



VMware Virtual SAN 6.2 with Virtual Desktop Infrastructure Workload

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Executive Summary	3
Introduction.....	3
Virtual SAN 6.2 New Features.....	3
Virtual Desktop Infrastructure Workload	4
VDI Testbed Setup.....	4
Content Based Read Cache (CBRC) in View Storage Accelerator	4
View Planner Benchmark.....	5
View Planner Scores.....	6
Virtual SAN Cluster Setup	6
Virtual SAN Hybrid Cluster Hardware Configuration	6
Virtual SAN All-Flash Cluster Hardware Configuration	6
Metrics	6
Virtual SAN Configurations.....	7
Performance of Hybrid Virtual SAN Cluster.....	8
View Planner Scores.....	8
CPU Utilization	8
Guest I/O per Second.....	8
Guest Average Latency	8
Impact of DOM Client Read Cache.....	9
Performance of All-Flash Virtual SAN Cluster	11
View Planner Scores.....	11
CPU Utilization	11
Guest I/O per Second.....	11
Disk Space Efficiency.....	11
Guest Average Latency	13
Impact of DOM Client Read Cache.....	13
Conclusion	14
Appendix A. Hardware Configuration for Hybrid Virtual SAN Cluster	14
Appendix B. Hardware Configuration for All-Flash Virtual SAN Cluster	14
References	15

Executive Summary

This white paper evaluates the performance of Virtual Desktop Infrastructure (VDI) applications with Virtual SAN 6.2. The Virtual Desktop Infrastructure delivers desktop service to end users by running virtual machines on consolidated clusters in the datacenter. The performance of the underlying storage solution is critical to the desktop service that VDI delivers. In this paper, we show that Virtual SAN 6.2 performs just as well as 6.1 with the VDI workload by meeting the I/O latency requirements from the applications. At the same time, Virtual SAN 6.2 provides data integrity and disk space saving benefits to users by way of the new features at a very small CPU cost.

Introduction

Virtual SAN is a distributed layer of software that runs natively as part of the VMware vSphere® hypervisor. Virtual SAN aggregates local or direct-attached storage disks in a host cluster and creates a single storage pool that is shared across all hosts of the cluster. This eliminates the need for external shared storage and simplifies storage configuration and virtual machine provisioning operations. In addition, Virtual SAN supports vSphere features that require shared storage such as VMware vSphere® High Availability (HA), VMware vSphere® vMotion®, and VMware vSphere® Distributed Resource Scheduler™ (DRS) for failover. More information on Virtual SAN design can be obtained in the [Virtual SAN design and sizing guide](#) [1].

Note: Hosts in a Virtual SAN cluster are also called nodes. The terms “host” and “node” are used interchangeably in this paper.

Virtual SAN 6.2 New Features

Virtual SAN 6.2 introduces new features to improve space efficiency and data integrity. These features provide benefits to users but may consume more resources. For a full review of Virtual SAN 6.2 new features, please refer to the [datasheet](#) [2], [white paper](#) [3], and [blog post](#) [4]. The data in this white paper demonstrates performance numbers with the following new features and illustrates the trade-off between performance and resource cost.

- **Data integrity feature: software checksum**
Software checksum is introduced to enhance data integrity. Checksum works on a 4KB block. Upon a write, a 5-byte checksum is calculated for every 4KB block and stored separately. Upon a read operation, a 4KB data element is checked against its checksum. If the checksum doesn't match the calculation from data, it indicates there is an error in the data. In this case, the data is fetched from a remote copy instead, and the data with the error is updated with a remote copy.
- **Space efficiency feature: erasure coding (RAID-5/RAID-6)**
Previous Virtual SAN releases support only RAID-1 configuration for data availability. To tolerate 1 failure, 1 extra data copy is required and there is a 100% capacity overhead. Similarly, 200% capacity overhead is needed to tolerate 2 failures. However, in Virtual SAN 6.2, a RAID-5 configuration tolerates 1 failure by storing 1 parity from 3 different data objects. Therefore, only a 33% capacity overhead is needed. Furthermore, a RAID-6 configuration tolerates 2 failures by storing 2 parities for every 4 different data objects in a 6-node cluster. Hence, only 50% capacity overhead is needed to tolerate 2 failures.
- **Space efficiency feature: deduplication and compression**
The data stored in Virtual SAN may be de-duplicable or compressible in nature. Virtual SAN 6.2 introduces deduplication and compression features which reduce the space required while the data is being persisted to disks. Deduplication and compression are always enabled or disabled together. The scope of the features is per disk group. Deduplication works when the data is de-staged from the caching tier to the capacity tier, and its granularity is 4KB. Upon writing a 4KB block, it is hashed to find whether an identical block already exists in the capacity tier of the disk group. If there is one, only a small meta-data is updated. If no such identical block is available, compression is then applied to the 4KB block. If the compressed size of the 4KB

block is less than 2KB, Virtual SAN writes the compressed data to the capacity tier. Otherwise, the 4KB block is persisted to the capacity tier uncompressed.

- **Performance enhancement: DOM Client Read Cache**

Virtual SAN 6.2 provides an internal in-memory read cache in the distributed object manager (DOM) layer. It operates on incoming guest read requests to Virtual SAN stack to cache and reduce read latency. This cache resides at Virtual SAN DOM client side which means the blocks of a VM are cached on the host where the VM is located. The default size of this cache is 0.4% of the host memory size with a maximum of 1GB per host.

Erasur coding, and deduplication/compression features are available only on an all-flash configuration. The software checksum feature is available on both hybrid and all-flash configurations. Erasure coding and software checksum features are policy driven and can be applied to an individual object on Virtual SAN. Deduplication and compression can be enabled or disabled across clusters. DOM client read cache is supported on both Hybrid and All-Flash Virtual SAN 6.2 clusters and is enabled by default.

Virtual Desktop Infrastructure Workload

View Planner 3.5 is used with VMware View 6.2 to generate the Virtual Desktop Infrastructure workload. Experiments are conducted in both the hybrid and all-flash Virtual SAN 6.2 cluster.

VDI Testbed Setup

[Figure 1](#) gives a brief overview of the VDI setup used in View Planner benchmarks. There are two 4-node Virtual SAN clusters in the setup: the desktop cluster and client cluster. These two clusters are connected by 10GbE network. The desktop virtual machines are provisioned to the desktop cluster with Virtual SAN being the underlying storage. The client virtual machines are deployed to the client cluster, also using Virtual SAN. A VDI user logs into the desktop virtual machine from a client, and the client connects through the View Connection Manager to desktop virtual machines over the network by PCoIP¹ for remote display. The applications are executed in desktops and the user perceives the applications on the client. The Virtual SAN performance of the desktop cluster is of interest to this whitepaper. The provisioning of desktops in the VDI testbed aims to achieve the maximum number of virtual machines for the resource available. In our experiments, we use an automated desktop pool with a floating assignment in VMware View 6.2. Each user (client virtual machine) connects to a separate desktop virtual machine. We use Windows 7 desktops with 1GB memory and 1 virtual CPU. This minimal virtual machine configuration is recommended by [VMware Horizon 6 with View Performance and Best Practices](#) [5] for tests to achieve a maximum-sized desktop pool with optimized resource usage. The best practice [5] also describes the virtual machine configurations of resource-intensive applications. For desktops, we create View Composer linked clones² with View Storage Accelerator and CBRC enabled. The default settings for disposable disk (4 GB) and customization with QuickPrep are selected in View Pool creation.

Content Based Read Cache (CBRC) in View Storage Accelerator

The View Storage Accelerator is configured to use the Content-Based Read Cache³. CBRC caches common virtual disk blocks as a single copy in the memory for virtual machines on the same host. This way it reduces read

¹ PCoIP (PC over IP) provides an optimized desktop experience for the delivery of a remote application or an entire remote desktop environment, including applications, images, audio, and video content for a wide range of users on the LAN or across the WAN.

² Linked clones, generated from a parent virtual machine snapshot, are in general preferred over full-clones to conserve disk space. We also use linked clones in our experiments. On the other hand, we would like to note that deduplication feature in V SAN 6.2 will result in substantial amount of space reduction in full-clones. We see space reduction in linked-clones as well since View Planner uses same data files in each virtual machine.

³ View Storage Accelerator (CBRC) provides, by default 1GB in each host, in-memory offline content-based read cache. Virtual machine blocks are registered to CBRC filter during the virtual machine provisioning. This accelerates specifically boot time since each virtual machine typically shares the same content for operating system data.

latencies for cached reads. For details of CBRC, please refer to the [View Storage Accelerator white paper \[6\]](#). The default size of the CBRC cache is 1GB memory. Therefore, 2 in-memory read caches are used in the experiments: CBRC and DOM Client Read Cache. Upon a read from the virtual machine, CBRC is checked first and the DOM client cache is checked only when the request cannot be satisfied by the CBRC.

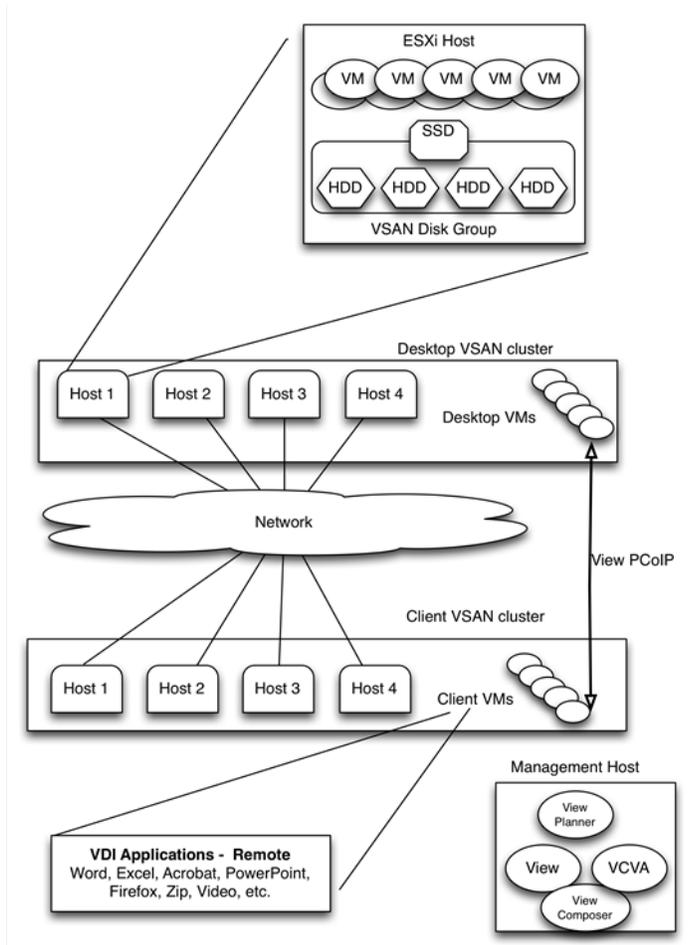


Figure 1. VDI setup desktop and client Virtual SAN clusters

View Planner Benchmark

View Planner simulates the end-user behavior in a typical VDI environment. The View Planner benchmark includes end-user activities such as writing in a Microsoft Word document, preparing PowerPoint slides, sorting data in Excel files, reading Outlook emails, browsing Web pages, and watching video. We use the standard benchmark with 2-seconds of think time between activities to mimic a heavy load environment. The standard benchmark runs 5 iterations with all activities (Word, PowerPoint, Web Browsing, Video, PDF view, Outlook, Excel). View Planner measures application latencies for each operation performed in the iteration, per virtual machine. Operations include open, save, edit, view, iterate, sort, and more. Measurements are taken both from desktops (local) and clients (remote). Remote latencies are used for calculations because they represent what users face in a real environment. First and last iterations are omitted, and the other 3 iterations are used in calculating the View Planner scores. For more information about View Planner, see the user’s guide, which is available from Download Now on the [View Planner product page \[7\]](#).

View Planner Scores

In View Planner, operations are divided into two main categories. **Group A** represents interactive and CPU-intensive operations, and **Group B** represents I/O-intensive operations. For example, a PowerPoint slide show operation is in the Group A category, but a PowerPoint save-as operation is in the Group B category.

The View Planner benchmark requirements are met if Group A score is less than or equal to 1 second and Group B score is less than or equal to 6 seconds. The Group A score is defined as the 95th percentile of latency values in the group, and it represents the latency of majority application activities in the group. Group B score is defined in the same fashion.

The upper boundary value of 1 and 6 seconds are known as the quality of service (QoS) limit for the application Group. Lack of enough compute power will result in more than 1 second Group A score. Similarly, storage performance indirectly affects the Group B score. Latency scores higher than the QoS limits indicate that a fewer number of virtual machines should be deployed to satisfy the View Planner benchmarking requirement.

Notes:

- Because View Planner mimics real-world VDI operations, it is normal to see a 1-2% difference in latency scores between multiple runs.
- We do not use VDI mark as a metric in this study. VDI mark describes the maximum number of users that can be served in a given setup. Instead, we measure Virtual SAN 6.2 performance metrics in the desktop cluster when there is a large number of desktop VMs and the CPU utilization in the cluster is close to saturation.

Virtual SAN Cluster Setup

The performance of the 4-node desktop Virtual SAN cluster in [Figure 1](#) is of interests to readers of this whitepaper. We tested with both a hybrid and all-flash Virtual SAN configuration.

Virtual SAN Hybrid Cluster Hardware Configuration

Each node is a dual-socket Intel Xeon CPU E5-2670 v2 @ 2.50 GHz system with 40 Hyper-Threaded (HT) cores, 256GB memory, and 2 LSI MegaRAID SAS controllers hosting one 400GB Intel S3700 SATA SSDs and 4x 900GB 10,000 RPM SAS drives per controller. Each node is configured to use a 10GbE port dedicated to Virtual SAN traffic. The 10GbE ports of all the nodes are connected to a 10GbE switch. A standard MTU size of 1500 bytes is used. A 1GbE port is used for all management, access, and inter-host traffic. Details on the hardware are available in [Appendix A](#).

Virtual SAN All-Flash Cluster Hardware Configuration

Each node is a dual-socket Intel Xeon CPU E5-2670 v3 @ 2.30 GHz system with 48 Hyper-Threaded (HT) cores, 256GB memory, 2x 400GB Intel P3700 PCIe SSDs, and 1 LSI MegaRAID SAS controller hosting 6x 800GB Intel S3500 SATA SSDs. Each node is configured to use a 10GbE port dedicated to Virtual SAN traffic. The 10GbE ports of all the nodes are connected to a 10GbE switch. Jumbo frames (MTU=9000 bytes) is enabled on the Virtual SAN network interfaces. A 1GbE port is used for all management, access, and inter-host traffic. Details on the hardware are available in [Appendix B](#).

Metrics

In the experiments, virtual desktop VMs and the Virtual SAN cluster hosting them are the systems under test. The View Planner scores, that is the Group A and Group B latency values, are collected as the performance metric. View Planner starts workloads at different times using a random distribution. All the metric collection exclude View Planner's warm-up and cool-off time, only reporting the values during a steady CPU utilization period.

In addition to View Planner metrics, Virtual SAN metrics are also collected. In all the experiments, I/Os per second (IOPs) and the average latency of each I/O operation are measured at the point where guest I/O enters Virtual SAN storage. Two CPU utilization metrics are recorded. The overall system CPU utilization implies how busy the system is under the workload. The CPU utilized by Virtual SAN reflects the resource overhead from the storage software to support the workload.

Finally, the storage space usage is presented in an experiment with the all-flash cluster. Storage space usage is measured by adding up the space consumed by the capacity tier disks in the whole Virtual SAN cluster. For accuracy, the measure is taken when all the data from the workload is de-staged to the capacity tier disks and no data is buffered in the caching tier for accuracy. The space usage number in the cluster is in gibibytes (GiB), and a space saving ratio in percentage is also presented when comparing with the baseline of the Virtual SAN 6.2 default configuration. This percentage directly reflects the benefit of the space efficiency features.

Virtual SAN Configurations

A baseline is first established on the Virtual SAN 6.1 release, and then several Virtual SAN 6.2 feature combinations are used. For the rest of the paper, the abbreviations in Table 1 are used to represent the configurations of features. NC stands for No Cache, R5 stands for RAID-5, and D stands for deduplication and compression.

Name	Testbed cluster configuration	Checksum	RAID level	Deduplication and compression	DOM client read cache
6.1	Hybrid, All-Flash	No	1	No	Feature not available
6.2	Hybrid, All-Flash	Yes	1	No	Yes
6.2 NC	Hybrid	Yes	1	No	No
6.2 R5	All-Flash	Yes	5	No	Yes
6.2 D	All-Flash	Yes	1	Yes	Yes
6.2 R5+D	All-Flash	Yes	5	Yes	Yes
6.2 R5+D NC	All-Flash	Yes	5	Yes	No

Table 1. Test name abbreviations and configurations

Unless otherwise specified in the experiment, the Virtual SAN cluster is designed with the following common configuration parameters:

- Failure to Tolerate of 1
- Stripe width of 1
- Default cache policies are used and no cache reservation is set
- 1 disk group for the tests on the hybrid cluster; each disk group has 4 capacity magnetic disks
- 2 disk groups for the tests on the all-flash cluster; each disk group has 3 capacity SSDs
- DOM client cache size: 1GB memory per host
- CBRC cache size: 1GB memory per host

Performance of Hybrid Virtual SAN Cluster

The first experiment is on the 4-node hybrid Virtual SAN cluster where the desktop pool size in the cluster is varied. We test with 400, 480, 500 and 520 virtual machines in the Virtual SAN 6.2 cluster and compare with a 6.1 baseline. Virtual SAN performs very well in all cases: Group A scores are all under 1 second and Group B scores are all under 6 seconds. The latency score numbers for 6.1 and 6.2 are still very similar. Then, we deep-dive to the 500 virtual machine pool size case and discuss the Virtual SAN performance in detail. Please note that each host in the 4-node hybrid cluster has 20 physical CPU cores that have hyper-threading enabled, and the 500 virtual machine pool translates into 6.25 virtual machines per physical core, which is a good consolidation ratio for VDI. Each host in the Virtual SAN cluster has 1 disk group of 1 SSD and 4 magnetic disks.

View Planner Scores

Virtual SAN results for the 500 virtual machine case are plotted in [Figure 2](#). View Planner scores for both Group A and B are very similar for Virtual SAN 6.1 and 6.2 as shown in [Figure 2](#) (a). Virtual SAN 6.2 achieves this score even though it has more work to do because of its software checksum feature that ensures data integrity, in which every read is verified against the checksum and every write introduces an extra checksum write.

CPU Utilization

The system and Virtual SAN CPU utilization in the cluster is shown in [Figure 2](#) (b). Here we see that the CPU resource is close to saturation for both 6.1 and 6.2 under the workload. Virtual SAN 6.2 uses 0.3% more CPU in the cluster, that is 0.48 of a physical CPU core. This small increase is due to the calculation overhead of software checksum introduced in Virtual SAN 6.2. Overall 6.2 provides data integrity protection for the workload at a very small cost.

Guest I/O per Second

The guest IOPs per host is shown in [Figure 2](#) (c). On each host, Virtual SAN 6.1 has 750 read and 740 write IOPs, while 6.2 gives 650 read and 720 write IOPs. The IOPs number in the View Planner VDI workload has a small run-to-run variation, but the slightly lower IOPs for 6.2 does not indicate a performance degradation since the latency score for 6.1 and 6.2 are very similar and both meet View Planner requirements.

Guest Average Latency

[Figure 2](#) (d) plots the average latency for all guest I/Os, and for read and write I/Os respectively. Virtual SAN 6.1 has an average guest latency of 10.1ms but Virtual SAN 6.2 has 17.2ms. Looking at the read and write latency, it can be inferred that the increase of average latency in 6.2 is from write I/Os. The average write latency climbs from 17.1ms to 29.6ms from Virtual SAN 6.1 to 6.2. The increase in write latency is the result of a scheduling quirk in the vSphere networking stack that manifests itself under high CPU utilization conditions (The VDI run has a CPU utilization close to 100%). However, even though the write latency is higher in Virtual SAN 6.2, the View Planner Score of application latency still meets the requirement, for both group A of latency sensitive applications and group B of I/O intensive applications.

The average read latency for Virtual SAN 6.1 and 6.2 are similar and both low (Virtual SAN 6.1 gives 3.2ms and 6.2 gives 3.8ms). The low read latency is because the Content-Based Read Cache is doing a good job caching guest I/Os. The CBRC cache hit rate is constantly above 80% on all 4 hosts in the cluster during the steady state while the VDI workload is running.

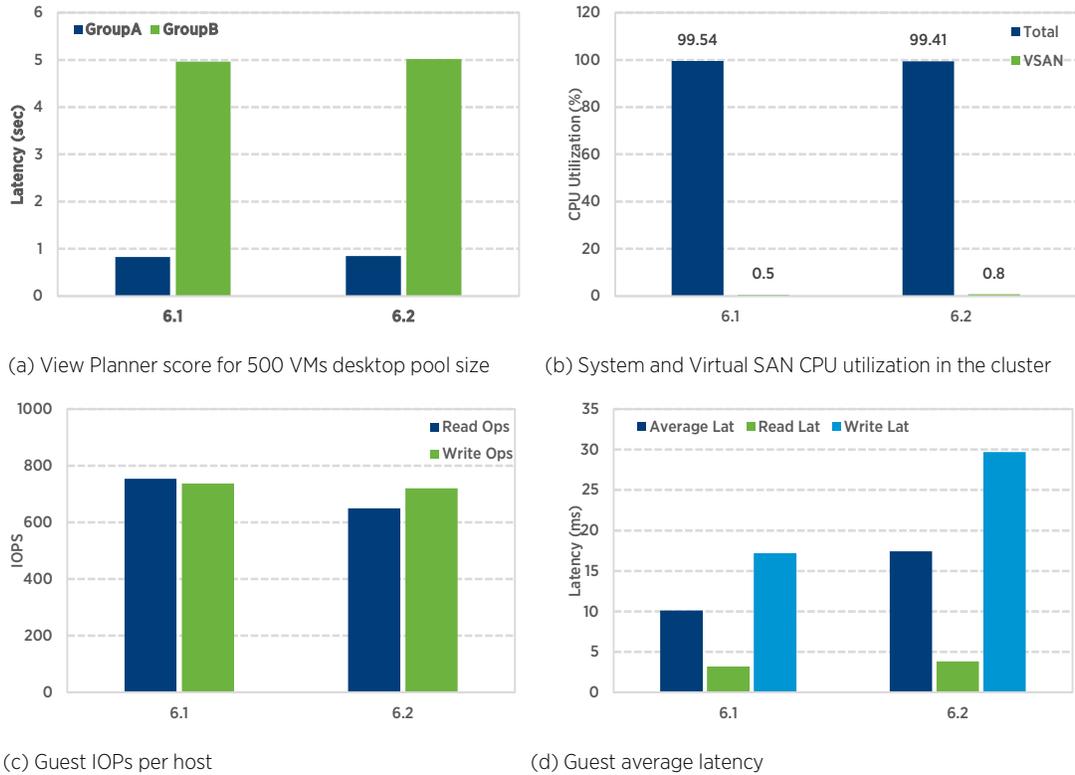


Figure 2. Virtual SAN performance on hybrid cluster for 500 VMs desktop pool size: (a) View Planner scores (b) CPU utilization (c) IOPs per host, (d) Guest average latency

Impact of DOM Client Read Cache

The DOM client read cache is enabled by default in Virtual SAN 6.2, and to illustrate its impact, the 6.2 NC run has the DOM client cache disabled. Figure 3 shows the results of 6.2 NC and compares them with Virtual SAN 6.1 and 6.2. The View Planner Score for 6.2 NC is very similar to 6.1 and 6.2 as illustrated in Figure 3 (a) and meets the latency requirement. The read latency results are shown in Figure 3 (b). Virtual SAN 6.1 has only CBRC, 6.2 has both CBRC and DOM client cache, and 6.2 NC has only CBRC.

There are 2 points worth noting:

- For all the three cases, guest read latency is similar and lower than 4ms, but the DOM client read latency is much higher (more than doubled). We already know this is because CBRC is caching a majority of guest reads.
- We can also see that the DOM client read latency is also reduced when its cache is enabled. Comparing 6.2 and 6.2 NC, 6.2 has DOM client cache and 6.2 NC doesn't. Clearly 6.2 has a lower DOM client read latency of 8.6ms while 6.2 NC has a higher value of 12.2ms. When the DOM client cache is enabled, the internal IOPs is also reduced because cached reads can skip the entire Virtual SAN I/O stack.

For this particular VDI workload, the effect of the DOM client cache is not showing up because the CBRC cache in the View Storage Accelerator is already caching the reads very well.

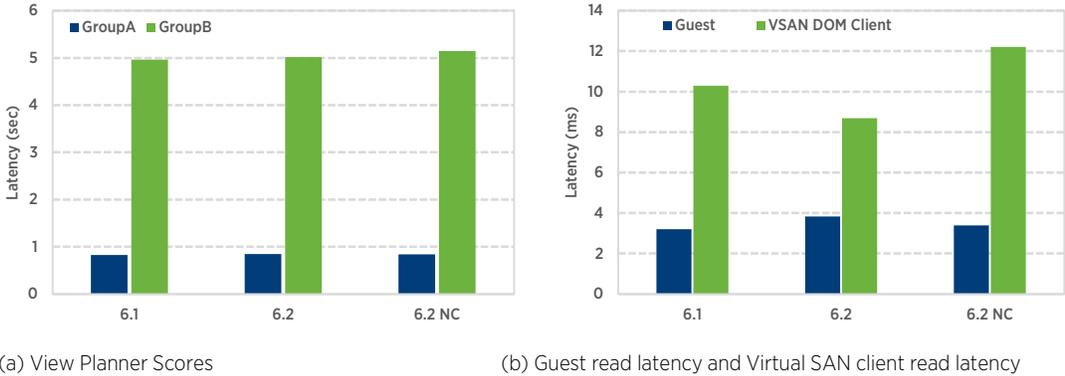


Figure 3. View Planner Scores and read latency comparison for DOM client cache disabled case on hybrid cluster for 500VMs

Performance of All-Flash Virtual SAN Cluster

In this experiment, the VDI workload is exercised on the 4 node all-flash cluster. Different combinations of Virtual SAN 6.2 new features are used in the experiments. We test with 500, 540, 580, and 600 virtual machines in the cluster as the desktop pool size to represent the typical VDI environment that has a high consolidation ratio but does not sacrifice application performance. Virtual SAN 6.2 performs very well in all cases by meeting the View Planner scores requirements for Group A and B applications. Next, we deep-dive to the 580 virtual machine pool size case and discuss the Virtual SAN performance in detail. Each host in the 4-node all-flash cluster has 24 physical CPU cores that have hyper-threading enabled, and the 580 virtual machine pool translates into 6.04 virtual machines per physical CPU, which is a good consolidation ratio for VDI. Every host in the Virtual SAN cluster has 2 disk groups, and each disk group has 1 caching SSD and 3 capacity SSDs. Please note this is not a configuration optimized for achieving lowest cost per VDI desktop virtual machine, but it represents a typical all-flash Virtual SAN setup.

View Planner Scores

Figure 4 shows the detailed results for 580 virtual machines to compare 6.2 new features. Figure 4 (a) shows the View Planner latency scores of Group A and B applications. Virtual SAN 6.2 performs well and meets View Planner requirements. Furthermore, we can see that enabling advanced feature does introduce a small latency overhead. Taking Group B latency as an example, 6.2 R5+D, which has all the Virtual SAN 6.2 new features, displays 5.2 seconds. 6.2 D gives 4.7 seconds and 6.2 R5 drops to 4.5 seconds. 6.2, which only has software checksum, gives 4.4 seconds. The overhead is from the extra I/O operations and computation that Virtual SAN 6.2 has to accomplish in the feature implementation. However, we are going to show that the new feature does have good benefit such as space savings.

CPU Utilization

The system and Virtual SAN CPU utilization in the cluster is shown in Figure 4 (b). The system CPU utilization in the cluster is extremely high and close to 100% for all 6.2 cases. The Virtual SAN CPU usage is very low. In the extreme case of 6.2 R5+D, Virtual SAN uses 1.42% of the CPU in the 4-node cluster. This is equivalent to 2.7 CPU cores out of the 192 CPUs available in the cluster.

Guest I/O per Second

Figure 4 (c) shows the guest IOPs per host. The read IOPs is in the approximate range of 1200 to 1300 and the write IOPs is the range of 900 to 1000. Overall, the IOPs number is similar but with a small variation for different 6.2 cases. The variation of IOPs occurs because the VDI workload driven by View Planner is mimicking a real life VDI environment and has built-in variations such as start an application at randomly generated time and shuffle between the operations. The IOPs per host variation does not indicate performance difference in this particular case.

Disk Space Efficiency

The space efficiency features in Virtual SAN 6.2 reduce the space taken on disk. Figure 4 (d) shows the space usage number in the cluster in gibibytes (GiB) and the savings ratio in percentage. Virtual SAN 6.2, rather than 6.1, is used as the baseline because its space usage is very similar to 6.1 except for a small meta-data overhead⁴. The 6.2 baseline has a RAID-1 mirroring configuration and takes about 326GiB of space in the cluster. 6.2 R5, which is using RAID-5, saves disk space by 33% and only takes 218GiB on disks. When the deduplication and compression feature is enabled, a substantial disk space saving is observed: 6.2 D which uses deduplication and

⁴ Virtual SAN 6.2 default configuration has the software checksum feature, in which there is a 5-byte checksum for a 4KB block. This accounts for the extra meta-data overhead compared with Virtual SAN 6.1.

compression, gives a 90% space saving (32GiB on disk) and 6.2 R5+D, which uses both RAID-5 and deduplication and compression, gives a 93% saving (22GiB on disk).

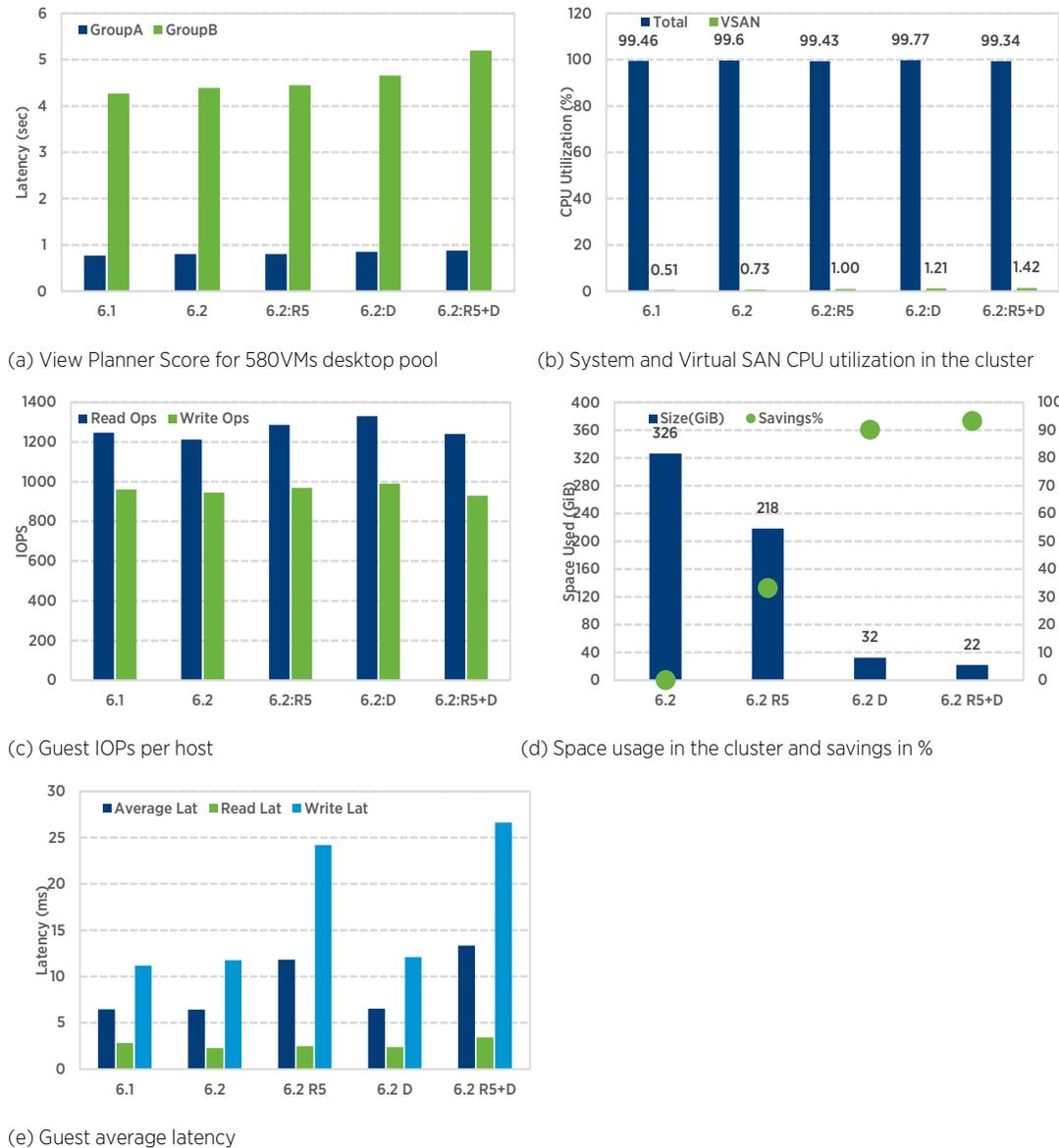


Figure 4. Virtual SAN performance on all-flash cluster for 580VMs desktop pool size: (a) View Planner scores (b) CPU utilization (c) Guest IOPs per host (d) Space usage and saving percentage (e) Guest average latency

The substantial space saving ratio is observed because the workload generator contains duplicable data: all the desktop virtual machines are cloned from a same base virtual machine before the workload starts. As a result these data are highly duplicable. However, once the workload starts, there is more unique data across the desktop virtual machines because that View Planner exercises application actions in the desktop virtual machines in a way to mimic real applications. The actual space saving ratio in a production environment will depend on the workload.

Guest Average Latency

Figure 4 (e) shows the average latency for read, write, and all guest I/Os respectively. Most noticeable, the 6.2 RAID-5 cases, (R5 and R5+D) have a similar latency of around 25ms, while other RAID-1 cases (6.1, 6.2, 6.2 D) have a latency of around 12ms. This doubled average latency is from write I/Os: read latency is low for all cases (2.5ms to 3.5ms), while the RAID-5 write latency is twice high as RAID-1 cases.

The jump of write latency in Figure 4 (e) occurs because a RAID-5 write in Virtual SAN 6.2 has an extra round of reads before issuing the actual write. RAID-5 generates 1 parity from 3 data objects. A write needs to update both the data and the parity. Assuming 4KB sized I/O, 1 guest write in RAID-5 spawns 2 reads because it has to get all the 3 data copies for the parity update. Then Virtual SAN calculates the parity and issues 1 write to persist data from the guest and 1 write to update the parity. In other words, RAID-5 has to wait for an extra round of 2 reads plus a parity calculation before it issues 2 writes compared to RAID-1.

In the current Virtual SAN 6.2 implementation, the 2 reads spawn by the 1 write have to be fetched from another host in the cluster. This adds an extra Round Trip Time (RTT) to the write. This explains why the average write latency in RAID-5 is double of write latency in RAID-1.

Impact of DOM Client Read Cache

Figure 5 plots the View Planner Scores and average read latency of guest and Virtual SAN DOM client. To illustrate the impact the DOM client read cache, we show 3 cases: 6.1, 6.2 R5+D, and 6.2 R5+D NC. All three cases have CBRC. 6.1 does not have DOM client cache feature available, 6.2 R5+D has DOM client cache enabled, and 6.2 R5+D NC disables DOM client cache. We have very similar observation to the Hybrid experiments: in Figure 5 (a) DOM client cache has minimal impact to the View Planner Score; and in Figure 5 (b) guest read latency is low for all 3 cases, but the DOM client read latency is much higher without the client cache. The latency difference is because CBRC is caching guest reads well in this workload. The CBRC hit rate is constantly above 80% on all 4 hosts in the cluster during the steady state while the VDI workload is running. On the other hand, comparing 6.2 R5+D and 6.2 R5+D NC, clearly 6.2 R5+D has a lower read latency of 5.6ms while 6.2 R5+D NC has a higher value of 11.1ms. The lower read latency is achieved because of DOM client cache. The Virtual SAN internal IOPs is also reduced because of the cache.

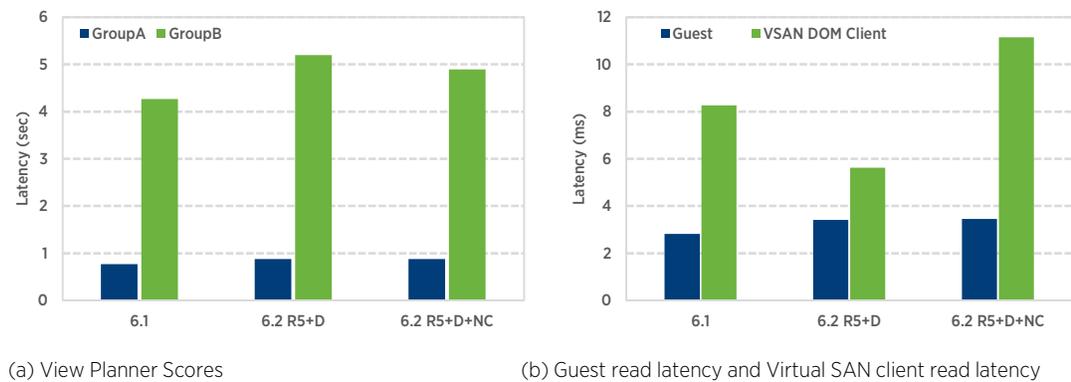


Figure 5. View Planner scores and read latency comparison for DOM client cache disabled case on all-flash cluster for 580VMs

Conclusion

This paper shows the performance results for running a VDI workload on Virtual SAN 6.2. Overall, Virtual SAN 6.2 performs well by meeting the latency requirements for both the group of latency-sensitive applications and I/O-intensive applications. Even when the system CPUs are close to 100% utilization in a large-sized desktop pool, Virtual SAN 6.2 performs just as well as 6.1.

More importantly, tests on the all-flash cluster show the new data integrity and space saving features in Virtual SAN 6.2 have minimal guest latency and CPU consumption overhead. At the same time, these features provide software-based checksum protection and reduce the space usage on disk to lower the storage cost per gigabyte. Lastly, the in-memory DOM cache reduces read latency for I/Os that reach the client layer.

Appendix A. Hardware Configuration for Hybrid Virtual SAN Cluster

The servers were configured as follows:

- Dual-socket Intel® Xeon® CPU E5-2670 v2 @ 2.50GHz system with 40 Hyper-Threaded (HT) cores
- 256GB DDR3 RAM @1866MHz
- 2x LSI / Symbios Logic MegaRAID SAS Fusion Controller with driver version: 6.603.55.00.lvmw, build: 4852043
- 1x 400GB Intel S3700 SATA SSDs
- 4x 900 GB Western Digital WD9001BKHG-02D22 HDDs
- 1x dual-port Intel 10GbE NIC (82599EB, Fibre Optic connector)
- 1x quad-port Broadcom 1GbE NIC (BCM5720)

Appendix B. Hardware Configuration for All-Flash Virtual SAN Cluster

The servers were configured as follows:

- Dual-socket Intel® Xeon® CPU E5-2670 v3 @ 2.30GHz system with 48 Hyper-Threaded (HT) cores
- 256GB DDR3 RAM @1866MHz
- 1x LSI / Symbios Logic MegaRAID SAS Fusion Controller with driver version: 6.603.55.00.lvmw, build: 4852043
- 2x 400GB Intel P3700 PCIe SSDs
- 6x 800 GB Intel S3500 SSDs
- 1x dual-port Intel 10GbE NIC (82599EB, Fibre Optic connector)
- 1x quad-port Broadcom 1GbE NIC (BCM5720)

References

- [1] John Nicholson. (2016, March) VMware Virtual SAN 6.2 Design and Sizing Guide.
<https://www.vmware.com/files/pdf/products/vsan/virtual-san-6.2-design-and-sizing-guide.pdf>
- [2] VMware Inc. (2016) VMware Virtual SAN 6.2 datasheet.
https://www.vmware.com/files/pdf/products/vsan/VMware_Virtual_SAN_Datasheet.pdf
- [3] VMware. (2016, February) What's New with VMware Virtual SAN 6.2 white paper.
<https://www.vmware.com/files/pdf/products/vsan/vmware-virtual-san-6-2-technical-white-paper.pdf>
- [4] VMware Inc. (2016, February) What's New VMware Virtual SAN 6.2.
<https://blogs.vmware.com/virtualblocks/2016/02/10/whats-new-vmware-virtual-san-6-2/>
- [5] VMware Inc. (2015, March) VMware Horizon 6 with View Performance and Best Practices.
<http://www.vmware.com/files/pdf/view/vmware-horizon-view-best-practices-performance-study.pdf>
- [6] VMware Inc. (2012, October) View Storage Accelerator in in VMware View 5.1.
<https://www.vmware.com/files/pdf/techpaper/vmware-view-storage-accelerator-host-caching-content-based-read-cache.pdf>
- [7] VMware, Inc. (2016) View Planner product page. <https://www.vmware.com/products/view-planner>

About the Authors

Zach Shen is a Member of Technical Staff in the I/O Performance Engineering group at VMware and he holds a PhD in Electrical Engineering from the University of Cambridge. **Mehmet Balman** is a Senior Member of Technical Staff in the same group and holds a PhD in Computer Science from Louisiana State University. Their work strives to improve the performance of storage products of VMware.

Acknowledgements

The authors would like to acknowledge Julie Brodeur, Duncan Epping, Christos Karamanolis, Vishal Kher, Eric Knauft, Lenin Singaravelu, Rajesh Somasundaran, Wenguang Wang, and Ruijin Zhou for their valuable contributions to the work presented in this white paper.

