 10 GbE networking at a minimum. For this configuration, we utilized two disk groups per node—with one cache drive per disk group and four capacity drives per disk group, resulting in two cache drives and eight capacity drives per node.

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

For further information on AF-6 requirements, see VMware's vSAN Hardware Quick Reference Guide.

Measuring Performance

Test Methodology

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

The benchmark tool used for this RA is HCIBench. HCIBench is primarily a wrapper around Oracle's Vdbench, with extended functionality to deploy and configure VMs, run vSAN Observer, and aggregate data, as well as provide an ergonomic web interface from which to run tests.

HCIBench is deployed as a VM template. In this case, there is a separate vSAN cluster set up for all infrastructure services, such as for HCIBench, DNS, routing, etc. 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 (115), 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 your storage policy. To understand how each of these affect performance, four test configurations were chosen:

Configuration	FT Method	FTT	Checksum	Dedup+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 6: 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 dataset sizes were used to show the difference in performance when the working set fits 100% in the cache tier, and one when it 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 the tests where the working set is spread across both cache and capacity tiers as a **capacity test**.

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 creates four VMs on each node. Each VM has eight VMDKs, each either 6GB or 128GB, depending on whether it is a cache or capacity test.

Upon deployment, each configuration is initialized (or preconditioned) with HCIBench using a 128K sequential write test that is run sufficiently long to ensure that the entire dataset is written over twice. This ensures that the VMDKs have readable data, instead of simply all zeros. This is particularly important when it comes to checksumming to ensure that the checksum is always calculated on non-zero data. A checksum is meaningless when your data is all zeros. Additionally, OSR is set to 100% for all tests—except for the density profile—and stripe width is left at the default value of 1, as per the vSAN policy described in an earlier section. This ensures that data is spread physically across the entire usable space of each disk, instead of potentially lying in only a subset of them, in a thin provisioned manner.

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 we select the **clear read/write cache before testing** option in HCIBench. We also allow each test to get to steady state performance before we start our performance measurements. Steady state is found by running a test, monitoring performance, and seeing when it is a consistent performance level without significant deviation. 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 is captured over a long enough time to ensure that a good average is collected, while not collecting too long, since many runs need to be conducted. For our testing, the data capture period is one hour.

The table below shows the HCIBench parameters used for all cache and capacity tests and summarizes all run options used for testing. We also selected four threads per VMDK[2] based on experimentation, as four threads seemed to be the best balance of high IOPS performances while keeping latency at an acceptable level.

HCIBench Test Parameter	Cache	Capacity
Threads per VMDK	4[2]	
Test Duration	1 hour	
Ramp Duration	2 hours	
%Read	0/30/50/70/100%	

HCIBench Test Parameter	Cache	Capacity
%Random	100	
Working Set Size	100%	
SSD Initialization	128K SEQ WRI	
Clear Cache Before Testing	Yes	

Table 7: HCIBench Test Parameters

[2] Two threads per VMDK for the density profile capacity tests at less than 100% read.

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. This results in roughly 2:1 writes at the vSAN level as compared to what the VMs think they are writing.

When moving to RAID-5/6 (Erasure coding) with FTT of 1, writes happen 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 think they are writing.

This is important to consider when studying performance differences between different storage policies. RAID-5/6 will write 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 does deduplication and compression in what is called near-line, and is performed in one operation while destaging from cache to capacity. During destaging, each 4K block is hashed. If that hash matches another block's hash in the capacity tier, it will simply skip that write entirely, and just write a pointer to the previously written block. If the block's hash does not match, it will try to compress the block. If the block is compressible to less than 2K, it will be written as a compressed block. If not, it will simply be written as the original uncompressed raw 4K block.

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

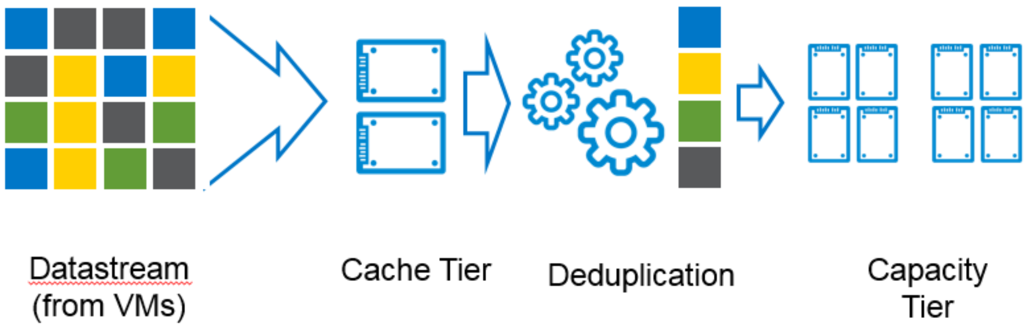
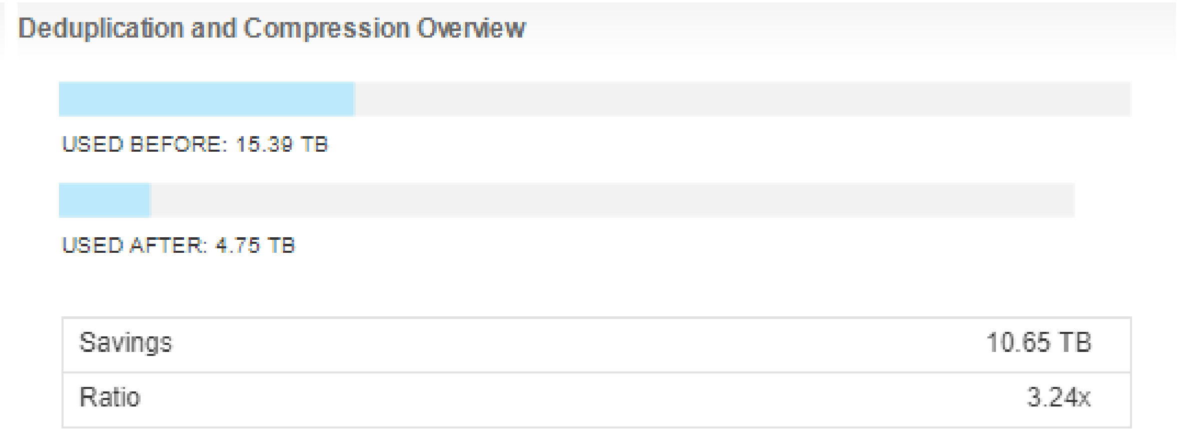


Figure 3: Data Deduplication

Testing with deduplication and compression enabled is slightly different from testing other profiles. Deduplication and compression offers no benefit if your data is not compressible, and defeats its purpose. For this reason, the dataset needs to be guaranteed to be compressible instead of purely random.

HCIBench utilizes Vdbench as its load-generating tool, which supports options for duplicable and compressible datasets. 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 RA. The settings used resulted in an approximately 3.24X data reduction ratio with deduplication and compression enabled for the capacity test (shown below), and 0X for the cache test since deduplication and compression only happens during destaging to the capacity tier. To get meaningful results, OSR was set to 0% for the density profile; otherwise, the deduplication and compression factor is not measurable by vSAN since it will reserve 100% of the raw capacity, regardless of how much of it gets utilized.



Deduplication and Compression Overview

USED BEFORE: 15.39 TB

USED AFTER: 4.75 TB

Savings	10.65 TB
Ratio	3.24x

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 removes the overhead from CPU-intensive policies such as RAID-5/6, checksum, and deduplication and compression. This will be the test by which we gauge each policy’s reduction in performance.

Note that this policy would not be recommended for most customers, since disabling checksum means there is a chance of getting a bit error and not being able to detect it. However, this does allow us to see just how much performance is lost by enabling checksumming and other features.

Each test—except for the density profile—is run with OSR of 100% to ensure that we are writing to the total amount of disk that we intend. Furthermore, all tests start 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 means you need twice the raw capacity of what you need for usable data. The density configuration improves upon this, and only requires an additional 33% more raw space than you need, but at a performance penalty.

The table below shows how much additional raw storage is needed for each option. Also note that when enabling deduplication and compression, capacity can be further extended, but it is highly dependent on how compressible your data is. 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 8: 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.

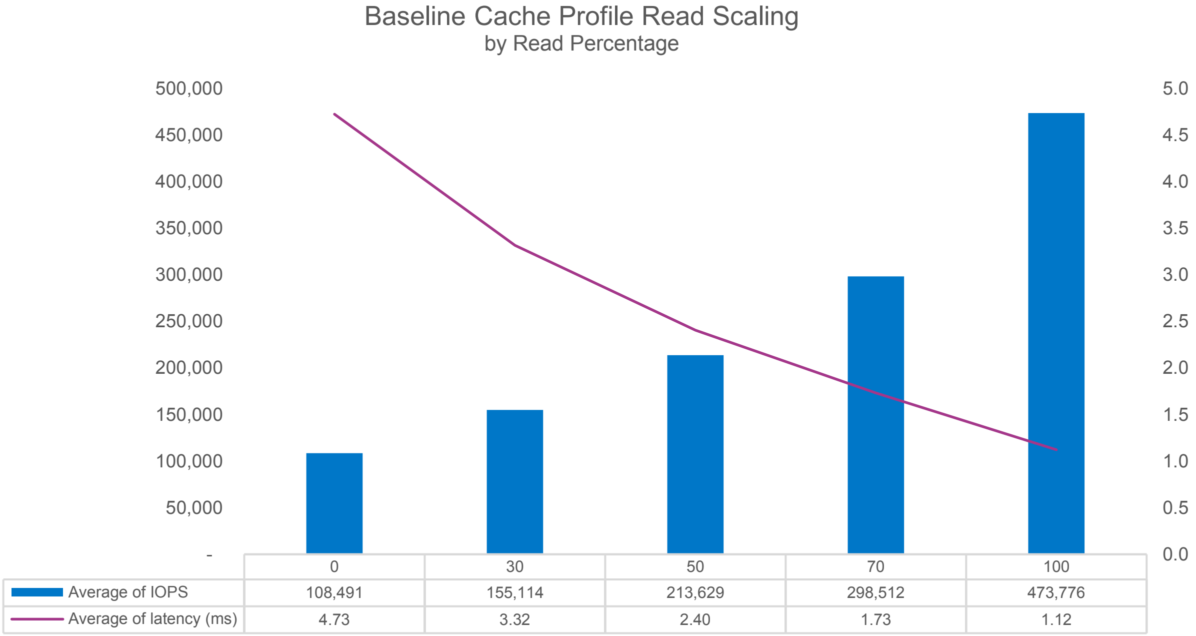
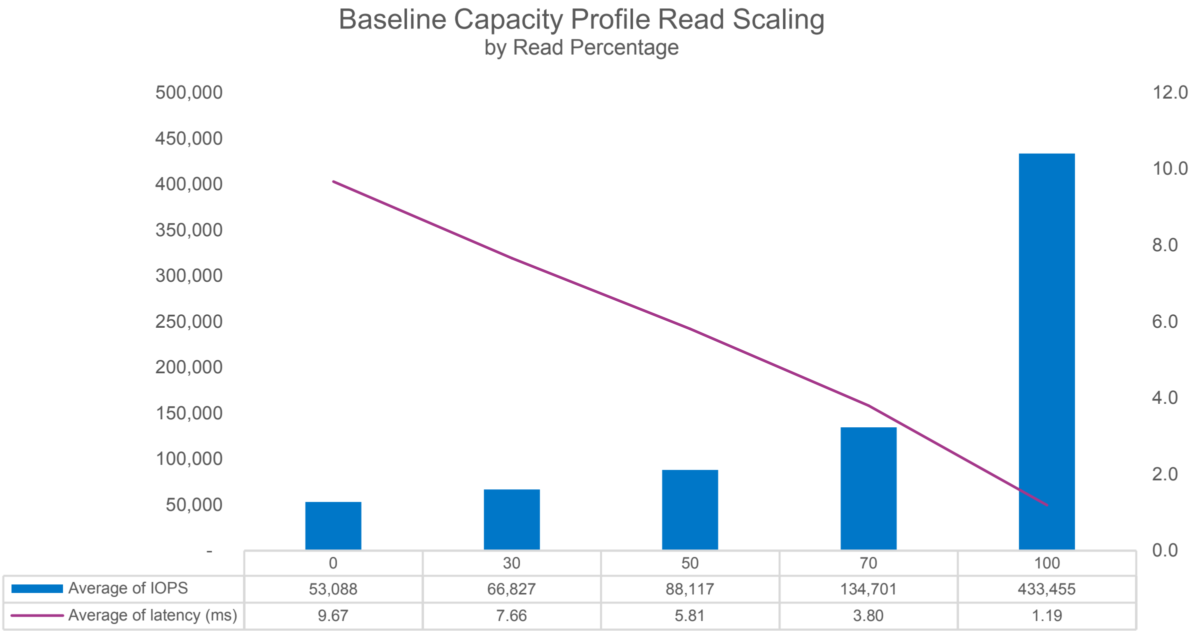


Figure 5: Baseline Cache Test

Figure 5 shows the IOPS and latency for the baseline for each read percentage mixture. Doing a pure write test produces 108K IOPS at an average latency of 4.73ms. As more reads are added into the mix, the performance begins to increase, netting higher IOPS and lower latency. At 100% read, IOPS are up to over 473K at 1.12ms latency. This mean each node can deliver over 118K IOPS, which is 137% more than what vSAN documentation states an AF-6 configuration should consistently be able to serve, at 50K IOPS.



	0	30	50	70	100
🟦 Average of IOPS	53,088	66,827	88,117	134,701	433,455
— Average of latency (ms)	9.67	7.66	5.81	3.80	1.19

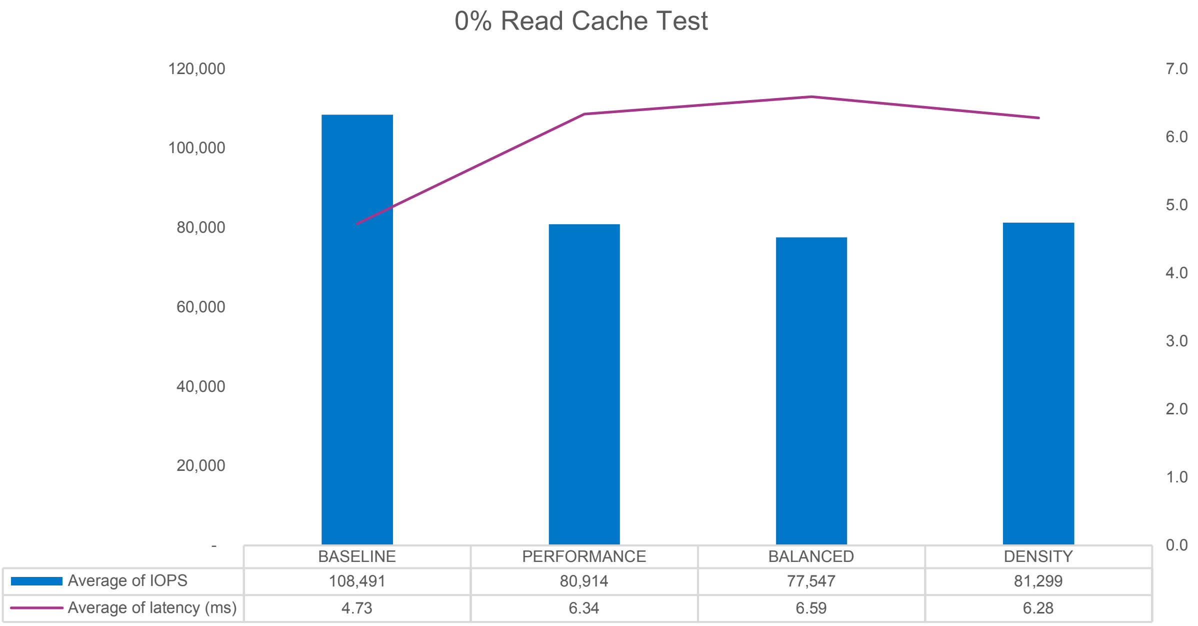
Figure 6: Baseline Cache Test

Figure 6 shows the IOPS and latency for the baseline capacity test. We see the same trend followed as with the cache test, but with slightly lower performance, especially for the write workloads. As the reads get closer to 100% of the workload, the difference in performance becomes minimal since all destaged reads come from the capacity tier in an all-flash configuration.

At 100% reads, both tests see minor performance differences. Read caching (in memory) is a large contributor to this observation, as vSAN dedicates a small amount of memory in each host for caching some data. The smaller working set size you use, the more apparent this feature will be.

Performance Results—Cache Test

The first comparison is with a working set size that fits 100% in cache (**cache test**). This test eliminates most destaging actions and increases performance for the mixed tests since the cache tier is much more performant than the capacity tier.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	108,491	80,914	77,547	81,299
Average of latency (ms)	4.73	6.34	6.59	6.28

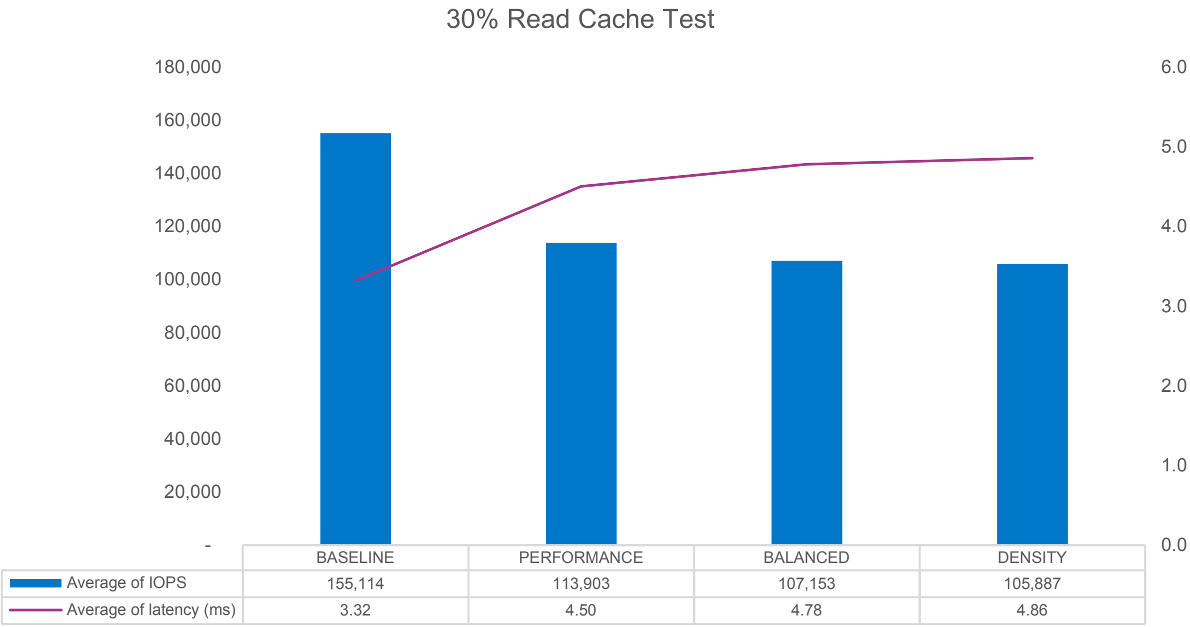
Figure 7: 0% Read Cache Test

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, since computing the checksum requires additional CPU cycles for each write operation.

For this test, enabling checksum reduces IOPS by roughly 25% from the baseline, as well as adding approximately 34% latency.

The balanced profile, which utilizes RAID-5/6, shows a negligible performance drop from the profile at 4% lower IOPS and 4% higher latency. We expect write performance to lag with enabling RAID-5/6, since parity calculations will be performed, but it appears to be very small here. This means the added latency associated with the calculation is very small compared to the disk write latency.

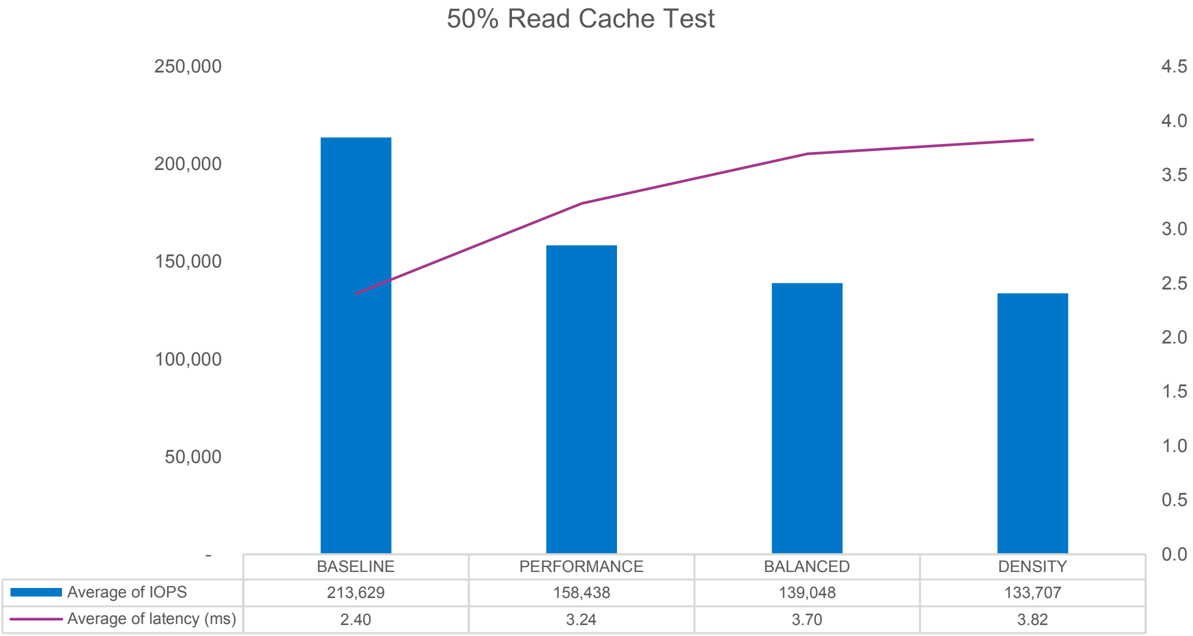
The density profile produces about 5% higher IOPS and 5% lower latency than the balanced profile. Typically, enabling deduplication and compression adds some CPU overhead. However, just like with enabling RAID-5/6, the CPU requirement is a small impact on the overall latency, with the disk write latency being a larger contributor to the overall latency. The performance and density profiles are all close enough that we can say their differences are statistically insignificant for this test case.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	155,114	113,903	107,153	105,887
Average of latency (ms)	3.32	4.50	4.78	4.86

Figure 8: 30% Read Cache Test

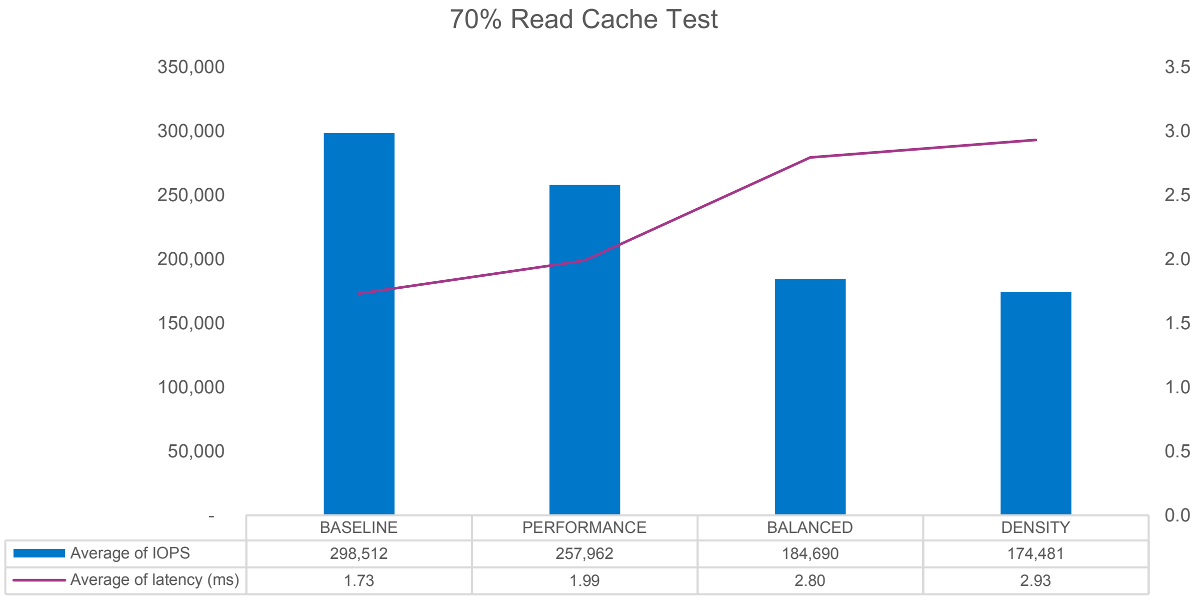
Adding 30% writes into the mix shows an increase in performance on all profiles, which is to be expected. We also see that the trend across profiles is closer to what we would expect from the various profile options, with performance decreasing with each option enabled. The performance profile shows a 27% reduction in IOPS and a 36% increase in latency. Switching to RAID-5/6 in the balanced profile shows another 6% reduction in IOPS and 6% increase in latency. Lastly, enabling deduplication and compression gives an additional 1% reduction in IOPS and 2% increase in latency.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	213,629	158,438	139,048	133,707
Average of latency (ms)	2.40	3.24	3.70	3.82

Figure 9: 50% Read Cache Test

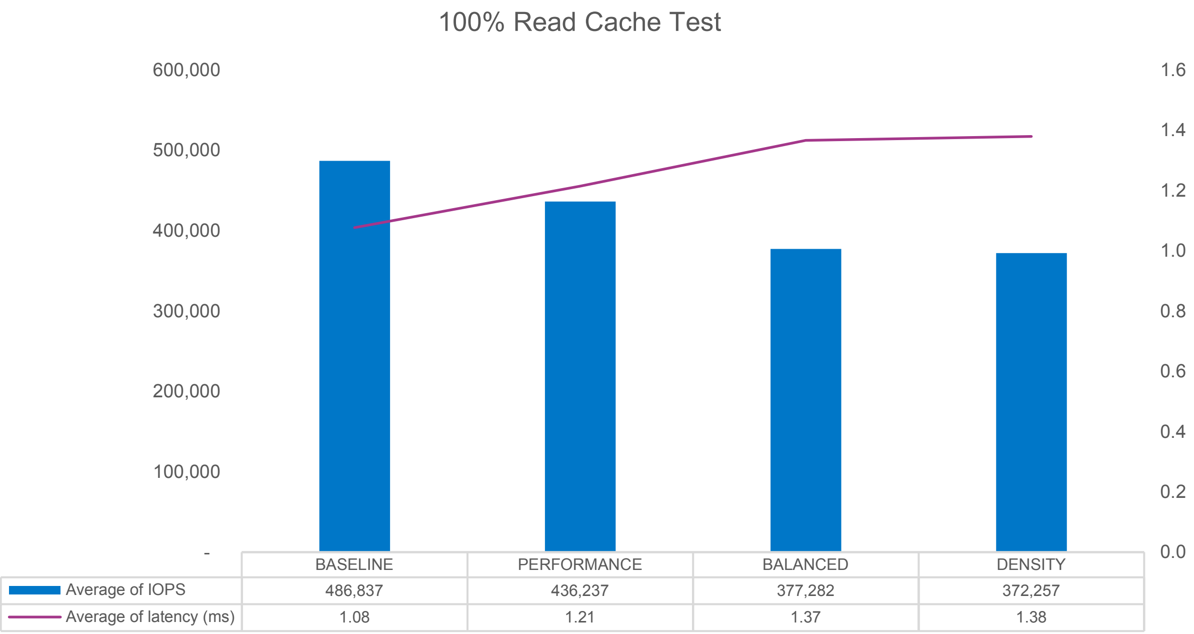
At 50% writes, performance is again higher for all profiles, and the difference between each profile becomes more apparent. The performance profile shows 26% less IOPS and 35% higher latency than the baseline. The balanced profile shows an additional 22% reduction in IOPS and 14% increase in latency. The density profile further reduces IOPS 4% and increases latency 3%. At this point, RAID-5/6 causes little performance degradation (nor does deduplication and compression).



	BASELINE	PERFORMANCE	BALANCED	DENSITY
▬ Average of IOPS	298,512	257,962	184,690	174,481
— Average of latency (ms)	1.73	1.99	2.80	2.93

Figure 10: 70% Read Cache Test

At 70% reads, we continue to see IOPS increase and latency decrease. The performance profile reduces IOPS 14% from the baseline and increases latency 15%. Here, the reduction in performance from RAID-5/6 becomes very apparent. The balanced profile reduces IOPS by 28% and increases latency by 41%. This is because read latency is inherently much lower than write latency, and thus CPU overhead becomes a larger contributor to the overall write latency than the disk latency. This is further witnesses by enabling deduplication and compression, which further reduces performance minimally at 6% reduction in IOPS and 5% increase in latency.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	486,837	436,237	377,282	372,257
Average of latency (ms)	1.08	1.21	1.37	1.38

Figure 11: 100% Read Cache Test

At 100% reads, we are at the highest performance for each profile. The baseline shows up to 486K IOPS at 1.08ms latency, which is almost 2 GB/s throughput. Enabling checksum for the performance profile reduces IOPS by 10% and increases latency by 12%. Changing to RAID-5/6 from mirroring (balanced) further reduces IOPS by 14% and increases latency by 13%. Enabling deduplication and compression (density) has minimal impact, reducing IOPS by less than 2% and increasing latency by less than 1%.

This first test shows that if you have a relatively small working set size—meaning it fits entirely (or mostly) in the cache tier—there is very little downside to utilizing RAID-5/6 with deduplication and compression, especially if your workload is mostly writes.

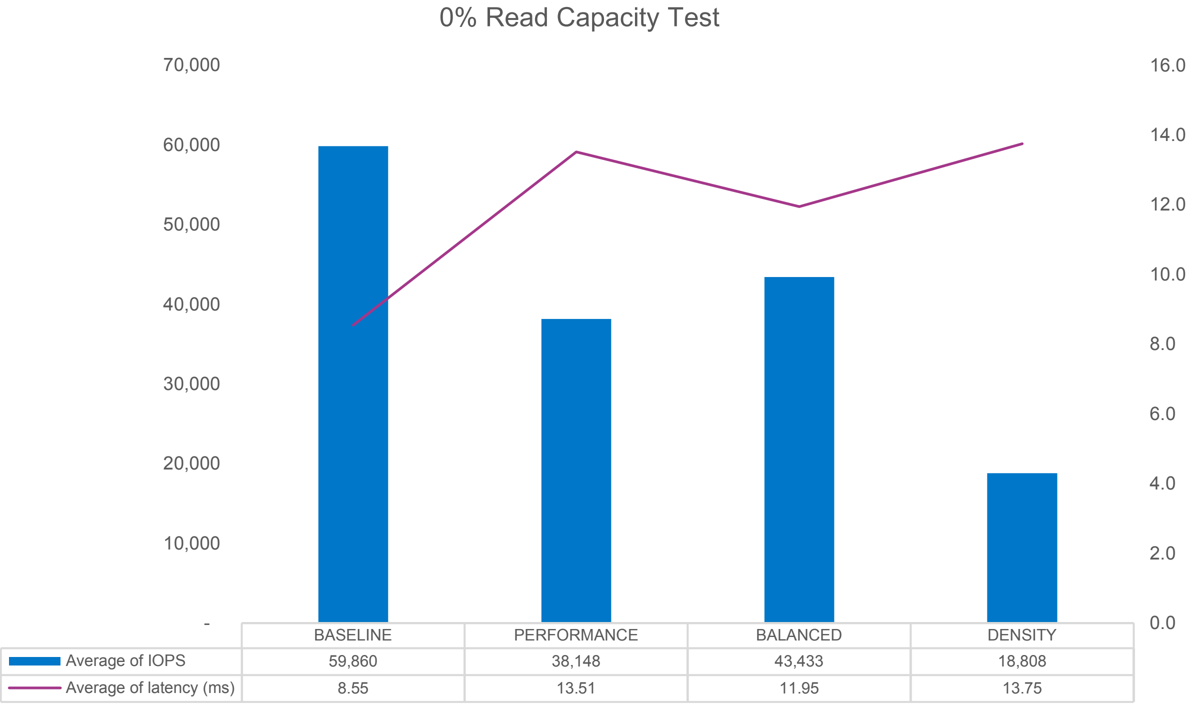
If increased usable capacity is your primary goal, the density profile can potentially give you much more usable capacity than your raw capacity, thanks mostly to deduplication and compression. It is important to note that not all workloads are the same. Some are more compressible and some are less compressible. If your workload is highly compressible, using deduplication and compression is strongly recommended. If your dataset is not very compressible, you may be better off not using deduplication and compression, as you will take a small performance penalty with little or no capacity benefit.

Tip: Workload Compressibility

If your workload is mostly reads and your data set is highly compressible, deduplication and compression is recommended. If your data set is far less compressible, you may see better results not using these features.

Performance Results—Capacity

The second comparison (**capacity test**) 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 is around 4TB, with each node only being able to use 600GB per disk group for cache. Therefore, only about 29% of user data can reside in the cache tier, while the rest must be held in the capacity tier. Consequently, this means that there will be considerable destaging operations for write-intensive workloads, which will reduce performance.



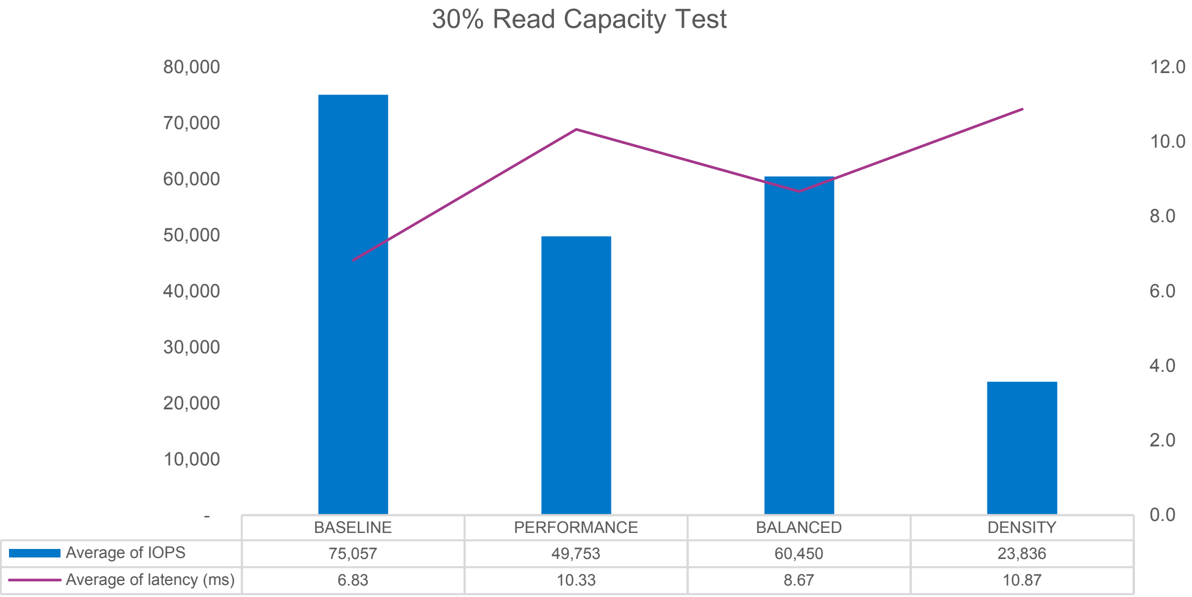
	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	59,860	38,148	43,433	18,808
Average of latency (ms)	8.55	13.51	11.95	13.75

Figure 12: 0% Read Capacity Test

Our first observation with the capacity test is that performance is lower across the board. This makes sense because we now have destaging occurring as well as less of a percent of our data set being cached in memory. Enabling checksumming drastically reduces performance, with a 34% reduction in IOPS and 58% increase in latency.

When enabling RAID-5/6, we see something strange. The balanced profile gives us a significant performance increase over the performance profile, with 14% higher IOPS and 12% lower latency. Even though RAID-5/6 requires some additional work by the CPUs, it doesn't have to write as many bits to the capacity tier, and thus we see higher performance. It is important to remember that the capacity tier is using 5100 ECO SSDs, which are lower performance and less expensive than the 5100 MAX SSDs used in the cache tier.

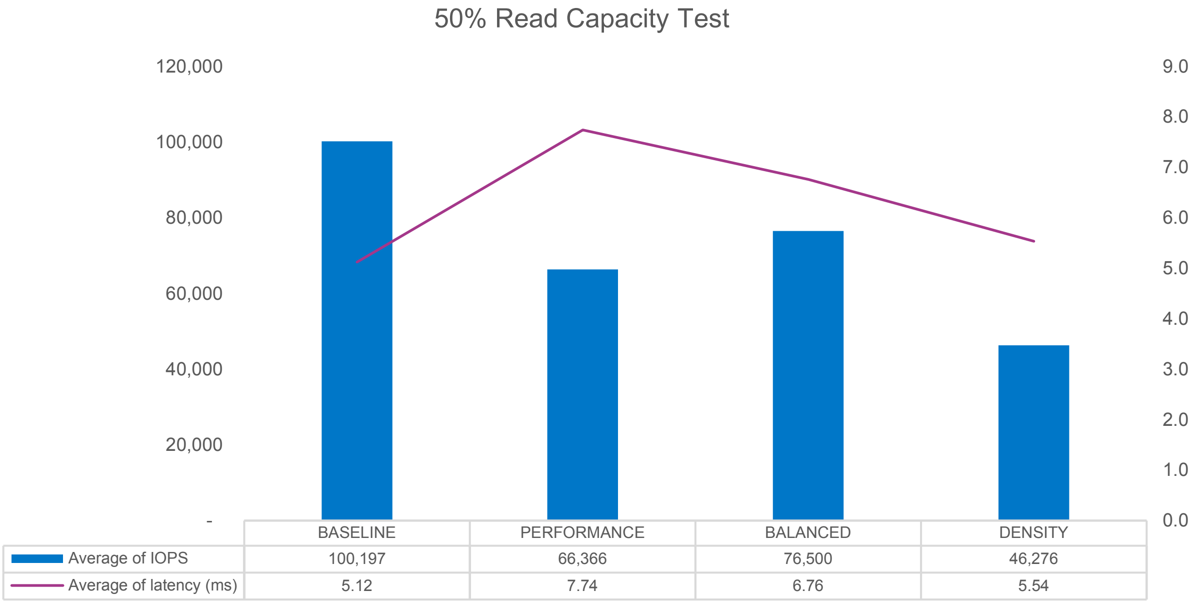
Using data from the balanced profile, one might expect deduplication and compression to show a minimal performance difference from the balanced profile. However, the chart above does not support that theory. The density profile shows a drastic performance decrease, with 57% fewer IOPS and 15% higher latency. This anomaly is discussed in Appendix B and is a very important consideration when enabling deduplication and compression on SATA drives. Note that because of this issue, the capacity tests for 0/30/50/70% read used two threads per VMDK, as opposed to four threads for all other tests.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	75,057	49,753	60,450	23,836
Average of latency (ms)	6.83	10.33	8.67	10.87

Figure 13: 30% Read Capacity Test

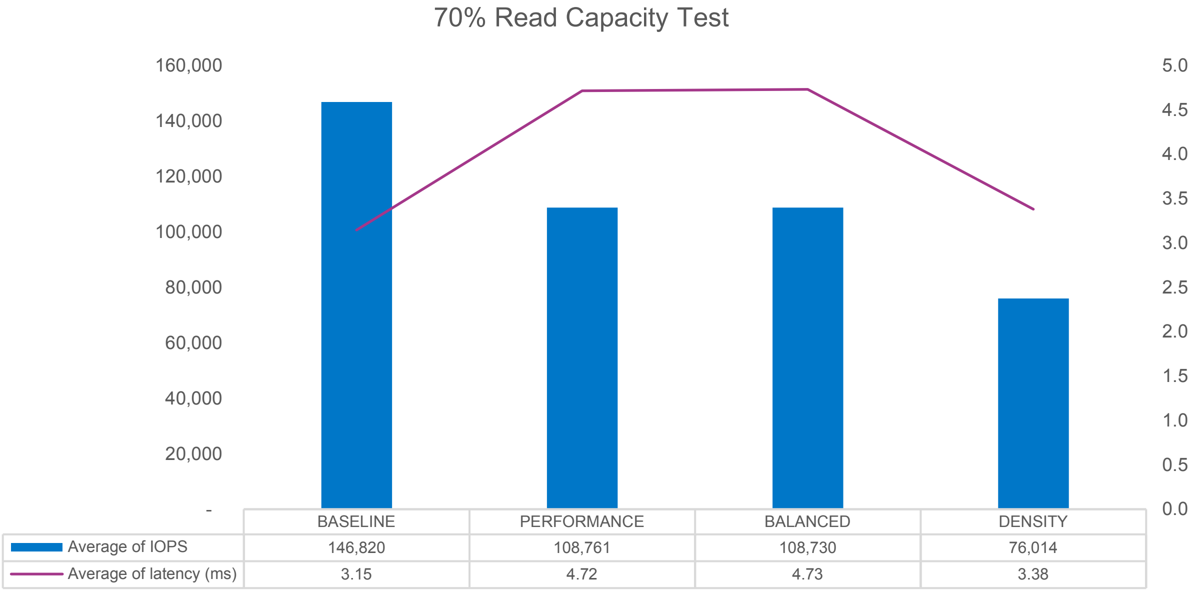
At 30% reads, we see the same trend as at 0%, but with higher performance. Enabling checksum (performance) reduces IOPS by 34% and increases latency by 51%. Switching to RAID-5/6 (balanced) gives some performance back, increasing IOPS by 21% and reducing latency by 16%. Enabling deduplication and compression results in a 61% reduction in IOPS and 25% increase in latency.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	100,197	66,366	76,500	46,276
Average of latency (ms)	5.12	7.74	6.76	5.54

Figure 14: 50% Read Capacity Test

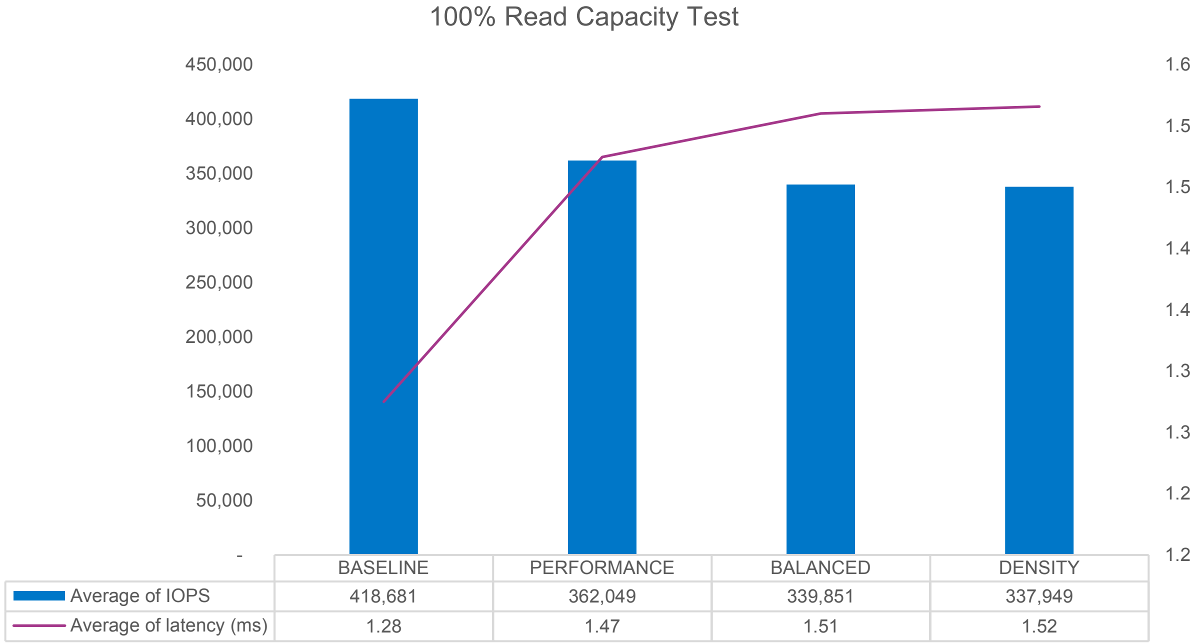
50% reads, again, follows the same trend, but with higher performance. The performance profile results in a 34% reduction in IOPS and 51% increase in latency. A balanced profile increases IOPS by 15% and decreases latency by 13%. A density profile reduces IOPS by 40% and decreases latency by 18%. Note that the gap between the balanced profile and the performance/density profiles is shrinking. In the case of the performance profile, we are starting to see the benefit of RAID-1 for read operations. For the density profile, we are seeing less log congestion due to fewer writes, so the performance impact is diminishing.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	146,820	108,761	108,730	76,014
Average of latency (ms)	3.15	4.72	4.73	3.38

Figure 15: 70% Read Capacity Test

At 70% reads, the performance profile has caught up to balanced profile, with almost identical IOPS and latency. The density profile still lags, with 30% lower IOPS and 29% lower latency. We are now seeing the effect of better read performance of RAID-1 compared to RAID-5/6.



	BASELINE	PERFORMANCE	BALANCED	DENSITY
Average of IOPS	418,681	362,049	339,851	337,949
Average of latency (ms)	1.28	1.47	1.51	1.52

Figure 16: 100% Read Capacity Test

At 100% reads, we see an expected trend: Enabling checksum (performance) results in 14% fewer IOPS and 15% higher latency. Switching to RAID-5/6 (balanced) shows 6% fewer IOPS and 3% higher latency. Lastly, enabling deduplication and compression (density) shows less than a 1% difference in both IOPS and latency.

Summary

The two test cases in this RA illustrate how the performance of a vSAN cluster is strongly dependent on working set size. If the working set fits mostly in the cache tier, you will see much higher performance than if only a small portion of it does, especially if you use deduplication and compression.

Typically, choosing RAID-5/6 reduces performance significantly (that is why we chose RAID-1 for performance). However, with these SATA drives and a large percentage of writes on a working set size that does not fit in the cache tier, performance shows improvement, along with space-saving benefits.

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 is likely not an option. This is a case where NVMe drives, or another low-latency alternative, would be recommended.

100% Read Workloads: Performance Depends on Working Set Size

▷ Enabling checksum (**performance**) results in 14% fewer IOPS and 15% higher latency.

▷ Switching to RAID-5/6 (**balanced**) shows 6% fewer IOPS and 3% higher latency.

▷ Enabling deduplication and compression (**density**) shows less than a 1% difference in both IOPS and latency.

Appendix A: vSAN Configuration Details

Tuning Parameters

vSAN's default tunings are configured 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, we followed VMware's performance document to alter three advanced configuration parameters. The table below shows the default value and the value used in this configuration.

Tunings[1]		
Parameter	Default	Tuned
/LSOM/blPLOGCacheLines	128K	512K
/LSOM/blPLOGLsnCacheLines	4K	32K
/LSOM/blLLOGCacheLines	128	32K

Table 9: vSAN Settings[1]

1. https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2150012

Note: Even with these performance tunings, vSAN occasionally experiences various forms of congestion. Congestion appears to occur more often during runs with 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 particular parameter file is the one used for testing deduplication and compression, using a deduplication ratio of four with 4K units, and a compression ratio of five. This resulted in an initial compression ratio of 3.25X, which after accounting for using RAID-5/6, puts the total ratio of usable capacity to raw capacity at 2.63. The highlighted section denotes the modifications that were 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=4
dedupunit=4k
compratio=5
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 100-102
  vlan 110-115
  interface ethernet 1/49/1 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/1 switchport trunk allowed-vlan add 100-102
  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 100-102
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 1
  interface ethernet 1/49/2 switchport trunk allowed-vlan add 100-102
  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 100-102
  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 100-102
  interface ethernet 1/49/4 switchport trunk allowed-vlan add 110-115
```

Appendix B: Deduplication and Compression on SATA Drives

As we saw previously, the density profile had some unexpected behavior on capacity tests with large amounts of reads. This section discusses why that is and how we determined the issue.

One of the perks of using HCIBench for testing is that every test starts an instance of vSAN observer and provides diagnostic data from the test run. This is incredibly useful in scenarios where performance does not seem to be correct. Below are some of the graphs that helped us determine what was occurring.

By looking at the vSAN observer charts, we immediately saw congestion was causing strange behavior.

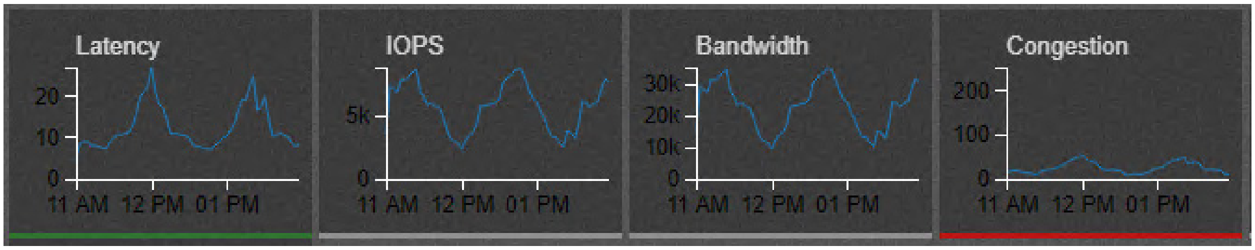


Figure 17: Host Performance

Figure 17 shows the IOPS, latency, and bandwidth with respect to the level of congestion introduced. Notice that as congestion is introduced, the IOPS decrease and latency increases. This is a cyclical process, where congestion increases as the log space fills up, and returns to zero once the log space frees.

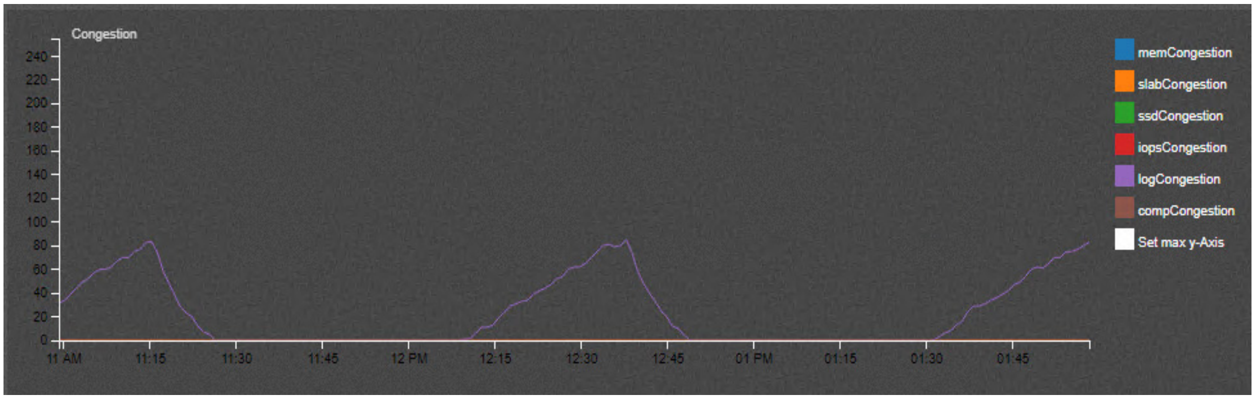
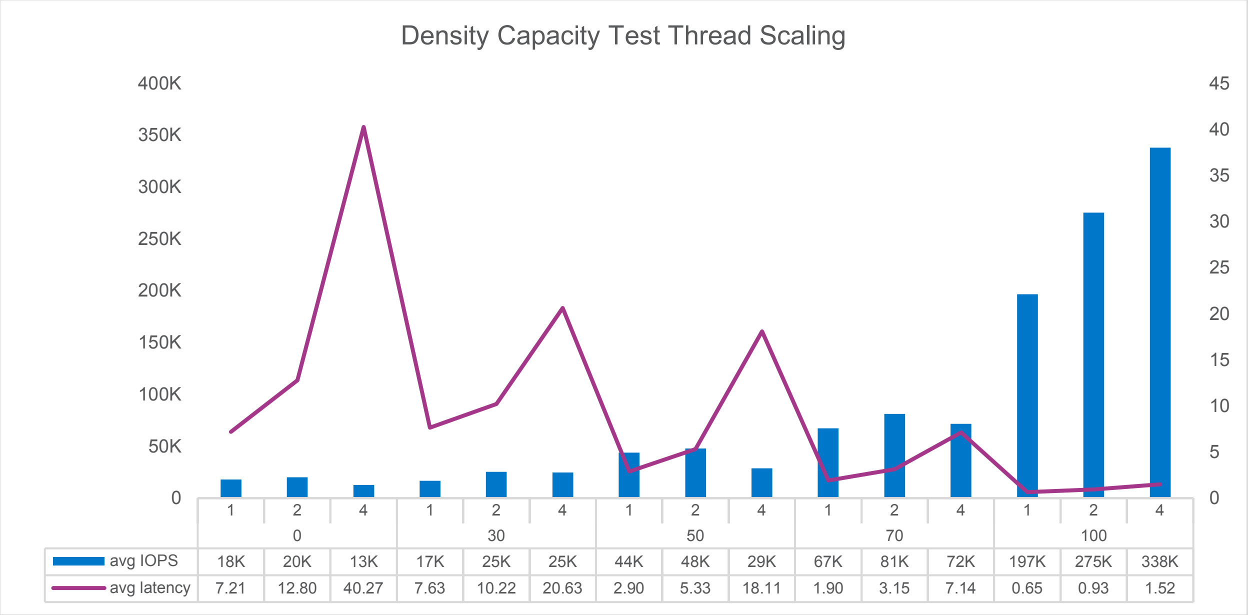


Figure 18: Congestion Type

Drilling down further into this congestion issue, we see there was only one type of congestion present: log congestion. This congestion occurs when the internal log in the cache tier of vSAN's Local Log Structured Object Manager (LSOM) runs out of space. This log is fixed in size, and when metadata is written faster than it can be purged, the log runs out of space. (Log congestion is introduced to slow down write transactions.)

Upon discovering this issue, an additional sweep of threads per VMDK was needed for the density profile to try and minimize this effect and increase performance.



	0			30			50			70			100		
	1	2	4	1	2	4	1	2	4	1	2	4	1	2	4
avg IOPS	18K	20K	13K	17K	25K	25K	44K	48K	29K	67K	81K	72K	197K	275K	338K
avg latency	7.21	12.80	40.27	7.63	10.22	20.63	2.90	5.33	18.11	1.90	3.15	7.14	0.65	0.93	1.52

Figure 19: Thread Scaling on Density Profile for Capacity Test

Figure 19 shows the thread scaling on the density profile for capacity tests. We see that maximum IOPS for all tests except 100% read is attained at two threads per VMDK. Going to four threads reduces IOPS while drastically increasing latency. Thus, we used two threads per VMDK for 0/30/50/70% read on the density profile for capacity tests and four threads for 100% read.

Appendix C: 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, configure them, run Vdbench files against each VM, 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 Ruby vSphere Console (RVC). HCIBench starts an observer instance with each test, and stores it alongside of 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.

- **Ruby vSphere Console (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 D: Bill of Materials

Component	Qty per Node	Part Number	Description
Server	1	868703-B21	HP Proliant DL380 Gen 10
CPU	2	BX806736148	6148 Gold 20 core 2.40GHz
Memory	12	MEM-DR432L-CL02-ER26	Micron 32GB DDR4-2666MHz RDIMM ECC
Boot Drive	1	MTFDDAK480TCB-1AR1ZABYY	480GB Micron 5100 PRO
Cache SSD	2	MK000960GWEZK	Micron 5100 MAX SATA 960GB SSD
Capacity SSD	8	VK001920GWEZE	Micron 5100 ECO SATA 1920GB SSD
Networking (NIC)	1	631FLR-SFP28	Broadcom BCM57414 NetExtreme-E 25GbE Dual Port

micron.com

Benchmark software and workloads used in performance tests may have been optimized for performance on specified components and have been documented here where possible. Performance tests, such as HCIbench, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

©2018 Micron Technology, Inc. All rights reserved. All information herein is provided on an “AS IS” basis without warranties of any kind. Micron, the Micron logo, and all other Micron trademarks are the property of Micron Technology, Inc. All other trademarks are the property of their respective owners. Products are warranted only to meet Micron’s production data sheet specifications. Products, programs and specifications are subject to change without notice. Dates are estimates only. Rev. C 6/18 CCM004-676576390-11062