WHITE PAPER

Ushering in a New Era of Hyper-Converged Big Data Using Hadoop* with All-Flash VMware® vSAN™

vmware®

Table of Contents

Executive Summary 1
1. Introduction 2
2. vSAN Cluster Configuration 2
2.1 Test Environment 2
2.2 Hadoop Configuration. 3
2.3 Parameter Tuning. 4
2.4 Workload. 4
2.5 Virtual Machine Configuration 4
2.6 vSAN Configuration Parameters 5
3. Performance Results 5
3.1 Number of Disk Groups. 6
3.2 Stripe Width. 7
3.3 Failures to Tolerate (FTT). . . 7
3.4 Host Affinity (Tech Preview) 8
4. Conclusion. 8
Appendix A. System Performance Tuning 9
BIOS Tunings. 9
Virtual Machine OS Performance Tuning 9
References. 10

Executive Summary

Since its introduction, VMware vSAN has been adopted by more than 5,500 customers. One of the main reasons for that success has been vSAN's ability to eliminate infrastructure silos. Customers are able to run most workloads at scale on top of vSAN, and reference architectures exist for the majority of business-critical apps, such as Microsoft SQL Server*, Oracle Database, and Microsoft Exchange*, including their clustering technologies.

Currently, the majority of big-data workloads are deployed in silos as well. Some of the major challenges in adopting and utilizing big data are related to infrastructure. Security, scalability, manageability, and performance are mentioned by survey respondents as top pain points, as shown in Figure1:[1]

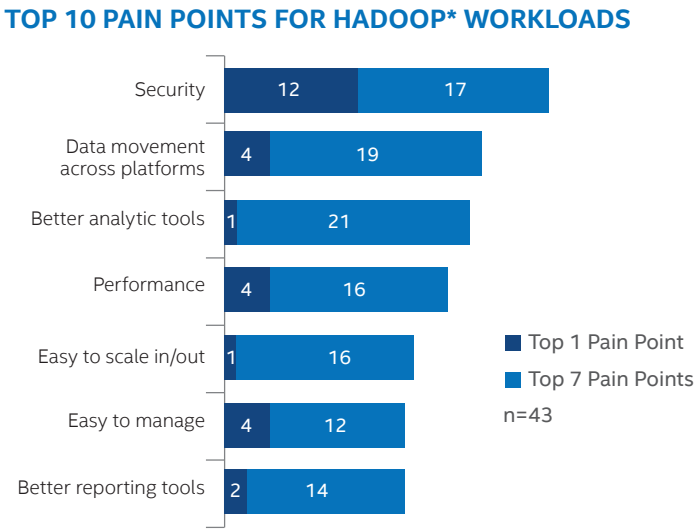

Figure 1. Some major customer challenges when adopting big data.

Many customers have asked for a vSAN big-data reference architecture. Their main objectives are to utilize one unified platform for all their workloads, to avoid silos, and to take advantage of the security, scalability, and manageability that the VMware vSphere® hypervisor offers.

The adoption of all-flash vSAN has significantly increased. There are three primary reasons for that growth among vSAN customers in commercial and enterprise market segments:

- Flash devices have grown in capacity and performance
- Prices of flash devices have decreased tremendously
- Customers want to future proof their infrastructure

This paper is a close collaboration between VMware and Intel to provide a reference architecture and details on deployment, design principles, best practices, and also a tech preview of what is in development for future enhancements. The goal of this document is to address current big-data infrastructure challenges in terms of security, scalability, manageability, and performance.

1. Introduction

Big data is a key enabler for customers to gain business insights, playing a significant role in key business areas such as IT, marketing, finance, security and compliance, and business operations.

vSAN is a software-defined storage solution built into vSphere, which runs on standard Intel® architecture-based servers. It pools together storage capacity into a single shared datastore across all hosts of the cluster. This capability eliminates the need for external shared storage and simplifies storage configuration and virtual machine (VM) provisioning operations. More information on vSAN can be obtained from the vSAN 6.2 Design and Sizing Guide (see References section, item [2]).

Combining Hadoop* with vSAN is appealing to IT organizations, as hyper-converged solutions make efficient use of resources for compute and storage workloads, and managing the Hadoop infrastructure through vSphere simplifies management. In addition, using one large storage pool with a single namespace enables multiple applications and workloads to contribute to the same datastore and make use of it for analytics, deriving additional insights for the business.

Enterprise and service providers are switching to all-flash server-based storage to make real-time decisions and drive differentiation using big data. Non Volatile Memory Express* (also known as NVMe* or NVM Express*), is a device-interface specification for accessing non-volatile storage media attached via the PCI Express* (PCIe*) bus. The NVMe interface is designed to take advantage of the low latency and internal parallelism of flash, while connecting directly to Intel® Xeon® processors, overcoming SAS/SATA SSD performance limitations. 3D NAND NVMe flash storage is becoming increasingly appealing for big-data workloads as densities continue to increase and prices continue to fall. Joining Intel Xeon processors, VMware vSAN, and the Intel® SSD Data Center (DC) Family using the NVMe interface can help reduce latency significantly and provide higher levels of IOPS.

This reference architecture showcases the feasibility of running Hadoop with vSAN in a hyper-converged infrastructure on Intel Xeon processor-based servers. It demonstrates the impact of multiple vSAN configuration parameters on Hadoop performance, including the following:

- Number of Disk Groups
- Stripe Width
- Failure To Tolerate (the number of node failures that a vSAN cluster can tolerate without data loss)

This paper also showcases a new vSAN feature called Host Affinity (currently in tech preview), which helps in making Hadoop clusters implemented on virtualized infrastructure fully aware of the topology on which they are running using vSAN.

2. vSAN Cluster Configuration

2.1 Test Environment

The vSAN and Hadoop testing environment consists of eight Intel® Server Systems, each equipped with the Intel® Server Board S2600WTTR (http://www.intelserveredge.com/intel-cloud-block-vSAN). Each system is two rack units (3.5") high. The first system is used to run the vCenter Server Appliance, which provides easy deployment of vSAN clusters. The other seven systems are configured as shown in Table 1.

Table 1. Server node configuration.

COMPONENT	QUANTITY/TYPE
Processor	2x Intel® Xeon® processors E5-2690 v4 2.6GHz with 14 cores each
Logical Processors (including Intel® Hyper-Threading Technology)	56
Memory	256 GB (16 x 16 GB) DDR4 2133 MT/s
Network	2x Intel® Ethernet Converged Network Adapters X520-DA2 (2x 10 Gbps)
Caching Storage	4x Intel® SSD Data Center P3700 Series 800 GB (NVMe*)
Capacity Storage	14x Intel® SSD Data Center S3610 Series 1.6 TB (SATA)
RAID Controller	2x LSI*-3008-8i 12 Gbps
Network Switch	Arista 7150S-64SFP+ Switch

Note: All the experiments shown in this paper are conducted using the ESXi 6.0 update 2 release. ESXi 6.0 update 3 has been released recently and is the latest and recommended version.

ESXi 6.0 update 2 is installed on each of the eight hosts (# 1-8 illustrated in Figure 2) on a separate Intel® SSD DC S3610 Series 400 GB. This SSD is used as a virtual machine file system (VMFS) datastore upon which the OS disk ISOs for the VMs on that host are stored. A single vSAN datastore is created by aggregating all of the underlying disks on each individual host (except the 400 GB OS disk). Intel SSDs DC S3610 Series 1.6 TB act as the capacity tier for vSAN, and Intel® SSDs DC P3700 Series 800 GB act as the caching tier. Intel® Hyper-Threading Technology is enabled on the servers, so the 28 physical cores are seen by the OS as 56 logical processors.

The servers are networked to an Arista 10 GbE switch. Three different networks are used for management: networking (vSwitch0), vSAN network (vSwitch1), and VM network (vSwitch2), as shown in Figure 2.

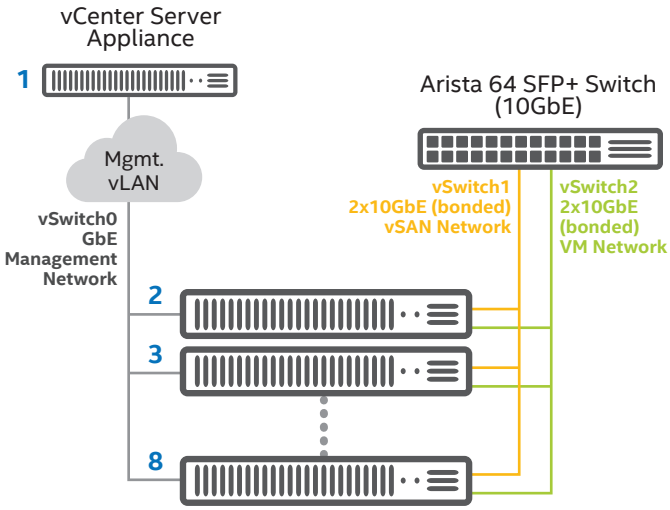

Figure 2. Eight Intel® Server Board S2600WTTR Systems used for testing.

vSwitch0 is used as a management network for managing all the hosts through vCenter appliance server (VCSA) as shown in Figure 2. This same switch is also used for vMotion® networking. vSwitch1 is created using two 10 GbE NIC ports, which are bonded together using route-based IP hash load balancing policy. The maximum transmission unit (MTU) of the bonded 10 GbE NIC is set to 9000 bytes to handle jumbo Ethernet frames. This switch is used for vSAN network traffic between nodes in the clusters. Similarly, vSwitch2 is created using two separate 10 GbE NIC ports that are also bonded together using route based on originating virtual port load balancing policy. This switch acts as a network for VM traffic. The MTU of the bonded 10 GbE NIC is set to 9000 bytes.

Note: The guest VM NIC interface MTU should also be set to 9000.

Four VMs are created on each physical host, for a total of 28 VMs in the cluster. Each VM consists of four 400 GB VMDKs, which are created from a single aggregated vSAN datastore. Each VM's OS disk (40 GB thin provision) also resides on the same vSAN datastore. One additional VM (29th VM) is created from the VCSA host (#1 in Figure 2). This VM is running Cloudera* Manager and is also used for running Cloudera management services.

2.2 Hadoop Configuration

The Cloudera CDH 5.7.0 distribution of Apache* Hadoop is installed in all VMs. This distribution supports both versions 1 and 2 of MapReduce. Version 1 of MapReduce (MR2) is used in this cluster. The Hadoop cluster consists of three types of nodes (gateway node, master node, and data node), which are used as shown in Figure 3.

One gateway node acts as client systems for Hadoop applications and provides a remote access point for users of cluster applications. The master node runs the Hadoop master services such as NameNode, ResourceManager, and their associated services (JobHistory Server, etc). 27 Data nodes/Worker nodes (1-27) run the resource-intensive Hadoop File System (HDFS) DataNode role and the YARN NodeManager role, which is responsible for monitoring container resources and reporting back to the resource manager. Data node 1 also runs as a secondary name node service. ZooKeeper* services support NameNode and ResourceManager high availability from three nodes: one gateway, one master, and one data node. Table 2 shows all of the different roles used in the cluster.

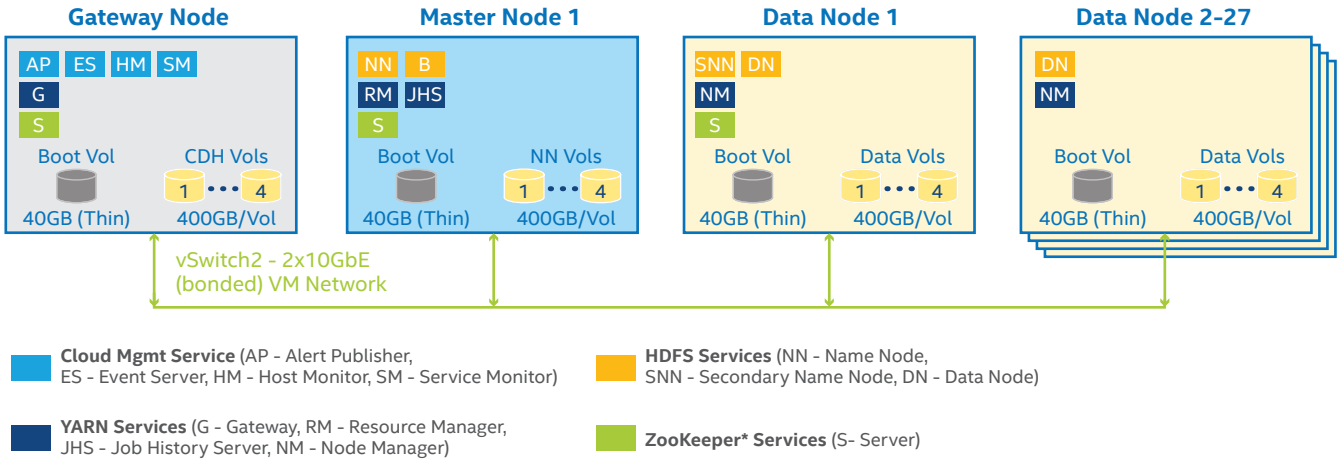

Figure 3. Hadoop* cluster configuration.

Table 2. Hadoop* roles.

NODE	ROLES
Gateway node	Cloudera* Manager, ZooKeeper* Server, HDFS Gateway, YARN Gateway, Alert Publisher, Event Server, Host Monitor, Service Monitor
Master node	HDFS NameNode, YARN ResourceManager, ZooKeeper Server, HDFS Balancer, Job History Service
Data node 1	HDFS secondary Namenode, HDFS Datanode, ZooKeeper Server, Yarn Node Manager
Data node (2-27)	HDFS Datanode, Yarn Node Manager

2.3 Parameter Tuning

HDFS block size is increased from the 128MiB default to 256MiB. The larger block size is chosen because it increases application efficiency by creating fewer but longer-running Hadoop tasks. The trade-off is that a larger block size needs more memory and may make balancing the workload across a large cluster more difficult for small datasets (see References section, item [1]). Other Hadoop parameters are changed from their defaults in order to further increase efficiency, as listed in Table 3.

Table 3. Hadoop* parameter tuning.

PARAMETER	VALUE
dfs.blocksize	256 MiB
dfs.client.use.datanode.hostname	TRUE
mapreduce.task.io.sort.mb	400 MiB
yarn.scheduler.minimum-allocation-mb	2 GiB
mapreduce.map.memory.mb	2.1 GiB
mapreduce.reduce.memory.mb	2.1 GiB
mapreduce.map.cpu.vcores	1
mapreduce.reduce.cpu.vcores	1
mapreduce.job.heap.memory-mb.ratio	0.8

2.4 Workload

TeraSort* is a built-in Hadoop distribution benchmark that is a good performance analog for an ETL operation. The "all to all" data manipulation and transform requires all aspects of the Hadoop cluster's resources to be involved: CPU, network, and disk.

The TeraSort benchmark is used to evaluate the data restructuring capability and performance of the cluster. It is a MapReduce program that executes within the Hadoop framework. It operates on a very large text-based dataset of records. Each record occupies 100 bytes and contains a string of random numbers and letters. The dataset is distributed in blocks across the individual nodes in the Hadoop cluster.

When the benchmark is invoked, the sort program executes a "map" and a "reduce" phase. The map phase executes independently on each node on separate data blocks, sorting the keys within the block. The results of the sort operation running in the map phase are written to intermediate data blocks in HDFS, which are then processed during the reduce phase. The reducer tasks run and consume the sorted output written in the map phase, creating a final sorted output of all keys. The TeraSort benchmark is extremely I/O-intensive, as all data in the randomized dataset is accessed and transformed into a sorted output. It utilizes disk, CPU, and network resources across the nodes.

This benchmark consists of three different phases of execution:

1. **TeraGen** is a 100-percent write workload that generates the user-requested data needed for TeraSort operation. It writes data directly to HDFS.
2. **TeraSort** is the operation where actual sorting of the generated data from TeraGen occurs. It is a combination of map and reduce phases, starting with the CPU-intensive map sort, with disk writes occurring as sorted data is spilled to disk. After that comes a shuffle phase, where data is transferred from maps to reduces. During this phase, the network I/O occurs. In the reduce copy phase of TeraSort, disk I/O occurs when sorted data gets written to HDFS by the reducers.
3. **TeraValidate** is a high-disk-read, low-CPU-operation phase, the purpose of which is to validate the results generated during the TeraSort phase (see Reference section, item [4]).

The TeraSort benchmark is shipped with the Apache and Cloudera distributions of Hadoop. The work in this paper uses the Cloudera distribution of Hadoop, along with running Cloudera's TeraSort benchmark suite.

2.5 Virtual Machine Configuration

Four VMs are created on each of the seven physical hosts, for a total of 28 VMs. The 56 virtual CPUs on each host (corresponding to the 56 logical processors with Intel Hyper-Threading Technology enabled) are uniformly spread across four VMs with 14 cores each. One important consideration to note is the memory size on the VMs. Out of 256 GiB available on each host, 192 GiB of memory is allocated and spread equally across all four VMs, and the remaining 64 GiB is left for system use.

The VM is configured with two vSCSI controllers: one (LSI Logic*) for the OS disk (40 GiB thin provision) and the other one (LSI Logic) is used by four data disks (VMDKs) of 400 GiB each. All of the data drives are formatted with the XFS* file system. Exact XFS mount parameters can be found in "Appendix A: System Performance Tuning." All of the VMDKs come from a single aggregated all-flash vSAN datastore. All of these data drives act as a source of HDFS data I/O. Tables 4 and 5 provide more details on VM configuration; each VM is configured the same way.

Table 4. Virtual machine configuration.

COMPONENT	VALUE
Virtual CPUs	14
Memory	48 GiB (192 GiB / 4 VMs)
Virtual NICs	Bonded 2x 10 GbE NICs – ESXi bonding
Disk Drives	1x 40 GB OS disk, 4x 400 GB data disks

The software configuration of each VM is shown in Table 5. The Cloudera Distribution of Hadoop (CDH) 5.7.0 along with CentOS* 6.7 is used. The Oracle Java* Development Kit 1.8.0_60 is installed on each node and then employed by CDH.

Table 5. Virtual machine software configuration.

COMPONENT	VALUE
OS	CentOS* 6.7 (64 bit)
Cloudera* Distribution of Hadoop*	5.7.0
Cloudera Manager	5.7.0
Hadoop*	2.6.0+cdh5.7.0+1335
HDFS	2.6.0+cdh5.7.0+1335
YARN	2.6.0+cdh5.7.0+1335
MR2	2.6.0+cdh5.7.0+1335

2.6 vSAN Configuration Parameters

The following section showcases performance results while varying the following vSAN configuration parameters:

- Number of disk groups
- Stripe width
- Failures to tolerate (FTT), which is the number of host failures that a vSAN cluster can tolerate without data loss
- vSAN Host Affinity (tech preview)

3. Performance Results

The Hadoop TeraSort performance benchmark is run on the gateway node, which acts as a gateway for Hadoop client applications. Time (in minutes) is the key performance metric used to compare different configurations.

The TeraSort benchmark is run after doing the following for each test case:

1. Clear any prior test data in HDFS.

2. Purge HDFS trash files.

Table 6 shows the list of commands run to capture TeraSort results for 1 TB, 2 TB, 3 TB, and 5 TB input sizes.

Table 6. TeraSort* commands run for 1 TB, 2 TB, 3 TB, and 5 TB input sizes.

COMPONENT
sudo -u hdfs hadoop fs –expunge sleep 2 hadoop fs -rm -r -skipTrash /user/root/tera* sudo -u hdfs hadoop fs –expunge sleep 20 sudo -u hdfs hadoop fs -rm -r -skipTrash /user/hdfs/.Trash/* sudo -u hdfs hadoop fs -rm -r -skipTrash /user/root/.Trash/* sleep 5 time hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar teragen -Dmapreduce.job.maps=377 -Dmapreduce.job.reduces=377 -Dmapred.map.tasks=377 -Dmapred.reduce.tasks=377 $teragensize /user/root/terasort-out1 time hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar terasort -Dmapred.map.tasks=377 -Dmapred.reduce.tasks=377 /user/root/terasort-out1/ /user/root/terasort-sorted/ time hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar teravalidate /user/root/terasort-sorted/ / user/root/terasort-validated-test1/ $teragensize was varied: 10000000000, 20000000000, 30000000000 and 50000000000 for different test runs

3.1 Number of Disk Groups

This experiment involves varying the number of disk groups from two to four on each of the seven physical hosts when building the vSAN cluster, to determine the optimally performing disk group number. It is important to note that while adjusting the number of disk-groups, size of the capacity tier remained constant.

The two-disk-group configuration consists of one Intel SSD DC P3700 Series 800 GB for caching, along with seven Intel SSDs DC S3610 Series 1.6 TB for capacity in each disk group. The three-disk-group configuration consists of each disk group having one Intel SSD DC P3700 Series 800 GB for caching, along with four Intel SSDs DC S3610 Series 1.6 TB for capacity. In the four-disk-group configuration, each disk group consists of one Intel SSD DC P3700 Series 800 GB for caching, along with three Intel SSDs DC S3610 Series 1.6 TB for capacity. These configurations are shown in Table 7.

This testing reveals that more disk groups help to spread I/O for faster response times. It should be noted that when creating more disk groups, more caching disks are needed for each individual vSAN node, which in turn will incur higher cost to the overall cluster, but at the same time help in increasing overall performance of the cluster.

Figure 4 shows the benefit of more disk groups when running Hadoop TeraSort.

As shown in Figure 4, with four disk groups per physical node, elapsed time is reduced from 84 minutes to 58.4 minutes for a 3 TB TeraSort operation, which translates to up to 30-percent performance improvement on the four disk group configuration when compared against the two disk group configuration. Having more disk groups helps performance, as the four-disk-group test achieves the lowest elapsed time compared to the two- and three-disk-group results when running the TeraSort operation. The four-disk-group configuration is therefore recommended.

Table 7. Configurations for various disk groups.

TEST CASES	CACHING DISKS	CAPACITY DISKS	CACHE-CAPACITY RATIO PER DISK GROUP
2 Disk Groups	2x Intel® SSD Data Center (DC) P3700 Series 800 GB	14x Intel® SSD Data Center (DC) S3610 Series 1.6 TB	1:7
3 Disk Groups	3x Intel SSD DC P3700 Series 800 GB	12x Intel SSD DC S3610 Series 1.6 TB	1:4
4 Disk Groups	4x Intel SSD DC P3700 Series 800 GB	12x Intel SSD DC S3610 Series 1.6 TB	1:3

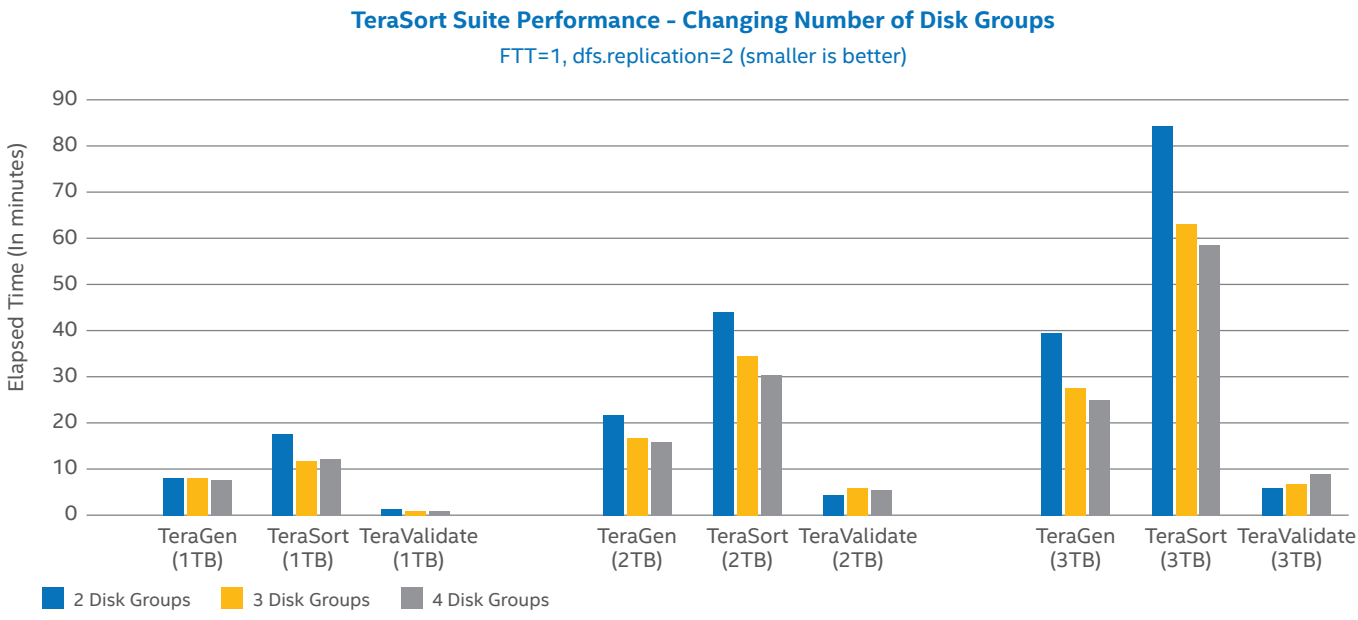

Figure 4. TeraSort* Suite performance with varying the number of disk groups.

3.2 Stripe Width

This experiment varies stripe width (SW) count from 1 to 3 while creating the VMDKs. With a higher SW setting, each VMDK gets striped across multiple SSDs. The maximum tested SW in the following test is 3, because that is the ratio of capacity-tier disks to cache-tier. However, the maximum supported SW in vSAN is 7.

Figure 5 shows Hadoop performance with different SW counts. While Hadoop performance doesn't change substantially when increasing SW count for 1 and 2 TB input data sets, there is up to a 37-percent performance improvement when going from an SW setting of 1 to 3 with a 3 TB TeraSort operation. The success of this result appears to be due to better object placement with SW=3 compared to SW=1.

3.3 Failures to Tolerate (FTT)

The FTT feature allows the user to set redundancy in a vSAN cluster. By default, FTT is set to 1, meaning that the cluster is designed to tolerate a one-node failure without any data loss. A higher level of redundancy can be set to ensure a higher degree of cluster reliability and resiliency. However, this comes at the expense of maintaining multiple copies of data, thereby impacting the number of writes needed to complete one transaction, as well as requiring greater capacity in the cluster (see Reference section, item [1]). In the first test case, Hadoop is configured to create one copy (dfs.replication = 1), while vSAN created three replicas with FTT set to 2. In the second test case, Hadoop is configured to perform 2 copies with (dfs.replication = 2), while FTT is set to 1 in vSAN to allow vSAN to create two replicas.

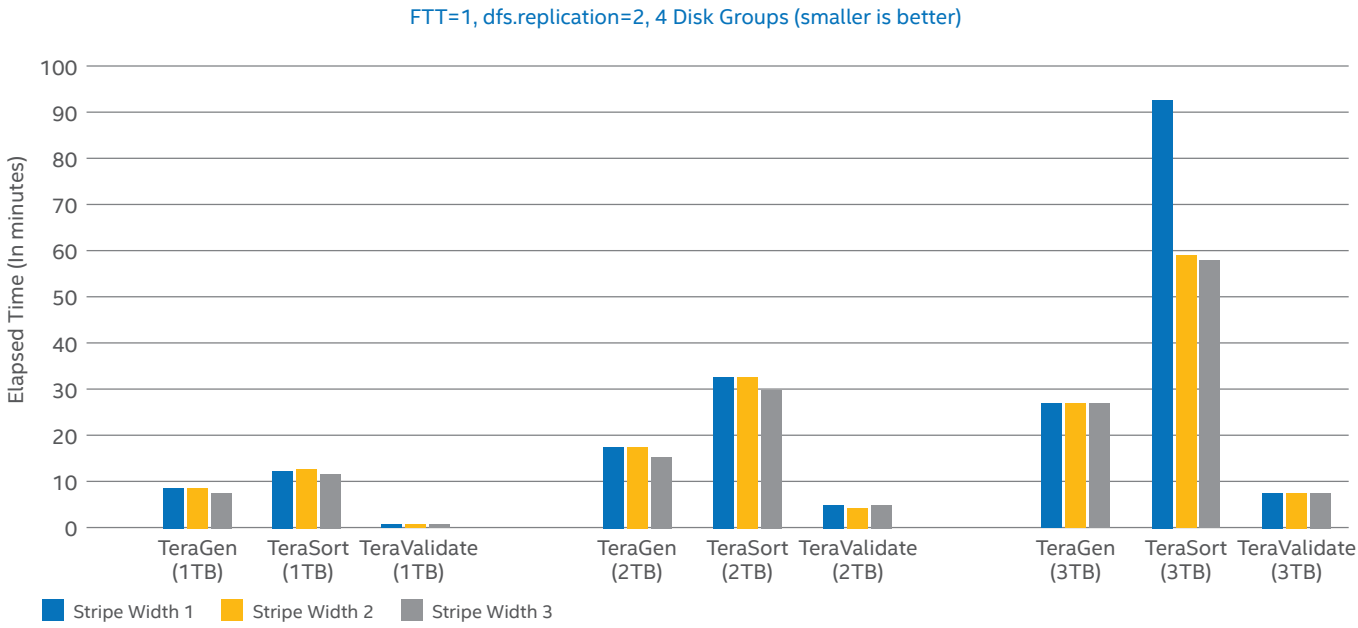

Figure 5. TeraSort* Suite performance with varying stripe width count.

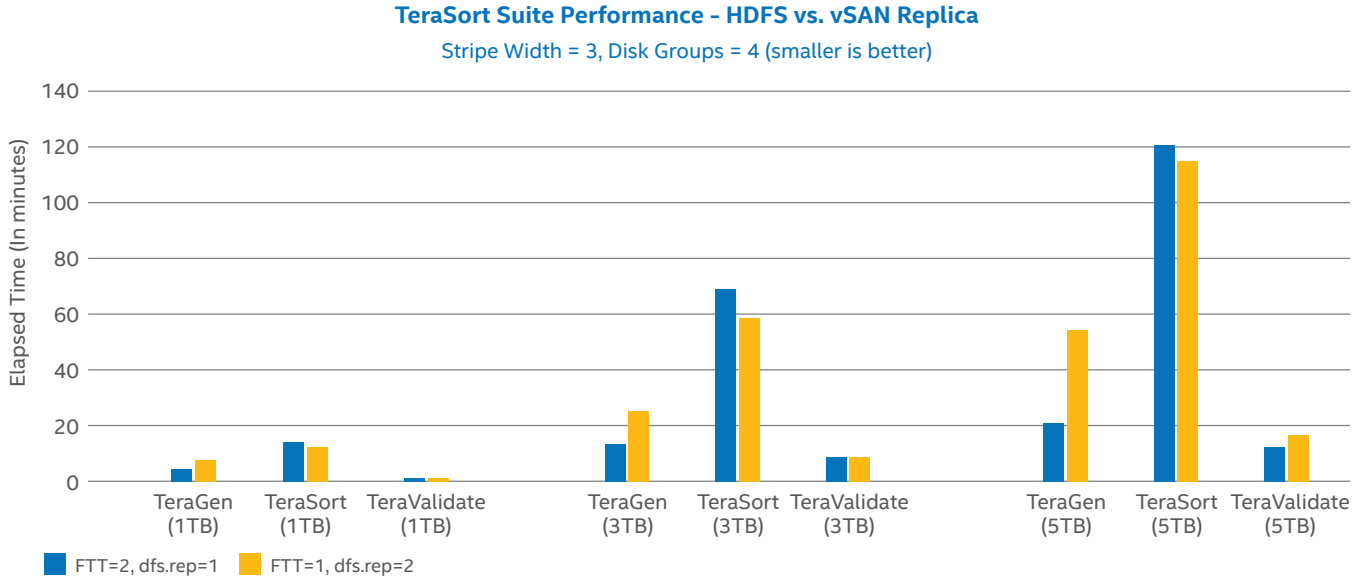

Figure 6. TeraSort* Suite performance when comparing Hadoop* versus vSAN replication.

Figure 6 shows how performance compared with these two test cases. There is not much noticeable performance difference between the two test cases when running the TeraSort operation. It should also be noted that in the second test case (FTT=1, dfs.replication = 2), four copies are actually created in total (two HDFS and two vSAN), versus three copies (three vSAN, zero HDFS) in the first test case (FTT=2, dfs.replication = 1). The impact on TeraGen performance is shown in Figure 6.

Figure 6 shows that no conclusion can be drawn from this testing in favor of either Hadoop or vSAN replication, and therefore either configuration can be chosen depending on tradeoffs between protection and capacity.

3.4 Host Affinity (Tech Preview)

One new feature in development on vSAN is called vSAN Host Affinity (currently in tech preview). vSAN Host Affinity allows VMs to be fully aware of the underlying vSAN topology, thus providing the ability to exploit better performance by having vSAN data stored on the same host as where the VM is performing an I/O transaction. In the default setup, vSAN distributes data across all available storage resources, aiming for reasonably uniform usage. While this simplifies the job of managing storage, it also reduces performance because of highly distributed I/O transactions.

vSAN Host Affinity allows specifying a storage policy that will attempt to place at least one vSAN copy on the same host as where the VM is located. When coupled with FTT=0 and SW=1, this approach ensures that I/O transactions are less distributed. However, VSAN Host Affinity also comes with the penalty that some storage resources may have very high usage, since vSAN data placement will depend on the placement of VMs.

Figure 7 demonstrates the value of Host Affinity when it is turned on versus when it is turned off. In the results shown here, for a 5 TB TeraSort test case, elapsed time drops from 73 minutes to 48 minutes when using Host Affinity, for up to a 34-percent performance improvement.

4. Conclusion

This paper shares some best practices and key learnings when running Hadoop in a virtualized manner with hyper-converged all-flash storage using vSAN. It further shows how different vSAN configuration parameters can be chosen for optimum Hadoop performance. The testing reported on here shows that more disk groups, a greater stripe width, and avoiding extra redundancy by letting Hadoop manage data replication offer the best performance. vSAN's new feature called vSAN Host Affinity (currently in tech preview) also offers potential for performance benefits based on data locality.

While there are trade-offs on a pure performance level when compared to running Hadoop on bare-metal servers, there are other advantages that make running them together on hyper-converged infrastructure attractive. The advantages of managing Hadoop and vSAN storage through a common VMware framework and tools, as well as pooling together data from multiple sources for analytics, potentially drive greater insights for business. Therefore, this architectural approach offers strong promise.

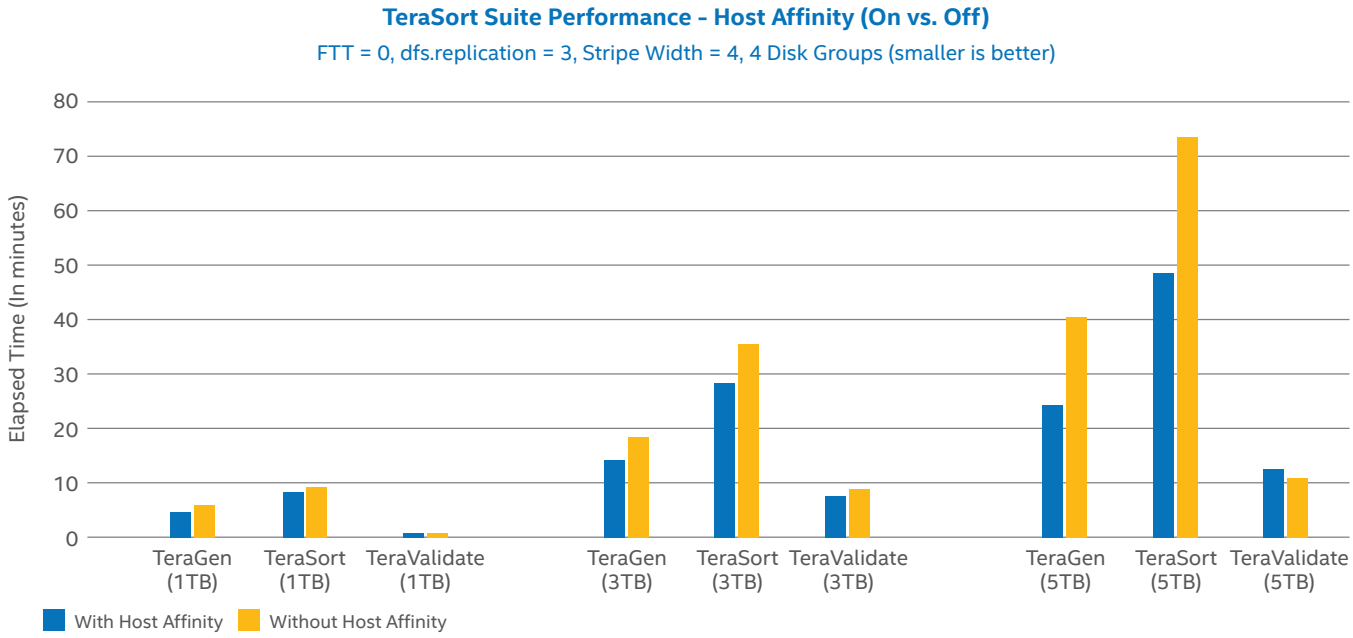

Figure 7. TeraSort* Suite performance with vSAN Host Affinity feature.

Appendix A. System Performance Tuning

BIOS Tunings

- Profiles:
  - CPU Power and Performance Policy: Performance
  - Workload Configuration: Balanced
  - Memory RAS configuration: Maximum Performance
  - Fan Profile: Performance
- Enabled:
  - Hyper-Threading
  - NUMA Optimized
  - Enhanced Intel® SpeedStep® Technology
  - Intel® Turbo Boost Technology
  - Performance P-Limit
- Disabled:
  - Cluster on Die
  - Early Snoop
  - CPU C States
  - Energy Efficient Turbo

Virtual Machine OS Performance Tuning

ALL VMS	CONFIGURATION CHANGE
All	**XFS filesystem** Created using 2048 inode for all data drive mount points. **XFS mount option**: xfs noatime,nodiratime,logbufs=8 0 0
All	**/etc/sysctl.conf** echo 'net.core.rmem_max = 16777216' >> /etc/sysctl.conf echo 'net.core.wmem_max = 16777216' >> /etc/sysctl.conf echo 'net.ipv4.tcp_rmem = 4096 87380 16777216' >> /etc/sysctl.conf echo 'net.ipv4.tcp_wmem = 4096 65536 16777216' >> /etc/sysctl.conf echo 'net.core.netdev_max_backlog = 250000' >> /etc/sysctl.conf echo 'vm.swappiness=0' >> /etc/sysctl.conf sysctl -p echo '* - nofile 65536' >>/etc/security/limits.conf echo '* - nproc 65536' >>/etc/security/limits.conf
All	**Disable Transparent Huge Page defrag** # echo never> /sys/kernel/mm/transparent_hugepage/defrag # echo never> /sys/kernel/mm/transparent_hugepage/enabled **Set NIC MTU to 9000 bytes** # ifconfig eth{} mtu 9000

References

[1] VMware, Inc. (2015, February) VMware vSAN 6.0 Performance:

http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/products/vSAN/vmware-virtual-san6-scalability-performance-white-paper.pdf

[2] John Nicholson, (2016, March) VMware, Inc. vSAN 6.2 Design and Sizing Guide:

http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsan/virtual-san-6.2-design-and-sizing-guide.pdf

[3] VMware, Inc. (2015, February) Virtualized Hadoop Performance with VMware vSphere 6 on High Performance Servers:

http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/virtualized-hadoop-performance-with-vmware-vsphere6-white-paper.pdf

[4] VMware, Inc. (2016, August) Big Data Performance on vSphere 6:

http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/bigdata-perf-vsphere6.pdf

[5] VMware vSAN Overview:

http://www.vmware.com/products/virtual-san.html

[1] Source: VMware internal focus groups. 43 responses from different companies, based on six focus groups in Europe and the US.