BEST PRACTICES FOR RUNNING SQL SERVER ON DELL EMC XTREMIO X2

Abstract

This White Paper explains the general characteristics of running SQL Server on DELL EMC's XtremIO X2 enterprise all-flash storage array. It describes the value added by XtremIO X2 in performance, storage efficiency, availability, and SQL Server Application lifecycle management. This document also provides general guidance and best practices of design, deployment, and optimization of SQL Server on XtremIO X2.

February, 2018
Contents

Abstract ................................................................................................................................................................. 1

Executive Summary ............................................................................................................................................... 4

Audience ......................................................................................................................................................... 4

Introducing XtremIO ........................................................................................................................................... 4

Architecture Overview ....................................................................................................................................... 5

Integrated Copy Data Management (iCDM) ........................................................................................................... 6

XtremIO Virtual Copy ....................................................................................................................................... 6

AppSync .......................................................................................................................................................... 8

XtremIO X2 ........................................................................................................................................................ 9

Running SQL Server on XtremIO X2 ................................................................................................................... 9

Performance .................................................................................................................................................... 9

IOPS Intensive ................................................................................................................................................ 9

Bandwidth Intensive ....................................................................................................................................... 11

Mixed OLTP and OLAP Workloads .................................................................................................................. 12

Log Write and XtremIO Write Boost .............................................................................................................. 14

Storage Efficiency ........................................................................................................................................... 15

Thin Provisioning .......................................................................................................................................... 15

Inline Data Deduplication ............................................................................................................................... 16

Data Compression ......................................................................................................................................... 19

Lifecycle Management .................................................................................................................................... 22

Use Case I: Test and Dev ................................................................................................................................. 22

Use Case 2: Database Maintenance ................................................................................................................ 23

Use Case 3: Query Tuning ............................................................................................................................... 23

Use Case 4: Reporting ................................................................................................................................... 24

Use Case 5: Upgrade ...................................................................................................................................... 24

High Availability and Data Protection ............................................................................................................. 25

Logical Data Protection and Fast Recovery .................................................................................................... 25

Storage Monitoring ......................................................................................................................................... 26

Advanced Analytics Reporting ...................................................................................................................... 28

Provision for SQL Server ............................................................................................................................... 30

RAID Configuration .................................................................................................................................... 30

Separation of Files ......................................................................................................................................... 31

Number of LUNs .......................................................................................................................................... 31

Number of Files .......................................................................................................................................... 32

Multipathing ................................................................................................................................................ 33

Queue Depth ................................................................................................................................................ 33

512B vs. 4K Physical Sector Size ................................................................................................................... 34
Executive Summary

In today's information age, businesses are increasingly relying on data for decision making. Business needs are driving rapid growth of data. At the same time, IT departments across organizations are being challenged to increase efficiency and improve services.

Microsoft SQL Server is the core database engine supporting many business-critical applications. Storage is a key component to the successful design, deployment, and optimization of a SQL Server environment. An effective storage subsystem is not only crucial to delivering fast application response time, but it also affects the availability and efficiency of the database.

The DELL EMC XtremIO X2 all-flash storage array effectively addresses the needs of modern database applications. XtremIO delivers impressive high IOPS, ultra-wide bandwidth, and consistent sub-millisecond latency for databases of all sizes, thus providing an ideal storage platform for running online transaction processing (OLTP), online analytical processing (OLAP), or mixed workloads. XtremIO built-in storage efficiency and integrated copy data management features bring new levels efficiency, agility, and availability to SQL Server database lifecycle management.

This white paper discusses:
- XtremIO X2 features, and added value for SQL Server
- Use case scenarios for using XtremIO X2 features to enhance SQL Server application lifecycle management, and increase productivity for database administrators, and Business Intelligence teams
- SQL Server on XtremIO X2 deployment considerations and best practices

Audience

This paper is intended for:
- SQL Server database administrators (DBAs)
- IT Administrators
- Storage / Data center architects
- Technical managers
- Other IT personnel responsible for design, deployment, and managing SQL Server databases, infrastructure, and data centers

Introducing XtremIO

XtremIO is a Dell EMC enterprise all-flash storage array that has been designed from the ground-up to unlock flash's full performance potential and deliver array-based capabilities that leverage the unique characteristics of flash-based SSDs.
Architecture Overview

The XtremIO Storage Array is an all-flash system, based on flexible scaling options. The system uses building blocks called X-Bricks. Each X-Brick is a high availability unit that consists of dual active-active storage controllers and a set of SSDs. X-Bricks are clustered together to grow performance, capacity, or both, as required. Interconnects between X-Bricks and storage controllers are provided by high speed, ultra-low latency Remote Direct Memory Access (RDMA) with InfiniBand. By leveraging RDMA, the XtremIO system is essentially a single shared memory space spanning all storage controllers.

The storage engine is content-aware. As data stream enters the system, it is broken down into data blocks. Each data block is fingerprinted with a unique signature based on the content of the data block. The system maintains a mapping table in memory. Each fingerprint of the incoming data block is checked against the mapping table for duplications. Only unique data blocks not previously written to the system are further compressed and stored on SSDs. Data is always stored efficiently with the smallest footprint the first time with no need for post processing data reduction. The mathematical process that calculates the fingerprints always results in a uniform distribution of fingerprint values, and the fingerprint mapping is evenly spread among all storage controllers in the cluster.

With its intelligent content-aware storage architecture, XtremIO provides:

- Even distribution of data blocks. Performance is inherently balanced across all storage controllers and all SSDs.
- Even distribution of metadata.
- Extended flash endurance by eliminating data and metadata hotspots.
- Easy setup with no tuning. LUNs are distributed across RAID groups or spindles without the need for complex capacity planning.
- Advanced data services, including inline data deduplication, inline data compression reduction, thin provisioning, advanced data protection (XDP), copy data services with snapshots, and more.
Integrated Copy Data Management (iCDM)

Building on its unique performance foundation, XtremIO arrays leverage XtremIO Virtual Copy (XVC) technology to deliver instant, high performance, space-efficient, writable copies of SQL Server database. XVC abstracts the copy operations as unique in-memory metadata operations with no impact on any back-end resources. XtremIO integrated copy data management (iCDM) integrates XVC with SQL Server to deliver space efficient, application consistent, and self-service copies. With XtremIO iCDM, you can provision SQL Server databases on top of physical or virtualized infrastructures in seconds with centralized management. iCDM significantly simplifies the procedure of making a development copy, instantly refreshing it to the latest production data, pushing the development copy to a QA host, pushing the QA copy to a scalability test-bed, and rolling the output back into production. For analytics processes, production data can be extracted and pushed to all downstream analytics applications on-demand as a simple in-memory operation.

As a no-cost, array-based application service, iCDM is unique to XtremIO and solves the most intractable CDM challenges. XtremIO iCDM copies are:

- Space-efficient, high-performance, and nearly unlimited
- Instantly created, refreshed, deleted or recovered
- Independent of other copies so as not to risk their SLAs
- Managed through a flexible, integrated application workflow

XtremIO Virtual Copy

XtremIO Virtual Copy (XVC) creates a copy of the data volume(s) that captures point-in-time state of a specified volume or set of volumes. An XVC can be writable or read only.

XVC is implemented by leveraging the array's content-addressable capabilities along with in-memory metadata and the system's dual-stage metadata. As a result, copies not undergo deduplication at the physical data block level, but they also share the in-memory metadata.
When creating a Virtual Copy, the system only generates a pointer to the ancestor metadata of the actual data in the system, making the operation very quick. This operation does not have any impact on the system and does not consume any capacity at the point of creation, unlike traditional snapshots which may need to reserve space or copy the metadata for each snapshot. Virtual Copy capacity consumption occurs only when changes are made to any copy of the data. Then, the system updates the metadata of the changed volume to reflect the new write, and stores its blocks in the system using the standard write flow process.

Figure 3. Sample XVC Metadata Tree Structure

XVC provides the following basic functions and workflows which enable iCDM to perform various operations on XtremIO X2:

- **Consistency Groups (CG)** – Grouping of volumes to allow Virtual Copies to be taken on a group of volumes as a single entity.
- **Snapshot Sets** – A group of Virtual Copies of volumes taken together using CGs or a group of manually-chosen volumes.
- **Protection Copies** – Immutable read-only copies created for data protection and recovery purposes.
- **Protection Scheduler** – Used for local protection of a volume or a CG. It can be defined using intervals of seconds/minutes/hours or can be set using a specific time of day or week. It has a retention policy based on the number of copies required or the permitted age of the oldest snapshot.
- **Restore from Protection** – Restore a production volume or CG from one of its descendant snapshot sets.
- **Repurposing Copies** – Virtual Copies configured with changing access types (read-write / read-only / no-access) as needed.
- **Refresh a Repurposing Copy** – Refresh a Virtual Copy of a volume or a CG from the parent object or from other related copies which include relevant updated data. Volume provisioning changes are not required for the refresh to take effect. Only host-side logical volume management operations are needed to discover the changes.
AppSync

EMC AppSync is an advanced copy management software package for DELL EMC storage array products. AppSync provides application integration and orchestration services for XVCs. It simplifies and automates the process of creating and consuming XVCs for iCDM use cases. With AppSync, it is possible to protect all critical applications in a single click, dial in the correct service level, and enable application owners to control data protection. AppSync helps with any copy management activity, including data repurposing for test/dev, backup acceleration using copies, or operational recovery.

**EMC APPSYNC**

**Single-Click Replica Management**

While AppSync provide integration for a wide range of applications and virtualized environments, it also includes several features specifically provided for SQL Server.

**Application Consistency**

AppSync uses the SQL Server virtual device interface (VDI) to create application consistent copies of the database that ensure recoverability. Through SQL VDI, AppSync freezes the SQL Server database while taking an XVC. To avoid any transaction loss, all I/O activity is pre-empted while a copy is made. The database is then released and regular I/O activity resumed.

An application consistent copy can be mounted on another server and recovered with several options:

- **Recovery** – This option brings the database copy online and ready to use.
- **No Recovery** – This option puts the database in a “restoring” state to allow DBA to restore transaction log backups. When in “No Recovery” mode, the database is unusable.
- **Standby** – This option restores the database in read only mode to allow the DBA to apply transaction logs or evaluate if the database copy is valid and usable.

The ability to recover the database into different states is an important feature of application consistent copy. It allows the copy to be used in the deployment of SQL Server AlwaysOn Availability Groups (AAGs), recovery of point-in-time transaction logs, and other important high availability scenarios.
**XtremIO X2**

X2 and XIOS v6.0 are the latest releases of XtremIO hardware and software. Building on the innovative metadata-centric, content-aware architecture and integrated copy data management (iCDM) technologies of the previous generation of XtremIO platform, X2 and XIOS v6.0 software continue to set new benchmarks for consistent performance, efficiency and agility.

- **Unmatched Storage Efficiency** – X2 offers an average of 25% better data reduction to minimize footprint even further – making XtremIO the ultimate platform. In addition, with support for twice the number of XtremIO Virtual Copies (XVC) per cluster, customers can make XVC's more frequently, and retain them for longer periods – expanding the incredibly popular iCDM capabilities. In fact, with over 3,000 XtremIO customers, we have found that more than half of the 1.5M XVCs created on XtremIO arrays are writable and being actively used for scenarios such as test/dev.

- **New Multi-Dimensional Scalability** – X2 provides over 100TB effective capacity per U and the ability to scale-up in increments as small as 7TB raw storage within an X-Brick. In addition, you can scale out with partially populated X-Bricks and scale up capacity as needed – up to 138TB raw per X-Brick. You can scale out to 8 X-Bricks in even or odd increments, for a total of 5.5PB effective capacity per cluster.

- **Extreme Performance** – New X2 hardware and XIOS v6.0 innovative software greatly improve performance with up to 80% better response times and twice as much bandwidth. X2 Write Boost reduce SQL Server small block write latency by 55%.

- **Dramatically lower TCO** – X2 software innovations along with a brand new XtremIO hardware platform provide up to 50% lower $/useable GB compared to the previous generation.

- **Consumer Simplicity with Enterprise Capabilities** – XIOS v6.0 introduces a new version of XtremIO Management Software (XMS) with a new HTML GUI and a variety of enhancements. These include innovative, contextual, and automated workflow suggestions for management activities, as well as advanced reporting and analytics for easier troubleshooting. In addition, new global search means you can quickly find that “needle in the haystack” function.

**Running SQL Server on XtremIO X2**

EMC XtremIO X2 brings value beyond that of traditional storage. XtremIO all-flash storage array not only sets new performance benchmarks, but also provides rich solutions to many data center challenges such as storage footprint, lifecycle management, high availability and data protection. In this section, we will present lab studies and use cases which demonstrate the valued added by XtremIO for SQL Server in those areas.

**Performance**

Different types of applications have varying requirements for reading and writing data. The nature of the application also drives different demands for storage performance. Some applications demand high IOPS with low latency. Other applications may not be latency sensitive, but may require the storage system to support high throughput/bandwidth.

To gain a thorough understanding of XtremIO X2's ability to meet the requirements of various types of applications, we conducted a series of lab studies which check IOPS in both bandwidth intensive and mixed workload scenarios. This section presents the results of these studies.

**IOPS Intensive**

SQL Server applications with high IOPS demands tend to generate large number of small reads and writes. Data access is typically random, over the entire database. The size of data being read and written is small, typically less than 8KB. A typical example of such an application is online transaction processing (OLTP). The key metric in measuring performance of OLTP workloads is the number of I/O per second (IOPS) which can be achieved while maintaining a good response time.
In this study, we used an OLTP-like workload that simulates a stock trading application to generates I/O transactions of a typical SQL Server. The workload generates 90% reads and 10% writes. Most of the data transactions are 8K random access. The study measures IOPS and latency as the SQL Server application scales.

Figure 5 summaries XtremIO X2 dual X-Brick performance as the number of application IOPS scales up beyond 500K. The average latency maintains below 500µs.

![XtremIO X2 Latency vs. IOPS](image)

Figure 5. XtremlIO X2 OLTP Application Performance

Figure 6 shows the XtremIO XMS dashboard performance view with sustained high IOPS and consistent performance.

<table>
<thead>
<tr>
<th>PERFORMANCE</th>
<th>CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOPS</td>
<td>474,603</td>
</tr>
<tr>
<td>LATENCY</td>
<td>0.43 ms</td>
</tr>
<tr>
<td>BW</td>
<td>3.63 GB/s</td>
</tr>
<tr>
<td>FREE</td>
<td>21.76 TB</td>
</tr>
<tr>
<td>DRR</td>
<td>5.71</td>
</tr>
</tbody>
</table>

![XMS – OLTP Performance](image)

Figure 6. XMS – OLTP Performance

See Appendix II: Test Configuration for additional details on test configuration.
Bandwidth Intensive

Another use case of SQL Server is for applications which require scan-intensive operations that access large portions of the data at a time. These operations result in smaller number of I/O access than OLTP applications, but the size of the data accessed with each I/O request is much larger – 64K or greater. This makes the throughput or gigabytes per second (GB/s) the critical metric, and ensuring there is enough connection bandwidth between server and storage is critical. Online analytical processing (OLAP) application SQL Server maintenance operations such as backup and index rebuild operations are typical representations of this type of workloads.

To assess how the SQL Server workload performs with the enhanced 16Gbps fiber channel connectivity support in XtremIO X2, we ran the SQL Server Backup to NULL device test. When a backup is run, it reads a large sequential block of data from disk into the database buffer, and then writes from the buffer to the backup media. Backing up to NULL device will cause full time read without pausing to write to the backup media. This is a common method used to assess throughput of a storage subsystem for SQL Server applications.

Figure 7 shows the XMS dashboard view of XtremIO throughput of over 12GB/s, indicating the SQL Server is able to take advantage of the maximum bandwidth that a dual X-Brick supports.
As shown in Figure 8, backing up a 5TB database to NULL device completed in only 4 minutes.

Figure 8. Backup to NULL Device

See Appendix II: Test Configuration for Additional Details on Test Configuration.

Mixed OLTP and OLAP Workloads

Typical SQL Server application workloads have moderate IOPS/bandwidth needs that rarely stretch the capability of a shared storage area network (SAN). Organizations consolidate multiple workloads on SAN to maximize resource usage, and take advantage of the availability, scalability and ease of management of a SAN. While storage consolidation delivers the promises of management efficiency, maintaining application performance and user service level has been a huge challenge for SQL Server running on SAN due to “noisy neighbor” issues for resource sharing, or issues arising from mixed workloads with different disk access patterns.

In this study, we measure the performance impact of mixing an OLAP and an OLTP workload. OLAP workload is throughput dependent. Bandwidth or gigabytes per second (GB/s) is the critical performance metric. OLTP workload is sensitive to the amount of I/O which can be processed per second (IOPS) with reasonable latency.
Figure 9 shows XMS dashboard view of an OLAP-only workload. The OLAP workload is driving about 2.88 GB/s throughput.

**PERFORMANCE**

<table>
<thead>
<tr>
<th>IOPS</th>
<th>LATENCY</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>43,432</td>
<td>0.62 ms</td>
<td>2.88 GB/s</td>
</tr>
</tbody>
</table>

**CAPACITY**

<table>
<thead>
<tr>
<th>FREE</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.47 TB</td>
<td>4.7:1</td>
</tr>
</tbody>
</table>

In conclusion, XtremIO X2 consistent performance for mixed OLAP and OLTP workloads provides an ideal storage platform for customers consolidating any SQL Server applications.
Log Write and XtremIO Write Boost

SQL Server uses write ahead logging (WAL) to ensure durability of transactional operations to the database. This means that every data modification (insert, update, or delete) needs to be written to the transaction log and hardened on disk before acknowledging the client. Transaction log write latency is one of the most critical factors that directly impacts SQL Server database performance.

XtremIO X2 Write Boost enhances the write flow algorithm for small I/O requests. Every block of information that is not committed to SSDs is protected by local and remote NVRAM. This allows a host write to be acknowledged immediately after information is written to the Write Boost area of NVRAM. The actual write to the physical SSDs can occur later. The write to physical SSDs also allows X2 to group smaller host writes into 16K block physical SSD writes, thus reducing the number of physical writes to SSDs, and enhancing SSD endurance.

See Figure 11 for X2 write flow with Write Boost.

![X2 Write Flow with Write Boost](image)

Writes to the SQL Server transaction log always aligns with the physical sector size (512 byte or 4 kilobytes, depending on the configuration). Log write block size can an integer multiple of the physical sector size with up to 60K. Many of the log write requests can be smaller than 16K, especially with OLTP type of workloads which performs smaller transactions and requires frequent commits. We ran the simple T-SQL scripts to compare the transaction log write latency in XtremIO X2 with Write Boost enhancement and XtremIO X1.

```sql
WHILE (@I < 100000)
BEGIN
  BEGIN TRAN
  INSERT INTO tblTest values ('A', @I)
  COMMIT TRAN
  SET @I = @I + 1
END
```

The average latency for transaction log write with Write Boost is 120 microseconds. That is over two times faster than X1, which has a latency of 259 microseconds.
Storage Efficiency

XtremIO offers a wide range of storage efficiency features that are global, inline, and always-on. These storage efficiency features not only significantly reduce the storage footprint for running SQL Server databases on XtremIO, but also effectively improve SSD performance and longevity by reducing the number of write operations to the SSDs. XtremIO storage efficiency features includes thin provisioning, inline data deduplication, and inline data compression.

Thin Provisioning

Legacy storage systems use thick instead of thin provisioning. Thick provisioning typically requires projection of potential application storage usage for three years, with full storage costs included at purchase time. In many cases, the storage is over-provisioned, resulting in a large amount of space wasted for potentially a long time. Thin provisioning is built to increase storage utilization and maximize organization storage investment.

XtremIO storage is natively thin provisioned, supporting 100% storage on demand. With thin provisioning, the IT administrator provisions storage to SQL Server as usual, but the XtremIO system consumes capacity only when it is actually needed. Thin provisioning optimizes storage utilization by eliminating any provisioning of storage which in the end will not be needed.

SQL Server Pre-Allocated Space and Thin Provisioning

A typical SQL Server deployment pre-allocates 10 to 30 percent of space to anticipate data growth and to avoid auto growth of data files during run time. The portion of pre-allocated space is zero initialized but not written by SQL Server.
The XtremIO native thin provisioning provides many advantages:

- Database administrators can be more generous in assigning pre-allocated space to avoid the performance overhead for growing the database files.
- IT administrators do not have to worry about storage space going to waste due to a large amount of pre-allocated space not being used.
- IT administrators have more flexibility in sizing. Applications like SQL Server rarely used the full provisioned capacity. The ability to consume capacity on demand allows IT administrators to more efficiently provision or overprovision the physical capacity of the storage.

Thin provisioning is an integral part of the XtremIO content-aware architecture. Unlike some disk-oriented architectures, XtremIO thin provisioning has no performance impact and does not cause any fragmentation issues.

While thin provisioning offers many benefits, it requires IT administrators to implement measures to monitor percentage of available physical capability. To avoid issues related to SQL Server operation, additional storage should be added early when the amount of available physical capability is low. XtremIO alerts and non-disruptive cluster expansion features allow for proactively monitoring and scaling of space capacity.

**Inline Data Deduplication**

Data is stored according to its content on XtremIO. When a data stream enters the system, it is divided into 8K data blocks and fingerprinted based on its content. Each data block is automatically globally checked for duplication across the entire XtremIO cluster. Only unique data that does not already exist on the XtremIO array will be written onto the SSDs.

XtremIO’s unique architecture using content fingerprints not only enables inline data deduplication, but also provides flexibility for data distribution. Data blocks within any SQL Server volume are distributed evenly across all SSDs in the cluster. Data is automatically balanced to avoid any hot spots on the SSDs for optimal flash wear leveling. Performance is automatically load-balanced across all storage controllers in all X-Bricks in the XtremIO cluster.

XtremIO Inline Data Deduplication and its intelligent data storage process ensure:

- Balanced usage of the system resources, maximizing the system performance
- Minimum amount of flash operations, maximizing the flash longevity
- Equal data distribution, resulting in evenly balanced flash wear across the system
- No system level garbage collection (as opposed to post-processing data reduction)
- Smart usage of SSD capacity, minimizing storage costs
**Single Database**

SQL Server stores data in pages. Each page is an 8K data unit. Figure 13 depicts the structure of a SQL Server page. Each page contains a header with fields that are unique to the page. The actual row data is located on the remaining portion of the page. When deploying a single database on XtremIO, the inline data deduplication is not expected to reduce the on-disk footprint of the database due to the unique page header on each data page.

![Structure of a SQL Server Data Page](image)

**Multiple Copies of Database**

There are often needs for more than one copy of a database due to reporting, test, development, or other needs. Figure 14 shows the effects of inline data deduplication when deploying multiple copies of the same database. The figure shows deployment of 1, 2, 4, and 8 copies of a database. While the provisioned capacity and the space reserved and used by SQL Server multiplies as the number of copies increases, XtremIO only consumes the physical space for a single database. When copies are initially created, there is no additional space required. Space consumption will only increase afterwards as changes are made to the content of the database.

![Effects of Deduplication on Copies of Database](image)
INITIAL XVC CREATION

When an XVC is created, the metadata for the production source volume becomes an “ancestor” entity that is shared between the production source volume and the copy. New empty containers are created for subsequent changes to the production source volume and the copy. Thus, the act of creating a copy is extremely efficient that requires in-memory metadata operation only. Zero space is consumed at the time an XVC is created. Virtual Copy capacity consumption occurs only if a change requires writing a new unique block.

![Initial XVC Creation Diagram](image)

POST XVC CREATION

After creation of an XVC, both the production source volume and the copy continue to be written to and read from. Any INSERT or UPDATE operation to the source or copy will result in new metadata entries in the corresponding volume. If unique data is identified, it will be compressed, and then written to the SSD. As shown in Figure 16, an UPDATE triggered data B on Block 1 to change on the production source volume and a new entry is written to the production metadata volume. As the content of the data block is identified as unique, it is compressed and subsequently written to the physical media.

XVC efficiently shares metadata across the source and its copies. Database operations such as UPDATE or DELETE essentially trigger a decrease reference count of the underlying physical data block(s). A data block is marked for removal from the physical media only when there are no more references to it.

Conventional copy-on-write technology requires data blocks to be copied before the source can be updated. Writes to a source volume could potentially trigger an explosion of space consumption when the number of copies is large. XVC effectively prevents data explosion with the unique redirect-on-unique-write copy technology—only changed data blocks that are unique and have never been written to the physical media will take up additional space.

As Figure 16 illustrates, existing copies of the source volume do not consume additional space due to changes in the source volume. They continue to share common metadata and physical data blocks. This enables XtremIO to support a large number of copies on any volume without worrying about an explosion of disk space.
Similarly, changes to a copy will affect the copy only. The space usage and performance of the production source volume or other copies will not be impacted.

A copy has the same space efficiency as a normal volume for any data written post XVC creation.

As the data on the production source volume and the copies diverge over time, the storage efficiency savings will decrease. As such, it is a good practice to refresh copies periodically, after major data loads, or maintenance operations, to receive up-to-date production data and regain the space savings.

**Data Compression**

As part of the write I/O flow, XtremIO automatically compresses data after all duplications have been removed. Compression is a global, inline, always-on operation that is performed only on unique data blocks. XtremIO uses an algorithm, based on Lempel-Ziv, which optimizes the balance between compressibility and resources allocation.
Compression complements data deduplication for SQL Server deployments. Whether you are deploying a single database, multiple copies of the same database, or multiple application databases, the inline compression automatically reduces storage footprint by storing data blocks in the most efficient manner. The XtremIO system can reach compression ratios of 16:1 for unique data as effective. Overall compressibility is determined by the nature of the data.

**Example Diagram: Data Savings**

- **Overall Efficiency**: 35.7:1
- **Thin and Copy Savings**: 94%
- **Logical**: 736.40GB
- **Duplication**: 1:1
- **Compression**: 2.1:1
- **Physical**: 347.62GB

**SQL Server Native Data Compression**

SQL Server database administrators are familiar with its native data compression features; namely row compression and page compression. Row compression stores fixed-length data types in variable-length format, and does not use storage for NULLs and white space. Page compression looks for further data reduction opportunities by minimizing redundant column data in one or more rows on an 8K page. Redundant column data is stored once on a page and referenced by multiple columns.

While implementing row compression and page compression effectively reduces the space used by SQL Server, there are a few challenges that prevent their wide adoption:

- Compression and de-compression consumes host CPU cycles. Especially for page compression, the CPU overhead could be substantial, and you may even see negative impact on query performance if the workload is doing many updates. A more detailed study by the SQL Server team is available at [https://technet.microsoft.com/en-us/library/dd894051(v=sql.100).aspx](https://technet.microsoft.com/en-us/library/dd894051(v=sql.100).aspx). Due to the potential performance overhead, enabling row or page compression should be carefully evaluated at a table, partition, or even index level.

- Enabling compression requires moving data to a new record format, and removing duplications. As you can imagine, the process is time and resource intensive. After first being enabled, the initial compression process can take hours or days depending on the size of your database. That could be a big burden when managing many databases.

- Space savings from compression cannot be reclaimed until the database or data files are shrunk. As is widely known, the shrink process is an intrusive operation which is both time and resource intensive.
XtremIO Array Compression and SQL Server Native Data Compression

XtremIO inline data compression feature can complement SQL Server native data compression in many ways.

In the lab, we studied storage footprint of a database under the following three scenarios:

- Disabling native SQL Server compress features
- Enabling row compression
- Enabling page compression feature turned on

We used the SQL Server Management Studio Disk Usage report to compare the actual database space used in each scenario. XtremIO physical capacity counter was used to measure physical storage consumption on XtremIO.

As shown in Figure 19, the space savings are 54%, 58%, and 60% respectively for the three scenarios above. The out-of-box inline data compression feature from XtremIO alone provides a 54% storage footprint reduction at no cost. XtremIO inline data compression also works well together with SQL Server native compression and can provide additional storage savings on top of the SQL Server native row and page compression features. Customers who use SQL Server native compression features can get the best of both worlds.

Figure 19. XtremIO X2 Compression Working with SQL Server Native Compression Features
Lifecycle Management

Typical SQL Server deployments have a separate non-production environment for continuous application development and lifecycle management. This environment is typically used for the following:

- **Test/Dev** – for continuous application feature development and testing
- **Maintenance** – to perform resource-intensive database maintenance tasks, such as dbcc and checkdb
- **Operational Management** – to support upgrades, performance tuning, and pre-production simulation
- **Reporting** – to serve as the data source for any business intelligence system or reporting

Maintaining a setup that resembles the production environment can significantly reduce risks associated with releasing new features, and can reduce time to market.

One of the key benefits of XtremIO iCDM technology is the ability to provide a cost-efficient lifecycle management environment. iCDM provides efficient storage layer copy data management to consolidate both primary data and its associated copies on the same scale-out, all-flash array thus providing unprecedented agility and efficiency. In short, XtremIO and iCDM together promise incredible potential to consolidate both production and non-production applications without impacting SLAs by providing the following advantages:

- Consistent IOPS and latency
- Linear scale-out all-flash performance
- On demand performance and capacity scaling with no application downtime

Use Case I: Test and Dev

As application features continue to be developed, tested, and rolled out to production, copies of the existing production database are needed for various purposes such as supporting development efforts and performing user acceptance testing. As the production database continues to change and grow, database administrators face some serious challenges including:

- How to deploy and refresh test and development environments efficiently as compared to traditional backup/restore methods which take too long and are too resource intensive
- Where to find the additional storage to host multiple test and development copies of the database
- How to ensure test and development are running on storage with similar performance

**Example:**

XYZ company has a 5 TB production database. The company's business processes require it to host 12 versions of the database to support 100 developers for development, staging, testing, and other needs. Before XtremIO, XYZ company provided its developers with copies of the database by restoring backups of the production database. This took a few hours for the database to be restored in each environment. In addition, it was a constant struggle to find disk space to accommodate the copies. Copies were restored into a separate storage array so that the development copies would not impact production. The developer always struggled for storage I/O because the spinning disk array just couldn't keep up.

After implementing XtremIO and taking advantage of XtremIO iCDM, XYZ company improved its production database performance. With iCDM, the company could deploy multiple copies of the database in seconds on the same array as the production database to support its development and test environments. With XtremIO inline storage efficiency and space-efficient XVC, the company achieved a 10:1 data reduction ratio. DBAs no longer had to beg and borrow storage space. With the consistent, superior performance, and space-efficient copy provided by XtremIO all-flash storage, XYZ company has plenty of performance and capacity to support its consolidated production, test, and development environment.

XtremIO iCDM can be leveraged to provide test and development copies of production data. Multiple master copies can be created, and each copy can be processed (such as an anonymization/sanitization process) to build a golden image for different development and test stages. Child copies can then be created from each master copy and presented to various development or test teams. Provisioning copies is an easy and instantaneous process.

For an example of a deployment model of repurposing copies for test and development environment using PowerShell, see Appendix I: Repurposing Copy for Test/Dev.
Use Case 2: Database Maintenance

dbcc checkdb is a maintenance task that checks for the logical and physical integrity of all the objects in the database. It is an important task to perform to ensure the health of the SQL Server database. However, dbcc checkdb can be very disruptive to a production SQL Server. The SQL Server buffer pool may be completely trashed as checkdb reads all the pages of all the objects. Running dbcc checkdb on a production server is not a recommended practice. As a result, many database administrators go through a lengthy process to restore a copy of the production database onto a non-production environment to perform dbcc checkdb. Others may not be able to run dbcc checkdb, thus incurring a higher risk of data corruption.

Example:
Tommy is the DBA of XYZ company. In the past, Tommy would run dbcc checkdb on the production SQL Server database every week during the regular maintenance window. Since the recent company merger, the size of the database has grown by 70 percent. The dbcc checkdb job now takes longer to complete. Tommy is receiving complaints from users on applications running slowly during certain hours of the day. Performance data identified the cause of performance degradation as the dbcc checkdb job. Fortunately, XYZ company recently moved its SQL Server environment to XtremIO storage. To reduce the performance impacts to the production environment, Tommy was able to create a copy of the production database instantly by using XtremIO iCDM, attaching the copy to a secondary SQL Server host, and offloading the dbcc checkdb job to run on the secondary host. The copy is deleted after the dbcc checkdb job is done.

With XtremIO iCDM, database administrators can easily attach a copy of the production database to a secondary SQL Server instance to run dbcc checkdb without worrying about trashing the production SQL Server buffer pool. In addition, with the XtremIO snapshot refresh feature, database integrity can be periodically checked on a copy to ensure the ongoing health of the database.

Use Case 3: Query Tuning

As an application continues to mature and new features added, the existing database indexes may cease to be sufficient for supporting user activities. Index tuning is one of the daily tasks for any SQL Server database administrator. Effective query tuning requires running on a production-like database and may entail many trials and errors. Tuning on a production environment is not practical. However, getting a copy of the terabyte-sized production database for query tuning is also very challenging, both from a storage capacity requirement perspective and a deployment time perspective. Many SQL Server database administrators face this dilemma with traditional storage.

Example:
Tommy is the DBA of XYZ company. Tommy has been receiving user complaints about performance issues of the production SQL Server database. Given the increase in the size of the database due to a recent company merger, Tommy suspects that the existing indexes on the database might no longer be sufficient. Tommy needs to do a thorough examination of the existing indexes, removing some old indexes, and evaluating new ones for possible performance improvement. He needs an environment with a production-like database to assess the impacts of the index changes before rolling out the changes to production.

With the production SQL Server database running on XtremIO storage, Tommy was able to create a copy of the database instantly, mount and recover the copy onto a secondary SQL Server host, and perform exhaustive index tuning on production data without worrying about impacting the performance of the production environment.

XtremIO iCDM can create a fully writable copy of a database instantly without using additional physical storage space. When attaching the copy to a secondary SQL Server instance, you create an environment dedicated for tuning without impacting the production workload.
Use Case 4: Reporting

One of the primary purposes of a database is service reporting. Report queries typically run for a long time, require a large amount of CPU and memory resources, and are heavy on disk I/Os. Mixing reporting and online user activities on the same SQL Server machine may cause resource contention, locking, and deadlocking issues. The ability to offload reporting, and separate reporting activities from primary online user workloads, helps to reduce load and improve performance on the production environment.

Example:
XYZ company provides online education services offering K-12 education to kids nationwide. August is the company's peak season with students returning to school and new student enrollments. Last summer, the company enrolled so many new students, that the CPU on the production SQL Server cluster was completely overwhelmed.
To alleviate the CPU pressure from the production SQL Server serving online user requests, XYZ company was able to leverage the XtremIO iCDM feature to create near real-time copies of the production database every 15 minutes and offload some heavy reporting workloads from the production SQL Server to a secondary SQL Server host.

XtremIO iCDM enables you to easily build a solution that scales out a database to provide near real-time copies of the production data for reporting purposes.

Use Case 5: Upgrade

As applications and SQL Server continue to mature, more features are released. Environments need to be upgraded to take advantage of these new features. XtremIO iCDM can enhance the upgrade experience by simplifying the upgrade process, minimizing planned downtime, and reducing the risks of the upgrade.

Example:
XYZ company is about to undergo a major upgrade of its online education system. Several schema and data changes are planned for the SQL Server database. As the lead DBA, Tommy is expected to provide database support during the application upgrade. Tommy has been challenged to provide a quick fallback plan for the 5 TB production database in case anything goes wrong during the upgrade.
Tommy immediately considered generating an application consistent copy of the database using AppSync. If the database needs to be rolled back for any reason, he can restore the older version of the database from the application-consistent copy in seconds. A copy takes virtually zero space on XtremIO, so there is no need to beg and borrow for additional storage.

With the capability to create an instant writable copy of the database without any additional storage space, you can enhance the upgrade process by using XtremIO iCDM to:

- Test the upgrade on a copy of the database to resolve any potential issues.
- Run through the upgrade process using a copy of the database to achieve a real sense of how long the upgrade will take.
- Protect the database by generating a copy prior to the upgrade; if the upgrade fails for any reason, simply point the SQL Server instance to the copy and recover.
High Availability and Data Protection

Many organizations deploy SQL Server AlwaysOn Availability Groups (AlwaysOn AG) to provide local high-availability protection for SQL Server databases. With the AlwaysOn AG architecture, each replica has its own copy of the database. SQL Server manages the data synchronization between the replicas. AlwaysOn AG has many benefits over other local high-availability solutions such as:

- Less than a five-seconds failover
- Easy setup and maintenance
- A readable secondary

However, organizations using AlwaysOn AG to protect their databases typically encounter the following issues:

- Storage costs multiply due to the number of replicas deployed.
- Seeding and reseeding secondary replicas can be time-consuming (especially for large databases or databases with high transaction rates) during which time the databases will be running without protection.

AppSync integrates with both XtremIO all-flash array and SQL Server to enable an application-consistent copy of the database. SQL Server has complete knowledge of the application consistent copy and treats the storage copy as a full database backup. Hence an application consistent copy can be mounted and recovered as a database with “No Recovery” mode and serves as an AAG secondary replica to continue receiving log stream synchronization from the primary. The nature of XtremIO iCDM enables not only rapid deployment, but also extreme storage efficiency.

Logical Data Protection and Fast Recovery

Traditionally, backup is performed using SQL Server native backup tools or third-party software backup solutions. Traditional backup solutions require reading through the data and writing the data back on disk. This approach is time consuming, and I/O and network resource intensive. The resulting backup has a large on-disk footprint. Because of these limitations, backups need to be scheduled during off peak hours, and only a limited number of backups can be retained on the local data center. Backups are typically stored off the primary storage to save storage cost even at the cost of higher RTO.

XtremIO integrates with EMC AppSync to provide application consistent snapshot backup / restore capability to SQL Server. AppSync interfaces with SQL Server Virtual Device Interface (SQL VDI) to coordinate the snapshot backup with XtremIO.

We compared the backup time of a 1TB database using traditional SQL backup method vs. application consistent snapshot backup using AppSync. The backup time was reduced from 28 minutes to 1 minute when using application consistent snapshot backup. Because the snapshot backup is a complete metadata operation which does not require copying or moving any physical data, the larger the database, the greater the reduction in backup time.

AppSync support full backup and copy-only backup with application consistent snapshot. With the full backup option, the data and the active part of the transaction log is backed up. A full backup can be restored into No Recovery state to allow additional transaction logs to be restored. This option supports point-in-time recovery with transaction logs.

The copy-only option backs up the database without affecting the sequence of a conventional backup. This option creates a backup of the database without interfering with third-party backup applications that may be creating full and/or differential backups of the SQL Server databases.

Creating full database backup using the AppSync application-consistent snapshot capability is an effective way of obtaining a fast-recoverable copy of the database. To further protect database in the events of catastrophic disaster or physical storage failure which is beyond recovery, additional measures such as replication or remote backup to offsite storage should be used for complete data protection.
Storage Monitoring

Monitoring, managing and optimizing storage health are critical to ensure performance of a SQL Server database. Simple and easy-to-use has always been the design principle for XtremIO Management Server (XMS). With XIOS 6.0, XMS delivers an HTML5 user interface for consumer-grade simplicity with enterprise-class features. The improved user interface includes:

- Contextual, automated workflow suggestions for management activities
- Advance reporting and analytics that make it easy to troubleshoot
- Global search to quickly find that proverbial needle in the haystack

The simple, yet powerful user interface drives efficiency by enabling administrators to manage, monitor, receive notifications, and set alerts related to the storage. With XMS, key system metrics are clearly displayed in an easy-to-read graphical dashboard. From the main dashboard, you can easily monitor the overall system health, performance and capacity metrics, and drill down to each object for additional details. This information allows you to quickly identify potential issues and take corrective actions.

Figure 20. XtremIO XMS
XtremIO collects real time and historical data (up to 2 years) for a rich set of statistics. These statistics are collected at the Cluster/Array level as well as the object level (Volumes, Initiator Groups, Targets etc.). This data collection is available from day one, enabling XMS to provide advanced analytics of the storage environment running SQL server.
Advanced Analytics Reporting

SQL Server data access pattern can be unpredictable depending on factors such as application operations and table indexes. Thus, storage sizing for an SQL Server environment can become very complex. XMS built-in reporting tracks data traffic pattern, thus significantly simplifying the sizing effort.

With X2 release, XMS provides a built-in weekly pattern reporting widget that tracks data traffic patterns. You can easily discover IOPS pattern at any time during the week, and understand if the pattern is sporadic or consistent over a longer period.

Figure 22. Weekly Patterns Reporting Widget
The CHANGE button on the widget tracks and displays changes (increasing or decreasing) of the past week relative to the previous 8 weeks. If there is no major change (i.e. if the hourly pattern in the past week was no different than the previous 8 weeks), then there will be no up/down arrow indication. However, if there is an increase/decrease in the traffic relative to the previous 8 weeks, a visual arrow indication will appear.

Figure 23. Weekly Reporting on Relative Data Pattern Changes

You can also select to drill down for data traffic statistics for a specific hour.

Figure 24. Drill Down Report on Specific Hour
Provision for SQL Server

SQL Server maps a database to set of files on disk. Data and transaction log information is stored in separate files. To maximize performance and operational efficiency of SQL Server, optimizing database file layout has been the focal point of discussion for many years. In this section, we will examine the considerations and recommendations for this mapping, and discuss relevancy for provisioning a SQL Server database on XtremIO.

Common considerations for SQL Server database file layout include:

- RAID configuration
- Separation of SQL Server files
- Number of LUNs
- Number of files

RAID Configuration

SQL Server typically gives the following RAID recommendations, considering performance, data protection, and cost:

- RAID 10 for user data and log files for best performance and availability. When cost is a key concern, RAID 5 or equivalent can be used.
- For write intensive TempDB, use RAID 10. RAID 0 can be used if cost must be reduced, although the system may become unavailable during disk failure if RAID 0 is used. RAID 5 is sufficient for non-write intensive TempDB.
- Use a large number of smaller disks instead of a smaller number of large disks.

When deploying SQL Server on XtremIO, RAID configuration consideration is no longer relevant.

- XtremIO system has built-in a "self-healing" double-parity RAID as part of its architecture.
- XtremIO Data Protection (XDP) is designed to take advantages of flash media specific properties and XtremIO content addressable storage architecture.
- With the content addressable storage architecture, SQL Server files are automatically distributed across all SSDs, and processed by all storage processors in the cluster.

Best Practices Tips:

- With XtremIO, no RAID configuration is needed. A flash optimized RAID (XDP) is built in and pre-configured on XtremIO.
Separation of Files

Common SQL Server recommendations:

- Allocate separate LUNs for data, log, and TempDB files. Allocate separate physical disks for LUNs used by data and log.
- Allocate separate LUNs for different workload types (e.g. OLTP vs. Analytics).

The reason for file separation is due to SQL Server I/O characteristics. In addition, it optimizes for the less efficient spinning hard disk media used in traditional storage.

SQL Server accesses data and log files with very different I/O patterns. While data file access is mostly random in nature, transaction log file access is sequential only. Traditional storage built with spinning disk media requires re-positioning of the disk head for random read and write access. Hence, sequential data is much more efficient than random data access. Separating files that have different random vs. sequential access pattern helps to minimize disk head movements, thus optimize storage performance.

This guideline is no longer applicable when deploying SQL Server on XtremIO all-flash array.

- XtremIO uses all solid-state disks (SSDs). SSDs have no moving parts. It does not matter where the physical data block is located as there is no disk head movement for read/write access.
- With XtremIO's unique architecture, data is evenly and intelligently distributed across all SSDs by its content addressable storage engine to benefit from the processing power of all the storage processors.

There is no performance gain in allocating different LUNs for SQL Server data, log, TempDB files, or different workload types. However, from an operational efficiency prospective, one may consider separating TempDB files from the database files (i.e. user data and transaction log files). TempDB is a global resource that is shared by all databases within a SQL Server instance. It is a temporary work space that is recreated each time an SQL Server instance starts. When using XtremIO snapshot technology to create backup, deploy copies of a database, or replicate database to a secondary site, separating TempDB would allow the snapshot to apply to database files only, thus removing any unnecessary noise.

Best Practices Tips:

- No performance consideration for file separation.
- For more efficient use of copy data services or storage replication, consider the following:
  - Separate TempDB from user data files and log.
  - Unless your business transaction spans multiple databases, do not share LUN across multiple databases; LUN/volume is the smallest unit for snapshot operation.

Number of LUNs

SQL Server allows database objects and files to be grouped in filegroups. Filegroups provide the means to separate user objects into different LUNs or physical disks. Because a single file cannot cross multiple filegroups, you can never have data from objects assigned to different filegroups residing in the same physical file. SQL Server storage best practices recommend:

- Use filegroups for administration requirements such as backup / restore, partial database availability, etc.
- Use data files to “stripe” the database across your specific I/O configuration (physical disks, LUNs, etc.).
- Unless you understand the application very well, avoid trying to over-optimize the I/O by selectively placing objects on separate spindles.
These guidelines can be significantly simplified for most SQL Server deployments.

- Generally, depending on the application, a single LUN is sufficient to utilize an XtremIO storage system to its maximum I/O capability.
- Multiple filegroups should continue to be used for enhancing database availability, managing table partitioning, etc.
- There is no need to over-optimize the I/O by selectively placing objects on separate spindles. A single LUN per database provides the performance needed for most SQL Server deployments. LUN configuration should be kept simple per standard recommendations by leveraging XtremIO capability of automatically balancing data and workload.

Best Practices Tips:
- A single LUN per database suffices for most deployment scenarios. XtremIO automatically takes care of even distribution of data, and ensures maximum parallel processing.
- For extremely large scaled I/O intensive SQL Server deployment, the following tuning options should be considered:
  - Increase LUN queue depth, as described in Queue Depth.
  - If I/O exceeds the maximum allowable queue depth, spread database files across multiple LUNs to increase overall queue depth beyond the maximum configurable value per LUN.
- If storage is dedicated to a single SQL Server database, a minimum of four LUNs/volumes should be created on the XtremIO cluster to maximize cluster performance by taking full advantages of the parallel processing powers of all the storage controllers.

Number of Files

SQL Server “stripes” allocations across files within a filegroup by using a proportional fill algorithm. Each file has its own Page Free Space (PFS), Global Allocation Map (GAM) and Shared Global Allocation Map (SGAM) pages. These special “administration pages” track the free space and allocation in the file. If the files in the group have the same size as recommended by SQL Server, the allocation is round-robin.

Having multiple data files provides scalability advantages for allocation intensive workloads. This is especially true for TempDB where activities tend to be allocation heavy.

SQL Server allocation behaviors do not change when running on XtremIO. This recommendation should be followed to optimize TempDB deployment, and improve performance of workloads with heavy insert activities.

Best Practices Tips:

When deploying SQL Server on XtremIO, be aware that many of the existing SQL Server storage considerations and recommendations center around the traditional spinning hard disk media. XtremIO all-flash array is not just built with the more advanced SSD media that is optimized for both random and sequential access, but also its intelligent content engine and unique scale-out architecture significantly simplifies the complexity of provisioning storage for a SQL Server. With XtremIO, database administrators should be free from worrying about RAID, spindle counts, or any of the underlined storage configurations.
Multipathing

Multipathing facilitates the I/O data routing over redundant hardware paths connecting a host to storage. If any component along the storage path fails, (for example, cabling, host bus adapters (HBAs), switches, storage controllers, or even power), the multipath software resets the connection and passes the request over an alternate path. Applications such as SQL Server can continue to service I/Os without any interruption. In additional to protecting application from hardware path failure, multipathing also enhances application performance by load balancing I/O across all available paths to optimize resource, maximize throughput, and reduce I/O latency.

XtremIO supports native multipathing software from vendors or multipathing using EMC PowerPath.

- XtremIO supports native multipathing using Microsoft Native Microsoft Multipath I/O (MPIO) with Windows Server 2008 and above. For optimal operation with XtremIO storage, configure the Least Queue Depth policy for MPIO for devices used with XtremIO. With this policy, I/O is sent down the path with the fewest outstanding I/O requests.
- XtremIO supports the VMware vSphere Native Multipathing (NMP) technology. For best performance, it is recommended to do the following:
  - Set the native round robin path selection policy on XtremIO volumes presented to the ESX host.
  - Change the vSphere NMP Round Robin path switching frequency for XtremIO volumes from the default value (1000 I/O packets) to 1.
- For details on installing and configuring PowerPath with XtremIO native class support on your host, refer to the EMC PowerPath on Windows Installation and Administration Guide or EMC PowerPath on VMware vSphere Installation and Administration Guide. The guide provides the required information for placing XtremIO volumes under PowerPath control, and ensures optimal load distribution and availability of I/O paths to the XtremIO storage.


Best Practices Tips:

- For Windows 2008 and above, use Least Queue Depth policy with Native Microsoft Multipath I/O (MPIO).
- For SQL Server on VMware vSphere, use VMware native round robin path selection policy, and set switching frequency to 1.
- For best practices with PowerPath, refer to EMC PowerPath on Windows Installation and Administration Guide or EMC PowerPath on VMware vSphere Installation and Administration Guide.

Queue Depth

SCSI device drivers have a configurable parameter called the queue depth that determines the maximum number of outstanding SCSI commands or I/O requests a given LUN can have at one time. If the queue depth value is too low, the excess I/Os are queued at the application side. SQL, Server I/O latency increases, and throughput can suffer. If the queue depth value is too high, the storage system cannot keep up with the I/O processing and will become overwhelmed, thus impacting I/O performance for all applications running on the storage system.

An optimal queue depth setting should strive for a balanced system design and account for the following:

- SQL Server IOPS and throughput requirements
- Number of LUNs used by SQL Server
- The maximum IOPS and throughput the storage system can support
- The number of hosts and initiators that are connected to the storage system
- Types of HBA (brand and bandwidth) used to connect the hosts and the storage system
The queue depth is per-LUN, and not per-initiator. Typical host bus adapter (HBA) vendors pre-configure the queue depth value to 32. The value may vary by vendor or virtualization implementation. Each initiator or host bus adapter (HBA) port can support much higher number of concurrent I/O requests, typically in the range of thousands, depending on specific vendor implementation.

Most of the vendor settings for default queue depth are optimized for traditional hard disk storage systems. An all-flash array like XtremIO can process I/O requests in micro-second. In addition, its unique scale-out architecture means it can process millions of concurrent I/O requests. Given the simplicity of XtremIO design, the number of LUNs can be reduced to a single LUN, so that the queue depth may become the limiting factor hindering SQL Server performance.

For optimal operation with XtremIO storage, set the queue depth to 256. Since queue depth setting is a global setting, if you have a mixed environment with a host connected to multiple storage platforms, check the recommendations for all storage platforms to avoid performance issues.

For deployment on VMware vSphere environment, queue depth is further controlled by limits set on the vSCSI adapter and VMKernel scheduler. To ensure proper configuration, queue depth configuration should be set on all layers, including physical HBA, vSCSI adapter, and VMKernel scheduler.

- For details on changing queue depth for vSCSI adapter, see: http://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=1267
- For details on configuring VMKernel scheduler admittance policy, see: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1268

We recommend setting Disk.SchedNumReqOutstanding to the maximum supported value of 256 for XtremIO volumes.

Refer to XtremIO Host Configuration Guide (https://support.emc.com/docu56210_XtremIO-Host-Configuration-Guide.pdf?language=en_US) for detailed steps on setting queue depth with different HBA vendor implementations.

**Best Practices Tips:**
- Set LUN queue depth to 256 for QLogic HBA or 128 for Emulex HBA.
- For SQL Server deployment on VMware vSphere, set queue depth on all layers, including physical HBA, vSCSI adapter, and VMKernel scheduler when configuring queue for SQL Server running on VMware vSphere.
- Set Disk.SchedNumReqOutstanding to 256.

**512B vs. 4K Physical Sector Size**

XtremIO can present either 512B or 4K physical sector size to Windows and SQL Server. As part of the “creation volume” options, you may specify for a volume to use either 512 LBs or 4KB LBs. Up until recently, Microsoft Windows and SQL Server were based on 512B disk sectors size. Microsoft started native support for 4K sector size with release of Windows 2012. More details on this support can be found on the following support sites:

- https://support.microsoft.com/en-us/kb/2510009
- https://support.microsoft.com/en-us/kb/926930

SQL Server stores data in pages of 8K unit. Using 4K sector size enables a SQL Server data page to occupy two physical sectors, instead of 16 physical sectors with 512B sector size. 4K sector size aligns efficiently with SQL Server data page 8-KB boundary and reduces the amount of metadata overhead. The effect of the metadata improvement is apparent with metadata-heavy operations such as ODX copy. With ODX copy of a set of SQL Server files with total size of 1TB, the average bandwidth for copying 512B to 512B volume is 6G/s. In contrast, the average bandwidth for copying from 4K to 4K volume is 12G/s.
While 4K sector size offers better storage capacity with less metadata overhead, there are a few caveats you should be aware of before presenting 4K sector size volumes to SQL Server.

- SQL Server transaction log writes always align with the physical sector size. Log write block size can be one or multiple of the physical sector size up to 60Ks. With 4K sector size, an application that performs a large number of small write operations with frequent commits may see increased usage of log space.

- Earlier versions of SQL server prevent restoring or attaching a database in an environment that has a larger physical sector size than the sector size the database was formatted with. See details on: http://blogs.msdn.com/b/psssql/archive/2011/01/13/sql-server-new-drives-use-4k-sector-size.aspx

- SQL Server versions prior to SQL Server 2005 ship with system and sample databases that are 512B based. Configuring legacy SQL Server versions with 4K sector volume will fail.

- When placing TempDB and user database on different volumes, be aware that each will use the sector size reported by the operation system at the time of creation based on the volume sector size in which the file is located. Variance of sector sizes can occur when database or TempDB files are located in volumes with different physical sector size. Variance of sector size in I/O path should be avoided to prevent any suboptimal data access.

- If you run SQL Server in a virtual environment, be aware that 4K physical sector size is not yet supported by all Hypervisor vendors. See the following links for more details on VMware vSphere and Windows Hyper-V support:
  - https://support.microsoft.com/en-us/kb/2515143

Before deploying a SQL Server database, always check on the physical sector size of the volumes provisioned for SQL Server. You can do that by running FSUTIL. Figure 25 shows sample FSUTIL output in case of 512B and a 4K sector volume size.

![Figure 25. FSUTIL](image-url)
As a SQL Server database administrator, you can also run dbcc file header to check the sector size of the database set during creation, and the physical sector size of the volume on which the database currently resides. Figure 26 shows sample outputs of dbcc file header for a database created on 4K and 512B sector size. The SectorSize shows the size with which the database was created. The ActualSectorSize is the current sector size of the volume on the database. For best performance, the SectorSize should be equal to the ActualSectorSize.

Figure 26 shows sample outputs of dbcc file header for a database created on 4K and 512B sector size. The SectorSize shows the size with which the database was created. The ActualSectorSize is the current sector size of the volume on the database. For best performance, the SectorSize should be equal to the ActualSectorSize.

Determine what to use:

4K is the more advanced technology to which the storage and software industry is heading. It enhances SQL Server data page storage and reduces metadata overhead. For any new database development, 4K sector size would be a good choice as the new standard. However, despite the advantages 4K sector size offer, 512B sector size remains the more widely supported sector size so that it is less likely to cause any application or tools support issue. Be sure to check support on software platform and storage integrated tools when deploying on 4K sector size.

Best Practices Tips:

- Use 4K sector size for physical Windows and Hyper-V deployments with standalone XtremIO deployment.
- Use 512B sector size if deployment includes products such as RecoverPoint for replication, or VPLEX.
- When moving existing database to the XtremIO platform, check the sector size on the existing database using dbcc fileheader. Align XtremIO volume sector size with the sector size of the existing database.

Allocation Unit Size

Allocation Unit Size is a configurable option for formatting an NTFS volume. It should not be confused with physical sector size. Physical sector size specifies the smallest unit that can be written by an application such as SQL Server. The allocation unit size is the smallest unit of storage that any individual file can occupy.

The default allocation unit size on a windows drive is 4KB. When you format the drive, you can set that to a larger size. 64KB is recommended for SQL Server data, log, and TempDB files. This is due to the following reasons:

- SQL Server files are typically much larger than 64K.
- SQL Server allocation is by extent. Each extent in SQL Server is eight 8K pages, which is 64K.
**Application vs. Crash Consistent Copy**

XtremIO copy data services can be used to enhance many SQL Server usage scenarios and simplify day-to-day operations for a database administrator. XtremIO copy data services is based on snapshot technology. It supports both crash consistent snapshot and application consistent snapshot. What is the difference between crash consistent snapshot and application consistent snapshot? When should each be used?

**Crash Consistent Copy**

Crash consistent snapshot captures the state of the data volumes at a particular point in time. SQL Server has no knowledge of the snapshot. As such, there is no impact on a running SQL Server instance. A crash consistent snapshot of a SQL Server database is equivalent to the state of a running database during a power failure. In that state, there may be data pages in memory (open transactions) not yet flushed to disk. SQL Server uses write ahead logging (WAL). All committed transactions are logged in SQL Server transaction log on disk. SQL Server can recover from a crash using information in the transaction log. Data pages not yet flushed to disk can be re-created. In flight transactions will be marked as failure.

**Application Consistent Snapshot**

Application consistent snapshot requires coordination of the snapshot with SQL Server. XtremIO integrates with EMC AppSync to support application consistent snapshot with SQL Server. AppSync coordinate between SQL Server (via SQL Server Virtual Device Interface) and XtremIO to take a snapshot. XtremIO v4.0 also ships with a native VSS Provider that allows developers or third-party vendors to write their own “AppSync” like utilities to take application consistent snapshots.
Before taking an application consistent snapshot, SQL Server is notified of a pending back up to enable it to prepare as follows:

- Commit or roll back any in-flight transactions.
- Run checkpoint to flush dirty pages to disk, noting the log sequence number. This helps synchronize the data and the log files to minimize the work during restore.
- Freeze I/O operations.
- Back up the metadata.

Application consistent snapshot is considered as a full backup for SQL Server. With AppSync, an application consistent snapshot can be mounted and restored into various states, i.e. Recovery, No Recovery, or Standby as shown in Figure 28. With No Recovery restore, the database is left in a non-operational state and does not roll back the uncommitted transactions, allowing additional transaction logs to be restored.

![Figure 28. Database Recovery Types](image)

**When to Use Application Consistent vs. Crash Consistent Snapshot**

To determine when to use each snapshot type, the following should be considered:

- Work needed during recovery - With application consistent snapshot, the data files and log files are in sync. There is no additional work required to redo or undo any transactions. While both application consistent snapshot and crash consistent snapshot have the same RPO (Recovery Point Objective), application consistent snapshot can offer better RTO (Recovery Time Objective).
- Impact of I/O freeze during snapshot - SQL Server allows up to 10 seconds for a snapshot to be taken. Application may experience a performance drop during the time the I/O is frozen with Application Consistent Snapshot.
- Metadata vs. full backup - Whether SQL Server has metadata backup of the snapshot operation or treats the snapshot as a full backup. An application consistent snapshot can be restored into No Recovery mode to continue roll forward log backups.

To determine which type of snapshot to use, you should consider the availability requirement, frequency of the snapshot, performance impact, and need to support log backups. The followings are recommended scenarios for using application consistent vs. crash consistent snapshots.
Application Consistent Snapshot
- Database backup
- Creating always-on availability groups, secondary replica, database mirror server, or secondary log ship

Crash Consistent Snapshot
- Repurposing for test, development, or reporting
- Offload processing, for example, tuning, dbcc

**Best Practices Tips:**
- Use Application Consistent Snapshot for backup, and HA that require rolling forward additional transaction log data.
- Use Crash Consistent Snapshot for repurposing, offload processing, or backup that does not require rolling forward additional logs.
Appendix I: Repurposing Copy for Test/Dev

This section describes the sample workflow for repurposing deployment test / development environment using PowerShell scripts. In this workflow, you will see how we use XtremIO multi-level crash consistent copies to create first generation copy, apply simple data masking, and deploy test / development environment of the sanitized first-generation copy.

The repurposing for test / development workflow can be broken down into 5 steps.
- Create first generation crash consistent copy from production source database prod.
- Mount and recover first-generation copy as stageDB.
- Execute scripts "c:\scripts\dataMask.sql" to mask customer SSN data on stageDB (see script details below).
- Create second-generation copy from the first-generation copy.
- Mount and recover second-generation copy as testDB.

```powershell
# Adding exception to accept a self - signed certificate or accepting an X509Certificate that previously did not exist
add-type @"
    using System.Net;
    public class TrustAllCertsPolicy : ICertificatePolicy {
        public bool CheckValidationResult(
            ServicePoint srvPoint, X509Certificate certificate,
            WebRequest request, int certificateProblem) {
            return true;
        }
    }
"

###
# A generic wrapper function that queries XtremIO REST API using HTTP/GET
# Retrieves and lists existing configuration of an object or multiple objects
###
function ExecuteGetRestQuery ($xmsip,$cfgOption,$headers)
{
    try
    {
        $baseUrl = "https://"+$xmsip
        $resUrl = '/api/json/v2/types/'
        $url = $baseUrl + $resUrl + $cfgOption
        Write-Host "ExecuteGetRestQuery()::"+$url
        $jsonserial.MaxJsonLength = [int]::MaxValue
        $result = $jsonserial.DeserializeObject((Invoke-WebRequest -Method GET -Uri $url -Headers $headers))
        return $result
    }
    catch{
        return $false
    }
}
```
# A generic wrapper function that queries XtremIO REST API using HTTP/POST
# Creates a new object with specified properties
###
function ExecutePostRestQuery ($xmsip,$cfgOption,$data,$headers)
{
    $baseUrl = "https://"+$xmsip
    $resUrl = '/api/json/v2/types/
$url = $baseUrl + $resUrl + $cfgOption
Write-Host $url
Write-Host $data
$jsonserialInput.MaxJsonLength = [int]::MaxValue
$jsonserialOutput.MaxJsonLength = [int]::MaxValue
$result = (Invoke-RestMethod -Method POST -Uri $url -Body $data -Headers $headers)
return $result
}
###
# A function to create XtremIO snapshot on a single volume or an existing snapshot
# It accepts name of the object (volume or a snapshot) name and snapshot suffix as arguments and invokes ExecutePostRestQuery
###
function createXtremSnapshot ($xmsip,$parentVolumeName,$snapSuffix,$ssName,$headers)
{
    Write-Host "+++++++++Creating snapshot from "$parentVolumeName
$cfgOption = 'snapshots'
$parentVolumes = '[$parentVolumeName]'
$parentVolumes = ($parentVolumes|ConvertTo-Json).ToString()
$snapSuffix = ($snapSuffix|ConvertTo-Json).ToString()
$ssName = ($ssName|ConvertTo-Json).ToString()
$data = @"{
    "volume-list":$parentVolumes,
    "snap-suffix":$snapSuffix,
    "snapshot-set-name":$ssName
}"
return ExecutePostRestQuery $xmsip $cfgOption $data $headers
}
###
# A function to create XtremIO snapshot from a consistency group
# It accepts name of the consistency group, target snapshot suffix, and resulting snapshot set name as arguments and invokes ExecutePostRestQuery
###
function createXtremSnapshotByCG ($xmsip,$cgName,$snapSuffix,$ssName,$headers)
{
    Write-Host "+++++++++Creating snapshot from "$parentVolumeName
$cfgOption = 'snapshots'
$cgName = ($cgName|ConvertTo-Json).ToString()
$snapSuffix = ($snapSuffix|ConvertTo-Json).ToString()
$ssName = ($ssName | ConvertTo-Json).ToString()
$data = @"{
  "consistency-group-id":$cgName,
  "snap-suffix":$snapsuffix,
  "snapshot-set-name":$ssName
}
"@
return ExecutePostRestQuery $xmsip $cfgOption $data $headers
}###

# A function to create XtremIO snapshot from a snapshot set
# It accepts name of the snapshot set, target snapshot suffix, and resulting snapshot set name as arguments and invokes ExecutePostRestQuery
###
function createXtremSnapshotBySS ($xmsip,$srcSSName,$snapSuffix,$targetSSName,$headers)
{
  Write-Host "+++++++++++Creating snapshot from "$parentVolumeName
  $cfgOption = 'snapshots'
  $srcSSName = ($srcSSName | ConvertTo-Json).ToString()
  $snapSuffix = ($snapSuffix | ConvertTo-Json).ToString()
  $targetSSName = ($targetSSName | ConvertTo-Json).ToString()
  $data = @"{
    "snapshot-set-id":$srcSSName,
    "snap-suffix":$snapsuffix,
    "snapshot-set-name":$targetSSName
  }
  "@
  return ExecutePostRestQuery $xmsip $cfgOption $data $headers
}###

# A function to check if an XtremIO volume copy exists
# It accepts the name of the copy volume and invokes ExecuteGetRestQuery to retrieve properties of the copy volume object
###
function checkSnapshot ($xmsip,$snapshotName,$headers)
{
  $cfgOption = 'snapshots?name='+$snapshotName
  $snapshot = ExecuteGetRestQuery $xmsip $cfgOption $headers
  if ( $snapshot -ne $null)
  {
    return $snapshot['content']
  }
  else
  {
    return $false
  }
}
### A function returns the device NAA name of a given volume

```powershell
function getVolDeviceID ($xmsip, $volName, $headers)
{
    $cfgOption = 'volumes?name=' + $volName
    $vol = ExecuteGetRestQuery $xmsip $cfgOption $headers
    if ($vol -ne $null)
    {
        $volContent = $vol['content']
        return $volContent['naa-name']
    } else
    {
        return $false
    }
}
```

### This function is used to create lun mapping between a set of initiators and volume

# It accepts a list of initiators, the volume name and invokes ExecutePostRestQuery to create a new lun map between every initiator in the list and the volume

```powershell
function createLunMap($xmsip, $initiatorList, $volName, $headers)
{
    $cfgOption = 'lun-maps'
    $volName = ($volName | ConvertTo-Json).ToString()
    foreach ($initiator in $initiatorList)
    {
        $initiator = ($initiator | ConvertTo-Json).ToString()
        $data = @'
        {
        "vol-id":$volName,
        "ig-id":$initiator
        }
        '"
        ExecutePostRestQuery $xmsip $cfgOption $data $headers
    }
}
```

### Mount the disk

```powershell
function mountDisk ($deviceID, $driveLetter, $volLabel) {
    $newDisk = Get-Disk | Where UniqueId -eq $deviceID
    Set-Disk -Number ($newDisk.Number) -IsReadOnly $false
    Set-Disk -Number ($newDisk.Number) -IsOffline $false
    Start-Sleep -Seconds 1
    ## Assign drive letter
    $newPar = Get-Partition -DiskNumber ($newDisk.Number)
    if (-not ($newPar.DriveLetter -eq $driveLetter)) {
        Set-Partition -DriveLetter $newPar.DriveLetter -NewDriveLetter $driveLetter
        Set-Volume -DriveLetter $driveLetter -NewFileSystemLabel $volLabel
    }
}
```
# This function does a SCSI rescan to discover the new snapshot volume, mounts the volume to
# the file system, and restore the SQL Server database from mdf & ldf file via the attach method.

```powershell
function mountAndRecover ($dataDeviceID, $logDeviceID, $dbName, $dataDrive, $volLabelData, $logDrive,
$logLabelLog, $dataPath, $logPath) {
    # Rescan and bring new disk online
    Update-HostStorageCache
    Start-Sleep -Seconds 1
    # Mount disks
    mountDisk $dataDeviceID $dataDrive $volLabelData
    mountDisk $logDeviceID $logDrive $logLabelLog
    # Attach database
    $sqlStmt = "create database "+$dbName+" on (filename = N'"+$dataPath+"'), (filename = N'"+$logPath+"') for attach"
    if ((Test-Path -Path $dataPath) -and (Test-path -Path $logpath)){
        # Attach Database
        Invoke-SqlCmd -Query $sqlStmt
    }
}
```

# Importing SQL powershell module. This module is essential to perform management operations on SQL server
Import-Module sqlps -DisableNameChecking

### Creating authentication header object for XMS. This header object is passed to every function call defined above

```powershell
$xmsip = "xmsip"
$xmsuser = "username"
$xmspwd = ConvertTo-SecureString "password" -AsPlainText -Force
$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($xmspwd)
$secPw = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
$basicAuth = ("{0}:{1}" -f $xmsuser,$secPw)
$EncodeAuth = [System.Text.Encoding]::UTF8.GetBytes($basicAuth)
$EncodeBase64Auth = [System.Convert]::ToBase64String($EncodeAuth)
$headers = @{Authorization=('Basic {0}' -f $EncodeBase64Auth)}
```

### Creation of header object is complete

## Step 1: Create generation 1 snapshot for stage env
```
createXtremSnapshotByCG $xmsip "cg-sql-prod" "g1" "ss-sql-prod.g1" $headers
```

## Step 2: Mount snapshot volume to stage host, create stage DB
```
createLunMap $xmsip "lgsc040" "sql-prod.g1" $headers
createLunMap $xmsip "lgsc040" "sql-prod-log.g1" $headers
$dataDeviceID = getVolDeviceID $xmsip "sql-prod.g1" $headers
$logDeviceID = getVolDeviceID $xmsip "sql-prod-log.g1" $headers
mountAndRecover $dataDeviceID $logDeviceID "stageDB" "M" "sql-stage" "N" "sql-stage-log" "M:\prod.mdf"
"N:\prod_log.ldf"
```

## Step 3: Run data scubbing scripts to perform data masking
```
Invoke-SqlCmd -InputFile "C:\scripts\dataMasking.sql"
```

## Step 4: Create generation 2 snapshot from stage
```
createXtremSnapshotBySS $xmsip "ss-sql-prod.g1" "g2" "ss-sql-prod.g1.g2" $headers
```

## Step 5: Mount snapshot volume to test host, create test DB
```
createLunMap $xmsip "lgsc040" "sql-prod.g1.g2" $headers
```

---

44 | Best Practices for Running SQL Server on Dell EMC XtremIO X2
© 2018 Dell Inc. or its subsidiaries.
Details of dataMasking.sql:

```sql
USE stageDB
GO
UPDATE Customer set SSN = ' XXX-XX-XXXX'
GO
```
Appendix II: Test Configuration

The section provides the hardware and software specifications of the products used in performance study.

Hardware Resource

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Quantity</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Array</td>
<td>1</td>
<td>Dual XtremIO X2 X-Brick w/ 36 x 400 GB SSDs each</td>
</tr>
<tr>
<td>Virtual Hosts</td>
<td>4</td>
<td>Intel(R) Xeon(R) CPU E5-2690 v2 2 x 10 cores @3GHz with hyperthreading enabled 385 GB RAM 1 x 1 GbE NICs (management network) 1 x 10 GBE NICs (data network) 2 x 16 GB FC HBAs</td>
</tr>
<tr>
<td>Physical Hosts</td>
<td>2</td>
<td>Intel(R) Xeon(R) CPU E5-2658 v2 2 x 10 cores @2.4GHz with hyperthreading enabled 512 GB RAM 1 x 1 GbE NICs (management network) 1 x 10 GBE NICs (data network) 2 x 4-port 16GB FC HBAs</td>
</tr>
<tr>
<td>Virtual Machines (VMs)</td>
<td>4</td>
<td>SQL Server virtual machines with 2 VMs per host.</td>
</tr>
<tr>
<td>LAN Switches</td>
<td>1</td>
<td>10GBE Ethernet Switch for application traffic</td>
</tr>
<tr>
<td>LAN Switches</td>
<td>1</td>
<td>1GBE Ethernet Switch for management traffic</td>
</tr>
<tr>
<td>SAN Switches</td>
<td>1</td>
<td>16GB Fiber Channel Switch</td>
</tr>
</tbody>
</table>

Software Resources

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>XtremIO</td>
<td>V6.0.0 build 55_X2</td>
<td>Storage OS</td>
</tr>
<tr>
<td>VMware vSphere</td>
<td>V6.5</td>
<td>Hypervisor hosting all SQL Server virtual machines</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Windows 2012 R2</td>
<td>Operating system hosting all SQL Servers</td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server 2014 Enterprise Edition</td>
<td>Database</td>
</tr>
<tr>
<td>SQL Server</td>
<td>SQL Server 2016 Enterprise Edition</td>
<td>Database</td>
</tr>
<tr>
<td>Microsoft BenchCraft TPC-E Toolkit</td>
<td>1.12.0-1026</td>
<td>Workload generator and load driver for OLTP testing</td>
</tr>
<tr>
<td>Microsoft StepMaster TPC-H Toolkit</td>
<td>MSTPCH.2.17.0-1030</td>
<td>Workload generator and load driver for OLAP testing</td>
</tr>
</tbody>
</table>
### Configuration for OLTP Study

<table>
<thead>
<tr>
<th>Property</th>
<th>SQL Server 2014</th>
<th>SQL Server 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM Configuration</td>
<td>2 x 20 vCPU w/ 20 GB memory</td>
<td>2 x 20 vCPU w/ 20 GB memory</td>
</tr>
<tr>
<td>Database Size</td>
<td>2 x 1 TB</td>
<td>2 x 1 TB</td>
</tr>
<tr>
<td>File Layout (per database)</td>
<td>OS: 1 x 400 GB (shared datastore) Tempdb: 1 x 500 GB (shared datastore) DB &amp; Log: 1 x 2 TB (dedicated)</td>
<td>OS: 1 x 400 GB (shared datastore) Tempdb: 1 x 500 GB (shared datastore) DB &amp; Log: 1 x 2 TB (dedicated)</td>
</tr>
<tr>
<td>SQL Max Server Memory</td>
<td>12 GB</td>
<td>12 GB</td>
</tr>
<tr>
<td>Max DOP</td>
<td>Set to 1</td>
<td>Set to 1</td>
</tr>
</tbody>
</table>

### Configuration for Bandwidth Intensive Workload Study

<table>
<thead>
<tr>
<th>Property</th>
<th>SQL Server 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Configuration</td>
<td>1 x Physical host</td>
</tr>
<tr>
<td></td>
<td>• Intel(R) Xeon(R) CPU E5-2658 v2</td>
</tr>
<tr>
<td></td>
<td>• 2 x 10 cores @2.4GHz with hyperthreading enabled</td>
</tr>
<tr>
<td></td>
<td>• 512 GB RAM</td>
</tr>
<tr>
<td></td>
<td>• 1 x 1 GbE NICs (management network)</td>
</tr>
<tr>
<td></td>
<td>• 1 x 10 GBE NICs (data network)</td>
</tr>
<tr>
<td></td>
<td>• 2 x 4-port 16GB FC HBAs</td>
</tr>
<tr>
<td>Database Size</td>
<td>1 x 5 TB</td>
</tr>
<tr>
<td>Number of Data Files</td>
<td>16</td>
</tr>
<tr>
<td>Tempdb</td>
<td>4 x 200 GB</td>
</tr>
<tr>
<td>Data and Log Files</td>
<td>8 x 1 TB</td>
</tr>
<tr>
<td>SQL Max Server Memory</td>
<td>128 GB</td>
</tr>
<tr>
<td>Max DOP</td>
<td>Set to 15</td>
</tr>
</tbody>
</table>

### Configuration for Mixed Workload Study

<table>
<thead>
<tr>
<th>Property</th>
<th>SQL Server 2014</th>
<th>SQL Server 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Type</td>
<td>OLAP</td>
<td>OLTP</td>
</tr>
<tr>
<td>VM Configuration</td>
<td>1 x 20 vCPU w/ 200 GB memory</td>
<td>2 x 20 vCPU w/ 20 GB memory</td>
</tr>
<tr>
<td>Database Size</td>
<td>1 x 1TB OLAP database</td>
<td>1 x 1 TB OLTP database</td>
</tr>
<tr>
<td>File Layout (per database)</td>
<td>OS: 1 x 400 GB (shared datastore) Tempdb: 1 x 500 GB (shared datastore) DB &amp; Log: 1 x 2 TB (dedicated)</td>
<td>OS: 1 x 400 GB (shared datastore) Tempdb: 1 x 500 GB (shared datastore) DB &amp; Log: 1 x 2 TB (dedicated)</td>
</tr>
<tr>
<td>SQL Max Server Memory</td>
<td>128 GB</td>
<td>12 GB</td>
</tr>
<tr>
<td>Max DOP</td>
<td>Set to 16</td>
<td>Set to 1</td>
</tr>
</tbody>
</table>
Reference

XtremIO

https://www.emc.com/collateral/white-papers/h16444-introduction-xtremio-x2-storage-array-wp.pdf
https://support.emc.com/docu86215_XtremIO_6.0_Storage_Array_User_Guide.pdf?language=en_US

Microsoft

https://support.microsoft.com/en-us/kb/2154845
https://support.microsoft.com/en-us/kb/2515143
https://support.microsoft.com/en-us/kb/2510009
https://support.microsoft.com/en-us/kb/926930

VMware

http://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=1267
EMC XtremIO X2 all-flash array is engineered to transform SQL Server through a unique metadata-centric, content-aware architecture. It not only provides exceptional performance to any SQL Server deployment, but also redefines efficiency and simplicity for storage and database administrators managing the SQL Server application lifecycle. It enhances SQL Server workload and resolves SQL Server database administrator challenges in many ways.

- **Performance** – It dramatically boosts any SQL Server workload performance with zero tuning. With consistent low latency for any random or sequential, small or large block data access, SQL Server database administrators can safely consolidate multiple SQL Server workloads or mixed SQL Server workloads on XtremIO without worrying about the “noisy neighbor” issue. Performance and/or storage capacity can increase on-demand without interruption of services with XtremIO’s unique multi-dimensional scalability and non-disruptive expansion capability.

- **Efficiency** – Storage efficiency is built into XtremIO architecture, offered as free data services to any SQL Server workload. Database storage footprint is automatically reduced with always-on thin provisioning, inline data deduplication, and inline data compression.

- **Agility** – XtremIO iCDM redefines agility for SQL Server lifecycle management. iCDM instantly creates full size, full performance, yet space-efficient database copies that can be repurposed for test/dev or reporting, or used to simplify database maintenance, backup, or upgrade workflows. This results in dramatic improvements of database administrator productivity, and increased workflow agility.

- **Simplicity** – XtremIO eliminates complex capacity planning for SQL Server workloads. Database administrators no longer need to worry about RAID, spindle counts, or any of the underlying storage configurations when planning for SQL Server deployments. A flash optimized RAID is built-in to provide protection for hardware failure. Databases automatically get the performance of all SSDs and all controllers with the inherently balanced nature of the scale-out architecture. Provisioning for SQL Server workload is as easy as specifying the size for the volumes.
How to Learn More

For a detailed presentation explaining XtremIO X2 Storage Array's capabilities and how XtremIO X2 substantially improves performance, operational efficiency, ease-of-use and total cost of ownership, please contact XtremIO X2 at XtremIO@emc.com. We will schedule a private briefing in person or via a web meeting. XtremIO X2 provides benefits in many environments and mixed workload consolidations, including virtual server, cloud, virtual desktop, database, analytics and business applications.