VMware Virtual SAN™ Snapshots in VMware vSphere® 6.0

Performance Study

TECHNICAL WHITE PAPER

vmware®

Table of Contents

Executive Summary

In VMware vSphere® 6.0, snapshots with VMware Virtual SAN™ have been completely redesigned. In this paper, we present the performance of Virtual SAN snapshots for two different applications, namely a source tree compilation workload to roughly mimic "test and development" scenarios, and an online transaction-based database application. Overall, we find that Virtual SAN snapshots in vSphere 6.0 perform reasonably with minimal performance degradation compared to base disk performance. We also address performance tunable configurations and sizing recommendations to optimize performance with Virtual SAN snapshots.

Introduction

Virtual SAN is a distributed layer of software that runs natively as part of vSphere. Virtual SAN aggregates local or direct-attached storage disks in a host cluster and creates a single storage pool that all hosts in the cluster share. This eliminates the need for external, shared storage and simplifies storage configuration and virtual machine provisioning operations. In addition, Virtual SAN supports VMware features that require shared storage such as high availability (HA), vMotion, and distributed resource scheduling (DRS) for failover. You can find more information on Virtual SAN in the *Virtual SAN Design and Sizing Guide* [1].

In this paper, we describe the performance of Virtual SAN snapshots in vSphere 6.0 for two different application workloads: a "test and development" workload (Linux Kernel Compilation) and an online database transaction workload (DVD Store benchmark) [2].

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

Virtual SAN Snapshot Overview

In vSphere 6.0, VMware has redesigned the virtual machine snapshot implementation for Virtual SAN to provide better I/O performance. Unlike vSphere 5.5, creating a snapshot does not create redo logs. Instead, when a virtual machine snapshot is taken, additional "delta" objects get created which are identical to the base objects in the Virtual SAN. There are metadata structures that are local to every snapshot level of a VMDK that contain information about blocks that are newly written at that particular level.

Write I/Os to a snapshot always go to the top-most objects in the snapshot tree. Read I/Os, however, may be serviced from one or more of the layers in the snapshot tree depending on what parts of the data were written in a particular snapshot level. Therefore, when a read I/O request arrives at Virtual SAN, the snapshot logic traverses through the levels of the snapshot tree and composes the information as to which Virtual SAN object and what offset contain the data requested. This addressing information is then cached in the Virtual SAN snapshot metadata cache. The snapshot metadata cache exists in memory, and it is critical to the read performance of snapshots. This is because, for every miss in this cache, the address information has to be obtained by traversing through multiple levels of snapshot, which increases latency. This cache also prefetches the addressing information so that workloads with localized access patterns can benefit. This metadata cache, however, is limited in size and is shared across VMDKs that are open in the system. Therefore, when the cache is full, the least recent address information is flushed out. These considerations are discussed in greater detail in the "vsanSparse – Tech Note for Virtual SAN 6.0 Snapshots" paper [3].

Experimental Setup

Hardware Configuration

Experiments used 8 or 16 hosts depending on the intended size of the Virtual SAN cluster. For the Linux Kernel Compilation workload, each host in the Virtual SAN cluster is a dual-socket Intel® Xeon® E5-2650 v2 @ 2.6 GHz system with 16 cores (32 Hyper-Threads), 128GB memory, a LSI 9207-8i controller. The controller is hosting one 400GB Intel S3700 SSD, and seven 1.1TB, 10,000 RPM SAS drives. For DVD Store experiments, each host in the Virtual SAN cluster is a dual-socket Intel® Xeon® CPU E5-2670 v2 @ 2.50GHz system with 40 Hyper-Threaded (HT) cores, 256GB memory, and three LSI MegaRAID SAS controllers. Each controller hosts one 400GB Intel S3700 SSD, four 1.1 TB 10,000 RPM SAS drives per controller. In both the setups, we configured each node to use a 10GbE port dedicated to Virtual SAN traffic. We connected the 10GbE ports of all the nodes to an Arista 10bE switch. We used a 1GbE port for all management traffic. Appendix A provides more details for the hardware configuration.

ESXi/Virtual SAN Configuration

VMware vSphere 6.0 was used in this study for both DVD Store and Linux Kernel Compilation experiments. Both setups used homogenous hardware and software configurations and stored the entire workload virtual machine on the Virtual SAN datastore. We configured the Virtual SAN datastore with the default policy of HostFailuresToTolerate=1 and StripeWidth=1.

Workload/Virtual Machine Configuration

Linux Kernel Compilation

In order to understand the performance implications of using Virtual SAN for test and development environments, we ran the Linux Kernel Compilation workload on Virtual SAN snapshots. For this workload, the virtual machines each had 2 virtual CPUs (vCPUs), 2GB of memory, a 12GB virtual disk for OS and system, and additional 10GB virtual disk for the tests. We used a 64-bit Ubuntu 12.03 guest operating system. We also used and 8-node and a 16-node Virtual SAN clusters in our experiments. For both configurations, we ran 8 virtual machines per host, totaling 64 virtual machines in an 8-node cluster and 128 virtual machines in a 16-node cluster. Each cluster ran the kernel compilation job in parallel.

DVD Store Benchmark

To study the performance aspects of running a transactional database workload with Virtual SAN snapshots, we used the open-source DVD Store version 2.1 as the benchmark tool. DVD Store simulates an online ecommerce DVD store, where customers log in, browse, and order products. The benchmark tool is designed to use a number of advanced database features, including transactions, stored procedures, triggers, and referential integrity. The primary performance metric of DVD Store is orders per minute (OPM).

The DVD Store benchmark driver outputs a moving average of orders per minute and a cumulative number of transactions every 10 seconds. Because a moving average result would not show fine-grained changes in performance during scenarios like Virtual SAN snapshot create operations, we computed the absolute orders per minute from the cumulative number of transactions that the system printed every 10 seconds.

We encapsulated the entire DVD Store benchmark tools, including the query generator and the database backend in a single virtual machine, which ran the Microsoft Windows Server 2008 R2 operating system and Microsoft SQL Server 2008. We configured the virtual machine with:

- 4 vCPUs and 4GB of memory
- 3 virtual disks (VMDKs):
  - a 50GB boot disk containing Windows Server 2008 R2 and Microsoft SQL Server 2008
  - a 40GB database disk
  - a 10GB log disk

Figure 1 illustrates the experimental setup in detail.

The DVD Store workload used a database size of 15GB with 20 million customers and one million products. In all our DVD Store experiments, 4 virtual machines were used per host, totaling 32 virtual machines in the 8-node cluster and 64 virtual machines in the 16-node cluster. You can find a more detailed workload configuration in Appendix C.

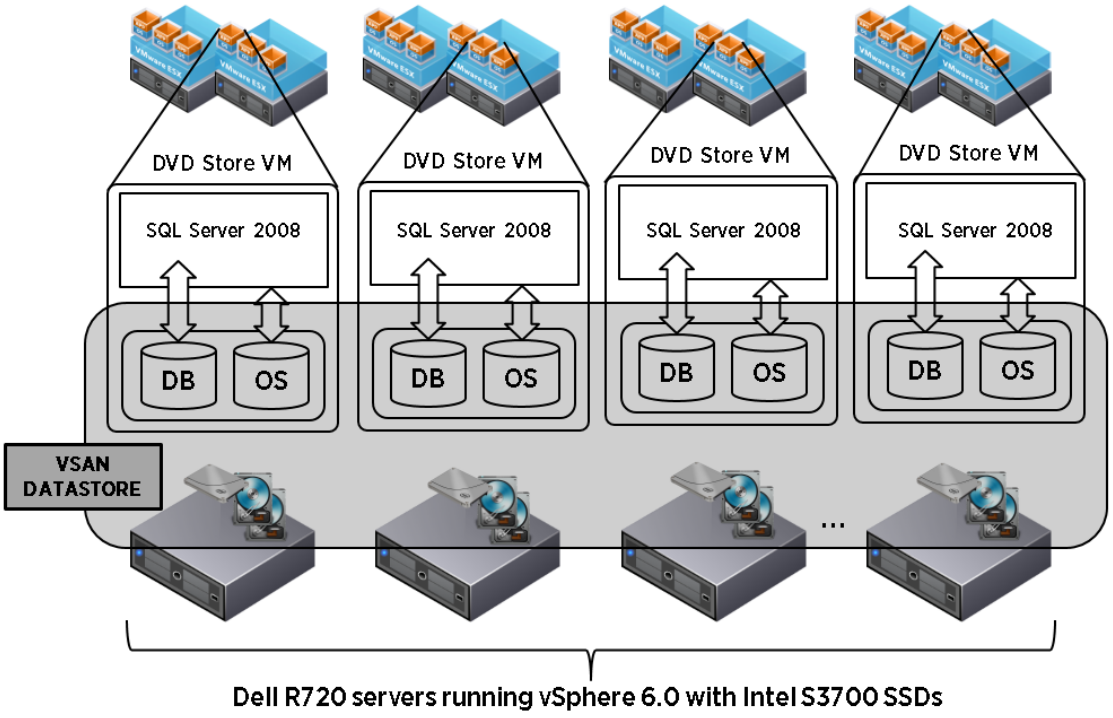


Figure 1. DVD Store Experimental setup

Test Methodology

Linux Kernel Compilation

We ran a Linux Kernel Compilation workload on Virtual SAN snapshots because we wanted to reproduce a "test and development" scenario. In this set of experiments, the workflow ran, in order:

1. A source code sync to a particular version of the Linux Kernel source tree
2. The `make clean` command
3. The `make` command.
4. The `time make` command, in order to gather the application metric "time elapsed" to compile the kernel

We aggregated the "time elapsed" value of each kernel compilation and computed the average time taken to build the particular kernel. The compilation process had a good mix of CPU cycles, read I/Os, and write I/Os.

We wanted to measure the performance of multiple levels of Virtual SAN snapshots, but compiling the same source code for every level resulted in most of the data in the snapshot chain to be available in the base disk. To solve this, for each level of snapshot, we synced the source tree to a slightly different version, which was about a month newer than the source code compiled in the previous snapshot level. This way, part of the source code was modified within the particular snapshot level, leading to reads encompassing multiple levels of snapshots. This is also a more realistic approach to mimic a practical "test and development" environment.

DVD Store Benchmark

In order to evaluate DVD Store benchmark performance with Virtual SAN snapshots, we ran the benchmark for a long duration and created periodic snapshots of the virtual machine without stopping the benchmark program. In our experiments, we took periodic snapshots of the virtual machine every hour and created snapshots 32 levels deep, which is the maximum depth of snapshots vSphere 6.0 supports. Therefore, we ran the DVD Store benchmark for a total of 33 hours where the initial 1-hour pertains to the base disk and subsequent hours each pertain to each level of the 32 snapshots. The DVD Store benchmark application queried the Microsoft SQL server database and provided the "orders per minute" metric as a result. However, we found that over the time period of run, the DVD Store benchmark itself degraded in performance with respect to orders per minute at the rate of about 1.6% per hour. The gradual drop in orders per minute in DVD Store, even with a baseline, is an artifact of the application itself and not an issue with Virtual SAN because we also found the same phenomenon with direct-attached storage exposed to the virtual machine as an RDM disk. Therefore, instead of having a single "average orders per minute" value as a baseline, we ran the benchmark for 33 hours without creating any virtual machine snapshots in order to obtain a baseline for comparison.

Between each of these 33-hour runs, we powered down the virtual machine and flushed out the Virtual SAN read cache. We also restored the entire database from a backup before every run. We took these steps in order to maintain the same initial state for all runs and avoid the impact of database cache inside the virtual machine, Virtual SAN cache state, and so on.

Performance Results and Analysis

This section presents the results of the performance study considering two different application workloads: a "test and development" workload, namely Linux Kernel Compilation, and a transactional database workload, namely DVD Store. Our performance study has two focus areas:

- Virtual SAN snapshot performance compared to base disk performance on Virtual SAN
- Impact of the number of Virtual SAN nodes on Virtual SAN snapshot performance

Linux Kernel Compilation

First, we present the results of testing the performance of the Linux Kernel Compilation workload with Virtual SAN snapshots. Figure 2 shows the degradation percentage comparing the average time spent to build a Linux kernel source on a base disk versus on a virtual machine snapshot in Virtual SAN. We discuss the detailed test methodology for this experiment in the previous section, and we provide the workload configuration in Appendix C. We tested two Virtual SAN cluster sizes of 8 hosts and 16 hosts. The results in the graph show that the overhead introduced by Virtual SAN snapshots is very minimal and even with the maximum of 32 levels of snapshots, the overhead seen is less than 4%.

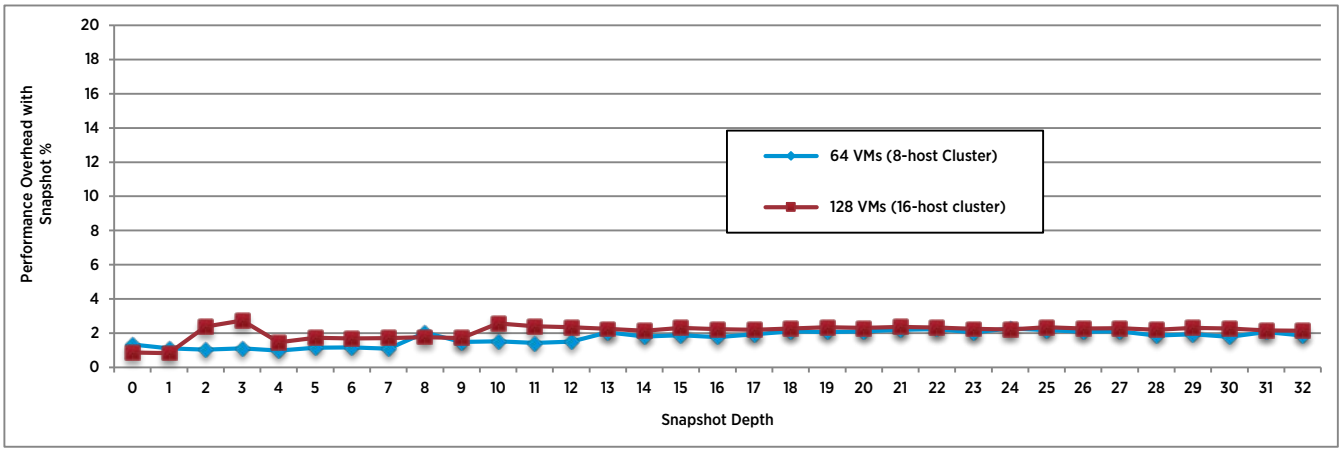


Figure 2. Performance of Kernel Compilation workload on Virtual SAN snapshots

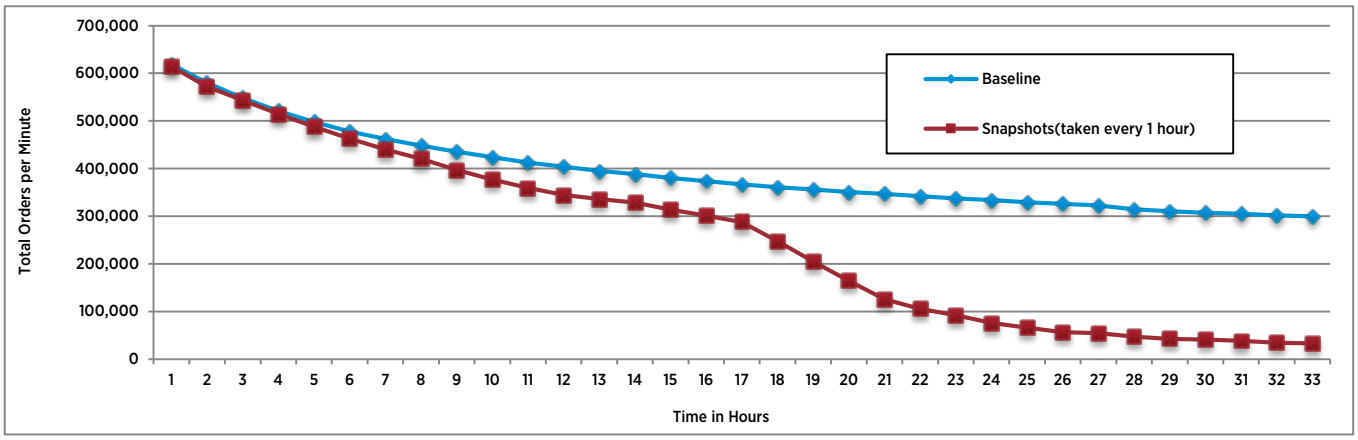
For the Linux Kernel Compilation workload, the overhead is minimal irrespective of the snapshot depth because the working set size of this workload is smaller than the size of the Virtual SAN read cache size and Virtual SAN snapshot metadata cache.

Microsoft SQL Server (DVD Store Benchmark)

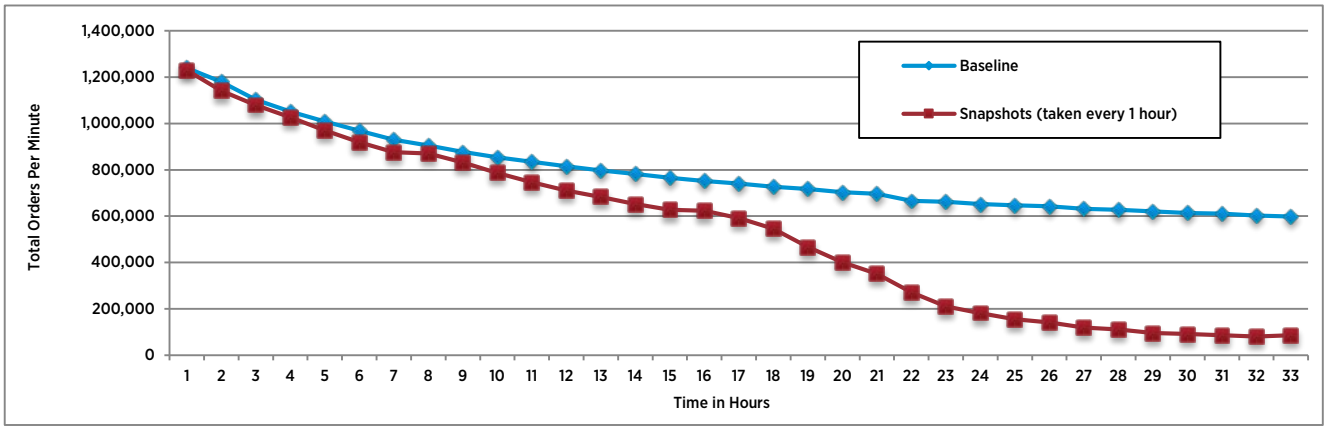
Next, we consider a transactional database application using DVD Store, which operates on a Microsoft SQL Server database. We ran the DVD Store benchmark on two Virtual SAN clusters of 8 nodes and 16 nodes in size and captured the performance for different snapshot depths. Figure 3 (a) and (b) show the total orders per minute of DVD Store for the 8-node and 16-node Virtual SAN clusters. In both configurations, 4 virtual machines were run on every host. Therefore, the total orders per minute value is the sum of "orders per minute" values obtained from 32 DVD Store instances in the case of the 8-node cluster and 64 DVD Store instances in the case of 16-node cluster. The x-axis shows the time (in hours) of running DVD Store. As mentioned in the "Test Methodology" section, "baseline" corresponds to base disk performance and "snapshots" corresponds to the performance of DVD Store when creating virtual machine snapshots at regular intervals of 1 hour. As shown in

the graphs, the size of the Virtual SAN cluster does not impact the performance behavior of the DVD Store workload for increasing the depth of snapshots.

Note: The gradual drop in orders per minute in DVD Store, even with a baseline, is an artifact of the application itself and not an issue with Virtual SAN because we also found the same phenomenon with direct-attached storage exposed to the virtual machine as an RDM disk.



(a) DVDstore on 8-node Virtual SAN cluster



(b) DVDstore on 16-node Virtual SAN cluster

Figure 3. Performance wrt. Virtual SAN cluster size

With respect to the performance of Virtual SAN snapshots compared to base disk performance, we see a steady decrease in performance as the snapshot depth increases by 1 every hour. We observe that performance degradation with Virtual SAN snapshots is approximately 1% for every increasing snapshot depth up to 16 levels of snapshot. Also, as the snapshot depth increases, the performance degradation slightly increases. However, beyond 16 snapshot levels, there is a much steeper reduction in performance and, finally, at 32 levels of snapshots, we see about 85% degradation compared to base disk performance. Virtual SAN metadata cache evictions and read cache space allocation cause the performance degradation. We discuss these issues in the following sections.

Virtual SAN Metadata Cache Evictions

As described in the "Virtual SAN Snapshot Overview" section, Virtual SAN maintains an in-memory cache of the mapping to locate the latest data blocks in the snapshot chain. However, this cache is limited and can contain a maximum of 65,536 entries per VMDK by default. Because Virtual SAN stores the mapping information in terms of extents, workloads with localized access patterns can store more mapping information compared to workloads that are completely random (as in the case of DVD Store). Therefore, as the workload progresses over time, writing to random blocks, the cache eventually becomes full and starts to evict the cache entries. This leads to cache misses for subsequent I/Os leading to gathering this information by scanning through the entire snapshot chain. This increase in latency impacts every I/O that ends up being a cache miss. That is the reason we see an increase in performance degradation with increasing snapshot depths. Figure 4 shows the impact of Virtual SAN metadata cache on DVD Store performance. With the cache sized to hold a maximum of 1 million entries, the performance degradation is less than 5% up to a snapshot depth of 19. However, beyond the depth of 19, the degradation is higher and eventually the DVD Store performance drops by about 56% in the 32nd level. We discuss details about changing the Virtual SAN metadata cache size and the corresponding memory overhead considerations in the section "Performance Tunables and Best Practices." We discuss the reason for a steeper drop in performance from a snapshot depth of 20 in the following section.

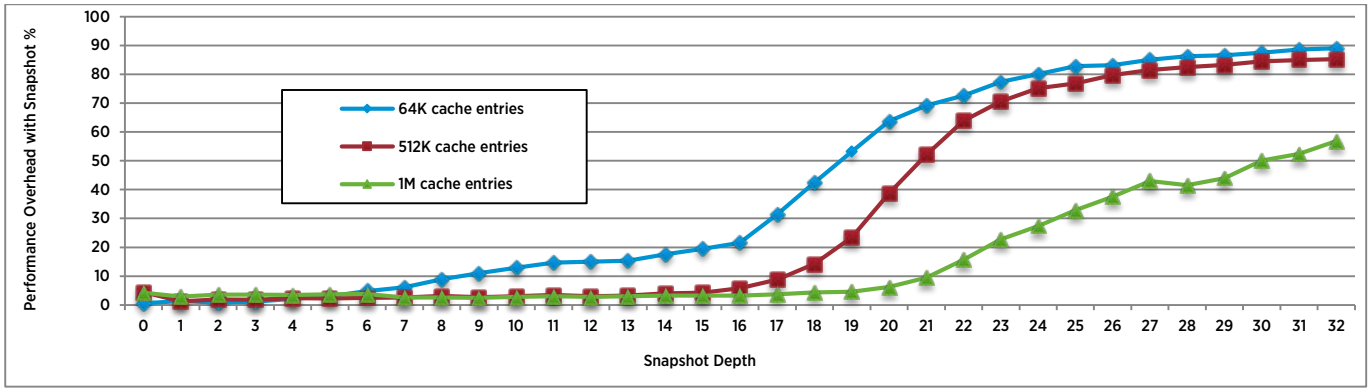


Figure 4. Impact of Virtual SAN metadata cache (DVDstore)

.

Read Cache Space Allocation

Virtual SAN read cache stages recently read blocks of data in the flash storage for faster subsequent retrieval. The Virtual SAN read cache space is shared across Virtual SAN objects. For each virtual machine, a Virtual SAN object is used to hold the "namespace" file system of the virtual machine and an object is used to store each VMDK base disk. Upon creating a virtual machine snapshot, separate Virtual SAN objects are created to store the contents of each delta disk. The benefit of this approach is that the delta disks (for example, for each linked clone

in VMware View) may have a different policy than that of the base disk, as the requirements and workload for each are different. For this reason, as more snapshots are created, the Virtual SAN read cache space per Virtual SAN object gets shrunk because of an increase in the number of Virtual SAN objects. Because the write access patterns of the DVD Store workload are mostly random, and the Virtual SAN read cache block size is 1MB, the already-shrunk space for the objects gets quickly filled up. Figure 5 shows the impact of the Virtual SAN read cache hit rate on snapshot performance.

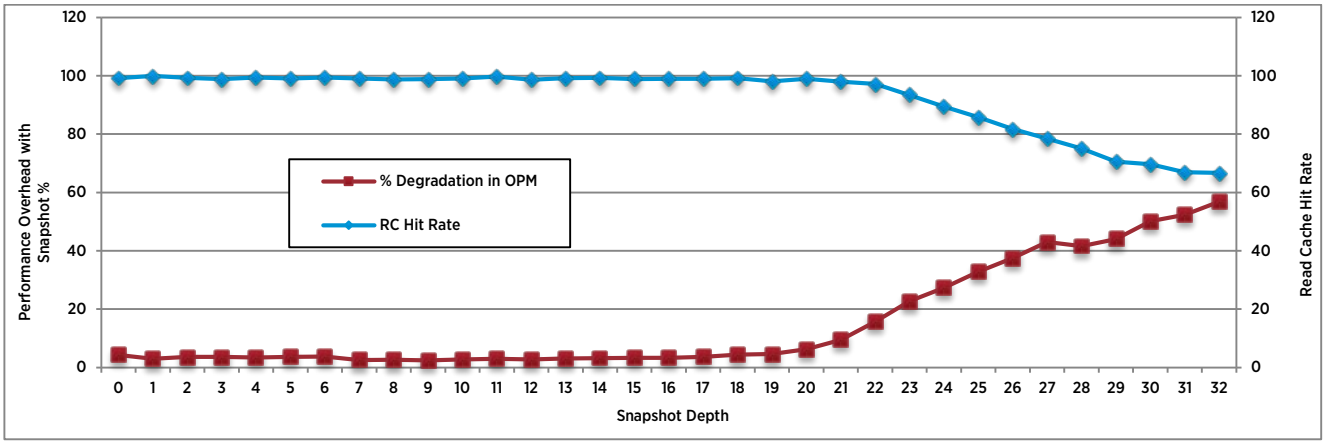


Figure 5. Impact of Read Cache hit rate on snapshot performance

In the experiment shown above, the Virtual SAN read cache hit rate is more than 98% until snapshot level 19, and beyond that the hit rate drops steadily leading to I/Os being served from the magnetic disk layer, increasing latencies. Therefore, we see a corresponding drop in the DVD Store orders per minute. You should carefully size the Virtual SAN read cache according to the working set size of the application and the desired depth of snapshots to have the least overhead. We discuss more details about sizing the read cache in the "Performance Tunables and Best Practices" section, next.

Performance Tunables and Best Practices

In this section, we discuss the effect of some of the tunable configuration parameters in Virtual SAN snapshots along with some high-level sizing guidelines. We deal with three parameters in this section. We provide the exact commands for these tunables in Appendix D.

Virtual SAN Snapshot Metadata Cache

You can configure the size of Virtual SAN metadata cache by setting appropriate values in the `/VSAN/VsanSparseCacheThreshold` parameter through the `esxcli` command in every host participating in the Virtual SAN cluster. `VsanSparseCacheThreshold` is a configuration parameter that sets the maximum number of mapping entries that can be held in the Virtual SAN snapshot metadata cache per VMDK. Increasing this value increases memory usage in return for a better application performance with snapshots. Every entry in the Virtual SAN snapshot metadata cache consumes 64 bytes of memory and the `VsanSparseCacheThreshold` parameter contains the maximum number of such entries. By default, the `VsanSparseCacheThreshold` value is set to 65,536. This means when data fills all entries in the cache, the cache consumes 4MB of memory.

Because the mappings in the cache are extent-based, the actual address space mapped by the entries in this cache depends on the workload access patterns. Assuming completely random write patterns, after a virtual

machine snapshot is taken, initially the number of entries stored in the cache can be high because of potentially small extent sizes. However, as the workload progresses by writing to more locations over time, the extents can get merged and the number of entries in the cache also gets consolidated. The default value of 65,536 works well for most of the common case scenarios, but you may increase this value when you detect low hit rates. You can find the hit rates of the Virtual SAN metadata in the vsanSparse tab, in Observer.

Virtual SAN Snapshot Metadata Cache Prefetch Size

The Virtual SAN snapshot metadata cache also does prefetching of the addressing information so that subsequent I/O requests can already find the mapping information in the cache. There is a configurable parameter to specify the size of the pre-fetch operation and you can set it in `/VSAN/VsanSparseSpeculativePrefetch` through the `esxcli` command in every host participating in the Virtual SAN cluster. The default value for this parameter is 4,194,304 bytes or 4MB. This parameter decides the size of the data region for which the extent information should be prefetched on every request for a particular extent information. There is a parameter named `VsanSparseMaxExtentsPrefetch` that caps the maximum number of extents to prefetch even if the value specified in `VsanSparseSpeculativePrefetch` includes a higher number of extents. The default value of this parameter is 64, and you can increase this value up to 128

The benefits of prefetching highly depend on the workload access patterns, and we recommend increasing this parameter only in cases when the workload has local access. In cases where the workload locality is high, but the data is fragmented leading to more extents, you can increase the `VsanSparseMaxExtentsPrefetch` value. For completely random workloads, increasing this value might only result in evicting more important information from the cache. Setting `VsanSparseSpeculativePrefetch` to 0 might give better performance in this case.

Virtual SAN Read Cache Sizing

The read cache is shared among the Virtual SAN objects. When creating a virtual machine snapshot in Virtual SAN, new objects are created for each VMDK present in the virtual machine and this increases the number of objects in a Virtual SAN node. As the number of objects increase, the space allotted to individual objects decreases, leading to contention. Therefore, you should determine the flash storage capacity for Virtual SAN read cache based on two factors:

- The active working set size of the workload
- The expected number of snapshot levels and the amount of writes that happen between two successive snapshot levels

In addition, the Virtual SAN read cache block size is in 1MB granularity and, therefore, if the workload is completely random, even smaller I/O sizes can take up a substantial amount of space in the read cache due to an internal fragmentation of data. However, in steady state, when more random writes are made, the internal fragmentation becomes less problematic and the space utilization becomes better.

Space Over-Commitment for Virtual SAN Snapshots

With Virtual SAN in vSphere 6.0, allocation for writes happens in 4MB granularities. Therefore, after creating a snapshot, subsequent writes to the virtual machine disks result in new allocations in the corresponding Virtual SAN objects of 4MB granularities. As a result of this, depending on the access pattern of the workload, datastore space usage can quickly increase. For example, if the workload does completely random small writes, this could result in many 4MB allocations, leading to disparity between the actual amount of data written versus the amount of space allocated for it.

One such example is DVD Store. In our experiments, every DVD Store virtual machine contained a 40GB VMDK for the database. Within the first 10 minutes after taking a virtual machine snapshot, the additional space allocation for the snapshot objects was about 94% of the base VMDK size because the workload had about 45% writes with completely random access patterns. For such workloads, you should plan for 1x the capacity-overcommit per snapshot level. However, in such a random workload case, when the workload stabilizes over time, the internal fragmentation in the write allocations gradually decrease and eventually most of the 4MB regions fill up.

On the other hand, in the case of Linux kernel compilation experiments, we used 10GB VMDKs. Upon creating a snapshot, the increase in space allocations for the snapshot object is linear as the compilation progresses. The total time taken for a compilation job was around 2400 seconds on an average across virtual machines. After completion of the compilation process, the total space consumed was around 6GB, requiring a capacity-overcommit of 0.6x per snapshot level. When committing capacity during snapshots, you should consider the access patterns and the actual write-footprint of the workload over time.

Conclusion

In this paper, we presented the results of running two different workloads: a test and development workload using parallel compilation of the Linux Kernel, and a transactional database workload using DVD Store. We analyzed the results obtained and discussed the factors impacting performance, along with a brief summary of the few performance tunables and sizing guidelines to obtain better performance in contentious scenarios.

Overall, we find that snapshots in Virtual SAN perform close to base disk performance for the Linux Kernel Compilation workload. While the DVD Store workload is not particularly cache-friendly because of its completely random access patterns, we showed how the performance can be improved by tuning some Virtual SAN parameters and by following optimal sizing recommendations.

Appendix A. Hardware Configuration

Our servers had the following configuration.

- Dual-socket Intel® Xeon® CPU E5-2670 v2 @ 2.50GHz system with 40 Hyper-Threaded (HT) cores
- 256GB DDR3 RAM @1866MHz
- 3x LSI / Symbios Logic MegaRAID SAS Fusion Controller with driver version: 6.603.55.00.1vmw, build: 4852043
- 3x Intel S3700 SSDs
- 12x SMC 2208 HDDs
- 1x Dual-Port Intel 10GbE NIC (82599EB, fibre optic connector)
- 1x Quad-Port Broadcom 1GbE NIC (BCM5720)

Appendix B. Virtual Machine Configuration

The DVD Store virtual machines were homogenous and had the following configuration:

- 64-bit Microsoft Windows Server 2008 R2
- VMXNET3 driver version 1.1.30.0, PVSCSI driver version 1.1.1.0
- 50GB disk (for the operating system) on LSI Logic controller
- 40GB database disk and 10GB log disk, both on PVSCSI controller
- Microsoft SQL Server 2008

The Linux Kernel Compilation workload, the virtual machines used had the following configuration:

- 64-bit Ubuntu 12.03 version
- VMXNET3 driver version 1.1.30.0, PVSCSI driver version 1.1.1.0
- 12GB disk with operating system on LSI Logic controller
- 10GB disk that contains the kernel source connected via PVSCSI controller

Appendix C. Workload Configuration

The DVD Store benchmark was configured as follows for all the experiments:

```
target=localhost
n_threads=8
ramp_rate=100
run_time=1485
db_size=15GB
warmup_time=2
think_time=0.002
pct_newcustomers=0
n_searches=15
search_batch_size=15
n_line_items=15
virt_dir=ds2
page_type=php
windows_perf_host=
linux_perf_host=
detailed_view=n
```

For Linux Kernel Compilation, the following kernel versions were run for different levels of snapshots:

0: 'v3.6.4',
1: 'v3.6.8',
2: 'v3.7',
3: 'v3.7.4',
4: 'v3.7.8',
5: 'v3.8',
6: 'v3.8.4',
7: 'v3.8.8',
8: 'v3.9',
9: 'v3.9.4',
10: 'v3.9.8',
11: 'v3.10',
12: 'v3.10.4',
13: 'v3.10.8',
14: 'v3.11',
15: 'v3.11.4',
16: 'v3.11.8',
17: 'v3.12',
18: 'v3.12.4',
19: 'v3.12.8',
20: 'v3.13',
21: 'v3.13.4',
22: 'v3.13.8',
23: 'v3.14',
24: 'v3.14.4',
25: 'v3.14.8',
26: 'v3.15',
27: 'v3.15.4',
28: 'v3.15.8',
29: 'v3.16',
30: 'v3.16.4',
31: 'v3.16.8',
32: 'v3.17'

Appendix D: Performance Tunable Commands

Virtual SAN Snapshot Metadata Cache Size

The unit of this parameter is the number of extent entries. By default, the value is 65,536. To set the parameter to a different value, run

```
esxcli system settings advanced set -o /VSAN/VsanSparseCacheThreshold -i 1048576
```

Virtual SAN Snapshot Metadata Cache Prefetch Parameters

As described in the "Performance Tunables and Best Practices" section, two parameters decide the amount of data to prefetch. The unit of `VsanSparseSpeculativePrefetch` is in bytes and the unit of `VsanSparseMaxExtentsPrefetch` is in number of extents.

The default value of `VsanSparseSpeculativePrefetch` is 4,194,304 and the default value of `VsanSparseMaxExtentsPrefetch` is 64. To set a different value, run

```
esxcli system settings advanced set -o /VSAN/VsanSparseSpeculativePrefetch -i 16777216
esxcli system settings advanced set -o /VSAN/VsanSparseMaxExtentsPrefetch -i 128
```

References

[1] Cormac Hogan, VMware, Inc. (2015, March) VMware Virtual SAN 6.0 Design and Sizing Guide. http://www.vmware.com/files/pdf/products/vsan/VSAN_Design_and_Sizing_Guide.pdf

[2] Todd Muirhead and Dave Jaffe. DVD Store 2.1. https://github.com/dvdstore

[3] Cormac Hogan, VMware, Inc. (2015, March) vsanSparse--Tech Note for Virtual SAN 6.0 Snapshots. https://www.vmware.com/files/pdf/products/vsan/Tech-Notes-Virtual-San6-Snapshots.pdf

About the Author

Dr. Sankaran Sivathanu is a staff engineer in the VMware Performance Engineering team. His work focuses on the performance aspects of the ESXi storage stack and characterization/modeling of new and emerging I/O workloads. He has a PhD in Computer Science from the Georgia Institute of Technology.

Acknowledgements

The author would like to thank Shilpi Agarwal, Mark Sterin, Julie Brodeur, Chuck Hollis, Christos Karamanolis, Lenin Singaravelu, Amitabha Banerjee, Todd Muirhead and Duncan Epping for their valuable feedback and reviews for this paper.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com