vSAN 6.2 for Remote and Branch Office Deployment

October 31, 2017
1. Executive Summary

This section covers the Business Case, Solution Overview and Key Results of the vSAN 6.2 for Remote and Branch Office Deployment document.
1.1 Business Case

Many customers today operate with multiple locations, which are remote or branch offices. Customers with VMware Remote Office Branch Office™ (ROBO) environments are faced with the challenges such as lack of infrastructure or staff to adequately manage the onsite IT resources. Minimizing costs for operations and maintenance while still providing an adequate platform capable of both performance and resiliency in such environments can be challenging.

ROBO deployment environments can benefit from shared storage while minimizing costs by using the vSAN ROBO deployment solution. vSAN ROBO deployment allows customers to deploy vSAN on two or more physical vSphere hosts using inexpensive and industry-standard server components to avoid large and upfront investments in purpose-built storage hardware. vSAN and storage policies are easily configured and managed using the familiar and user-friendly vSphere Web Client, so there is no need for ROBO administrators to learn a separate tool just for storage management. Administration can be performed from a central location or can be delegated through the use of VMware vCenter Server® roles and permissions.

The vSAN ROBO deployment solution delivers the same class of performance and availability as the enterprise edition does. This reference architecture demonstrates vSAN performance while running common and various ROBO user scenarios and helps customers architect the vSAN ROBO deployment solution.

1.2 Solution Overview

This solution describes the test strategy and architecture of the vSAN for ROBO deployment solution, focusing on test scenarios, test methodologies, and performance results. The solution validates a 2-node hybrid vSAN Cluster running typical ROBO deployment applications.

**Note:** vSAN ROBO deployment is a license instead of an architecture. The node number limit for vSAN ROBO deployment is 64, which is the same number for vSAN Cluster. We use the term ‘2-node ROBO’ in this reference architecture to indicate that we run the tests with the minimal configuration requirement. In addition to the CPU-socket-based license, VMware also provides a vSAN ROBO deployment license with 25 virtual machines. Users can apply this license to multiple ROBO sites and each ROBO site can have 2 to 64 physical hosts.

1.3 Key Results

Figure 1 highlights that vSAN is an enterprise-class storage solution suitable for ROBO deployment.
Figure 1. Key Results for 2-node Hybrid vSAN ROBO Deployment

- Three workloads are running concurrently
- Login VSI score of 640 to 700
- More than 650 database transaction per second (TPS)
- More than 50 MB/s file server service

To summarize, the tests prove that a 2-node hybrid vSAN ROBO Cluster can support consolidated applications and mixed workloads. The results show a predictable and sustainable high performance in a 2-node hybrid vSAN ROBO deployment.
2. vSAN ROBO Deployment Reference Architecture

This section details vSAN ROBO Deployment Reference Architecture used for solution validation.
2.1 Purpose

This reference architecture validates the ability of 2-node hybrid vSAN ROBO deployment to support common and various mixed workloads such as database, virtual desktop infrastructure (VDI), and file sharing. We run multiple tests to validate the performance, manageability, and reliability of a 2-node hybrid vSAN ROBO deployment.

The results validate that a 2-node hybrid vSAN ROBO deployment can handle hardware failures in a production environment. Data integrity is maintained while performance is minimally affected. By leveraging VMware vSphere High Availability (vSphere HA), virtual machines can recover from an entire host failure. A 2-node hybrid vSAN ROBO deployment is proved to be a robust configuration.

2.2 Scope

This reference architecture:

- Demonstrates storage performance scalability and resiliency of typical ROBO mixed workloads in a hybrid vSAN ROBO deployment environment.
- Illustrates the resiliency of hybrid vSAN ROBO deployment in various failure scenarios including disk, disk group, witness host, and storage host failures.
- Introduces vSAN Application Programming Interfaces (APIs) to ease the management work.

2.3 Audience

This reference architecture is intended for information technology architects and storage architects involved in planning, architecting and administering an environment with vSAN ROBO deployment.
3. Technology Overview

This section provides an overview of the technologies used in this solution.
3.1 Overview

This section provides an overview of the technologies used in this solution:

- VMware vSphere 6.0 Update 2
- VMware vSAN 6.2
- VMware Horizon 7
- Microsoft SQL Server 2014
- Windows File Sharing Server

3.2 VMware vSphere 6.0 Update 2

VMware vSphere is the industry-leading virtualization platform for building cloud infrastructures. It enables users to run business-critical applications with confidence and respond quickly to business needs. vSphere accelerates the shift to cloud computing for existing data centers and underpins compatible public cloud offerings, forming the foundation for the industry’s best hybrid cloud model.

3.3 VMware vSAN 6.2

VMware vSAN is VMware’s software-defined storage solution for hyper-converged infrastructure, a software-driven architecture that delivers tightly integrated computing, networking, and shared storage from x86 servers. vSAN delivers high performance and highly resilient shared storage by clustering server-attached flash devices and hard disks (HDDs).

vSAN delivers enterprise-class storage services for virtualized production environments along with predictable scalability and All-Flash performance, all at a fraction of the price of traditional, purpose-built storage arrays. Just like vSphere, vSAN provides users the flexibility and control to choose from a wide range of hardware options and easily deploy and manage it for a variety of IT workloads and use cases.
vSAN can be configured as a hybrid or an All-Flash storage. In a hybrid disk architecture, vSAN leverages flash-based devices for performance and magnetic disks for capacity. In an All-Flash disk architecture, vSAN can use flash-based devices (PCIe SSD or SAS/SATA SSD) for both caching and persistent storage. It is a distributed object storage system that leverages the vSAN SPBM feature to deliver centrally managed, application-centric storage services and capabilities. Administrators can specify storage attributes, such as capacity, performance, and availability, as a policy on a per VMDK level. The policies dynamically self-tune and load balance the system so that each virtual machine has the right level of resources.

3.4 VMware Horizon 7

VMware Horizon desktop and application virtualization solutions provide organizations with a streamlined approach to delivering, protecting, and managing desktops and applications while containing costs and ensuring that end users can work anytime, anywhere, and across any device.

With the introduction of Horizon 7, VMware is drawing on the best of mobile and cloud, offering greater simplicity, security, speed, and scale in delivering on-premises virtual desktops and applications with cloud-like economics and elasticity of scale.

3.5 Microsoft SQL Server 2014
Microsoft SQL Server is one of the most widely deployed database platforms in the world, with many organizations having dozens or even hundreds of instances deployed in their environments. The flexibility of SQL Server, with its rich application capabilities combined with the low costs of x86 computing, has led to a wide variety of SQL Server installations ranging from large data warehouses to small, highly specialized departmental and application databases. The flexibility at the database layer translates directly into application flexibility, giving end users more useful application features and ultimately improving productivity.

3.6 Windows File Sharing Server

The file sharing functionality in Windows Server allows users to centrally manage file shares on a computer. The Server Message Block (SMB) protocol is what Windows uses to share files, printers, serial ports, and communicate this information between computers by using named pipes and mail slots. In a networked environment, servers make file systems and resources available to clients. Clients make SMB requests for resources and servers make SMB responses.
4. Solution Configuration

This section introduces the resources and configurations for the solution including architecture configuration, hardware resources and other relevant VM and storage configurations.
4.1 Overview

This section introduces the resources and configurations for the solution including:

- Architecture configuration
- Hardware resources
- Infrastructure VM configuration
- VM configuration for different workloads
- vSAN disk group and fault domain configuration
- Network configuration
- VMware ESXi Server: Storage Controller Mode
- Storage policy settings

4.2 Architecture Configuration

It is not necessary to set up a one-to-one mapping between the management cluster and the ROBO Cluster. From the cost-effective perspective, we highly recommend you to use one management cluster for managing multiple ROBO Clusters as shown in Figure 3.
Because each of the ROBO sites is standalone and does not interfere with each other, we use one vSAN ROBO Cluster for demonstration and testing.

As shown in Figure 4, the ROBO Cluster architecture consists of two parts: the data center that contains the management cluster and a ROBO site. The data center is where central IT management occurs and the remote 2-node ROBO hybrid vSAN Cluster carries the real workloads.

There are several management virtual machines residing in the management cluster:

- The vCenter manages the ROBO sites.
- The Active Directory virtual machine acts as the User Identity, DNS, and DHCP Server.
- The Witness virtual appliance is a mandatory part of a vSAN ROBO Cluster and is used for arbitration in case ‘split-brain’ occurs.
• The Composer, Connection Server, and database Virtual machines are part of Horizon 7.

For the ROBO Cluster architecture, only one Horizon View management cluster is needed because it can manage multiple desktop pools in different ROBO sites. This can ease the management of Horizon View in a ROBO deployment environment.

For the ROBO site, the architecture shows one SQL Server, one Windows Server 2012 acting as a file sharing server, and a VMware Horizon desktop environment hosted on two servers with vSAN 6.2 running on vSphere 6.0 update 2. The vSAN ROBO Cluster only carries the real workloads but no management workloads.

The link between the management cluster and vSAN ROBO Cluster is shared by all types of traffic, including ESXi management traffic, vSAN traffic, and Horizon management traffic. The link has a bandwidth limit of 1.5 Mb/s. The maximum network latency is 500ms. We used a network latency of 250ms and 500ms respectively in this solution.

![vSAN ROBO Cluster Architecture](image)

**Figure 4. vSAN ROBO Cluster Architecture**

### 4.3 Hardware Resources

Table 1 shows the configuration of each vSAN ESXi Server in vSAN ROBO Cluster.

**Table 1. Server Configuration**

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESXi Server model</td>
<td>2 x Intel based Servers</td>
</tr>
<tr>
<td>ESXi host CPU</td>
<td>2 x 20 Intel(R) Xeon(R) CPU E5-2690 @ 3.0GHz v2</td>
</tr>
</tbody>
</table>
### PROPERTY SPECIFICATION

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESXi host RAM</td>
<td>512GB</td>
</tr>
<tr>
<td>ESXi version</td>
<td>ESXi 6.0 build 3620759</td>
</tr>
<tr>
<td>Network adapter</td>
<td>2 x 10Gbps SFI/SFP+ (set ports to 1G for vSAN traffic) 1 x 1Gbps (used for VMware vSphere vMotion®)</td>
</tr>
<tr>
<td>Storage adapter</td>
<td>12Gbps SAS HBA</td>
</tr>
<tr>
<td>Disks</td>
<td>Cache SSD: 1 x 800GB (6Gbps) Capacity HDD: 6 x 1.2TB (6Gbps)</td>
</tr>
</tbody>
</table>

### 4.4 Infrastructure VM Configuration

- Single vCenter Server with following roles:
  - vCenter
  - vCenter single sign-on (SSO)
  - vCenter Inventory Service
  - Windows Active Directory Server

### Table 2. Infrastructure VM Configuration—Hosted on Management Cluster

<table>
<thead>
<tr>
<th>INFRASTRUCTURE VM ROLE</th>
<th>VCPU</th>
<th>RAM (GB)</th>
<th>STORAGE (GB)</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Controller (and DNS)</td>
<td>2</td>
<td>8</td>
<td>40</td>
<td>Windows Server 2012 Datacenter 64-bit</td>
</tr>
<tr>
<td>vCenter appliance, build 3634794</td>
<td>4</td>
<td>16</td>
<td>415</td>
<td>SUSE Linux Enterprise 11</td>
</tr>
<tr>
<td>vSAN Witness Appliance</td>
<td>2</td>
<td>16</td>
<td>380</td>
<td>VMware ESXi 6</td>
</tr>
<tr>
<td>View Composer Server</td>
<td>4</td>
<td>8</td>
<td>100</td>
<td>Windows Server 2012 Datacenter 64-bit</td>
</tr>
<tr>
<td>View Connection Server</td>
<td>4</td>
<td>8</td>
<td>100</td>
<td>Windows Server 2012 Datacenter 64-bit</td>
</tr>
</tbody>
</table>
Table 3 shows the virtual machine configuration details. The memory allocation considers the memory reservation for the OS and test client to avoid high memory pressure after each test run.

### Table 3. VM Configuration

<table>
<thead>
<tr>
<th>VM ROLE</th>
<th>NUMBER OF VMS</th>
<th>VCPU</th>
<th>MEMORY (GB)</th>
<th>VERSION</th>
<th>OPERATING SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>Enterprise Edition, SP1</td>
<td>Windows Server 2012 Datacenter 64-bit</td>
</tr>
<tr>
<td>VMware Horizon virtual desktops</td>
<td>20</td>
<td>2</td>
<td>4</td>
<td>N/A</td>
<td>Windows 7 64-bit</td>
</tr>
<tr>
<td>Windows File Sharing Server</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>N/A</td>
<td>Windows Server 2012 Datacenter 64-bit</td>
</tr>
</tbody>
</table>

Table 4 shows the disk layout of the SQL server virtual machine. Because there are seven virtual disks, it is recommended to connect them to different virtual SCSI controllers. Based on the role of virtual disks, we separated them to four SCSI controllers as shown in Table 4.

### Table 4. SQL Server Virtual Disk Layout

<table>
<thead>
<tr>
<th>VM1 DISK LAYOUT—50 GB DATABASE</th>
<th>SCSI CONTROLLER</th>
<th>DATASTORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>100GB Drive C: Windows OS disk</td>
<td>LSI Logic SAS</td>
<td>vsanDatastore</td>
</tr>
<tr>
<td>100GB Drive E: Database disk</td>
<td>PVSCSI Controller 1</td>
<td>vsanDatastore</td>
</tr>
<tr>
<td>100GB Drive F: Database disk</td>
<td>PVSCSI Controller 1</td>
<td>vsanDatastore</td>
</tr>
</tbody>
</table>
Table 5 shows the test image used to provision desktop sessions in the View environment with Login VSI. We used optimization tools according to [VMware OS Optimization Tool](https://www.vmware.com/products/optimization-tool.html).

**Table 5. Virtual Machine Test Image**

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>LOGIN VSI IMAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop OS</td>
<td>Windows 7 Enterprise SP1 (64-bit)</td>
</tr>
<tr>
<td>Hardware</td>
<td>VMware Virtual Hardware version 11</td>
</tr>
<tr>
<td>CPU</td>
<td>2</td>
</tr>
<tr>
<td>Memory</td>
<td>2,048 MB</td>
</tr>
<tr>
<td>Memory reserved</td>
<td>0 MB</td>
</tr>
<tr>
<td>Video RAM</td>
<td>35 MB</td>
</tr>
<tr>
<td>3D graphics</td>
<td>Off</td>
</tr>
<tr>
<td>NICs</td>
<td>1</td>
</tr>
<tr>
<td>Virtual network adapter</td>
<td>VMXNet3 Adapter</td>
</tr>
<tr>
<td>Virtual SCSI controller</td>
<td>Paravirtual</td>
</tr>
<tr>
<td>Virtual disk—VMDK</td>
<td>30 GB</td>
</tr>
<tr>
<td>Applications</td>
<td>Adobe Acrobat 11</td>
</tr>
<tr>
<td></td>
<td>Adobe Flash Player 16</td>
</tr>
<tr>
<td></td>
<td>Doro PDF 1.82</td>
</tr>
<tr>
<td></td>
<td>FreeMind</td>
</tr>
<tr>
<td></td>
<td>Internet Explorer 11</td>
</tr>
<tr>
<td></td>
<td>FreeMind</td>
</tr>
<tr>
<td></td>
<td>Internet Explorer 11</td>
</tr>
<tr>
<td></td>
<td>Microsoft Office 2010</td>
</tr>
<tr>
<td>VMware Tools™</td>
<td>10.0.6.3560309</td>
</tr>
</tbody>
</table>
4.6 vSAN Disk Group and Fault Domain Configuration

We deployed a 2-node hybrid vSAN Cluster to support the vSAN ROBO deployment environment. Each server was deployed with an identical configuration and the ESXi Server was booted from local disk.

The maximum storage configuration depends on the hardware. And the fault domain setting uses the default one: one ESXi host is one vSAN fault domain.

Table 6 shows the disk group configuration on the ESXi hosts. Each host has one disk group including one SSD and six HDDs.

**Table 6. Disk Group Configuration**

<table>
<thead>
<tr>
<th>CONFIGURATION</th>
<th>NUMBER</th>
<th>CAPACITY (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk groups per host</td>
<td>1</td>
<td>7,200GB raw capacity in total.</td>
</tr>
<tr>
<td>HDD as the capacity tier per disk group</td>
<td>6</td>
<td>1,200 GB</td>
</tr>
<tr>
<td>SSD as the cache tier per disk group</td>
<td>1</td>
<td>800 GB</td>
</tr>
</tbody>
</table>

4.7 Network Configuration

As shown in Figure 5, we used two virtual machines acting as the software routers. Each VM has three vNICs. Bandwidth limit and latency were added to the vNIC connecting to VLAN5 by using the Linux ‘tc’ command (see this blog for details about how to set up bandwidth limit and latency). Table 7 shows the configurations of different VLANs. In the test setup, we set the bandwidth limit to 1.5 Mb/s. The latency was configured to round-trip time (RTT) 250ms and 500ms respectively to simulate WAN connection between the central data center and a ROBO site. Both management and vSAN traffics share the same VLAN5.

1.5 Mb/s is the minimal bandwidth requirement for vSAN ROBO deployment. We chose this configuration to validate the performance in the supported worst case with regards to network bandwidth.

500ms is the maximum supported network latency in a 2-node ROBO deployment environment. The tests with 500ms network latency are supposed to generate the worst result regarding network. In addition, we chose 250ms as a modest network latency to test the 2-node vSAN ROBO deployment in a normal situation.

**Table 7. VLAN Configuration**
<table>
<thead>
<tr>
<th>VLAN NAME</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN1</td>
<td>Management traffic in central data center</td>
</tr>
<tr>
<td>VLAN2</td>
<td>vSAN witness traffic in central data center</td>
</tr>
<tr>
<td>VLAN3</td>
<td>Management traffic in ROBO site</td>
</tr>
<tr>
<td>VLAN4</td>
<td>vSAN traffic in ROBO site</td>
</tr>
<tr>
<td>VLAN5</td>
<td>Internal traffic between software router</td>
</tr>
</tbody>
</table>
5. Solution Validation

In this section, we present the test methodologies and processes used in this operation guide.
5.1 Overview

The solution validated the performance and functionality of mixed workloads in a virtualized VMware environment running in the 2-node hybrid vSAN ROBO Cluster.

The mixed workloads include:

- VMware Horizon virtual desktops
- Microsoft SQL Server
- Microsoft Windows File Sharing Server

5.2 Test Overview

The solution tests include:

- Application performance testing with network latency and bandwidth constraint
  - 250ms round-trip time (RTT) delay between ROBO site and central data center
  - 500ms RTT delay between ROBO site and central data center
- Application performance with 2-node vSAN ROBO deployment with error injection and network latency
  - Failure of HHD (disk failure)
  - Failure of SSD (disk group failure)
  - Host failure
  - Witness failure
- Host maintenance test
  - vSAN performance during vSphere vMotion and host maintenance

5.3 Benchmark and Performance Data Collection Tools

Benchmark Tools

Benchmark Factory for Databases

Benchmark Factory for Databases is a database performance testing tool that enables users to conduct database industry-standard benchmark testing and scalability testing. See Benchmark Factory for Databases for more information.

Use Benchmark Factory scale factor of 5 to test SQL Server and get TPC-E-like performance in terms of TPS and latency. The scale factor as defined by Benchmark Factor allows for the use of larger user loads. This places a greater stress on the system-under-test.

Login VSI

Use the Login VSI in Benchmark mode with 20 sessions to measure VDI performance in terms of Login VSI baseline performance score (also called VSIbase or Login VSI index average score). The Login VSI baseline performance score is based on the response time reacting to the Login VSI workloads. A lower Login VSI score is better because it reflects that the desktops can respond with less time. In the tests, the workload type is ‘Knowledge Worker * 2vCPU’. For the various Login VSI notations, see VSImax.

DBENCH

DBENCH was used to measure file server performance in terms of bandwidth and latency.

A shared folder and 20 files were created. Each file was 100MB and the files were placed in the shares folder. Two virtual clients act as users to access the share folder concurrently. Each virtual user picks a
random file and performs read and write operations to the share folder. Then the user picks another file and goes into next iteration.

**Performance Data Collection Tools**

We used the following testing and monitoring tools in this solution:

- **vSAN Observer**
  
vSAN Observer is designed to capture performance statistics and bandwidth for a VMware vSAN Cluster. It provides an in-depth snapshot of IOPS, bandwidth and latencies at different layers of vSAN, read cache hits and misses ratio, outstanding I/Os, and congestion. This information is provided at different layers in the vSAN stack to help troubleshoot storage performance. For more information about the VMware vSAN Observer, see the [Monitoring VMware vSAN with vSAN Observer](#) documentation.

- **esxtop**
  
esxtop is a command line tool that can be used to collect data and provide real-time information about the resource usage of a vSphere environment such as CPU, disk, memory, and network usage. The ESXi Server performance is measured by this tool.

- **vSAN Performance Service**
  
Performance Service collects and analyzes performance statistics and displays the data in a graphical format. vSAN administrators can use the performance charts to manage the workload and determine the root cause of problems. When the vSAN Performance Service is turned on, the cluster summary displays an overview of vSAN performance statistics, including IOPS, throughput, and latency. vSAN administrators can view detailed performance statistics for the cluster, for each host, disk group, and disk in the vSAN Cluster.

5.4 Performance Testing of vSAN ROBO Deployment

**Test Scenarios**

**Baseline test**

In the baseline test, the mixed workloads were running concurrently for one hour and no errors were injected during the test. Benchmark Factory, Login VSI, and DBENCH were started at the same time. For Benchmark Factory and DBENCH, there was a warmup time of around 15 minutes. So we considered that the test was stabilized 15 minutes after the test was initialized. For the following failure tests, we also injected one error during the last 45 minutes of each test.

**SSD failure test**

In the SSD failure test, the mixed workloads were running concurrently for an hour. Benchmark Factory, Login VSI, and DBENCH were started at the same time. Then, we injected a permanent disk error to the SSD on one of the two nodes. The injection occurred 30 minutes after the test was initiated. Note that if an SSD was down, the whole disk group would be down as well. Then all the IOs moved to the other healthy node.

**HDD failure test**

In HDD failure test, the mixed workloads were also running concurrently for an hour. Benchmark Factory, Login VSI, and DBENCH were started at the same time. Then, we injected a permanent disk error to one of the hard disks. The injection occurred 30 minutes after the test was initiated. Since this was a permanent error, vSAN would start to rebuild the affected components onto other hard disks immediately after the error was detected.
**Witness appliance failure test**

In the witness appliance failure test, the mixed workloads were also running concurrently for an hour. Benchmark Factory, Login VSI, and DBENCH were started at the same time. The witness host was powered off 30 minutes after the test was initiated. Since two vSAN nodes were both healthy, vSAN can still function properly.

**vSphere vMotion and host maintenance test**

All the workload virtual machines were distributed evenly on the two nodes. To put a host into maintenance mode, we must vMotion all the virtual machines from one host to another host first. In this test, the mixed workloads were running concurrently for one hour. Benchmark Factory, Login VSI, and DBENCH were started at the same time. We started the vMotion 20 minutes after the test was initiated. After vMotion was completed, we put that host into maintenance mode immediately. After the host was in maintenance mode for 10 minutes, we made that host exit maintenance mode. All these actions could be done during the one-hour test.

**Host failure test**

In the host failure test, the mixed workloads were also running concurrently for an hour. Benchmark Factory, Login VSI, and DBENCH were started at the same time. We hard powered off one of the hosts 30 minutes after the test was initiated. Then we monitored whether vSphere HA restarted the affected virtual machines on the healthy host and whether vSAN was still functioning properly.

---

**250ms RTT Latency between Data Center and ROBO Site**

**Test Overview**

The network connection between the central data center and the ROBO site was configured with 250ms delay and 1.5 Mb/s bandwidth.

We ran the tests one by one. If one error was injected in the previous test case, the error was first cleared and vSAN was brought back to a healthy state before the next test case, so there was at most one error at any given time. Although the benchmark tools were running concurrently, we grouped the results by tool and showed the results in the next section.

**Test Results**

Before any tests, we first deployed 20 virtual desktops in Horizon. The desktop pool type was 'Linked Clone' with View Composer. Test result shows that the deployment took 32 minutes to finish.

**Note:** We used the baseline test result as a representative for all the detailed figures in the following sections. This was based on the observation that the results of failure tests were similar to those of the baseline testing.

Figure 6 shows the Login VSI results. After each test run, Login VSI gives a score indicating how the virtual desktops perform. A lower Login VSI score is better. In addition, Login VSI gives a VSIMax threshold that should not be reached during a test. Figure 6 and Figure 7 shows VSIMax threshold was not reached in each scenario.
Figure 6. Login VSI Score for 250ms Latency

Figure 7 shows the details of the Login VSI performance score. Twenty sessions ran successfully. VSIbase was 655 and VSImax v4.1 threshold was 1,655. VSImax (v4.1) Knowledge Worker was not reached with the baseline performance score of 655.

![Login VSI Score Graph]

Figure 7. Login VSI Score for 250ms Latency in the Baseline Test

Figure 8 shows the Benchmark Factory result of the TPC-E-like workload running on Microsoft SQL Server:

- TPS was between 675 and 705.
- The average response time was 3ms across all the tests.
- The average transaction time was 29ms in the HDD failure test and 28ms in the other tests.
Figure 8. Benchmark Factory Result for 250ms Latency

Figure 9 shows the Benchmark Factory result of the TPC-E-like workload running on Microsoft SQL Server in the baseline test. TPS jittered around 700 and the average response time varied from 1ms to 4ms.

Figure 9. Benchmark Factory Results for 250ms Latency in the Baseline Test
Figure 10 shows the DBENCH result for Windows File Sharing Server. The bandwidth was around 50~55 MB/s in all the tests. The average read latency was 0.61~0.65ms and the average write latency was 0.26~0.29ms.

![File Sharing Server Result](image)

Figure 10. File Sharing Server Result for 250ms Latency

Figure 11 shows the vSAN backend Performance Service page for 250ms in the baseline test. When the test was stable after 15 minutes' warmup, the peak write IOPS was around 3,640 and peak read IOPS was around 1,820. The peak write throughput was 65.88 MB/s and the peak read throughput was around 32.94 MB/s. Both the write and read latencies were around 0.6ms. There was no congestion during the whole test and the outstanding IO stayed between 6 and 12.

The result shows that the 2-node vSAN ROBO Cluster was not congested. Moreover, it could deliver a consistent excellent performance and could handle the mixed workloads properly.
Figure 11. vSAN Performance Service for 250ms Latency in the Baseline Test

For the host failure test, the vSphere HA automatically restarted the affected virtual machines on the healthy host in two minutes. After the restart, SQL Server, Windows File Sharing Server, and all the virtual desktops were properly working. However, Benchmark Factory, Login VSI, and DBENCH lost...
their test target to SQL Server respectively and did not generate any results. Therefore, for the host failure test, data in vSAN was intact and all virtual machines could be restarted by vSphere HA.

500ms RTT Latency between Data Center and ROBO Site

Test Overview

The network connection between the central data center and the ROBO site was configured with 500ms delay and 1.5 Mb/s bandwidth.

Test Results

Before any tests, we first deployed 20 virtual desktops in Horizon. The desktop pool type was ‘Linked Clone’ with View Composer. Test result shows that the deployment took 43 minutes to finish.

Figure 12 shows the Login VSI results. The Login VSI score of each scenario was between 655 and 670. In each test, VSIMax was not reached.

![Login VSI Score for 500ms Latency](image1)

Figure 12. Login VSI Score for 500ms Latency

Figure 13 shows the details of Login VSI performance score. Twenty sessions ran successfully. The key result shows that VSIbase was 656 and VSImax threshold was 1,657. VSImax (v4.1) Knowledge Worker was not reached with the baseline performance score of 656.

![Login VSI Score for 500ms Latency in the Baseline Test](image2)

Figure 13. Login VSI Score for 500ms Latency in the Baseline Test
Figure 14 shows the Benchmark Factory result of the TPC-E-like workload running on Microsoft SQL Server:

- Among the tests, TPS was between 683 and 708.
- The average response time was 3ms across all the tests.
- The average transaction time was 29ms in the SSD and HDD failure tests, and 28ms in the other tests.

![Benchmark Factory Result](image)

**Figure 14. Benchmark Factory Results for 500ms Latency**

Figure 15 shows the details of Benchmark Factory result of the TPC-E-like workload running on Microsoft SQL Server in the baseline test. TPS jittered around 700 and the average response time varied from 1ms to 5ms.
Figure 15. Benchmark Factory Results for 500ms Latency in the Baseline Test

Figure 16 shows the DBENCH result for Windows File Sharing Server. The bandwidth was around 48~55 MB/s in all the tests. The average read latency was around 0.59~0.78ms and the average write latency was around 0.25~0.29ms.
Figure 16. File Sharing Server Result for 500ms Latency

Figure 17 shows the vSAN backend Performance Service page for 500ms in the baseline test. When the test was stable after 15 minutes’ warmup, the peak write IOPS was around 3,414 and peak read IOPS was around 1,707. The peak write throughput was 67.95 MB/s and the peak read throughput was around 33.98 MB/s. Both the write and read latencies were around 1.0ms. There was no congestion during the whole test and the outstanding IO stayed between 6 and 12.

The Performance Service page shows that the 2-node vSAN ROBO Cluster was not congested. Moreover, it could deliver a consistent excellent performance and could handle the mixed workloads properly.
For the host failure test, the vSphere HA restarted the affected virtual machines on the healthy host in two minutes. SQL Server, Windows File Sharing Server, and all the virtual desktops were properly working after vSphere HA restarted them. However, Benchmark Factory, Login VSI, and DBENCH lost their test target to SQL Server respectively and did not generate any results. Therefore, for host failure test, the data in vSAN was intact and all virtual machines could be restarted by vSphere HA.

**Performance Testing Summary**

From the test results, the different network latency of 250ms and 500ms showed no significant impact on performance. All the test results were similar across different configurations. In addition, there was just slight impact on performance in failure tests comparing to the baseline test. This proved that a 2-node vSAN ROBO Cluster was robust and could deliver a consistent performance for mixed workload in various failure scenarios.
6. Appendix: vSAN Management API Reference

This section provides details on the vSAN Management API.
6.1 vSAN Management API Reference

For a typical vSAN ROBO deployment, the architecture includes a central data center and multiple ROBO sites. Since the ROBO sites are geographically distributed around the world, it is recommended for IT administrators to manage the ROBO sites from the data center. However, manual management is time consuming and is error prone. Therefore, it is recommended to leverage vSAN APIs and automate the vSAN management work.

vSAN 6.2 includes the vSAN API, which is an extension of the vSphere API. The vSAN API centers on a small set of managed objects, which enables administrators to query runtime state and to configure vSAN. The API is exposed as a Web service, running on both VMware vCenter Server systems and VMware ESXi systems. Managed objects are available for cluster-level and host-level operations.

**Note:** All the following operations in the code samples can also be done via vSphere Web Client. The purpose of the API introduction is to help administrators automate and ease the management of large number of ROBO sites. For medium and small deployments of vSAN ROBO, administrators can perform all the management tasks with vSphere Web Client without learning all the APIs.

For detailed information, see [VMware vSAN 6.2 vSAN Management API Cookbook for Python](#).

6.2 vSAN API Preview

vSAN 6.2 includes the following managed objects:

- VsanVcDiskManagementSystem
- VsanVcStretchedClusterSystem
- VsanVcClusterConfigSystem
- VsanVcClusterHealthSystem
- VsanPerformanceManager
- HostVsanHealthSystem
- VsanUpgradeSystemEx
- VsanSpaceReportSystem
- Objectiveness
- HostVsanSystem
- VsanUpgradeSystem

The collection of these managed objects is an extension to the vSphere managed objects. With these APIs, administrators can set up and configure all aspects of vSAN query runtime state. For API details, see the vSAN Management topic.

6.3 Typical API Usage for vSAN ROBO Deployment

vSAN API provides SDKs for various programming languages, including .NET, Java, Perl, Ruby, and Python. In this paper, we use Python as the demonstrating language.

**Prerequisites**

pyvmodm is the Python SDK for the VMware vSphere API. Since vSAN API is an extension to vSphere API, pyVmomd must be installed first before vSAN SDK.

For other programming languages, the corresponding SDK must be installed first. For example, install the [vSphere Perl SDK](#) before the vSAN Perl SDK.
vSAN SDK Utility and Sample

The downloaded vSAN SDK includes two useful files:

- vsanapiutils.py is the utility file that wraps all the necessary low-level method and provides high-level methods to users to reduce the programming efforts. By using methods defined in vsanapiutils.py, a user can easily connect to either a vCenter or an ESXi host and query vSAN information.

- vsanapisamples.py is the sample code that includes a whole procedure and complete codes to connect to a vCenter and query vSAN information. It is a complete sample so that a user can just slightly modify it to fit real-case needs.

For example, use the following code to connect to a vCenter. SmartConnect is a method defined in the vsanapiutils.py file:

```python
si = SmartConnect(host=args.host, 
                  user=args.user, 
                  pwd=password, 
                  port=int(args.port), 
                  sslContext=context)
```

Get a vCenter instance:

```python
vcMos = vsanapiutils.GetVsanVcMos(si._stub, context=context)
```

Get a cluster instance:

```python
cluster = getClusterInstance(args.clusterName, si)
```

Access the vSAN Health Service:

```python
vhs = vcMos[‘vsan-cluster-health-system’]
```

Then ‘vhs’ is the returned stub and can be used to call vSAN Health Service APIs.

vSAN ROBO Health Management

As stated above, the returned stub ‘vhs’ can be used to call vSAN Health Service APIs. The

```python
healthSummary = vhs.QueryClusterHealthSummary(
                  cluster=cluster, includeObjUuids=True, fetchFromCache=fetchFromCache)
```

‘healthSummary’ contains all the health information about a cluster. For example, retrieve

```python
clusterStatus = healthSummary.clusterStatus
```

Get each host’s health status:

```python
for hostStatus in clusterStatus.trackedHostsStatus:
    print("Host %s Status: %s" % (hostStatus.hostname, hostStatus.status))
```

By decomposing the structure in ‘healthSummary’, all aspects of the vSAN Cluster health information can be gathered, including hardware compatibility list (HCL) health information, network health information, disk health information, and so on.
vSAN ROBO Cluster Management

vSAN ROBO deployment is a special case of vSAN Stretched Cluster. To manage a vSAN ROBO Cluster, get to the ‘VsanVcStretchedClusterSystem’ stub first:

```python
vscs = vcMos['vsan-stretched-cluster-system']
```

After getting the ‘vscs’ ('vsan-stretched-cluster-system') stub, it can be used to do the management task related to vSAN ROBO deployment such as:

```python
witnessHosts = vscs.GetWitnessHosts(cluster)
print(" Unicast IP address used by witness host: %s" % (witnessHosts[0].unicastAgentAddr))
```

Check whether a specific host is a witness host or not:

```python
isWitnessHostResult = vscs.IsWitnessHost(host)
print('This host is%sa witness host.' % (' ' if isWitnessHostResult else ' not '))
```

Or, remove a witness host from a vSAN ROBO Cluster:

```python
removeWitnessTask = vscs.removeWitnessHost(cluster)
```

vSAN ROBO Deployment Space Management

The vSAN space report system is another useful API method. By calling this method, the administrator can get detailed capacity usage information about the whole cluster.

```python
vsrs = vcMos['vsan-cluster-space-report-system']
```

Then retrieve space related information:

```python
CapacitySummary = vsrs.QuerySpaceUsage(cluster=cluster)
clusterTotalCapacity = CapacitySummary.totalCapacityB
clusterFreeCapacity = CapacitySummary.freeCapacityB
```

6.4 vSAN Management API Summary

vSAN 6.2 released a complete vSAN Management API for administrators to manage, monitor, and troubleshoot vSAN. In the vSAN ROBO deployment environment, vSAN API is more required since the ROBO sites are geographically distributed. It is easier for administrators to manage the distributed sites through an automated way by leveraging vSAN API. This section describes the main procedure of working with vSAN API and a brief workflow of managing and monitoring vSAN ROBO Cluster. For a complete list of vSAN APIs and their usage, see the vSAN Management topic.
7. Conclusion

This section summarizes on how solution reference architecture confirms that a 2-node hybrid vSAN ROBO deployment can support a mixed workload that is scalable and high performing.
7.1 Conclusion

This solution validates the performance and functionality of mixed workloads, including Horizon virtual desktops, Microsoft SQL Server, and Windows File Sharing Server, in a virtualized VMware environment running in the 2-node hybrid vSAN ROBO Cluster.

vSAN and vSphere is an ideal and cost-effective platform for running a variety of virtual machine workloads requiring predictable performance and availability in ROBO environments. Important services and business-critical applications can benefit from shared storage without cost and complexity of dedicated storage hardware. vSAN makes it simple to add capacity using a scale-up or scale-out approach without incurring downtime so that maintenance windows are easier to schedule. In addition, the vSphere HA feature enables rapid recovery from unplanned failures. Moreover, vSAN 6.2 comes with a new suite of vSAN APIs, which can automate daily administrative tasks and allows for ease of management.

In sum, this solution reference architecture confirms that a 2-node hybrid vSAN ROBO deployment can support a mixed workload that is scalable, resilient, highly available, and high performing.
8. Reference

This section lists the relevant references used for this document.
8.1 White Paper

For additional information, see the following white papers:

- Microsoft SQL Server 2014 on VMware vSAN 6.1 Hybrid
- VMware Horizon View 6.0.2 and VMware vSAN 6.0 Hybrid

8.2 Product Documentation

For additional information, see the following product documentation:

- vSAN Management
- VMware vSAN 6.2 Stretched Cluster and 2-Node Guide
- VMware vSAN Network Design Guide

8.3 Other Documentation

For additional information, see the following document:

- VMware Compatibility Guide
9. About the Author and Contributors

This section provides a brief background on the author and contributors of this document.
9.1 About the Author and Contributors

- **Victor Chen**, Solution Architect in the Storage and Availability, Product Enablement team wrote the original version of this paper.
- **Catherine Xu**, Technical Writer in the Product Enablement team, edited this paper to ensure that the contents conform to the VMware writing style.