Micron® Accelerated All-Flash NVMe™ and SATA vSAN™ 6.6 Solution

Reference Architecture

Contents

Executive Summary

This reference architecture (RA) describes an example configuration of an all-flash VMware vSAN™ platform that combines NVM Express® (NVMe™) SSDs in the cache tier and enterprise SATA SSDs in the capacity tier into standard x86 rack-mount servers with 10 GbE networking. The combination of high-performance NVMe SSDs and lower-cost SATA SSDs with standard servers provides an optimal balance of performance and cost.

Similar to an AF-8 configuration, this VMware vSAN 6.6 all-flash enables:

- Fast deployment: The configuration has been pre-validated and is thoroughly documented to enable fast deployment.
- Balanced design: The right combination of cache and capacity SSDs, DRAM, processors and networking ensures subsystems are balanced and performance-matched.
- Broad deployment: Complete tuning and performance characterization across multiple IO profiles for broad deployment across multiple workloads

This RA details the hardware and software building blocks and measurement techniques to characterize performance and composition, including vSphere configuration, network switch configurations, vSAN tuning parameters, and Micron reference nodes and Micron SSD configuration.

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—a pragmatic blueprint for administrators, solution architects and IT planners who need to build and tailor a high-performance vSAN infrastructure that scales for I/O-intensive workloads.

Note that the performance shown was measured using the components noted. Different component combinations may yield different results.

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 solution leverages decades of technical expertise as well as direct, engineer-to-engineer collaboration.

Micron's Reference Architectures

Micron Reference Architectures are optimized, pre-engineered, enterprise-leading solution templates co-developed by 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 a number of 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
- Detect and recover from faults and failures
- Redistribute data dynamically (rebalance)
- Ensure data integrity (scrubbing)

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, 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 datastore, enabling creation of high-performance VMDKs for things such as database log files, and high-capacity/availability disk groups for things such as critical data files.

Figure 1 shows the logical layers of the vSAN stack, from the hosts down to the vSAN datastore.

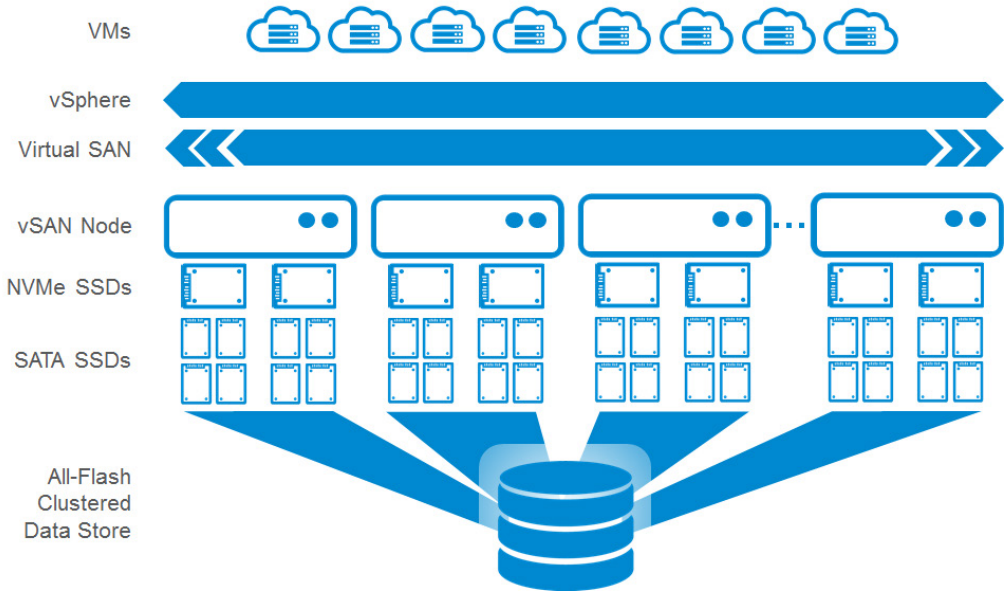


Figure 1: vSAN Architecture

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 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). 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 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 that 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.

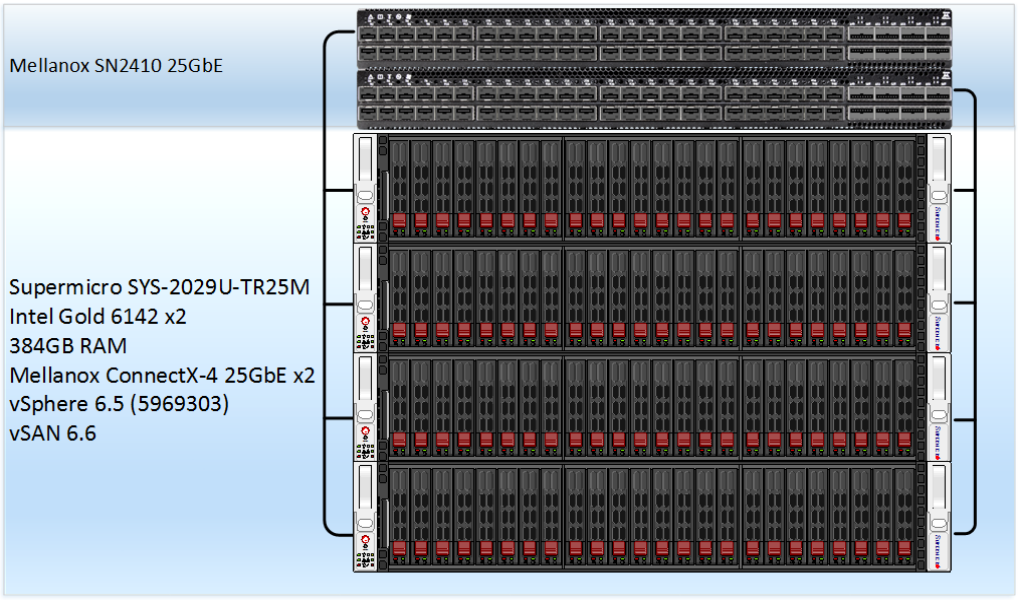


Figure 2: vSAN RA Hardware

Software

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

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:

- **Flash-optimized IOPS:** 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.
- **vSAN Encryption:** Native to vSAN, vSAN Encryption provides data-at-rest security at the cluster level; built for compliance requirements and offers simple key management.

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

According to 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 a hybrid configuration, 70% of the cache tier capacity is dedicated to the read cache and the remaining 30% is dedicated for 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® SSDs with NVMe™ for the cache tier and SATA SSDs for the capacity tier. This solution also utilizes Micron DRAM (the details of which are not discussed further in this document).

SSD	Use	Random Read	Random Write	Read Throughput	Endurance (TBW)
9200 MAX	Cache Tier	700,000 IOPS	256,000 IOPS	2,850 MB/s	3.5 PB
5100 ECO	Capacity Tier	93,000 IOPS	10,000 IOPS	540 MB/s	8.8 PB

Table 1: Micron SSDs

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

Server Platforms

This RA utilizes standard rack-mount 2U dual-socket Intel-based servers. Each server is configured with two Intel® Xeon® Gold 6142 processors, each with 16 cores at 2.60 GHz. These processors align with VMware's AF-8 requirements (VMware's nomenclature for a large-sized all-flash configuration).

Solution Network

Switch

vSAN utilizes commodity Ethernet networking hardware. This RA uses two Mellanox® SN2410 switches for all cluster-related traffic. Both switches are connected with a single QSFP+ cable between them. Spanning Tree is enabled to avoid loops in the network. All ports are configured in general mode, with VLANs 100-114 allowed. Each server is connected via a QSFP+ quad-port breakout cable.

According to VMware, vSAN requires at least three separate logical networks which 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 101, subnet 192.168.1.X/24
- vSAN: VLAN 102, subnet 192.168.2.X/24

While using different subnets or VLANs alone would suffice, adding both ensures 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.

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.

Network Interface Cards

Each server has a single dual-port Mellanox MT27630 ConnectX®-4 25 GbE NIC, with one port of each 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.

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	
2U, 2-socket standard rack mount server (note: Supermicro SYS-2029U-TR25M tested, other platforms may give different results)	
2X Intel® Xeon® Gold 6142 16-core 2.60GHz CPUs	1X 240GB Micron Enterprise SATA SSD (OS drive)
Micron 384GB 2666 MHz DRAM (32GB x 12)	3X LSI 9300-8i SAS/SATA HBA
3X Micron 1.6TB NVMe SSDs (9200 MAX)	1x Mellanox ConnectX-4 Dual-port 25 GbE SFP+ NIC (MT 27630)
12X Micron 3.84TB SATA SSDs (5100 ECO)	

Table 2: Node Hardware Components

Network Infrastructure

Network Components	
2X Mellanox SN2410 25 GbE switches	Mellanox QSFP+ copper breakout cables

Table 3: Network Infrastructure Components

Solution Design - Software

Software Components	
vCenter Server Appliance 6.5.0.10000	HBA driver 6.9.10.18.00-1vmw.650.0.0.4564106
ESXi build 5969303	HBA firmware 24.21.0-0015
vSAN 6.6	9200 driver 1.2.0.32-4vmw.650.1.26.5969303
Disk format version 5.0	9200 firmware 100007C0
	5100 firmware D0MU410

Table 4: Software Components

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 may mean missing performance requirements.

This RA targets 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 12 capacity drives, and 10GbE networking at a minimum. For more information on AF-8 requirements, see VMware's vSAN Hardware Quick Reference Guide.

This configuration utilizes three disk groups per node, with four capacity drives per disk group, resulting in three cache drives and 12 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 because of the 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 RA is moderately sized for good performance at a balanced price point.

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 RA focuses on using synthetic benchmarking to gauge storage performance.

The benchmark tool used 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 (114), and the HCIBench VM's virtual NIC was assigned to this network to ensure 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 5. Storage Policies

For each configuration, five different workload profiles were run, all generating 4K random read/write mixtures. Since read and write performance differs 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 enough to ensure the entire dataset is written over twice. This ensures the VMDKs have readable data instead of simply all zeros. This is particularly important when it comes to checksumming to ensure 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 earlier. This ensures 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 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 becomes stable. For all tests conducted in this paper, the time to reach steady state was about one hour—called rampup time or duration. After rampup, performance data is captured over a long enough time period 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 30 minutes.

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 based on trial and error, as four threads seemed to be the sweet spot where IOPS was near its highest value while keeping latency at an acceptable level.

HCIBench Test Parameter	Cache	Capacity	HCIBench Test Parameter	Cache	Capacity
Threads per VMDK	4		%Random	100	
Test Duration	30 minutes		Working Set Size	100%	
Ramp Duration	1 hour		SSD Initialization	128K SEQ WRI	
%Read	0/30/50/70/100%		Clear Cache Before Testing	Yes	

Table 6: HCIBench Test Parameters

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 we see roughly 2:1 writes at the vSAN level as compared to what the VMs think they are writing.

When using 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 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 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 they call 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 simply 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.

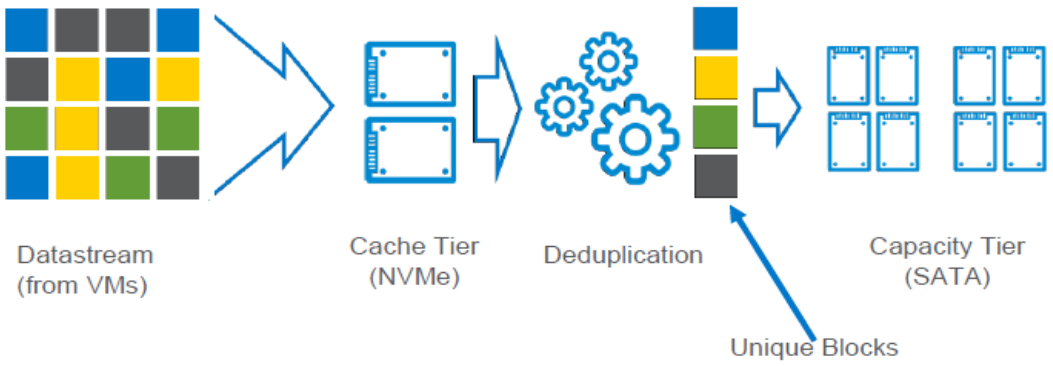


Figure 3. Data Deduplication

Testing deduplication and compression 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 must 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, 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 approximately a 3.5x deduplication and compression ratio 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.

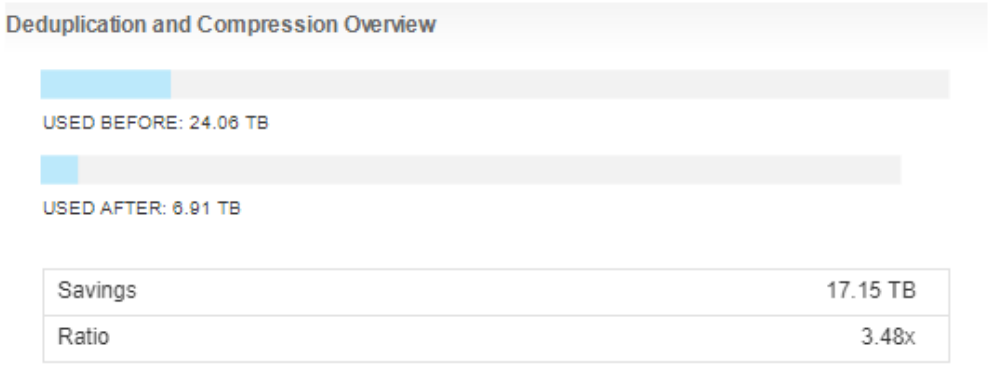


Figure 4. Deduplication and Compression ratio

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 requires twice the capacity the data set occupies. The density configuration improves upon this, requiring an additional 33% more space than the data set occupies, but at a performance penalty.

The table below shows how much additional 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 7. Additional Storage (by option)

Performance Results: Baseline

To obtain 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 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.

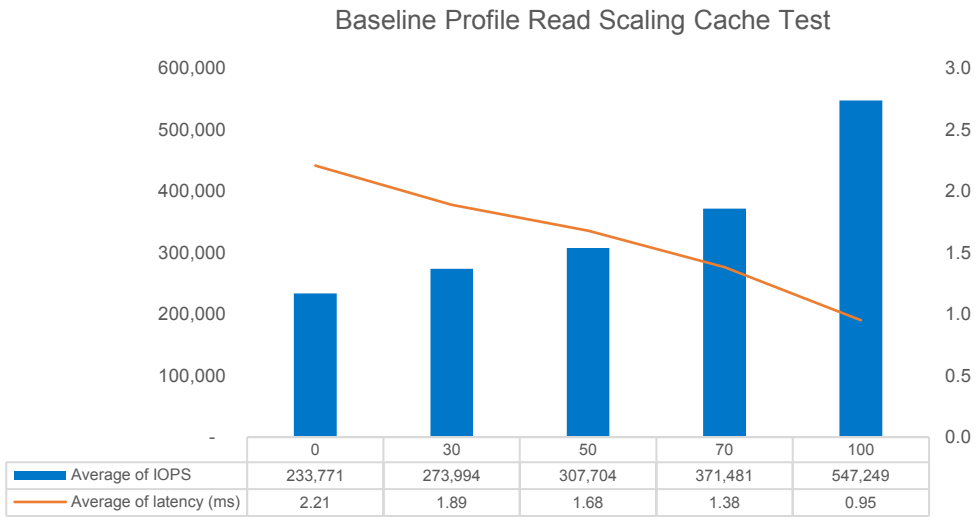
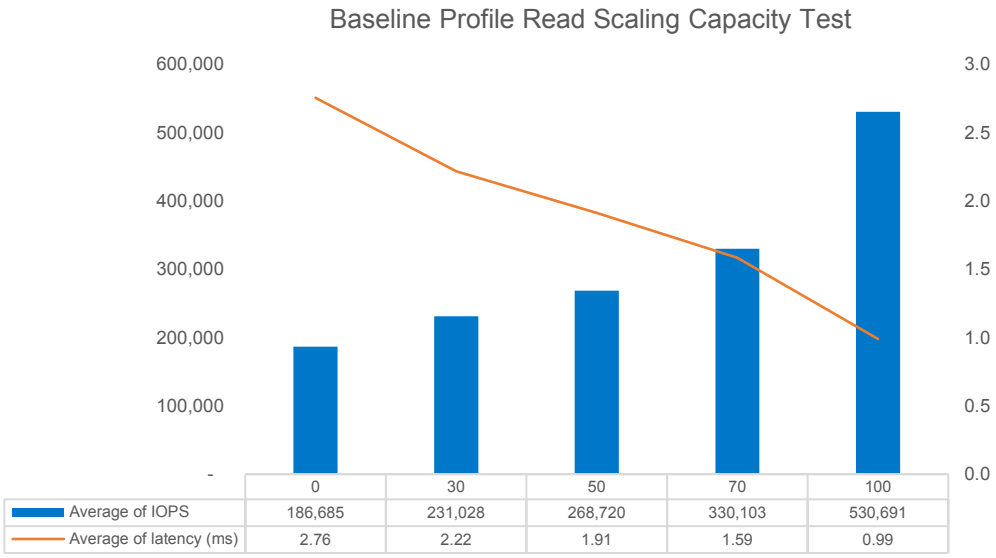


Figure 5. Baseline Cache Test

Figure 5 shows the IOPS and latency for the baseline for each read percentage mixture. Doing a pure write test produces 233K IOPS at an average latency of 2.21ms. 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 547K at 0.95ms latency. This means each node can deliver over 136K IOPS, which is 70% more than what vSAN claims an AF-8 configuration should consistently be able to serve, at 80K IOPS.



	0	30	50	70	100
Average of IOPS	186,685	231,028	268,720	330,103	530,691
Average of latency (ms)	2.76	2.22	1.91	1.59	0.99

Figure 6. Baseline Capacity 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 negligible since all destaged reads come from the capacity tier in an all-flash configuration.

Above 70% reads, both tests see minor performance differences. This means that below 30% writes, the capacity tier is mostly able to keep up with the rate at which the cache tier tries to destage writes. Above 30%, the capacity drives begin to slow down and can't keep up with the writes to the cache tier. Read caching (in memory) also comes into play, as vSAN dedicates a small amount of memory in each host for caching some data.

The smaller working set size, the more apparent this feature will be.

Tip: Deduplication, Compression and Write Performance

Deduplication and compression have almost no performance penalty with 100% writes, enabling additional capacity with little or no performance penalty.

Performance Results: Cache Test

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

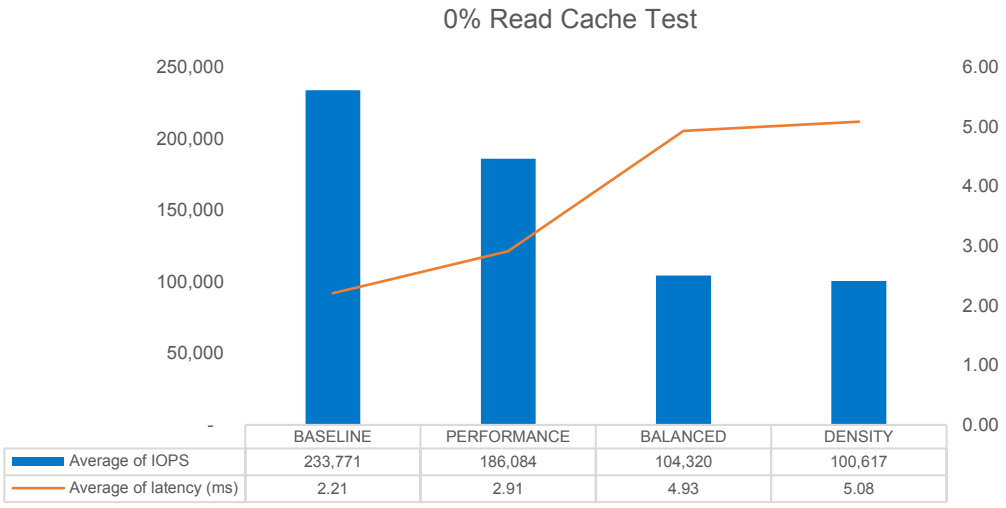


Figure 7. 0% Read Cache Test

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

For this test, enabling checksum reduces IOPS by roughly 20% from the baseline, as well as adding approximately 31% latency. The density storage profile shows its worst-case performance here. We expect write performance to lag with enabling RAID-5/6, since parity calculations will be performed.

The performance of the balanced profile is about 55% lower than the baseline profile, with 123% higher latency. Enabling deduplication and compression shows very little performance difference. The performance of the density storage profile shows a difference of less than 4% from the balanced profile.

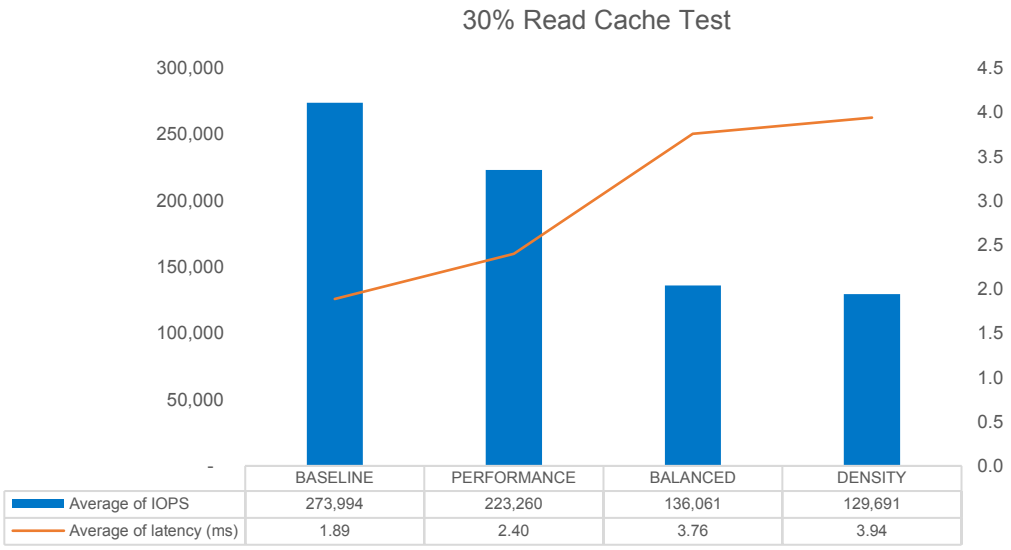


Figure 8. 30% Read Cache Test

Figure 8 shows 30% read results. The trend is similar to the 0% read test, but with slightly higher performance. The performance profile shows a 19% decrease in IOPS and 27% increase in latency. The balanced profile is about 50% lower IOPS and 99% higher latency. The difference between balanced and density is less than 5%.

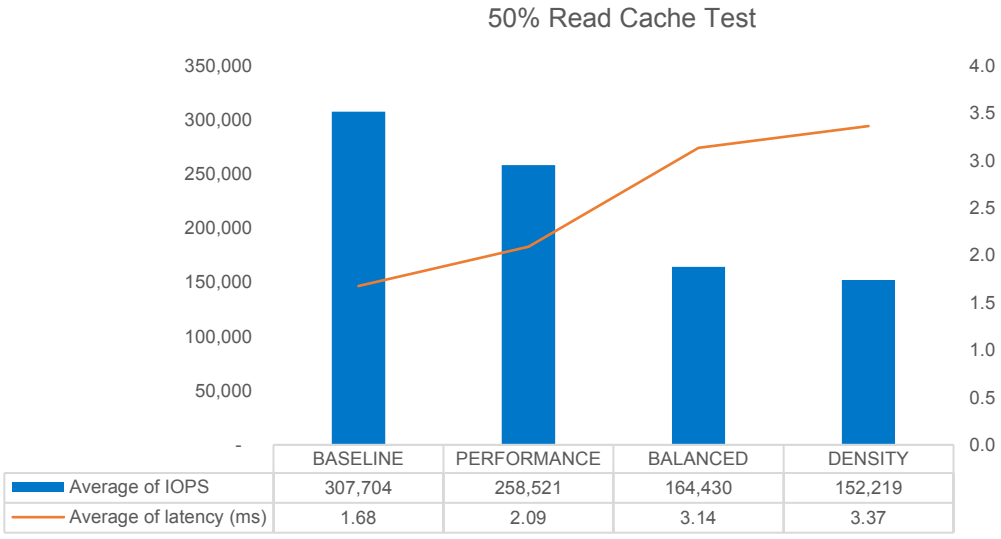


Figure 9. 50% Read Cache Test

As we start to induce more reads into the mix (Figure 9), we start to see that the difference between each profile with respect to the baseline begins to diminish. The performance profile is now only about 16% lower in IOPS than the baseline, and 24% higher latency. Balanced is 47% lower IOPS and 87% higher latency. Density is about 7% lower IOPS than balanced with 7% higher latency.

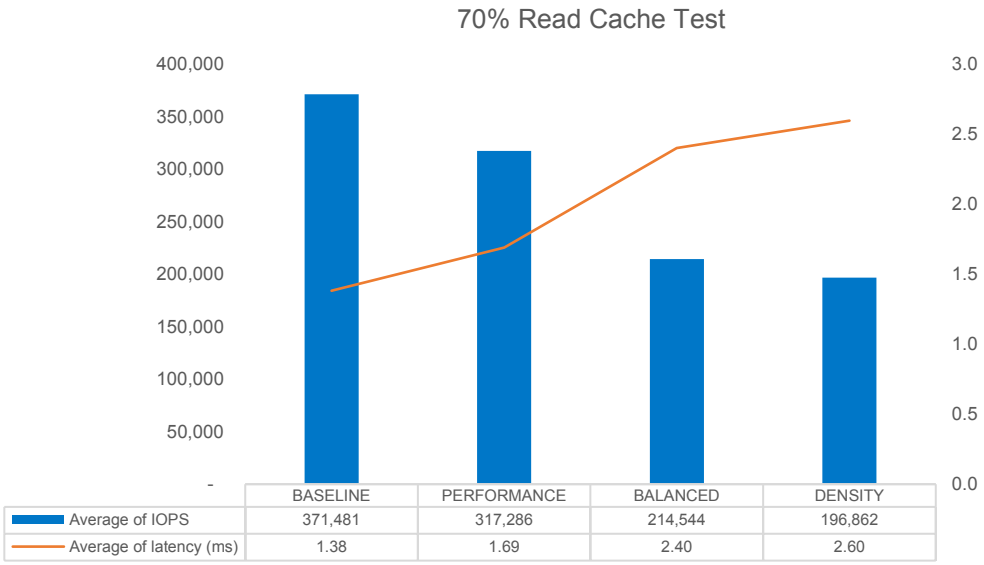


Figure 10. 70% Read Cache Test

With 70% reads in the mix (Figure 10), the difference diminishes further. The performance profile sees a 15% reduction in IOPS and 22% increase in latency. The balanced profile shows a 42% decrease in IOPS and an 88% increase in latency. The density profile shows an 8% decrease in IOPS and 8% increase in latency from the balanced profile.

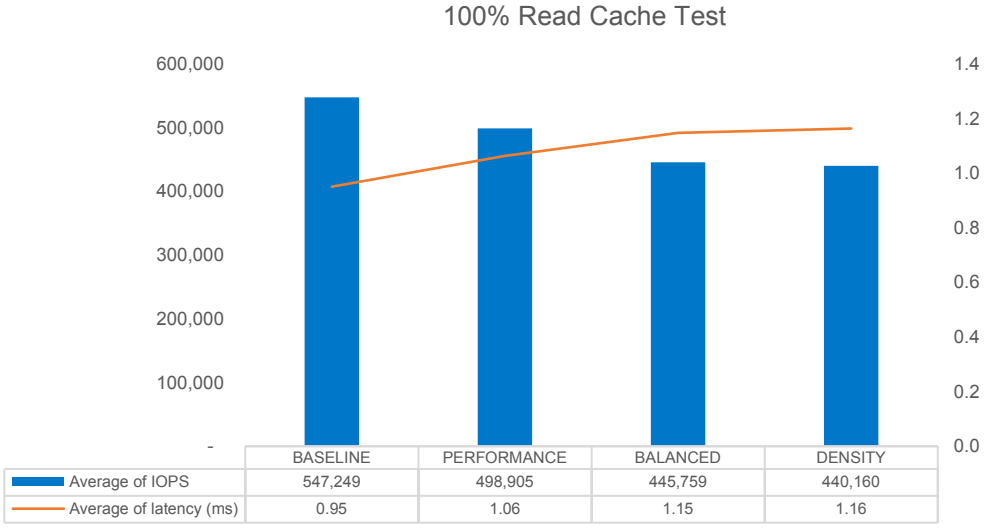


Figure 11. 100% read cache test

When we run a test with 100% reads (Figure 11), we see relatively small differences in performance between each profile. Enabling checksum (performance profile) only reduces IOPS by about 9%, and increases latency by 12%. The balanced profile shows about 19% lower IOPS, and 21% higher latency than the

baseline. The difference between the balanced and density profile is minimal, with a less than 2% reduction in IOPS and less than 1% increase in latency.

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

If increased usable capacity is your primary goal, the density profile will give you roughly 2.63 times more usable capacity vs raw capacity (based on this workload; see the tip below on workload compressibility). This is due mostly to deduplication and compression. It is important to note that not all workloads are the same. Some are more compressible, some are less compressible. If your workload is highly compressible, using deduplication and compression is recommended. If your dataset is far less compressible, you may see better results 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, using 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 looks at performance differences when the working set does not 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 44% of user data can reside in the cache tier, while the rest must be held in the capacity tier. Consequently, this means there will be considerable destaging operations for write-intensive workloads, which will reduce performance.

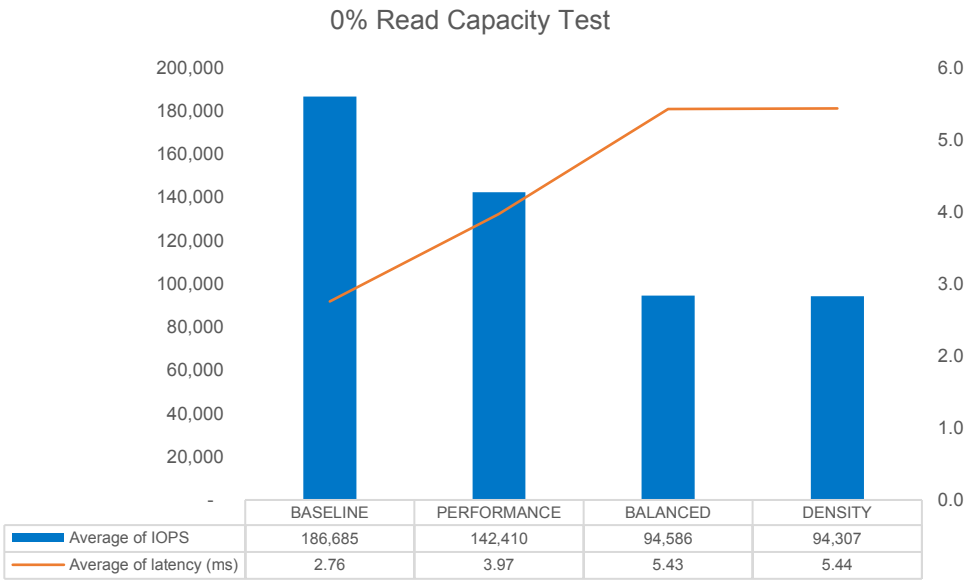


Figure 12. 0% Read Capacity Test

Figure 12 shows that the capacity test follows a trend similar to the cache test, but with lower performance numbers. This is to be expected because all writes now propagate from the cache tier to the capacity tier. The baseline profile for the capacity test is only able to reach 186K IOPS, as opposed to the 233K that the cache test reached. The performance profile saw a similar reduction in performance—at around 24% of the IOPS of the baseline profile—with 43% higher latency. The balanced and density profiles offered practically identical performance, at 50% lower IOPS and 97% higher latency than the baseline.

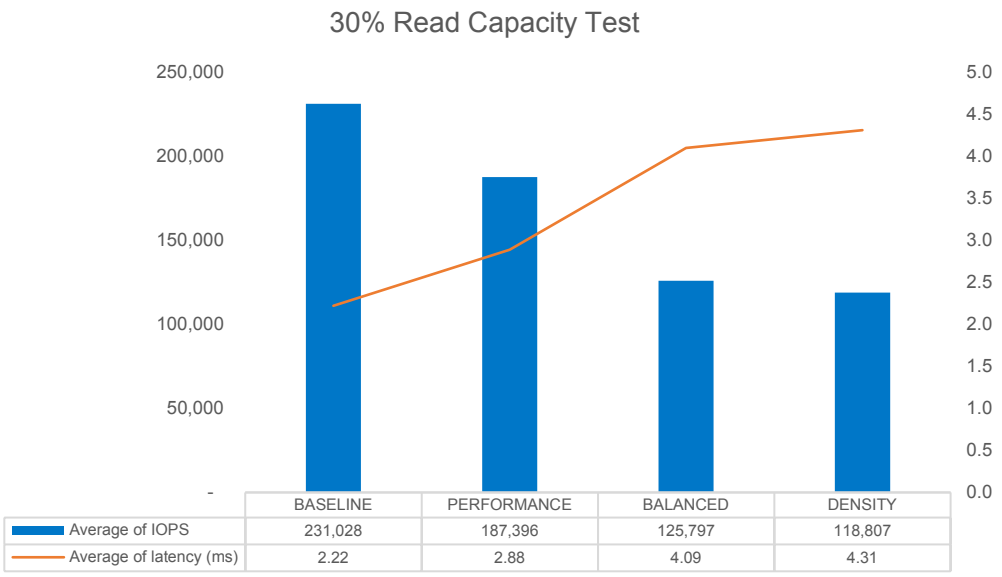


Figure 13. 30% Read Capacity Test

Figure 13 shows that with 30% read transactions, the trend is the same, but with slightly higher performance. Again, the balanced and density profiles are very similar.

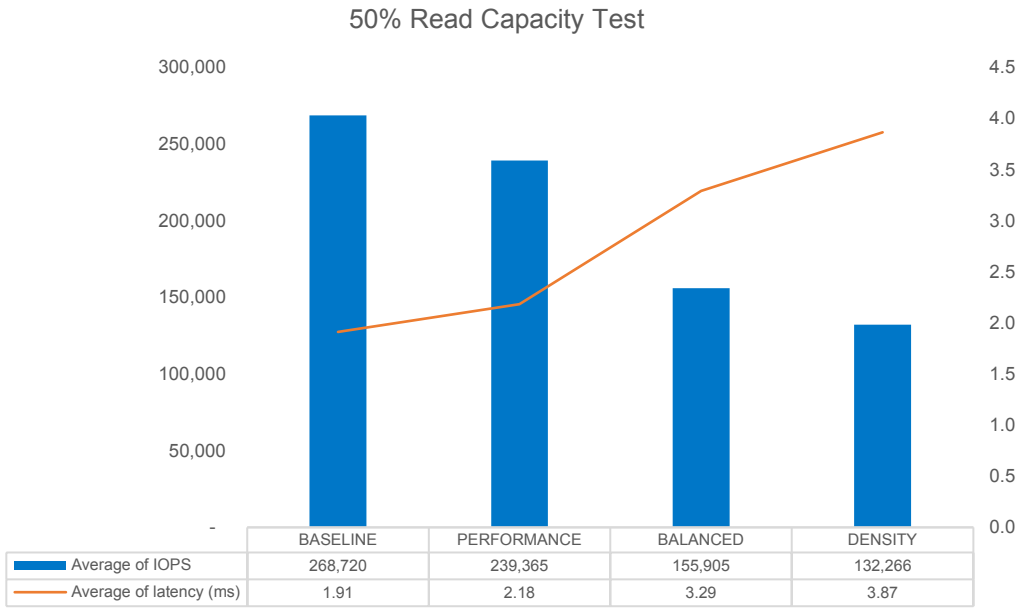


Figure 14. 50% Read Capacity Test

With 50% reads (Figure 14), we see a trend that is similar to the cache test, where the performance penalty of the RAID-5/6 profiles (balanced and density) become less pronounced.

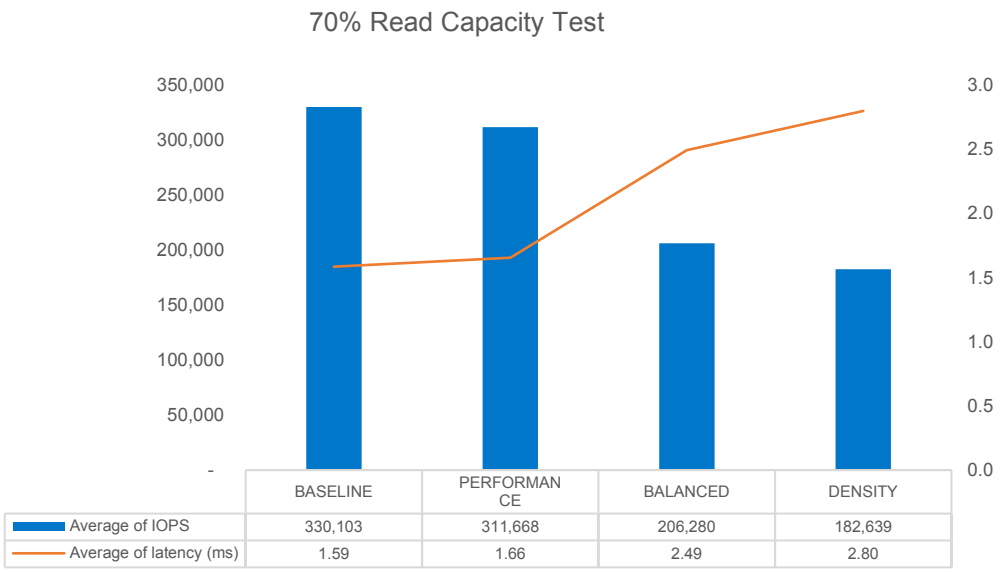


Figure 15. 70% Read Capacity Test

Figure 15 shows 70% read. Here, we see there is very little performance difference between the baseline and the performance profile, with only 6% fewer IOPS and 4% higher latency. The balanced profile yields 38% lower IOPS and 19% higher latency. The density profile reduces IOPS by an additional 12% and increases latency by 12% over the balanced profile.

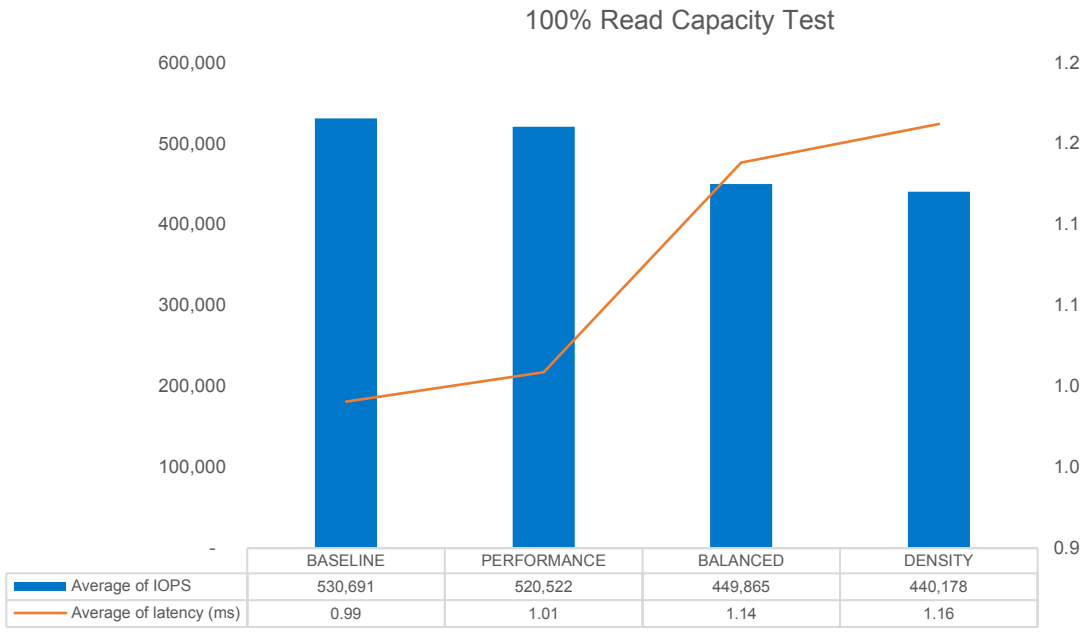


Figure 16. 100% Read Capacity Test

With 100% reads shown in Figure 16, we again see that there is very little performance difference across all the profiles.

The performance profile shows only a 2% decrease in IOPS and a 2% increase in latency over the baseline. The balanced profile gives 15% lower IOPS and 15% higher latency than the baseline. The density profile only reduces IOPS by about 2% and increases latency by less than 2% over balanced.

Summary

Both the cache and capacity profiles showed very similar trends in performance. Write performance is significantly worse than read performance in both cases, which is to be expected.

Enabling checksum showed minimal performance penalties in most tests, and it seemed to have a slightly higher impact on the cache tests than the capacity tests. Switching from RAID-1 to RAID-5/6 showed a significant performance reduction in almost all tests where any write mix was involved. This option is beneficial in that it extends your usable capacity, but at the cost of performance.

Enabling deduplication and compression showed only a very minor performance implication, which means that customers who need the extra capacity need not worry about losing performance by enabling this feature.

Tip: 100% Read Workloads

Enabling checksum with 100% read provides greater data integrity while showing minimum performance impact. So does enabling compression and deduplication with RAID-5/6.

To increase effective capacity, enable dedupe and compression with RAID-5/6 and 100% read.

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, the value this configuration used, and the documents referenced for the tunings.

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

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

Note: vSAN congestion was occasionally observed during runs with high write percentage. When observed, it was not alleviated with these performance tunings. Since congestion was transitory and inconsistent, results in this paper are focused on data observed when congestion was not present.

Vdbench Parameter File

Below is a sample Vdbench parameter file for a 0% read test against 8 VMDKs with a run time of 30 minutes and a warmup (ramp) time of one hour. This particular parameter file is the one used for testing deduplication and compression, using a deduplication ratio of 4 with 4K units and a compression ratio of 5. This resulted in an initial compression ratio of 3.5x, 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=1800,warmup=3600,interval=30
```

Switch Configuration (Sample Subset)

Below is a collection of sample sections of one of the switch configs. 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 110-114
   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 110-114
   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 110-114
   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 110-114
   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 110-114
```

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, 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 C: Bill of Materials

Component	Qty per Node	Part Number	Description
Server	1	SYS-2029U-TR25M	Supermicro 2U Ultra Server
CPU	2	BX806736142	6142 Intel Xeon Gold 16 core 2.60 GHz
Memory	12	MEM-DR432L-CL02-ER26	Micron 32GB DDR4-2666MHz RDIMM ECC
Boot Drive	1	MTFDDAK240TCB-1AR1ZABYY	240GB Micron 5100 PRO SSD
NVMe SSD	2	MTFDHAL1T6TCU-1AR1ZABYY	Micron 9200 MAX NVMe 1600GB SSD
Networking (NIC)	1	AOC-2UR68-M2TS-O	Mellanox ConnectX-4 25GbE Dual Port

Appendix D: About

Micron

Micron Technology is a world leader in innovative memory solutions that transform how the world uses Information. Through our global brands — Micron, Crucial and Ballistix — we offer the industry's broadest portfolio, and are the only company that manufactures today's major memory and storage technologies: NVMe™ and SATA SSDs, DRAM, NAND, NOR, and 3D XPoint™ memory.

VMware

VMware (NYSE: VMW), a global leader in cloud infrastructure and business mobility, helps customers realize possibilities by accelerating their digital transformation journeys. With VMware solutions, organizations are improving business agility by modernizing data centers and integrating public clouds, driving innovation with modern apps, creating exceptional experiences by empowering the digital workspace, and safeguarding customer trust by transforming security. With 2016 revenue of $7.09 billion, VMware is headquartered in Palo Alto, CA and has over 500,000 customers and 75,000 partners worldwide.

micron.com

Rev. A 3/18 CCM004-676576390-11005