SQL Server Failover Cluster Instance on VMware vSAN Native

Table of contents

SQL Server Failover Cluster Instance on VMware vSAN Native

Executive Summary

Business Case

The modern-day CIOs are concerned with high performance, high availability, and low cost when planning their Database Management Systems (DBMS). Over the years, Clustering Databases have become the mainstream choice over standalone databases in production environments. Clustering improves the availability of SQL Server instances by providing a failover mechanism to a new node in a cluster in the case of physical or operating system failures. The most used model is the active-passive nodes creating an embedded base of clustering legacy systems that could benefit from today's new technologies, such as Hyperconverged Infrastructure (HCI) infrastructures.

VMware vSAN™ vSAN has been widely adopted as an HCI solution for business-critical applications like Microsoft SQL Server. vSAN aims at providing a highly scalable, available, reliable, and high-performance HCI solution using cost-effective hardware, specifically direct-attached disks in VMware ESXi™ hosts. vSAN adheres to a policy-based storage management paradigm, which simplifies and automates complex management workflows that exist in traditional enterprise storage systems with respect to configuration and clustering.

With the introduction of vSAN 6.7 Update3 (U3), vSAN now supports Windows SQL Server Failover Clusters Instances (FCI). This allows data center administrators to run workloads using legacy clustering technologies on vSAN. vSAN 6.7 can support the shared target storage locations when the storage target is exposed using the vSAN native for SQL servers.

vSAN Native Support for WSFC

vSAN 6.7 Update 3 introduced lots of new enhancements including unified management for VM and container storage with similar operational modes, intelligent operations to make more informed decisions, and optimize automated operations, and enhanced performance by increasing throughput and lower latency for business-critical applications. Specially, vSAN 6.7 Update 3 and later releases provides native support for virtualized Windows Sever

Failover Clusters (WSFC). It supports SCSI-3 Persistent Reservations (SCSI3PR) on a virtual disk level required by WSFC to arbitrate an access to a shared disk between nodes. Support of SCSI-3 PRs enables configuration of WSFC with a disk resource shared between VMs natively on vSAN datastores.

Currently the following configurations are supported:

- Microsoft SQL Server 2012 or later running on Microsoft Windows Server 2012 or later
- Up to 6 failover clustering nodes per cluster.
- Up to 64 shared virtual disks per vSAN cluster.
- vSAN Stretched Clusters with SCSI3-PR for shared disks are fully supported.

Note that Windows 2008 R2 (Data Center Edition) and SQL Server 2008 R2 RTM were successfully validated, including the deployment on vSAN and the migration steps from SAN to vSAN. It is recommended rechecking with Microsoft for general support lifecycle for a given version of a SQL Server and Windows OS.

Solution Overview

This technical white paper validates the solution of a Microsoft SQL Server Failover Clustering Instance using shared disks backed by vSAN native.

Key Highlights

This solution provides:

- Demonstration of the supportability of running an SQL Server Always On Failover Clustering Instance (FCI) on vSAN native.
- Showcase of the different sized OLTP database performances on an SQL Server Failover Clustering Instance on vSAN native.

- Validation of the application role failover.
- Migration from SAN to vSAN.

Solution Architecture

This solution is designed to size databases on vSAN. To demonstrate different purposed workloads, we emulated a tier-1 application database with the size of 300GB with RAID-1 for the data and log disks as well as emulated a tier-2 applicatoin database with the size of 600GB with RAID-1 for log disk but using RAID-5 for data disks. We used the same test client configuration to compare the performance difference, expecting the tier-1 database can have better performance on TPS and lower latency, and tier-2 is for higher capacity but light performance-oriented application.

To form the vSAN all-flash cluster we used four DELL PowerEdge R640 servers. 1U platform was employed for density, performance and scalability, all optimized for high application performance. Each VMware ESXi™ host contains two disk groups, and each disk group consists of one cache-tier NVMe SSD and four capacity-tier SAS SSDs. We configured pass-through mode for the capacity-tier storage controller.

The configuration consisted of seven Windows 2016 virtual machines on a vSAN Cluster as shown in Figure 1. We have three SQL Server clusters created using shared disks from vSAN native. Two clusters were for the performance test and one cluster is for migration validation purpose. For tight integration between WSFC and Active Directory, a domain controller was created in the vSphere and vSAN cluster.

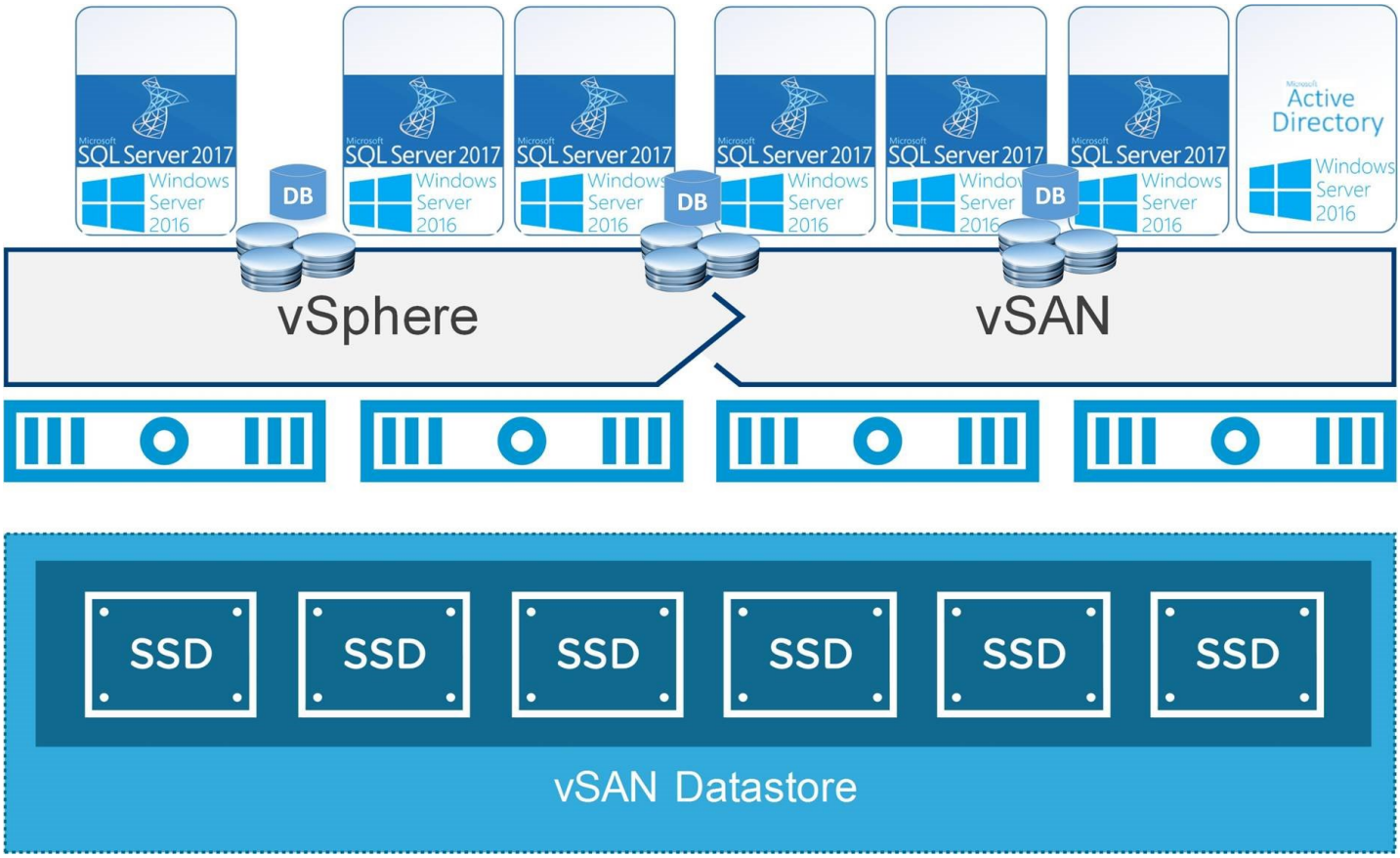

Figure 1. Solution Architecture

Solution Validation

This section covers configuration, SQL Server Failover Cluster performance on vSAN, application role failover test result and how to migrate SQL Server Failover Cluster from traditional SAN to vSAN.

We accomplished the following tests in this paper:

- Performance of FCI on vSAN
- Application role failover
- Migration of SQL Server from SAN to vSAN

Configuration

This section covers the performance test tool, SQL Server 2017 virtual machine configuration, database space consumption and vSAN configuration on VMware vSAN 6.7 U3 all-flash.

Performance Test Tools and Configuration

We used the Benchmark Factory for Databases to run tests with the desired parameters. We focused on the All-Flash vSAN aggregate performance of the 4-node cluster. Each test duration was set to one hour with 15-minute preconditioning and 45-minute sample period. We also used the vSAN Performance Service to monitor the vSAN performance.

SQL Server 2017 Virtual Machine Configuration

We followed Best Practice of SQL Server on vSphere to achieve the optimal performance of SQL Server on vSphere and vSAN cluster.

Database Space Usage

Configuring Scale Factor for database sizing. We configured three sized DB servers for the performance tests. We set SF=32 for medium-sized database, which created 300GB database; or a TPC-E-like OLTP user database with customer number of 32,000, which generated around 327GB data in the data files. We use SF=64 for 1TB database, or a TPC-E like OLTP user database with customer number of 64,000, which generated 625GB data in the data files.

vSAN Configuration and SPBM Policy

Deduplication, compression, encryption, and iSCSI target service have been disabled, while enabling performance service on the vSAN configuration. Two SPBM policies for different service level VMs were set to meet certain business requirement.

We used the RAID-1 with 100 percentage space reservation for both data and log virtual disks, for the 300GB database.

We used Erasure Coding (RAID-5) on the data file disks, but keep the RAID-1 for the log virtual disk, with 100 percentage space reservation for both data and log virtual disks, for the 600GB database.

SQL Server Virtual Machine and Disk Layout

Various sized databases on vSAN and different platforms were tested. Following are the virtual machine configuration and disk layouts.

For the TPC-E-like OLTP workload, the database size is based on the actual disk space requirement and additional space for the database growth. We designed 500GB virtual disk for 600GB database and 250GB virtual disk for 300GB database. Table 1 is the disk layout of the two-sized databases.

Table 1. SQL Server Disk Layout

Purpose	Number x Size (GB)
Operating system	1 x 100
Log disk	1 x 80
Data disks	2 x 250 (for 300GB database) 4 x 500 (for 600GB database)
Tempdb	1 x 60

SQL Server Failover Cluster Performance on vSAN Native

This test measured the impact of stressing a SQL Server 2017 on a Windows Server 2016 guest VM using Benchmark Factory for Databases to generate the TPC-E-like OLTP workloads.

Outline of the VM and vSAN Configuration to support SQL Server Failover Cluster Instance

The outline of the configuration of the virtual machines to support SQL Server FCI is as follows.

24 vCPUs and 64GB RAM with memory reservation on the 300GB and 600GB database VMs, for both active and passive node of the two WSFC. Both SQL Server instances were set with the maximum and minimum memory of 51GB. We used the Paravirtual SCSI controllers for the data and logs disks, and set the **SCSI Bus Sharing** to **Physical**. Figure 2 illustrates settings for one of the nodes with 600GB database.

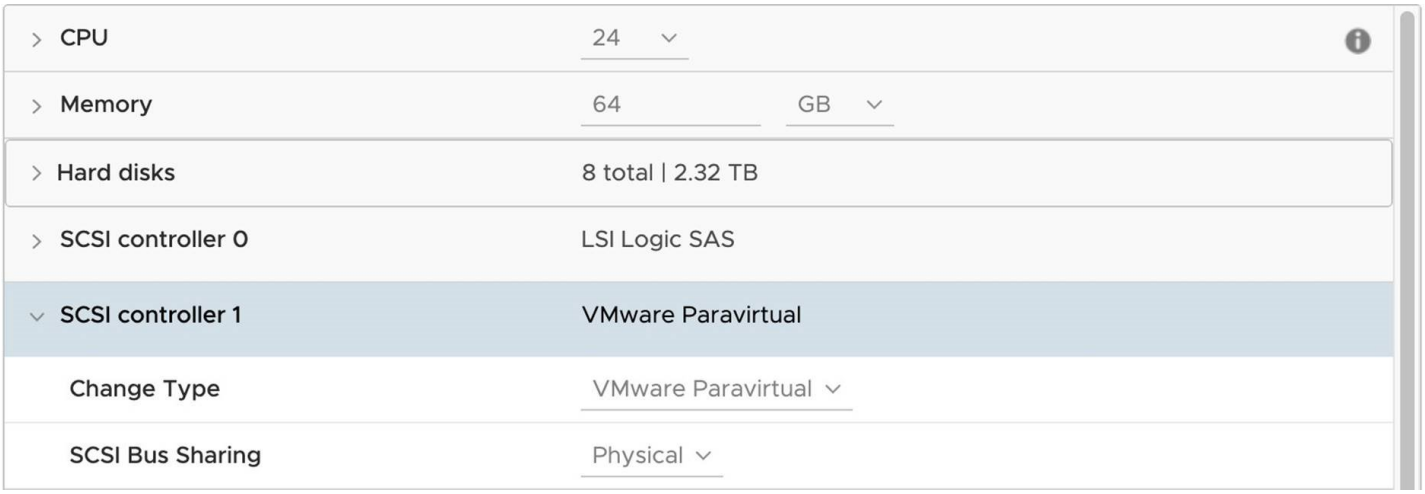

Figure 2. SCSI Controller Setting for WSFC Shared Disks

For the 300GB database VMs, the vison policy is RAID-1 and 100% space reservation for both data and log disks, and for the 600GB database VMs, the vison policy is RAID-1 for log and RAID-5 for data disks, and 100% space reservation for both data and log disks. Figure 3 is an example for the policy and SCSI controller setting for one of the data disks of the 600GB database VM.

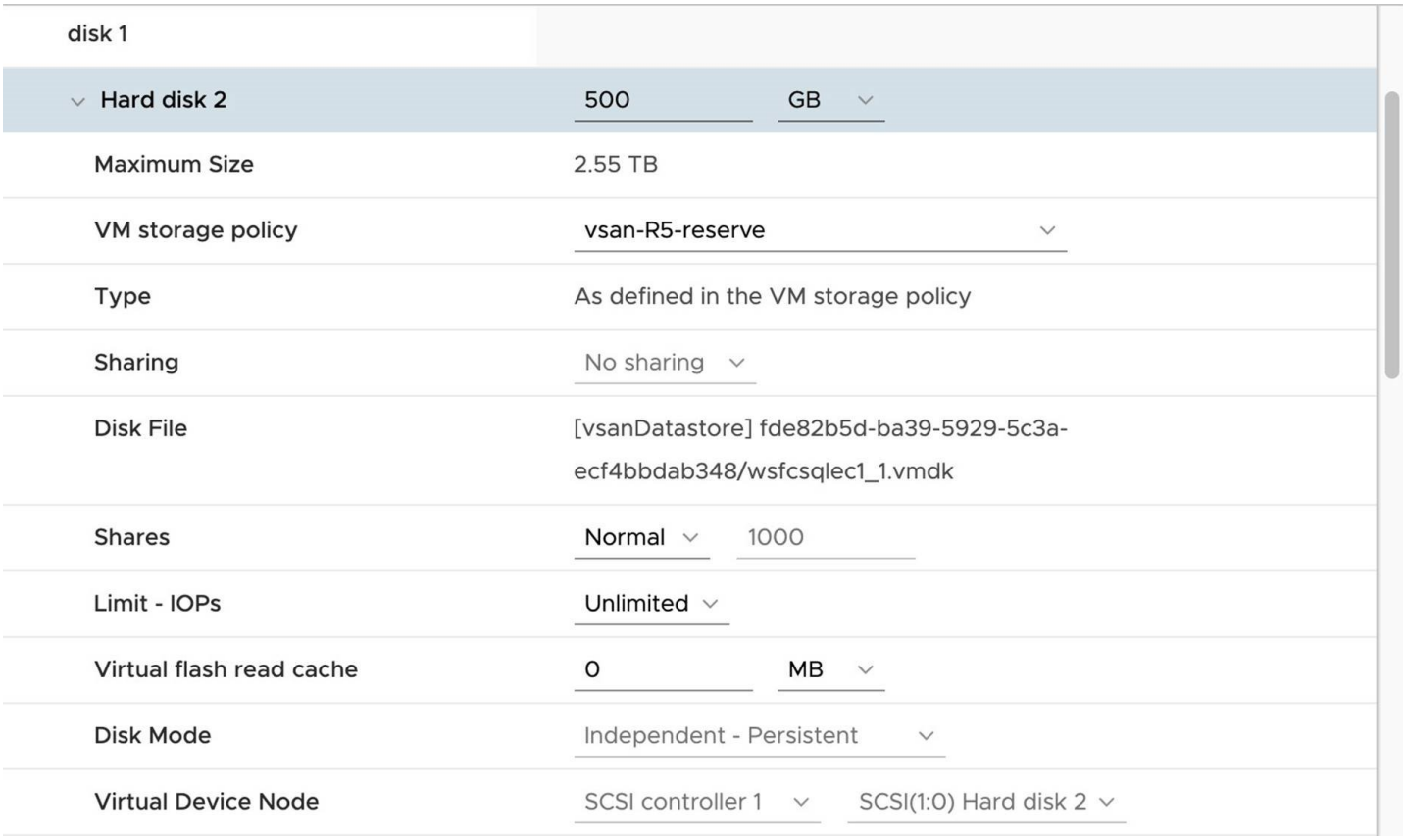

Figure 3. Policy Setting for Data Disk (RAID-5 with 100% Space Reservation)

Figure 4 shows the SQL Server Failover Cluster Instance and 7 shared disks from vSAN native supporting the 600GB database as an example.

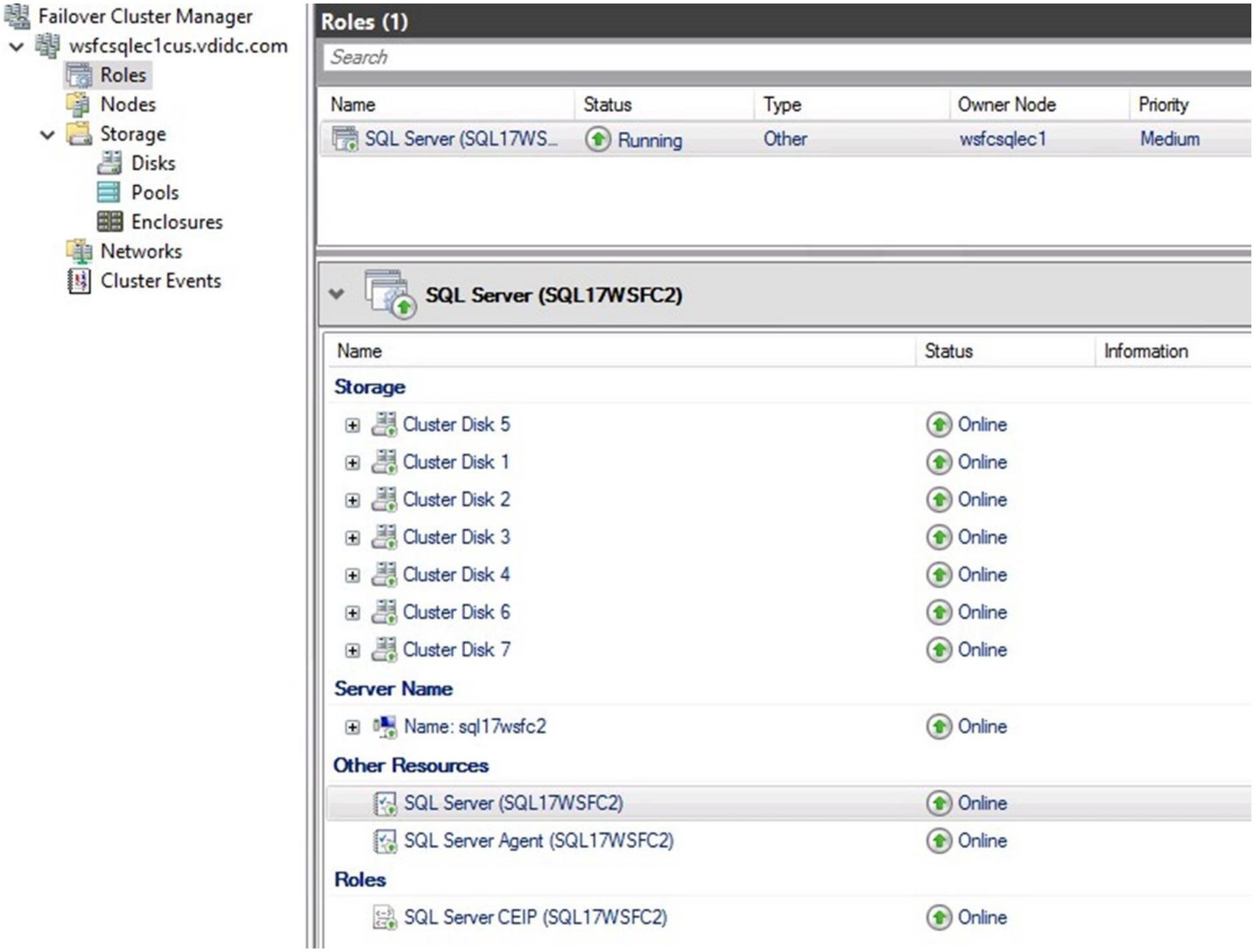

Figure 4. SQL Server Failover Cluster Instance and Cluster Disks

Performance Results

We ran the TPC-E-like OLTP workloads concurrently against the two different-sized databases on the active node of the two SQL Server clusters. The number of users, which generate OLTP activity, were adjusted as needed to saturate the host. Repeatability was ensured by restoring a backup of the database before each run. We kicked off the test concurrently from the test clients on the management cluster remotely, to avoid the side effect of the clients. The test duration was one hour with 15-minutes warm-up duration. Figure 5 shows the test results, and you may find that the TPS of the 300GB was 2,431 on average, and the average transaction time in milliseconds was 0.032 (32.41ms). TPS of the 600GB was 2,271 on average, and the average transaction time in milliseconds was 0.035 (34.50ms).

The OLTP performance is highly compute and storage capability dependent. We demonstrated that with different SPBM settings on different sized databases, vSAN provided the capability to keep TPS beyond 2,200, which kept the average transaction time at around 0.03 seconds.

Table 2. TPS and Average Transaction Time in Milliseconds

DB Size	TPS	Average Transaction Time (ms)
300GB	2,431	32.41
600GB	2,271	34.50

Figure 5. Failover Cluster SQL Server Cluster OLTP Performance

Table 3 is the vSAN VM Performance of the 300GB and 600GB databases with different SPBM policies.

Table 3. vSAN VM Performance of the Concurrent Workload of the 300GB and 600GB Databases

vSAN VM Performance			
Read IOPS	Write IOPS	Read Latency (ms)	Write Latency (ms)
11,200-21,300	1,670-2,700	0.47-0.52	1.99-3.43

Application Role Failover across WSFC Nodes

Application Roles are clustered services such as Failover Cluster Instance (FCI) or generic file server. An FCI is a SQL Server instance that is installed across nodes in a WSFC. This type of SQL Server Instance depends on resources for storage and virtual network name. The storage can use Fibre Channel, iSCSI, FCoE, SAS, or use locally attached storage for shared disk storage. Those storages should provide SCSI-3 locking and the requirement applies to other service running on WSFC too. And this is now supported on native vSAN.

We verified the SQL Server role failover and demonstrated the capability of Failover Cluster and advantages of vSAN native.

Application Role Failover

We verified the role failover for both SQL Server FCI and File Server; developed an application using BMF REST API to detect the job status, and to check that the job was running. If the job was stopped, we restarted the job immediately.

Our test results showed:

- The SQL Server application role failover duration was nearly 20 seconds to bring online both instances hosting two-sized databases with different number of shared disks.
- Using REST API can detect and trigger the job restart in 8 seconds, but the test application needs some time to prepare and restart the test. Shown in the table below, test client needed 48 seconds and 77 seconds to restart the application. This duration was longer than the instance failover time. That means if the failover duration (20 seconds) can be shorter than the restart duration of the application (48 seconds and 77 seconds), the failover will not cause the connection issue. Developer can use the failover duration as reference for their application timeout setting or try-catch-retry logic.

Table 4. Application Role Failover Duration

Database size and disk number	**Failover cluster bring online the resource**	**BMF restart job duration**
300GB (5 shared disks)	20 seconds	48 seconds
600GB (7 shared disks)		77 seconds

Migration from SAN to vSAN for SQL Server Cluster on WSFC

In this scenario, we accomplished migration for a SQL Server cluster with two nodes, from traditional SAN, emulated by using DellEMC UnityVSA using pRDM through iSCSI interface (it also support FC or FCoE configured pRDMs), to a vSAN. This procedure is applicable to the migration using real SAN storage in the same configuration.

Configuration before Migration

We deployed UnityVSA for pRDM emulation purpose. Four LUNs were created with the capacity and usage described in Table 5. We created two nodes Windows Failover Cluster, and put the two VMs to a shared Datastore from UnityVSA, and created 3 LUNs being as the pRDM to be added to the two VMs for Cluster witness disk, SQL Server system databases and user database. We created a 50GB OLTP database using Benchmark Factory for Databases and stored it into the 100GB LUN.

Table 5. LUNs for pRDM Demonstration from UnityVSA

Name	Size (GB)	Purpose
VM Home LUN	400	Shared datastore to store VM OS
SQL Home Directory LUN	35	SQL Server system database disk
SQL Database LUN	100	SQL Server user DB data and log disk
Cluster Witness LUN	25	WSFC quorum disk

Figure 6 shows the created LUNs from UnityVSA for the preparation of the migration demonstration.

Figure 6. LUNs for pRDM Demonstration

We created iSCSI interface on the UnityVSA and connected to the interface from the four ESXi hosts. As shown in Figure 7, the disks can be accessed by each ESXi host through the iSCSI software initiator.

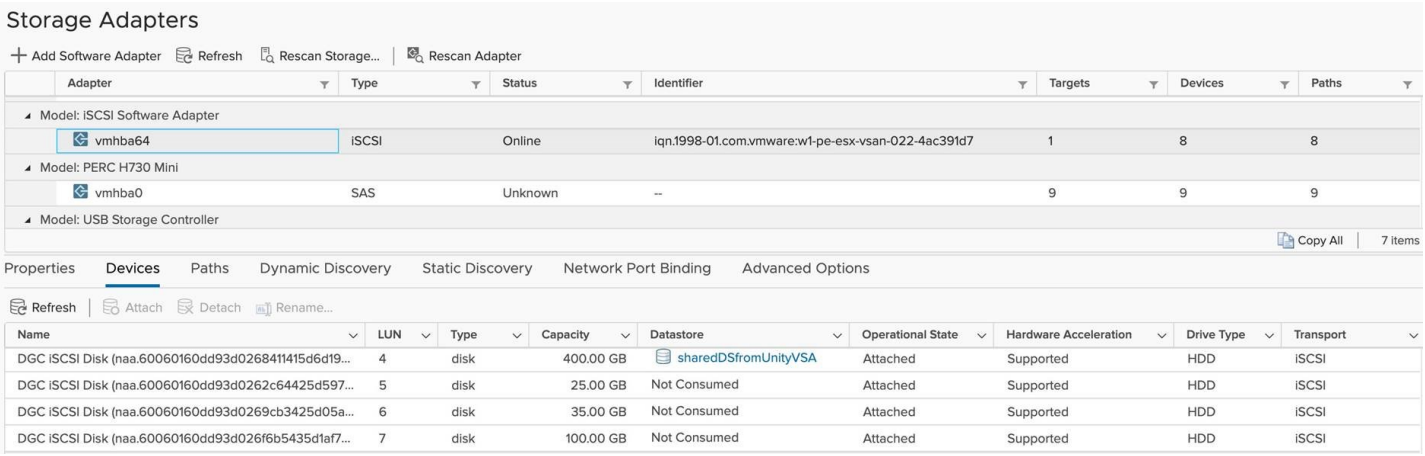

Figure 7. Connect to ESXi Host via iSCSI Interface

The disk exposed as RDM in physical compatibility mode or pRDM to SQL Server, virtual machine will have fixed capacity and using the physical mode of the SCSI Controller.

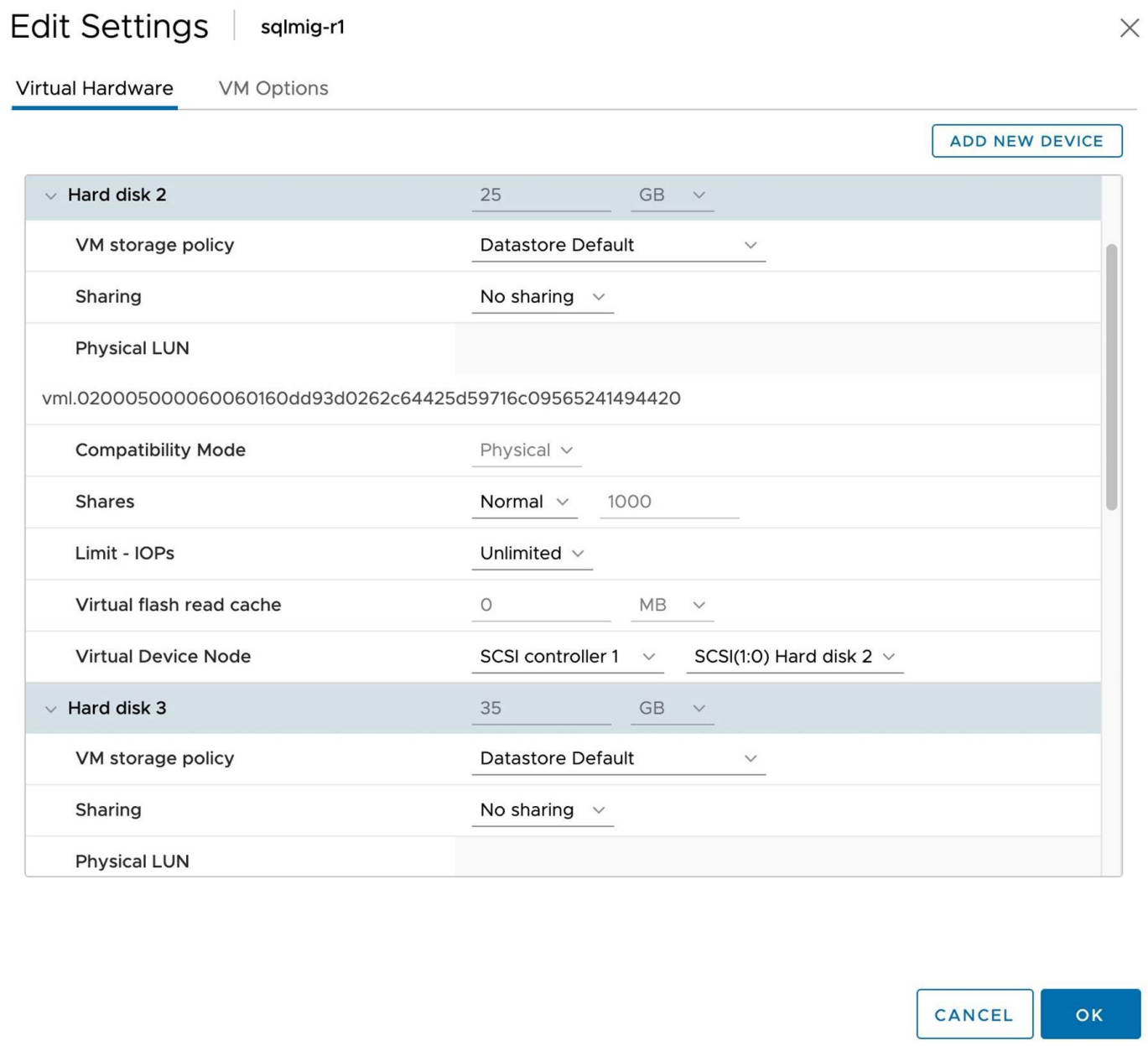

Figure 8. Add pRDM Disks to SQL Server Virtual Machine

Migration Steps

Note that before migration, backup is highly recommended to avoid potential data loss. This migration operation is offline, and the duration is mainly for data copy. Make sure the offline time window is enough before moving forward.

To migrate the SQL Server FCI cluster, using pRDMs as clustered disk resources to vSAN, follow the steps below:

1. Stop the SQL Server Cluster Role from the Windows Failover Cluster Manager
2. Shut down all the VMs hosting nodes of Windows Failover Cluster gracefully, by clicking Power-> shut down guest OS or within the Guest OS
3. Migrate the first node of a cluster to vSAN by choosing Change storage only in the Migrate wizard. The migration process will convert pRDMs to VMDKs, and apply the desired vSAN policies for clustered disks in the migration wizard.

4. Power on the first node and validate that clustered disk resources are visible in the Windows Failover Cluster Manager and SQL Server Cluster Role can be started, and you may keep it online.
5. Detach pRDMs used to host clustered disk resources from all remaining nodes of the cluster, which are not migrated to vSAN yet.
6. (Optional) Migrate the remaining nodes to vSAN, if non-shared disks are planned to migrated to vSAN as well.
7. Attach disk resources back to remaining nodes of the cluster pointing to VMDKs from the first node stored on the vSAN datastore. Ensure that vSCSI controllers hosting disks are configured to use physical mode, and share the VMDKs across virtual machines for the previous pointing disks on the cluster nodes.
8. Start up the virtual machines one by one and make sure the SQL Server Cluster Role is online on the first node, try failover from the active node to the passive node to check if the other nodes can start SQL Server Cluster Role normally.

Figure 8 shows the disk was provisioned by vSAN instead of from UnityVSA and you can change the policy and size according to the requirement.

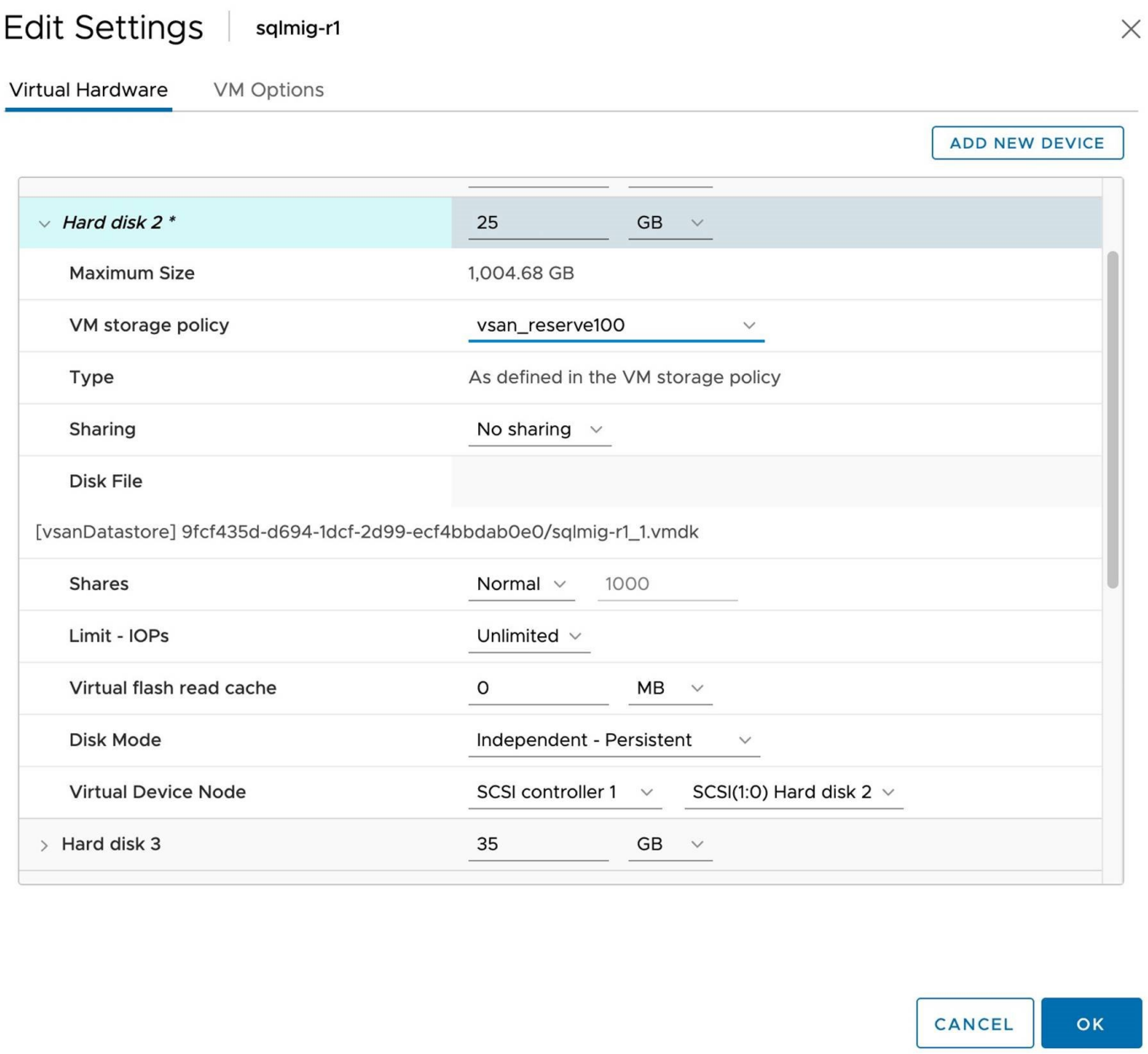

Figure 9. Disks on vSAN after vMotion

After following the steps above, the shared virtual disks of SQL Server cluster were on vSAN. Users can manage the virtual disks by

using vSAN policy including expending the disk, changing policy to follow the best practice to run SQL Server on vSAN to meet different business requirements.

Failback or Rollback from vSAN to SAN

The recommended migration and rollback way is to migrate a first node, power on the VM, and start up the SQL Server Cluster role. If it is failed due to the disk issue, all other nodes still have the pRDMs attached and can be started immediately. You may add back the pRDM disks to the first node following removing the virtual disks from vSAN.

If you want to roll back the configuration from vSAN to SAN, given the previous steps did not delete data from the RDM disks, you may follow the steps below to migrate disks back to SAN using pRDMs.

1. Stop the SQL Server Cluster Role from the Windows Failover Cluster Manager
2. Shut down all the VMs of the Windows Failover Cluster gracefully, by clicking Power-> shut down guest OS
3. Remove all the virtual disk from the VMs
4. Share the RDM disks across virtual machines using previous setting for disks under different SCSI controllers.
5. Start up the virtual machine and bring the SQL Server Cluster Role back online from Windows Failover Cluster Manager

Conclusion

VMware vSAN is optimized for modern all-flash storage with efficient near-line deduplication, compression, and erasure coding capabilities that lower TCO while delivering incredible performance.

- Demonstrated that with different SPBM settings on different sized databases hosted by FCI, vSAN provided the predicatable performance at TPS and average transaction time.
- Verified the SQL Server role failover and demonstrated the capability of SQL Server Cluster and advantages of the supportiblity of vSAN native to WSFC.
- Provided the detailed steps and guide to migration SQL Server cluster from traditional SAN to vSAN and failback methods and validated the steps in this solution.

vSAN 6.7 U3 and later releases support running WSFC natively by sharing VMDKs through SCSI-3 Persistent Reservations (SCSI3-PR). This is configured in VM configuration only without complicated LUN settings, and there is not needs for pRDMs from legacy storage. The generic support for WSFC without limitation provide the maximum flexibility for you to deploy Windows Failover Clustering on vSAN native.

References

- Microsoft Windows Server Failover Clustering on VMware vSphere 6.x: Guidelines for supported configurations
- Architecting Microsoft SQL Server on VMware vSphere Best Practices Guide
- Benchmark Factory for Database
- vSAN Resource

About the Author

Tony Wu, Senior Solution Architect in the Solution Architecture team of the HCI Business Unit wrote the original version of this paper.

Oleg Ulyanov, Senior Solution Architect in the Cloud Platform Business Unit also contributed to this paper.