Dell EMC Ready Solutions for Microsoft SQL: Design for Dell EMC XtremIO

With PowerEdge R840, XtremIO X2, Windows Server 2016, and RHEL 7.6

May 2019

H17593.1

Reference Architecture Guide

Abstract

This guide describes a highly scalable architecture for SQL Server using XtremIO X2 all-flash storage with PowerEdge servers. It details design principles, configuration best practices, and validation with Windows Server 2016 and RHEL 7.6.

Dell EMC Solutions

DELL EMC

Copyright

The information in this publication is provided as is. Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Intel, the Intel logo, the Intel Inside logo and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Other trademarks may be trademarks of their respective owners.
Published in the USA 05/19 Reference Architecture Guide H17593.1.

Dell Inc. believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

Contents

Executive summary

Overview

Today's IT organizations are under pressure to achieve cloud-like elasticity, scalability, and ease of provisioning while lowering the total cost of ownership—a challenging goal, particularly with the complexity of databases. To achieve the anticipated business outcomes requires a clear statement of associated success metrics that often must be negotiated with the executive sponsor. After the success metrics have been defined, the next challenge for the IT organization becomes selecting the technologies that will meet or exceed the success metrics.

Traditionally, selecting a database infrastructure was a long, exhaustive process with a wide variety of complexities that took months to resolve. What if there was a solution where all the components were validated and tested together using real SQL Server workloads? Prevalidation engineering means that the database solution is a proven platform, thus removing most of the complexity and time that is associated with manual integration work. Testing such a solution is more complex. A simple online transaction processing (OLTP) workload test only shows how the system performs if it is solely dedicated to one database. This approach is great for showcasing strong performance but falls short when you want to measure the performance of a multiple-database ecosystem.

A better test is to show how the database solution scales while supporting several SQL Server databases. Scalability is the capability of the database solution to support existing workloads with the potential to accommodate more databases for future growth. The traditional challenge with servers and storage has been growth of the database ecosystem, which has among the most resource-demanding and latency-sensitive applications. For example, as more databases are added to existing infrastructure, processor and storage contention can affect performance. Scalability in the cloud era means greater growth potential while performance remains consistent. Tradeoffs, such as sacrificing application responsiveness due to the growth of the database ecosystem, affect performance and cost of ownership. Today's IT organizations are looking for database solutions that offer far greater scalability and performance to meet their success metrics.

The new Ready Solutions for Microsoft SQL Server reference architecture has been validated with Dell EMC PowerEdge servers and Dell EMC XtremIO X2 all-flash storage. In addition to validating the solution with SQL Server, the Dell EMC labs pushed the boundaries of scalability testing by running 16 virtualized databases in parallel—8 on Windows Server 2016 VMs and 8 on Red Hat Enterprise Linux (RHEL) VMs. Key test findings include:

- The PowerEdge R840 servers demonstrated strong scalability. The database load on each of the two PowerEdge servers meant oversubscription of CPUs to the virtual machines. Each server had significant unused processor resources while delivering on performance.
- The XtremIO X2 array delivered sub-500-microsecond latencies while supporting 275,000-plus IOPS with 72 flash drives. The achievable IOPS per the XtremIO X2 specification sheet is 220,000 IOPS. We found no tradeoff between IOPS and latency during our tests on the XtremIO X2 array.

- XtremIO X2 inline data reduction technology reduced the size of a 1 TB SQL Server database to 239 GB, for a data reduction ratio of 3.52 to 1.
- The reference architecture delivered substantial consolidation savings:
  - One PowerEdge R840 server supported eight virtualized SQL Server databases.
  - XtremIO X2 inline data reduction savings enabled greater consolidation on all-flash storage.

This guide provides a detailed overview of the test findings, including a review of performance differences between SQL Server running on Windows Server and RHEL.

This reference architecture offers a great degree of sizing flexibility to meet business requirements. You can start with a minimal configuration that can grow incrementally or with larger configurations to support hundreds of databases. Having been validated with SQL Server, the architecture enables accurate sizing and faster time-to-value.

Audience

This guide is for database administrators, system engineers, IT managers, system administrators, storage administrators, and architects who design and maintain database infrastructures. Readers should have some knowledge of Microsoft Windows Server, Microsoft SQL Server, VMware virtualization, Dell EMC PowerEdge servers, Dell EMC storage, and Dell EMC networking products.

We value your feedback

Dell EMC and the authors of this document welcome your feedback. Contact the Dell EMC Solutions team by email or provide your comments by completing our documentation survey.

Authors: Sanjeev Ranjan, Mahesh Reddy, Vaani Kaur, Sam Lucido, Karen Johnson

Note: The Microsoft SQL Info Hub for Ready Solutions on the Dell EMC Communities website provides links to additional documentation for Ready Solutions for Microsoft SQL.

Architecture overview

Physical architecture

The following figure shows the physical architecture of this reference architecture for Microsoft SQL.

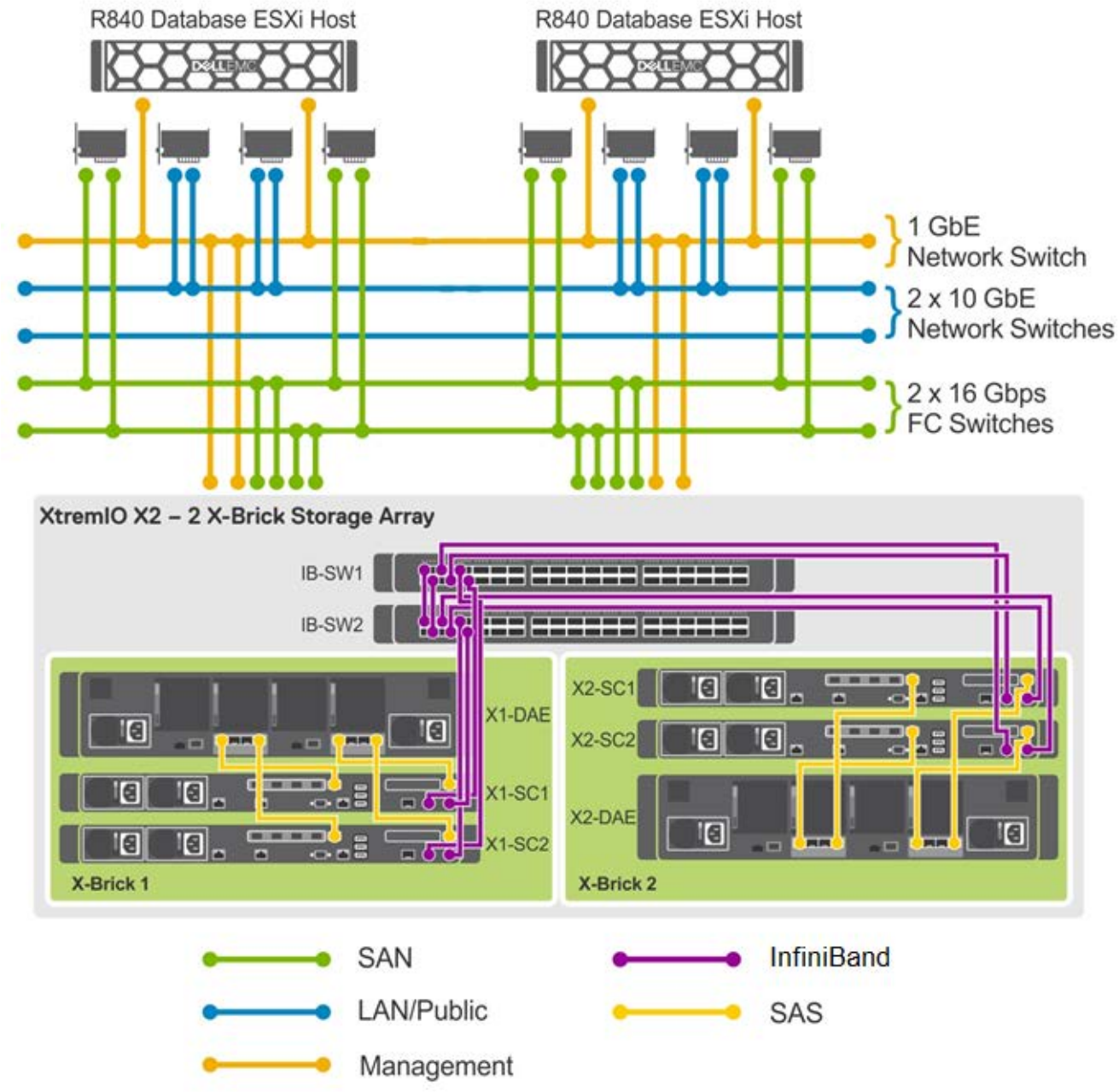

Figure 1. Physical architecture

Server layer

The server layer consists of two PowerEdge R840 servers. We configured ESXi 6.7 bare-metal hypervisors on both nodes to create two separate virtual environments—one with RHEL 7.6 and the other with Microsoft Windows Server 2016.

The following table lists the network components of each host node.

Table 1. Host-node network components

Component	Function
2 x dual-port 10 GbE network interface controllers (NICs)	SQL Server traffic
2 x dual-port 16 Gbps host bus adapters (HBAs)	SAN traffic
1 x 1 GbE remote Network Daughter Card (rNDC) port	In-band management of the server from within the operating system
2 x 10 GbE ports	Quest Benchmark Factory benchmarking traffic
1 x 1 GbE Integrated Dell Remote Access Controller (iDRAC) Ethernet port	Out-of-band management of the server

Network layer

The network layer consists of:

- **Two 10 GbE network switches**—Connect to two 10 Gb ports on the database server to route the SQL Server traffic
- **Two 16 Gbps Fibre Channel (FC) fabric switches**—Route SAN traffic between the R840 database/ESXi host and the XtremIO X2 storage array
- **One 1 GbE network switch**—Routes all management traffic between the components—ESXi hosts, management server, switches, and XtremIO X2 storage array

Storage layer

We used one XtremIO X2 storage array as FC SAN storage to test the SQL Server 2017 databases. The storage layer consists of:

- An XtremIO X2 cluster with two X-Brick modules, with a total of 72 x 2 TB flash-based Serial Attached SCSI (SAS) solid-state drives (SSDs)
- Two controllers and one disk array enclosure (DAE) on each X-Brick module
- Four 16 Gbps front-end FC ports
- Two InfiniBand switches for two X-Brick connections
- An XtremIO management server (XMS) for managing the storage array on the PowerEdge R640 management server

The LAN and SAN design includes redundant components and connectivity at every level to ensure that no single point of failure exists. This design ensures that the application server can reach the database server and the database server can reach the storage array if a component fails. The design provides protection even if a failure occurs in one or

more NICs or HBA ports, one LAN or FC switch, one or more XtremIO X2 front-end ports, or one XtremIO X2 X-Brick controller.

Logical architecture

The following figure illustrates the logical architecture of the database environment, including the multiple layers of infrastructure components.

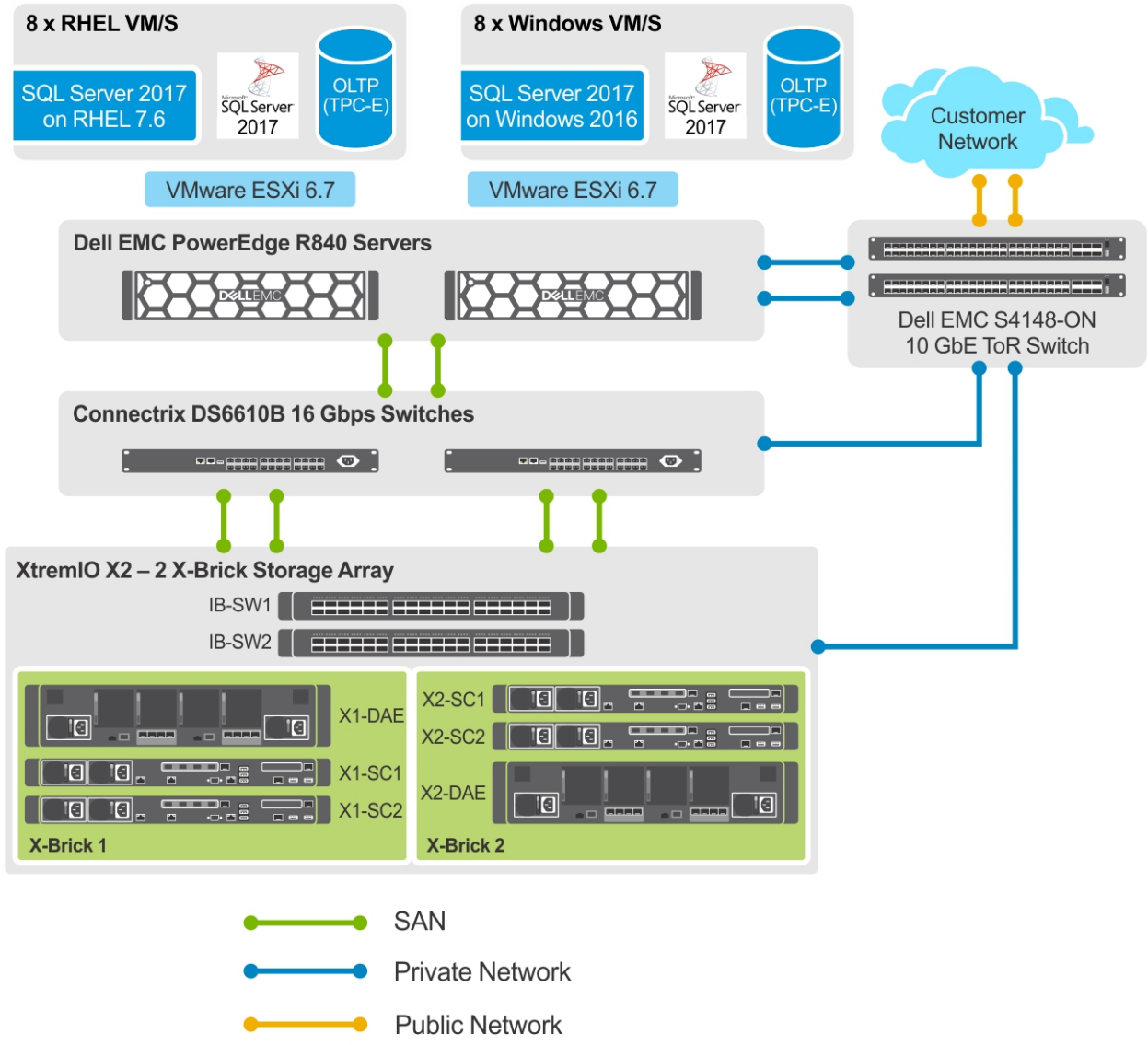

Figure 2. Logical architecture

We tested and validated this reference architecture for SQL Server with eight virtual machines (VMs) in these environments:

- SQL Server 2017 running on the RHEL 7.6 operating system
- SQL Server 2017 running on the Windows Server 2016 operating system

Each PowerEdge R840 server, which hosts ESXi 6.7, has four 18-core CPUs and 1,536 GB RAM. The VMs use RHEL 7.6 or Windows Server 2016 as the guest operating system that runs a SQL Server 2017 stand-alone database.

In this reference architecture, the management server runs VMware vCenter Server Appliance (VCSA), XMS, and the Benchmark Factory benchmarking tool that are deployed in separate VMs.

The storage layer hosts the storage volumes of all 16 databases. For more details about the storage layout for both Windows and RHEL environments, see Storage layer on page 7.

Hardware and software components

The following two tables summarize the hardware and software components that we used in testing and validating this reference architecture.

Table 2. Hardware components

Component		Details
2 x PowerEdge R840 servers	Chassis	4-CPU configuration
	Memory	24 x Samsung DDR4 Quad Rank 64 GB @ 2666 MHz
	Processor	4 x Intel Xeon Gold 6154 CPUs @ 3.00 GHz with 18C
	FC HBA	2 x QLE2692 dual-port 16 GB FC to PCIe Gen3 x8
	rNDC	Intel 4P X550-t
	Add-on NIC	2 x Intel Ethernet X520 server adapters
	Power supply	2 x Dell 2260W power supply modules
	RAID controller	Dell H740P
	iDRAC	iDRAC9 Enterprise
	Physical disk	3 x 1.2 TB SAS HDDs
Network switch	2 x Dell EMC Networking S4148F-ON 10 GbE	
FC switch	2 x Dell EMC Connectrix Gen6 6610B	
All-flash SAN storage	1 x Dell EMC XtremIO X2 array • 72 x 2 TB SAS flash drives • 2 x XtremIO X-Brick modules	

Table 3. Software components

Component	Details
Hypervisor	VMware ESXi 6.7 with vCenter 6.7
Windows operating system	Microsoft Windows Server 2016
Linux operating system	RHEL 7.6
DBMS	Microsoft SQL Server 2017
XtremIO operating system	6.2.0-85
XMS	6.2.0-85

Design considerations

ESXi host configuration

We configured the R840 database server/ESXi host as follows:

- Installed ESXi 6.7 U1 using the Dell EMC customized ISO image: Version A03, Build# 10764712.
- Zoned two dual-port 16 Gbps HBAs, four initiators in total, and configured them with the XtremIO X2 front-end FC ports for high bandwidth, load balancing, and highly available SAN traffic. For details, see FC fabric connectivity and zoning.
- Configured one 1 Gb Ethernet rNDC or LAN on motherboard (LOM) port for the management traffic and two 10 GbE ports for the SQL Server traffic. For details, see Virtual network design.
- Created multiple VMs with RHEL 7.6 or Windows Server 2016 as the guest operating system for the virtual SQL Server stand-alone databases. For more details, see Virtual machine configuration.

We configured, monitored, and maintained the ESXi host, virtual networking, and the VMs using VMware vSphere Web Client, ESXi Shell access, and VMware vCenter Server Appliance. We deployed vCenter Server Appliance as a VM on the management server.

For configuration procedures, see Appendix B: Configuring vSphere.

Multipathing configuration

The XtremIO X2 array supports vSphere Native Multipathing (NMP) technology. Multipathing increases efficiency of sending data over redundant hardware paths that connect PowerEdge servers to XtremIO X2 storage. Benefits include alternating I/O by using round-robin to optimize use of the hardware paths and more evenly distribute the data. Also, if any component along the storage path fails, then NMP resets the connection and passes I/O using an alternate path. For the XtremIO X2 array, we followed these best practices:

- Retained the default selection of round-robin as the native path selection policy (PSP) on the XtremIO X2 volumes that are presented to the ESXi host
- Changed the NMP round-robin path switching frequency from the default value (1,000 I/O packets) to 1

FC fabric connectivity and zoning

The following figure shows the recommended FC connectivity between the HBAs and the FC switches and the connectivity between the FC switches and the XtremIO X2 storage array. As shown, each server HBA port connects to two separate FC switches, and the two front-end ports on each XtremIO X2 array controller connect to the same FC switches.

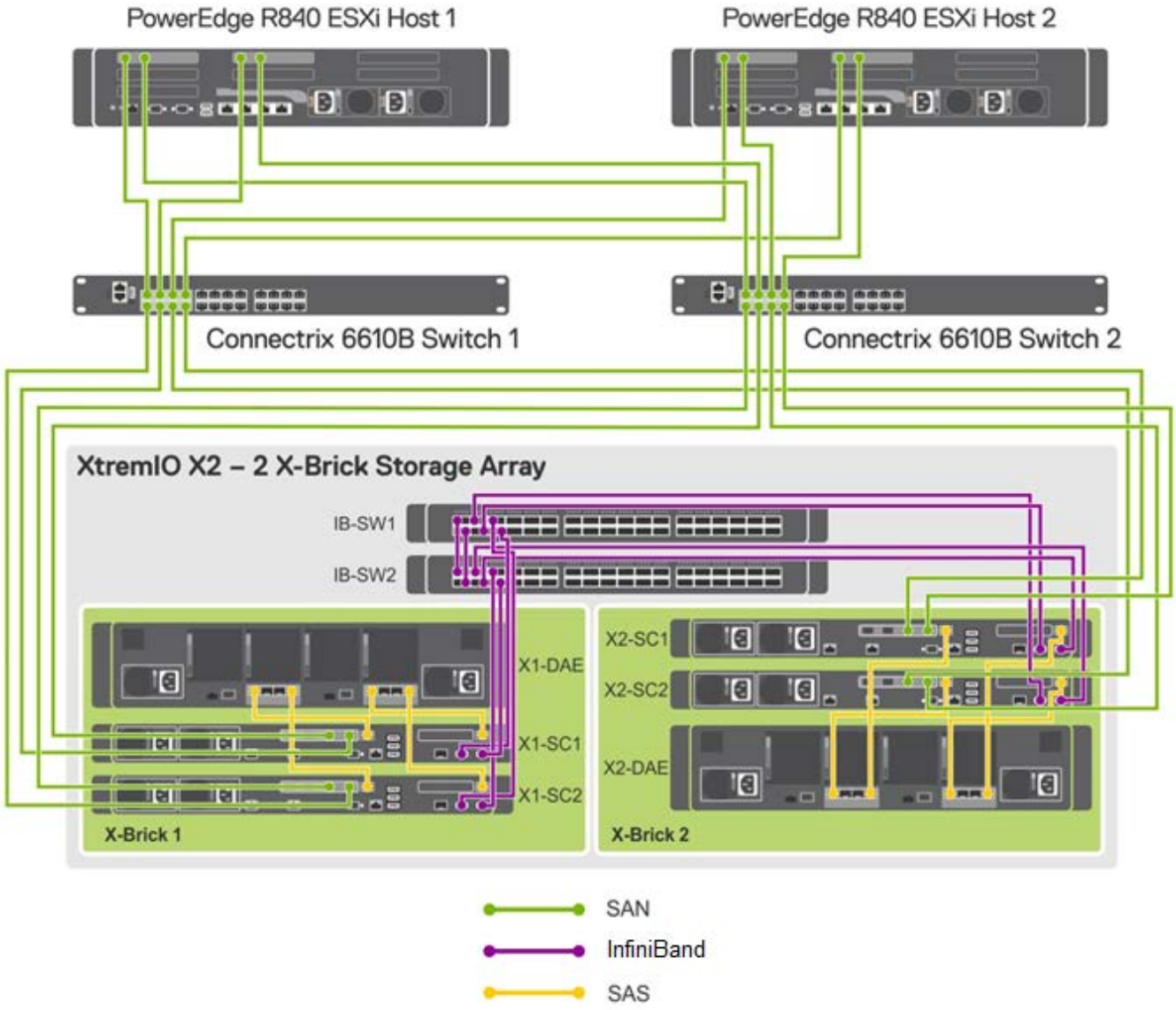

Figure 3. FC fabric connectivity design

Virtual network design

The following diagram provides a high-level overview of the virtual network design that we implemented in the ESXi hosts. The diagram also shows the mapping between the virtual switches and the physical switches.

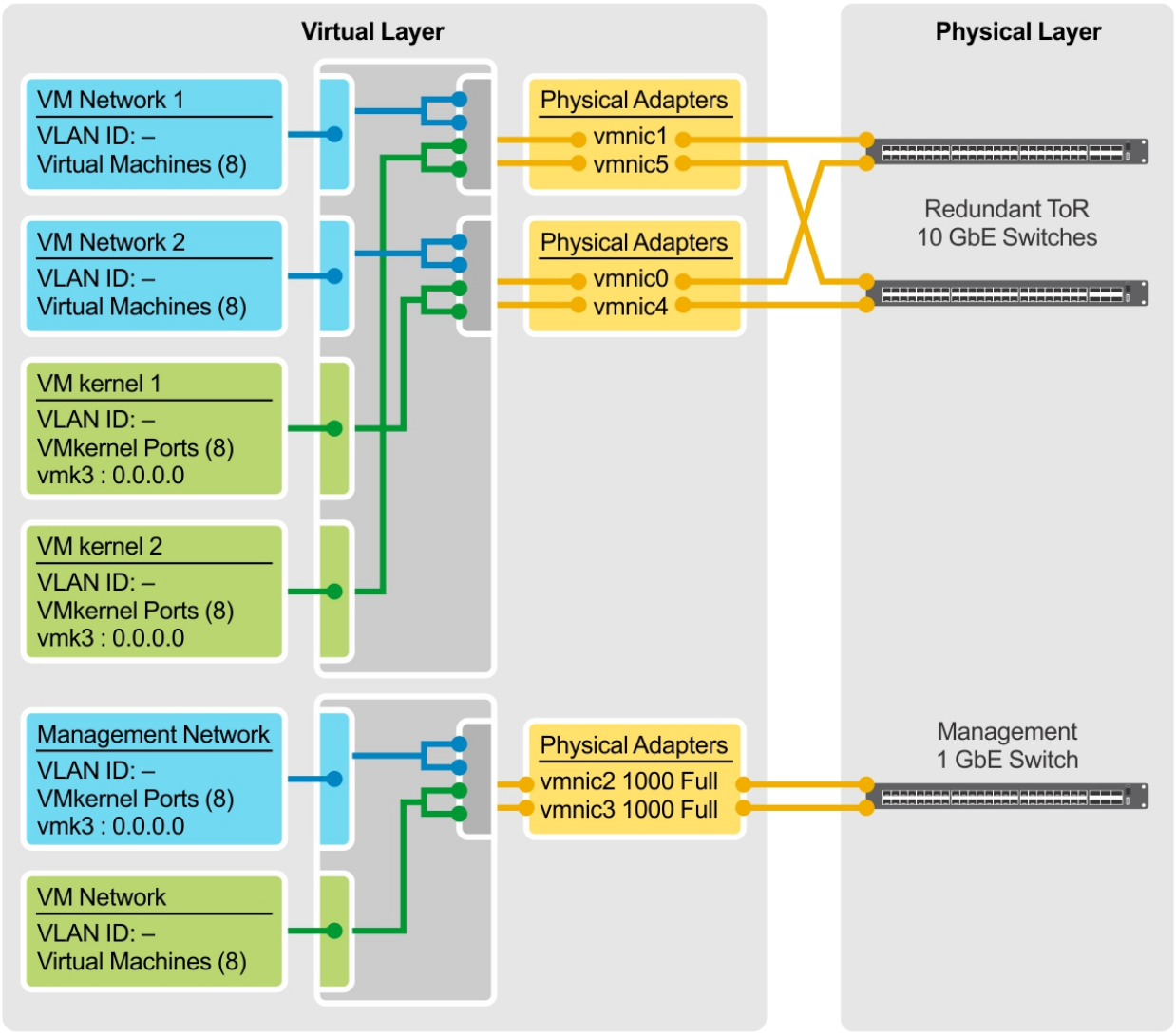

Figure 4. Virtual network design in the ESXi hosts

We configured the virtual networking as follows:

1. Created vSwitch0 for the management network on top of the vNIC1 and vNIC2.
2. Created another two standard switches—vSwitch1 and vSwitch2—one for VM migration and other network traffic, and the other for the SQL Server workload. We created these virtual switches on top of the two active adapters—the first port from the first network card and the first port from the second network card.
3. Created VMkernel adapters—VMkernel 1 and VMkernel 2—for vSwitch 1 and vSwitch2.
4. Added the network adapters to all the VMs.

Virtual machine configuration

We used the following design principles and best practices to create the VMs in this reference architecture:

- **SCSI controllers**—We created multiple SCSI controllers to optimize and balance the I/O for the different database disks, as shown in the following table.

Table 4. SCSI controller properties set in VMs

Controller	Purpose	Controller type
SCSI 0	Guest operating system disk	VMware ParaVirtual
SCSI 0	Database backup disk	
SCSI 0	tempdb data and log file disk	
SCSI 1	Database data file disk 1	
SCSI 2	Database data file disk 2	
SCSI 3	Database log file disk	

- **Datastore mappings**—We assigned the following properties to all database-related virtual disks such as DATA, Log, TempDB, and Backupdata:
  - **Type: Thick provision eager zeroed**

    We selected **Thick provision eager zeroed** to ensure that the space required for the virtual disks is allocated at creation time and the data on the physical device on the storage is zeroed out.

  - **Sharing: No sharing**

    We selected **No sharing** because the deployed databases are stand-alone databases that do not require sharing database virtual disks with another VM, unlike clustered VMs that share virtual disks between two or more VMs in the database cluster.

  - **Disk mode: Independent persistent**

- **VM vCPU and vMem**—The following table lists the distribution of virtual CPU (vCPU) and virtual memory (vMem) to the database VMs.

Table 5. VM configuration: vCPU and vMem

Number of vCPUs	vMem	
	Reservation	Total
16	128 GB	128 GB

- **Enable disk UUID**—In each of the VM options, we added the configuration **disk.EnableUUID** parameter and set it to **TRUE**. This setting ensures that the VMDK always presents a consistent UUID to the VM.

Guest operating system configuration

To install and configure the Windows Server 2016 and RHEL 7.6 guest operating systems, see the following VMware documents:

- *Guest Operating System Installation Guide: Windows Server 2016*
- *Technical Note: Installing and Configuring Linux Guest Operating Systems*

SQL Server configuration

To install and configure the SQL Server 2017 stand-alone database, see the following Microsoft instructions:

- Install SQL Server (Windows)
- Quickstart: Install SQL Server and create a database on Red Hat

Storage layout

To validate this reference architecture, we created six separate volumes for each VM on the XtremIO X2 storage array to ensure segregation of different I/O patterns on separate volumes. We placed operating system, backup, database data files, database log files, and tempdb files on their own dedicated volumes, as shown in the following table. This segregation not only separates and balances the I/O but also helps in efficiently monitoring, managing, and troubleshooting the volumes.

Table 6. Storage layout configuration summary

Volume details	Quantity	Size/LUN	Multipathing	Storage logical block size	SCSI controller	Windows file system	RHEL file system	Operating system file block size
Operating system	1	1 TB	vSphere NMP	512 bytes	VMware ParaVirtual	NTFS	Ext4	64 KB
Backup	1	2 TB				NTFS	Ext4	
PRD database data file	2	900 GB				ReFS	Ext4	
PRD database log file	1	500 GB				ReFS	Ext4	
tempdb data and log file	1	400 GB				ReFS	Ext4	

We configured each Windows and RHEL VM with six volumes in their own consistency group. We then mapped the volumes in vCenter with native mutipathing. We created separate datastores in vCenter for each volume and then created and added virtual disks on those volumes for the VM with the VMware ParaVirtual SCSI controller. After installation of the operating system in the VMs, we used the ReFS and Ext4 file systems, for Windows and RHEL operating systems respectively, with a block size of 64 KB for database data and log file drives.

For configuration procedures, see Appendix C: Creating and mapping storage to the VM.

The following figure shows the storage configuration for Windows and RHEL virtual machines.

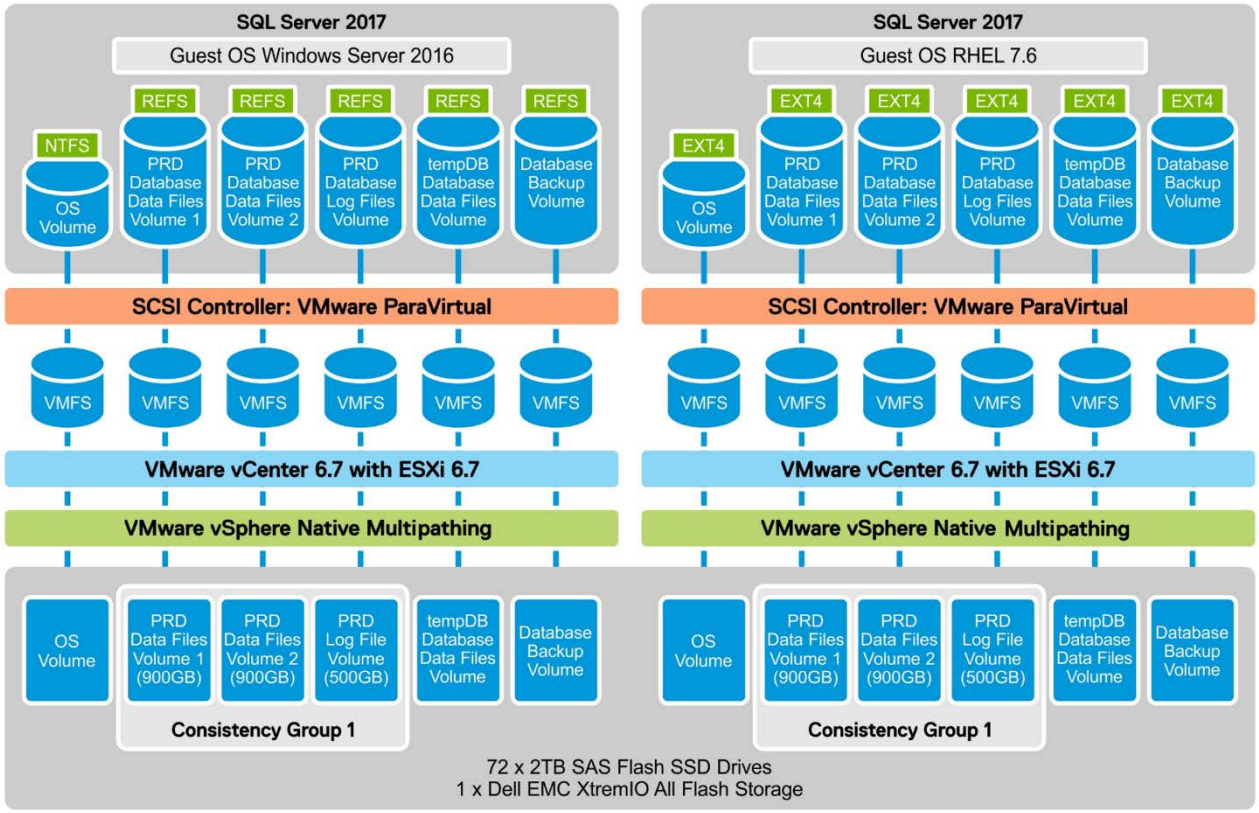

Figure 5. Windows and RHEL VM storage layout

Validation and use cases

Introduction

We extensively tested this reference architecture by running virtualized SQL Server in both Windows Server 2016 and RHEL by using PowerEdge R840 servers and the XtremIO X2 all-flash storage array.

Infrastructure configuration

The PowerEdge R840 servers are designed to maximize compute density for enterprise applications and X2 databases with a small 2U, 4-socket rack configuration. For this reference architecture testing, we used two identical PowerEdge R840 servers with one dedicated to virtualized RHEL and the other dedicated to virtualized Windows Server 2016.

The following table provides the infrastructure configuration details for the physical server and VMs in both the Windows and RHEL environments. It also includes workload execution details for the SQL Server instances.

Table 7. Environment setup details

Category	Specification	Configuration
Physical server	Number of physical servers	1
	Number of processors per server	4
	Number of physical cores per server	72
	Number of vCPUs per server	144
	Memory per server	1.5 TB
	Number of VMs per server	8
VM	Number of vCPUs per VM	16
	Memory per VM	128 GB
	Operating system LUNs per VM	1 x 1 TB
	PRD DB data file per VM	2 x 900 GB
	PRD DB log file per VM	1 x 500 GB
	tempdb data and log file LUN per VM	1 x 400 GB
	Backup LUN per VM	1 x 2 TB
SQL Server	Number of SQL Server instances per VM	1
	Number of vCPUs per SQL Server instance	16
	Memory per SQL Server instance	12 GB
	Database size per SQL Server instance	1,024 GB
	Number of concurrent users per SQL Server instance	100

We configured both PowerEdge servers with four Intel Xeon Gold processors with each CPU having 18 cores for a total of 72 cores. We enabled Intel Hyper-Threading on the R840 servers, doubling the number of cores from 72 to 144 cores. When Hyper-Threading is enabled, the common term that is used for processors is *logical cores* because each core can process more than one instruction per clock cycle.

VMware recommends that the allocation of vCPUs not exceed physical CPUs for production workloads. In this validation testing, the goal was to show how the design can accelerate databases under a highly consolidated workload. Therefore, we reserved 16 vCPUs for each VM; thus, with eight VMs per server, the total number of reserved vCPUs was 128. In our testing, we did exceed VMware's recommended number of physical CPUs, but the number of logical cores remained under the recommended number.

For memory sizing, the goal was to push I/O to the XtremIO X2 all-flash storage array and evaluate array performance. Because over-commitment of VM memory can cause memory contention, VMware recommends configuring VM memory so that it does not exceed the available physical memory on the server. Another best practice is to use memory reservations that account for SQL maximum server memory plus thread stack, plus VM overhead. Thus, we configured each VM in the reference architecture tests with a

memory reservation of 128 GB. With eight VMs on each server, the total memory reservations used 1 TB of a total of 1.5 TB of server memory.

Dell EMC XtremIO X2

The XtremIO X2 storage array is the new generation of the XtremIO family, providing 25 percent better data reduction and 80 percent better latency than the previous generation.[1] XtremIO X2 is ideal for consolidating SQL Server environments. The XtremIO X2 array provides a combination of all-flash for consistently fast performance and all-the-time inline data reduction for high efficiency, and it accelerates database provisioning with application-integrated copy services. Our XtremIO X2 test configuration included 72 SAS flash drives across two X-Brick modules for a total of 112.32 TB. X-Brick modules enable seamless expansion of the XtremIO X2 storage array.

According to XtremIO X2 specifications, one fully populated X-Brick module, with 72 drives, supports a maximum of 220,000 IOPS at 0.5 milliseconds (ms) of latency with a 70 percent read to 30 percent write mixture using 8 KB disk-space blocks. In our testing, we exceeded the achievable IOPS to obtain greater storage array consolidation for the databases. We used best practices when configuring XtremIO X2 storage for SQL Server. SQL Server has different I/O patterns for accessing data and log files:

- Data files have a mostly random I/O pattern for OLTP workloads.
- Transaction logs during normal database operations have a sequential I/O pattern.

In our testing, we separated data, log, tempdb, and operating system files into separate LUNs for each virtualized database:

- 1 x 1 TB operating system LUN
- 2 x 900 GB data LUNs
- 1 x 500 GB log file LUN
- 1 x 400 GB tempdb LUN

By separating parts of the database on the XtremIO X2 array, the database administrator can monitor the database and make changes independently of other database files. Additionally, when cloning and replicating a SQL Server database, copying or protecting all parts of the database is unnecessary. For example, the tempdb LUN does not have to be cloned or replicated to copy or protect the SQL Server database.

Data collection

One of the primary tools that Dell EMC uses for data collection during validation and use case tests is Dell EMC Live Optics, which is a free, agentless software for collecting data from PowerEdge servers. In just minutes, an engineering team can set up Live Optics to collect a wealth of information for configuration and resource utilization analysis. The Live Optics dashboard is intuitive and enables DBAs to monitor and collect data across the server and VMware virtualization layers.

The following figure shows an example of a Live Optics dashboard. The left side shows performance at the project, hypervisor, virtual server, and shared disk levels. The right side shows the collected data and graphs that enable a quick visual analysis.

[1] https://www.dellemc.com/en-us/storage/xtremio-all-flash.htm

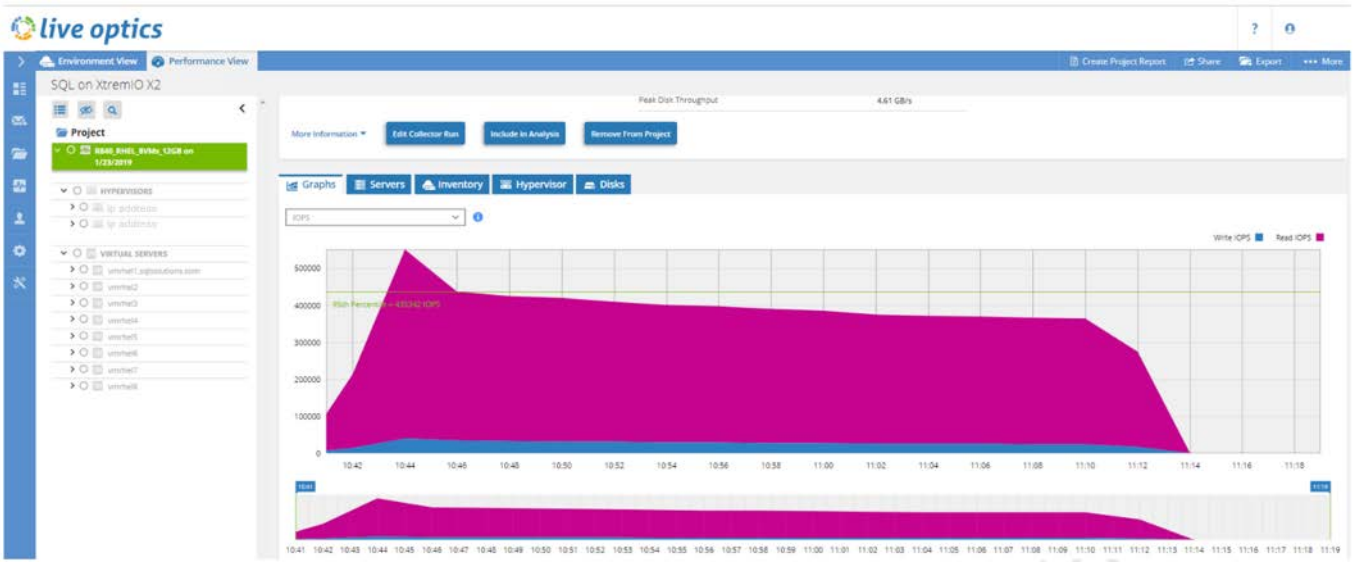

Figure 6. Live Optics dashboard

OLTP use case

Using Quest Benchmark Factory, we configured workloads to simulate many small environments. The TPC-E benchmark is an OLTP workload that the Transaction Processing Council (TPC) designed to test database performance with a mixture of read-only and update-intensive transactions. Our TPC-E-like database workload enables customers to objectively measure performance of the database infrastructure by simulating transactions in enterprise OLTP applications. The following table outlines our modified TPC-E workload, which is not directly comparable to official benchmarks submitted to and validated by the TPC.

Table 8. Modified TPC-E workload for OLTP use case

Benchmark Factory TPC-E parameter	Value
Database scale factor	105
Database size	1,024 GB (1 TB)
Number of users	100
Keying time delay	20 ms
Think time delay	20 ms
Test duration	10 min presampling and 20 min run time

Peak CPU utilization

Peak CPU utilization indicates the high-water mark of CPU usage as a percentage during the incremental use case tests. Each VMware virtualized database used 16 vCPUs for the OLTP workload tests. The following figure shows the peak CPU utilization across all 16 VMs—eight running Windows Server 2016 and eight running RHEL.

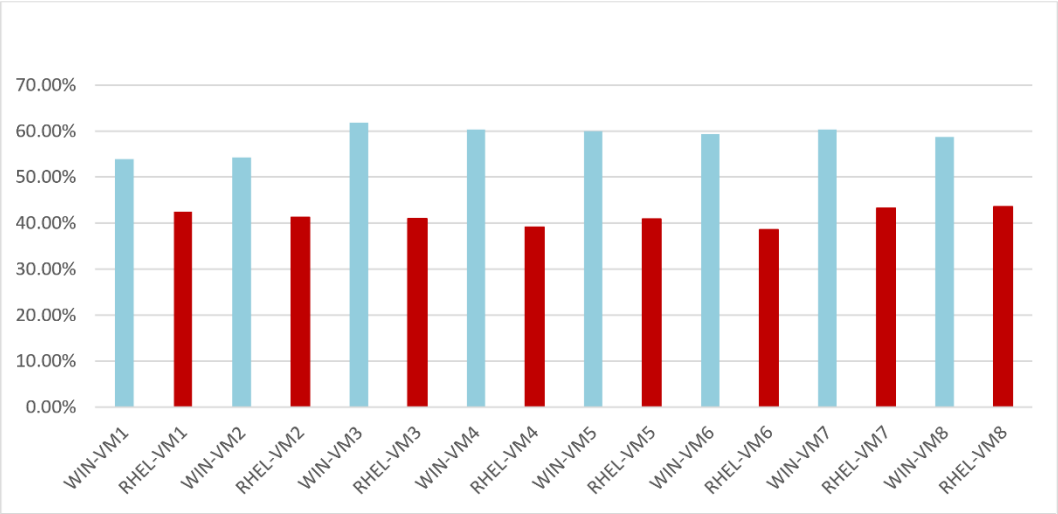

Figure 7. Peak CPU Utilization for Windows and Linux virtual machines

As shown in Figure 7, there was a noticeable peak CPU difference between the Windows VMs (blue bars) and Linux VMs (red bars). The Windows VMs had an average peak CPU utilization of 58.6 percent. The Linux VMs were approximately 17 points lower in peak CPU utilization, with an average of 41.2 percent.

In our testing, there were no statistics that explained the difference in peak CPU utilization between the two operating systems. Our recommendation is to closely monitor your virtualized SQL Server databases for peak CPU utilization and make the appropriate change in the number of reserved vCPUs. In cases of oversubscription of CPU resources, customers have found that using fewer vCPUs reduces processor instruction scheduling by the hypervisor and can increase performance.

Average IOPS

IOPS is a metric that indicates the load on a storage array. Our goal in running 16 VMs was to test the scalability of the architecture and maximize IOPS on the XtremIO X2 storage array. We expected the test findings to demonstrate that the XtremIO X2 array can scale to support several SQL Server databases running concurrently with low submillisecond latencies.

The following figure shows the average IOPS test findings for Windows and Linux. The Windows VMs generated 19,700 or more IOPS with an average of 20,500-plus IOPS. Notably, the Linux VMs generated 12,691 or more IOPS with an average of 13,900-plus IOPS—approximately 6,600 fewer average IOPS than that generated by the Windows VMs.

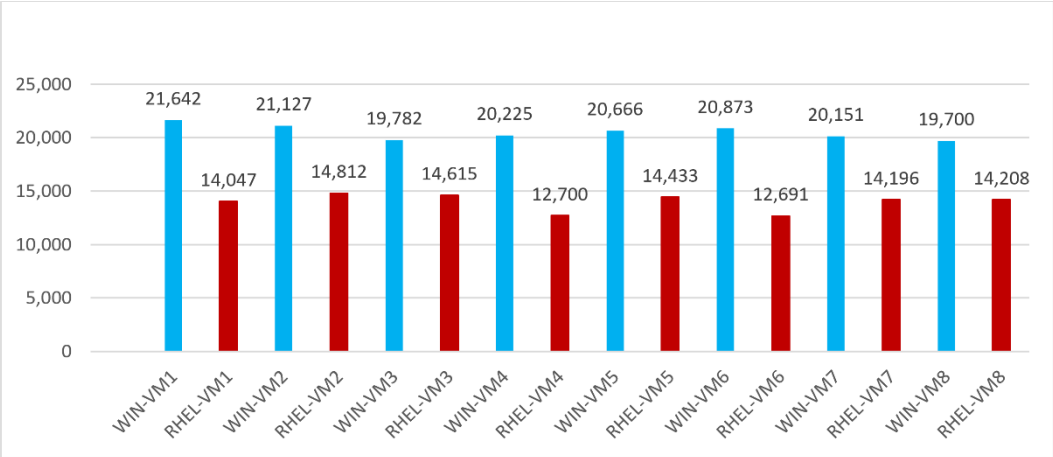

Figure 8. Average IOPS for Windows and Linux VMs

The Benchmark Factor workload configuration was the same between the two operating systems. Nothing in our test data indicates why there was a difference in IOPS between the Windows and Linux VMs. IOPS alone are only an indicator of load on the storage array, and, in this case, the XtremIO X2 array easily supported all the virtualized databases running in parallel.

The eight Windows VMs generated a total of 164,166 IOPS and the Linux VMs generated an additional 111,704 IOPS, for a grand total of 275,870 IOPS on XtremIO X2. The following figure shows the amount of IOPS for each operating system as part of the whole. According to the XtremIO X2 specifications, a fully populated X-Brick module with 72 flash drives supports 220,000 IOPS. In our testing of the reference architecture for SQL Server, the IOPS load exceeded the maximum by 55,870 IOPS.

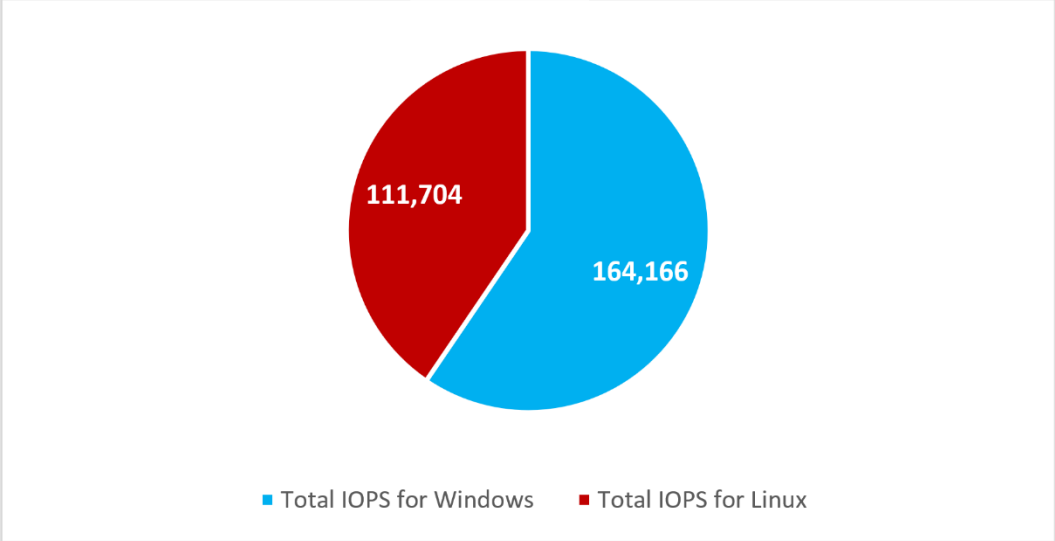

Figure 9. IOPS per operating system

Read and write latency

OLTP workloads are characterized predominately by small reads and writes to storage. A key indicator of storage performance for OLTP workloads is physical read and write latencies. The lower the latency the less time the database waits for reads and writes from storage. The gold standard for all-flash storage arrays is an average of 1 ms or less for all physical database operations. Modern storage arrays from Dell EMC have improved on the gold standard by accelerating storage operations, frequently achieving latencies of 0.75 ms or less.

In this guide, we review XtremIO X2 latency findings for SQL Server in microseconds, 1,000 of which equals 1 ms. The XtremIO X2 storage reports use microseconds as the measure for latencies.

Storage latency

For OLTP workloads, physical reads from storage are generally random small-block I/O. Database and application performance depend on how quickly data can be read from storage. Thus, the lower the read latency the faster the application users can access critical data. SQL Server commonly performs thousands or millions of reads per hour depending on the business load.

Windows latency

The following figure shows the average storage latency for the Windows Server VMs. The blue bars indicate the average read latency for accessing data. The high-water mark for physical reads is near 450 microseconds (0.45 ms) in WIN-VM1 and the low is 350 microseconds (0.35 ms) in WIN-VM5, with an average of 382 microseconds across all eight databases. The findings for physical reads demonstrate that the XtremIO X2 array consistently delivered low latencies for accessing database data.

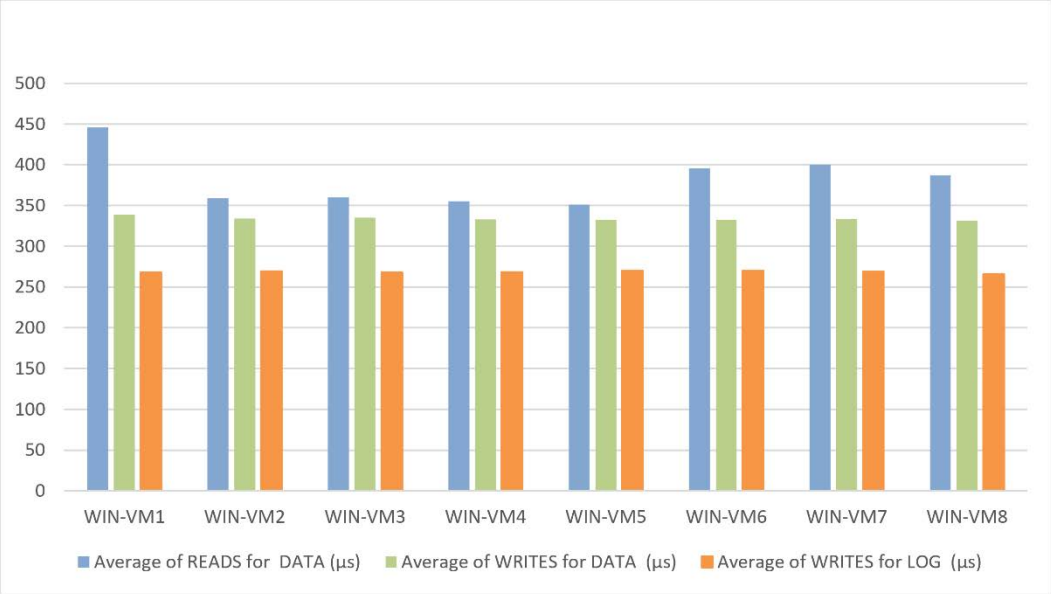

Figure 10. Windows average storage latency in microseconds (µs)

Write latency is critically important because it supports the durability aspect of Atomicity, Consistency, Isolation, Durability (ACID)-compliant databases. In terms of writes, our findings are separated into writes to the data LUNs and writes to the log LUNs. In Figure 10, the green bars indicate physical data write latency to the XtremIO X2 array. Physical writes

ranged from a high of 339 microseconds for WIN-VM1 to a low of 331 microseconds for WIN-VM8, with an average 334 microseconds (0.33 ms) across all eight databases. The XtremIO X2 array showed highly consistent physical write performance with low latencies.

By default, beginning with SQL Server 2016, the database issues a checkpoint every minute. A checkpoint is a process in which all in-memory modified pages are saved to storage and the active portion of the transaction log is updated for persistence. Slow checkpoints can affect database performance because the database engine waits for acknowledgement that the checkpoint has been completed. Because of the importance of the checkpoints, our test data collection included capturing writes to the database log files. The orange bars in Figure 10 show average write times for the log files. Physical log writes ranged from a high of 270 microseconds to a low of 266 microseconds, with an average of 269 microseconds across all eight databases.

Finally, this overview excludes physical log read data because reads from the log files are minimal during normal SQL Server database operations. For example, our test data for physical log reads did not exceed an average of 15 microseconds. Thus, the log files contained no substantial read load.

Linux latency

The following figure shows the average storage latency for the Linux VMs. The blue bars indicate the average read latency for accessing data. The high-water mark for physical reads is near 463 microseconds (0.46 ms) in RHEL-VM2 and the low is 341 microseconds (0.35 ms) in RHEL-VM8, with an average of 377 microseconds across all eight databases. The findings for physical reads demonstrate that the XtremIO X2 array consistently delivered low latencies for accessing database data.

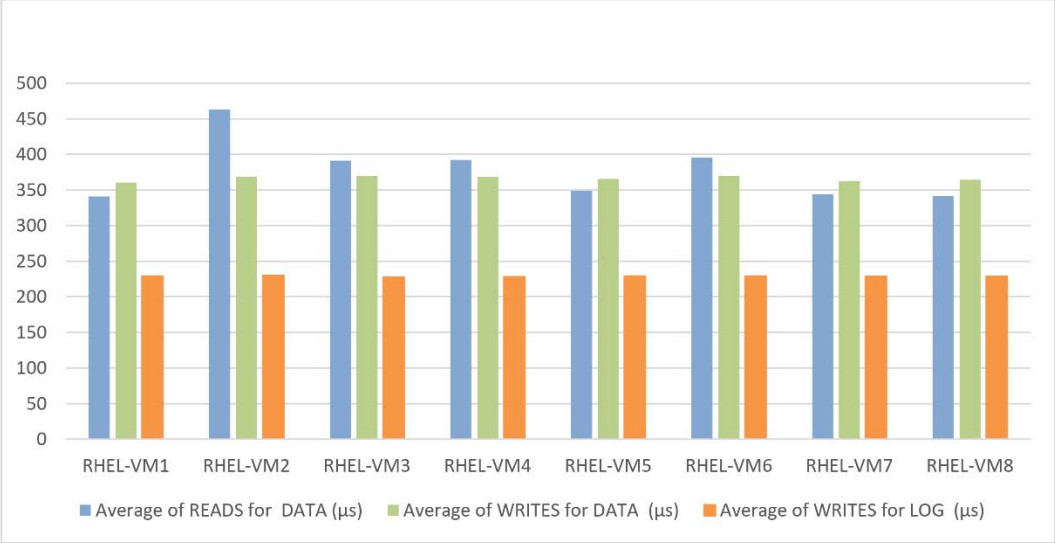

Figure 11. Linux average storage latency in microseconds (µs)

The green bars in Figure 11 indicate physical data write latency to the XtremIO X2 array. Physical writes ranged from a high of 370 microseconds for RHEL-VM6 to a low of 361 microseconds for RHEL-VM1, with an average 366 microseconds (0.36 ms) across all eight databases. The XtremIO X2 array showed highly consistent physical write performance with low latencies.

The orange bars in Figure 11 show average write times for the log files. Physical log writes ranged from a high of 230 microseconds to a low of 229 microseconds, with an average of 229 microseconds across all eight databases. Consistently low latency for physical writes to log files accelerate database checkpoints.

Combined performance of IOPS and latency

IT organizations and DBA teams typically deal with tradeoffs between IOPS and latency. For example, the greater the number of SQL Server databases the more IOPS on the storage array, resulting in higher latency times. This tradeoff between IOPS and latency happens over time. Initially, storage performance is good, and databases have low latency times. With time, more applications are added to the array and the tradeoff is weighted towards IOPS, thus impacting database and application performance.

In testing this architecture for Microsoft SQL Server, we wanted to aggressively consolidate databases to determine where the tradeoff between IOPS and latency was on the XtremIO X2 array. With 16 databases running in parallel, we surpassed the stated maximum of 220,000 IOPS for 72 flash drives by generating a total of 275,870 IOPS. The oversubscription of databases did not impact physical read and write latencies. Table 9 shows the average physical read and write latencies for the Windows and Linux VMs.

Table 9. Average latency for physical reads and writes by operating system

Type of reads/writes	Windows	Linux
Physical reads for data (µs)	382	377
Physical writes for data (µs)	334	366
Physical writes for log (µs)	269	229

Our findings show that there was no tradeoff between IOPS and storage latencies despite the oversubscription of databases. Customers can be confident that a properly sized SQL Server solution that is based on PowerEdge servers and XtremIO X2 arrays can scale while providing strong storage performance.

Database consolidation

IT organizations consolidate servers and storage arrays in data centers to control costs and limit data center expansion. Consolidating databases is more complex, however, because of their dependencies and performance requirements. In validating and testing this architecture, we followed proven strategies for consolidating databases.

Virtualization strategy

Virtualization enables the isolation of disparate applications and granular management of server and storage resources. Therefore, all the virtualized databases can have different versions of operating systems and database engines without having an impact on each other. Virtualization facilitates consolidation and enables more efficient use of server and storage resources. We used vSphere 6.7 as the virtualization layer for consolidating the SQL Server databases, as described in Multipathing configuration on page 10.

CPU utilization strategy

In our tests, virtualization was used to reserve CPU resources per VM. In vSphere, a reservation for CPU or memory means that the VM is guaranteed that server resource. For example, each VM in our tests had a vCPU reservation of 16 virtual cores, meaning

that this is the minimum acceptable amount of processor resources for the VM. The average peak processor utilization was 58.6 percent for the Windows server VMs and 41.2 percent for the Linux VMs. Figure 12 shows that average peak utilization for both Windows and Linux left significant unused processor resources for additional workloads on the PowerEdge R840 servers.

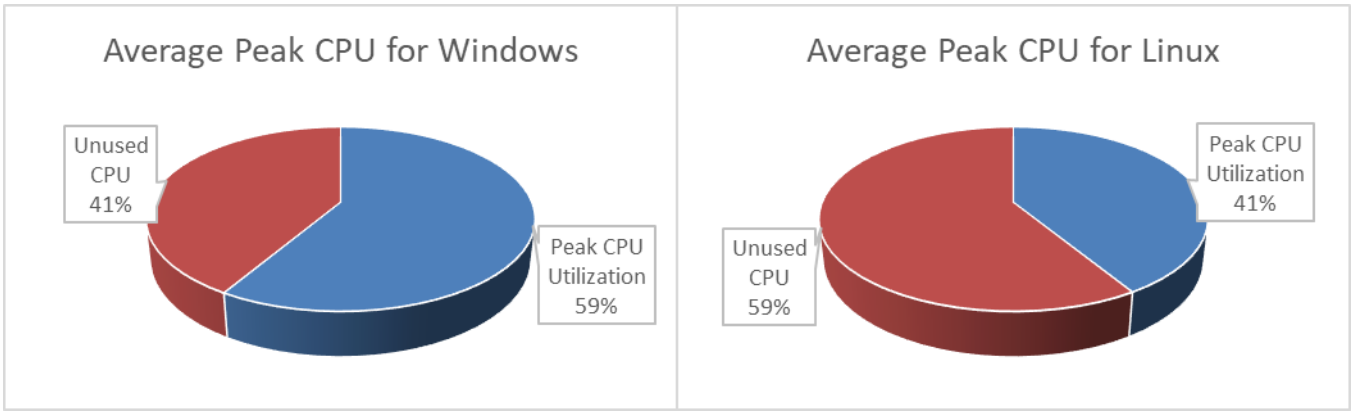

Figure 12. Average peak CPU utilization for Windows and Linux VMs

One strategy is to lower the number of reserved vCPUs on the server and monitor the VMs. The benefit of lowering the amount of vCPU resources is reducing the possibility of CPU resource contention. Customers have been successful with using a strategy of lowering vCPU reservations and increasing memory reservations, thus gaining more performance. We highly recommend that you use monitoring tools to collect and analyze changes in server resources for virtualized databases to determine if this strategy works.

The other benefit of looking for opportunities to save on CPU resources is the capability to maximize use of system resources and achieve even higher consolidation ratios. In the case of the PowerEdge R840 server tests, results show a consolidation ratio of eight virtualized SQL Server databases to each server. Additional testing, reducing the vCPU reservations on the VMs, can determine if the servers can support more databases.

XtremIO X2 inline data reduction

The XtremIO X2 all-flash array's wide range of features includes inline data reduction. Data reduction services work by first deduplicating data: If two blocks are the same, deduplication uses metadata to track the duplicate block. Metadata is maintained in memory for fast access, and only unique blocks are written to storage, thus saving space on the flash drives.

After deduplication, the XtremIO X2 array compresses data, reducing the amount of array space that is used by unique blocks of data. The compression space savings means that data blocks are stored in the most efficient manner. The value of inline data reduction is the resulting ability to consolidate more databases to the XtremIO X2 array. As part of our tests, we captured inline data reduction savings for SQL Server on RHEL. Table 10 shows the storage configuration and inline data reduction savings for SQL Server.

Table 10. XtremIO X2 configuration and data reduction ratios

Data volume	Volume size (GB)	Host-accessible size (GB)	Unique physical space (GB)	Data reduction ratio
RHEL DATA1	900	529.42	152.24	3.48 to 1
RHEL DATA2	900	505.18	141.66	3.57 to 1
Totals	1800	1034.6	239.9	3.52 to 1

We provisioned two 900 GB data LUNs for each database. Of the 900 GB of LUN space, the database used 529.42 GB on DATA1 and 505.18 GB on DATA2. In Table 10, *Host-accessible size* represents the space that is used without any data reduction. If DBAs were to query the amount of space used, the database would report a total of 1034.6 GB. The storage array's data reduction technology is transparent to all applications, meaning that it presents no complexity or special requirements. However, it also means that DBAs should work with the XtremIO X2 administrator to gain an understanding of the data reduction savings.

As shown in Table 10, the actual space that is used on the XtremIO X2 array is 152.24 GB for DATA1 and 141.66 GB for DATA2. The benefit of data reduction is expressed in a ratio. For example, 529.42 divided by 152.24 is a 3.48 to 1 space savings on the DATA1 LUN. On the DATA2 LUN, the data reduction savings were similar at 3.57 to 1. Data reduction ratios are useful, but perhaps a more powerful way to express the data savings is to compare the host-accessible size to the actual space used—unique physical space. The host-accessible size was 1034.6 GB and the actual space used was 239.9 GB, a total space savings of 794.7 GB. Expressed as a percentage, the space savings achieved is 71.5 percent, which is substantial for SQL Server databases.

Conclusion

Summary

The benefits of virtualization combined with the raw processing power of the PowerEdge R840 server and the data reduction savings of the XtremIO X2 array show how this reference architecture was designed for consolidation. Consolidation provides the following benefits:

- Virtualization provides the capability to manage disparate applications on PowerEdge servers, increasing the overall efficiency of the hardware resources.
- The PowerEdge R840 provides powerful support for SQL Server consolidation. In our testing, CPU utilization was under 59 percent across all Windows Server VMs and 42 percent for all Linux VMs.
- XtremIO X2 inline data reduction storage technology provides significant space savings. In our testing, a 1 TB database used only 239.9 GB on the all-flash array.

Consulting and support

To help customers navigate the best path to achieve their businesses objectives with Microsoft SQL Server, Dell EMC Consulting provides strategic and tactical services, from platform upgrades to data modernization and migration and business intelligence and analytics.

Dell EMC's ProSupport Enterprise Suite provides component-level support for Dell EMC Ready Solutions for Microsoft SQL: Design for XtremIO and PowerEdge.

References

Dell EMC documentation

The following Dell EMC documentation provides additional and relevant information:

- *Best Practices for Running SQL Server on Dell EMC XtremIO X2 White Paper*
- Dell EMC XtremIO

For additional information about Dell EMC Ready Solutions for Microsoft SQL, see the Microsoft SQL Info Hub for Ready Solutions on the Dell EMC Community Network.

VMware documentation

The following VMware documentation provides additional and relevant information:

- *Guest Operating System Installation Guide: Windows Server 2016*
- *Technical Note: Installing and Configuring Linux Guest Operating Systems*

Microsoft documentation

The following Microsoft articles provide additional and relevant information:

- Install SQL Server (Windows)
- Quickstart: Install SQL Server and create a database on Red Hat

Appendix A: Best practice guidance

Introduction

Every production environment is distinct in terms of its requirements, performance expectations, user load, and so on. Determining proper configuration values at each layer in a database solution—physical server layer, storage layer, virtualization layer, operating system layer, and database layer—becomes tedious and highly dependent on the environment in which the solutions are used.

This section provides configuration best practices for each architecture layer. Perform a thorough test and take appropriate precautions before changing any configuration values at any layer in this reference architecture.

Physical server layer

For the PowerEdge R840 rack server:

- Set power management to high performance in the server BIOS.
- Enable Hyper-Threading in the server BIOS.
- Configure local disks in RAID 1 for ESXi hypervisor installation.
- Ensure that you have the latest stable version of the BIOS and firmware for all devices that are attached to the server, including FC cards.
- Maintain redundancy at card and port levels for both FC HBAs and Ethernet.

Storage layer

For the XtremIO X2 array:

- Upgrade the XtremIO X2 storage operating system and XMS software to the latest stable release.
- Use thick provision eager zeroed volumes to achieve better and consistent performance.
- Create volumes with a physical sector size of 512 bytes for all LUNs.
- For better performance and consistency, map each virtual disk to a single LUN.
- Configure database data files and log files to dedicated volumes.

For more information, see the *Best Practices for Running SQL Server on Dell EMC XtremIO X2 White Paper*.

Virtualization layer

For VMware ESXi 6.7:

- On the QLogic card, increase the LUN queue depth according to your workload requirements. During our benchmarking, we used the queue value of 256.
- For the ESXi host, change the power management policy to high performance.
- Configure multipathing properly to have better performance and high availability for the paths between server and storage. We used VMware NMP technology for storage multipathing.
  - Select the native round-robin path policy.
  - Change the NMP round-robin path switching frequency for the XtremIO X2 array from the default of 1,000 to 1.

- Use PVSCSI controllers when creating virtual disks on the datastore and assigning it to the VM.
- Create distributed vSwitches to help with load balancing and high availability.
- Assign vCPUs and memory within single physical NUMA nodes for the VM to achieve better utilization and performance of the VM.
- Use VM vCPU and vMemory reservations for better performance predictability and reliability.
- For high-performance workloads, avoid overprovisioning of memory and vCPUs. Keep ESXi overhead in mind while planning for VM deployment.

Operating system layer

For RHEL 7.6:

- Use the tuned-adm command-line tool to set the latency-performance profile.
- Follow Microsoft's Performance best practices and configuration guidelines for SQL Server on Linux. Add the Microsoft-recommended performance-related configuration parameters for the RHEL operating system to the latency-performance profile.
- Change the disk label (dos, by default) to GPT.
- Create disk partitions using the fstab or parted utility on storage devices. We chose the EXT4 file system while formatting the disks.
- Keep all the mounted file entries in `/etc/fstab` to enable automatic mounting when the server reboots.

For Windows Server 2016:

- Select the **High Performance** power profile in the guest operating system.
- Format the drives using the ReFS file system with 64 KB allocation for better performance and durability.
- Change the disk label to GPT drive, and set the queue depth to 254 and ring pages to 32.
- Enable **lock pages in memory** for the SQL Server service account.

Database layer

For SQL Server 2017:

- Set **min server memory** and **max server memory** to the same value while leaving room for operating system overhead. For more information, see SQL Server Max Memory Best Practices.
- Change the **max degree of parallelism** (MAXDOP) configuration option and **cost threshold for parallelism** option after proper validation because the query parallelism requirement changes according to the dataset and nature of the queries. For more information, see Recommendations and guidelines for the "max degree of parallelism" configuration option in SQL Server and Configure the cost threshold for parallelism Server Configuration Option. During our study, we set the MAXDOP value to 1 and kept the **cost threshold for parallelism** value at its default of 5.

- Enable instant file initialization by granting **perform volume maintenance tasks** permissions to the SQL Server service account. For more information, see Database File Initialization.
- Set the **max worker thread** value according to the workload and processor that are assigned to the SQL Server instance. For more information, see Configure the max worker threads Server Configuration Option. During our study, we set **max worker thread** to 704.
- Use multiple data files on different virtual disks and LUNs within the same filegroup.
- Allocate multiple tempdb data files to address tempdb contention issues. For more information, see Recommendations to reduce allocation contention in SQL Server tempdb database. For our study, we allocated eight files on a separate drive that was dedicated for tempdb with 8 GB size per file.
- Segregate database data files, database log files, and tempdb files on separate drives that are mapped to dedicated virtual disks and volumes. For our study, we created two data files and one log file on dedicated drives.

Appendix B: Configuring vSphere

Configuring the ESXi management network

After the ESXi installation is complete and the server restarts, configure the ESXi management network as follows:

1. Press F2 to log in to the Direct Console User Interface (DCUI).
2. At **Authentication Required**, type the credentials that you created during setup and press Enter.
3. At **System Customization**, select **Configure Management Network**.
4. Under **Configure Management Network**, select **Network Adapters**.
5. Ensure that the vmnic0 and vmnic1 NIC ports are displayed under **Network Adapters**, and then press Esc.
6. Select **VLAN (optional)** and press Enter.
7. On the **VLAN (optional)** page, type the VLAN ID for the management network and press Enter.
8. Select **IPv4 Configuration** and press Enter.
9. On the **IPv4 Configuration** page, select **Set static IPv4 address and network configuration**, and then press the spacebar.
10. Type the information for **IP Address**, **Subnet Mask**, and **Default Gateway**, and press Enter to confirm.
11. Select **DNS Configuration** and press Enter.
12. On the **DNS Configuration** page, type the IP address of the DNS servers and the fully qualified domain name (FQDN) of the host.
13. At **Configure Management Network: Confirm**, press Esc to return to the main menu and press Y to confirm changes and restart the management network.
14. Select **Test Management Network**.

    The **Test Management Network** page displays the items that will be tested.
15. Press Enter to continue.

Create a vSphere datacenter

Create a datacenter within vSphere for XtremIO X2 configuration, and then add hosts to vCenter:

1. Open a web browser and then open the vSphere Web Client:

   `https://<vCenter Server Administrator FQDN or IP>/vsphere-client`
2. Log in with an account that has administrator privileges.
3. Go to **Home** > **Inventory** > **Hosts and Clusters**.
4. On the navigator menu, right-click the top-level vCenter Server Administrator object and select **New Datacenter**.
5. Enter the name for the datacenter—`XTREM_RHEL_WINDOWS`—and click **OK**.

6. Add the hosts to vCenter by performing the following steps on each of the servers that will be part of the datacenter:
   a. Open a web browser and open the vSphere Web Client:

      `https://<vCenter Server Administrator FQDN or IP>/vsphere-client`

   b. Log in with an account that has administrator privileges.
   c. Go to **Home** > **Inventory** > **Hosts and Clusters**.
   d. Right-click the datacenter object and select **Add host**.
   e. Enter the DNS name or IP address for the first compute host, and then complete the remainder of the wizard.
   f. Repeat steps d and e for another host.

Create a virtual network

Create a virtual network in vSphere as follows.

Create a standard switch

To create a standard switch:

1. Select the ESXi host and, under **Configure**, expand the **Networking** tab and select **Virtual switches**.
2. Click the **Add host networking**.
3. For the connection type, select **Virtual Machine Port Group for a Standard Switch**, and then click **Next**.
4. Select **New standard switch** and click **Next**.
5. Add active adapters and click **Next**.
6. Keep the default settings and click **Next**.
7. Click **Finish**.

Create a VMkernel adapter

To create a VMkernel adapter:

1. Select the ESXi host and, under **Configure**, expand the **Networking** tab and select **VMkernel adapters**.
2. Click the **Add Networking** tab, keep the default settings, and click **Next**.
3. Click the browse button, select a vSwitch, and click **Next**.
4. Keep the default settings and click **Next**.
5. If you have a DHCP server, keep the default settings and click **Next**; otherwise, select **Use static IP**, enter the IP address, and click **Next**.
6. Click **Finish**.

Appendix C: Creating and mapping storage to the VM

SQL Server maps a set of database files on disk. To maximize the performance and operational efficiency of SQL Server, consider the following recommendations.

Configuring volumes on the XtremIO X2 array

To configure volumes on the XtremIO X2 array:

1. Create the following LUNs, each with a logical block size of 512 bytes:
   - **Database**—2 x 900 GB
   - **Log**—1 x 500 GB
   - **Tempdb and temp log**—1 x 400 GB
   - **Operating system**—1 x 1 TB
   - **Backup**—1 x 2 TB LUN
2. In the configuration window, review the volumes, and then click **Mapping**.
3. Select the initiator group and click **Next**.

Discovering LUNs in vCenter

To discover LUNs in vCenter:

1. Right-click the host, select **storage**, and then select **New Datastore**.
2. Select **VMFS** and then click **Next**.
3. Select the data LUN that was created on the XtremIO X2 array, and then enter the LUN name.
4. Select the default VMFS version—**VMFS 6**.
5. Click **Next**, and then click **Finish**.
6. Repeat the preceding steps for each of the remaining volumes.

Creating a VM

To create a VM:

1. Right-click the host and select **New virtual machine**.
2. Enter the VM name.
3. Select the datacenter name.
4. Select the host where you want to install VM.
5. Select the datastore where you want to install RHEL or Windows Server 2016.
6. Select the guest operating system—Linux or Windows Server.
7. Select the guest operating system version.
8. Click **Next**, and then click **Finish**.

Creating SCSI controllers

To create SCSI controllers:

1. Right-click the VM and click **Edit Settings**.
2. Click **Add Device**.
3. Select **SCSI Controller** and add up to three controllers.
4. Click **OK**.

Adding a datastore to a VM

To add a datastore to a VM:

1. Right-click the VM and click **Edit Settings**.
2. Click **Add Device**.
3. Select **New Hard Disk**.
4. Browse to the location of the datastore and select the datastore.
5. Change the disk capacity to 900 GB.
6. Change the sharing to **No Sharing**.
7. Change the disk mode to **Independent Persistent**.
8. Change the SCSI controller to **New SCSI Controller**.
9. Click **OK**.
10. Repeat the preceding steps for the remaining volumes.

Discovering LUNs within the RHEL guest operating system

To discover LUNs within the RHEL guest operating system:

1. Start the VM and verify that all the virtual disks that are attached to the VM are visible as devices in the RHEL guest operating system.

   Device names appear as `/dev/sda`, `/dev/sdb`, and so on.

2. Run the following command to see the devices:

   ```
   Cat /proc/partitions
   ```

3. Change the disk label to GPT:

   ```
   parted /dev/sdb mklabel gpt
   ```

4. Create the partition using the `fdisk/parted` command.
5. Create an ext4 file system on the partition:

   ```
   mkfs.ext4 /dev/sdg1 2>/dev/null
   ```

6. Check the UUID for the device:

   ```
   blkid /dev/sdg1
   /dev/sdb1: UUID="9e4d449e-62df-4c8d-bc21-d1d7f732f953"
   TYPE="ext4"
   ```

7. Add the UUID in fstab:

```
UUID=9e4d449e-62df-4c8d-bc21-d1d7f732f953 /log  ext4
auto,user,rw 0 0
```

8. Create the target directory for data and log files:

```
Mkdir /data
Mkdir /log
```

9. Change the owner and group of the directory to the mssql user:

```
chown mssql /data
chgrp mssql /log
```

10. Mount the directory:

```
Mount /data
Mount /log
```

11. Repeat the preceding steps for the remaining volumes.

Discovering LUNs within the Windows guest operating system

To discover LUNs within the Windows guest operating system:

1. Open the Run dialog and enter `diskmgmt.msc` to open the disk management application.
2. In Disk Management, right-click the disk that you want to initialize, and then click **Initialize Disk**.

   If the disk is listed as **Offline**, first right-click it and select **Online**.
3. In the **Initialize Disk** dialog box, ensure that the correct disk is selected and then click **OK** to accept the default partition style.
4. Right-click the disk and select **New Simple Volume**.
5. Select the maximum disk space, click **Next**, assign a drive letter, and click **Next**.
6. From the list menu, select **ReFS**, select **64Kb** for the allocation unit size, and then click **Next**.