Best Practices for Running SQL Server on EMC XtremIO

ABSTRACT
This White Paper explains the general characteristics of running SQL Server on EMC’s XtremIO enterprise all-flash storage array. It reveals studies in the areas of performance, storage efficiency, deployment, capacity planning, and SQL Server Application lifecycle management. This document also provides general guidance on deployment best practices and considerations.

October, 2015
To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store.

Copyright © 2015 EMC Corporation. All Rights Reserved.

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

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

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

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

VMware, VMware vSphere and VMKernel Scheduler are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other trademarks used herein are the property of their respective owners.

Part Number H14583
# TABLE OF CONTENTS

**EXECUTIVE SUMMARY** .............................................................................................. 5
  Audience.................................................................................................................. 5

**INTRODUCING XTREMIO** .......................................................................................... 6
  Architecture ............................................................................................................. 6
  Copy Data Services ................................................................................................... 7

**PERFORMANCE** ......................................................................................................... 8

**STORAGE EFFICIENCY** .............................................................................................. 9
  Thin Provisioning ....................................................................................................... 9
    SQL Server Pre-Allocated Space and Thin Provisioning .............................................. 9
  Inline Data Deduplication .......................................................................................... 11
    Single Database ........................................................................................................ 11
    Multiple Copies of Database ..................................................................................... 12
  Inline Data Compression ........................................................................................... 13
    SQL Server Data Compression .................................................................................. 13
    XtremIO Array Compression and SQL Server Native Data Compression ................. 13

**SQL SERVER AND XTREMIO COPY DATA SERVICES** ................................................ 15

**STORAGE PLANNING AND OPERATIONAL CONSIDERATIONS** ............................... 17
  Provisions for SQL Server ......................................................................................... 17
  RAID Configuration .................................................................................................... 17
  Separation of Files .................................................................................................... 17
  Number of LUNs ......................................................................................................... 18
  Number of Files ......................................................................................................... 19
  Servicing Downstream Data Needs ........................................................................... 19
    Test and Development ............................................................................................ 19
    Near Real-Time Reporting ...................................................................................... 20
    Refreshing Downstream Data with Snapshot Refresh ............................................. 23
  Database Maintenance .............................................................................................. 23
    Query Tuning ........................................................................................................... 24
    DBCC ...................................................................................................................... 24
  Backup/Restore ......................................................................................................... 24
EXECUTIVE SUMMARY

For many organizations, SQL Server is the core database engine infrastructure for business-critical applications. While IT departments across organizations are being asked to cut costs and increase efficiency, SQL Server applications are getting more complex, database sizes are exploding and service windows are shrinking. Businesses, however, are demanding for a better response. It is the job of IT administrators and SQL Server database administrators alike to seek solutions to keep up with Service Level Agreements (SLAs) and performance metrics, while increasing efficiency and workflow agility.

EMC’s XtremIO All Flash Storage Array is an ideal platform for business-critical applications like SQL Server. XtremIO’s unique scale-out architecture not only delivers ultra-high performance, seamless scalability with consistent low latency for any SQL Server deployment, but also brings new levels of storage efficiency, simplicity, and agility that significantly increase the productiveness of IT and database administrators.

This paper discusses:

- XtremIO features and added value for SQL Server
- Use case scenarios on using XtremIO features to enhance SQL Server application lifecycle management, increase productivity for both database administrators and Business Intelligence teams
- SQL Server on XtremIO deployment considerations, including best practices

A summary of best practices recommendations is included in Appendix I: Best Practices Summary.

AUDIENCE

This paper is intended for:

- SQL Server database administrators (DBAs)
- IT Administrators
- Storage/Data center architects
- Technical managers
- Any other IT personnel who are responsible for design, deployment, and management of SQL Server databases, infrastructure, and data centers
INTRODUCING XTREMIO

XtremIO is an enterprise class all-flash storage array with rich data services. It is designed from the ground-up to unlock flash’s full performance potential and deliver array-based capabilities that leverage the unique characteristics of Solid State Disks (SSDs).

ARCHITECTURE

The XtremIO All Flash Storage Array is built on a scale-out architecture. The system is designed to scale, using building blocks (X-Bricks). Each X-Brick is a High-Availability unit that consists of dual Active-Active Storage Controllers and a set of 25 SSDs. X-Bricks are clustered together to increase performance and capacity as required. Interconnections 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, in essence, a single shared memory space spanning all Storage Controllers. By linearly scaling all resources such as CPU, RAM, SSDs and host ports in a balanced manner, the array achieves any desired performance level, while maintaining consistency of performance that is critical to predictable application behavior.

With an architecture designed to support an unlimited number of X-Bricks, the current XtremIO version 4.0 release supports up to eight X-Bricks with varying SSD capacities.

Figure 1. XtremIO All Flash Storage Array Architecture

XtremIO 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 data blocks that are unique (never previously-written to the system) are further compressed and stored on SSDs. Data is always initially stored with its smallest footprint, without any 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.
- No data or metadata hotspots resulting in extended flash endurance.
- Easy setup and no tuning. No complex capacity planning is required to spread out LUNs across RAID group, or spindles.
- Advanced data services, including inline data deduplication, inline data compression, thin provisioning, advanced data protection (XDP), and copy data services with snapshots.
COPY DATA SERVICES

One of the most amazing pieces of technology XtremIO offers is the agile copy data services. XtremIO copy data services elevates writable snapshots beyond simple data protection. XtremIO copy data services instantly creates full size, full performance, yet entirely space-efficient volume copies. The entire application environment, including production and non-production instances, can be efficiently managed on a single array with amazing economics. This gives application and infrastructure teams breakthrough workflow and business process agility, by eliminating time-wasting, performance-sapping, and capacity-hungry brute force copying techniques. Application and infrastructure teams can work on more value-added projects and innovation.

Figure 2. XtremIO Copy Data Services
Performance

Typical application workloads like SQL Server drives moderate IOPS/bandwidth that rarely stretches 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 Servers running on SAN due to "noisy neighbor" issues for resource sharing, or mixing workloads with different disk access patterns.

XtremIO is an enterprise class all-flash array that is designed to solve many challenges of a traditional SAN. XtremIO not only offers high IOPS/bandwidth, but also consistently maintains sub-millisecond latency as the number of workloads increase, or as the system scales. The system’s unique scale-out architecture also allows for linear scaling of performance as well as capacity. XtremIO’s architecture allows performance and capacity to be increased by adding X-Bricks. Each X-Brick offers 150K IOPS with 70:30 mixed reads/writes of 8K or 250K of 100% read 8K IOPS. The architecture is built to support an unlimited number of X-Bricks. The current Ver. 4.0 release supports up to eight X-Bricks with over 1.2 million IOPS with mixed reads/writes.

XtremIO’s unique scale-out architecture offers excellent flexibility for SQL Server deployments. You can start out small, and as your business grows, add more performance and capacity by adding additional X-Bricks. When you scale-out by adding X-Bricks, your SQL Server data is automatically rebalanced across all the SSDs in the XtremIO cluster. With XtremIO’s online cluster expansion capability, workloads automatically benefit from the additional SSDs and Storage Controllers, without any service interruption.

Figure 3 shows the results of a study conducted on consolidation of up to eight SQL Server workloads on a dual X-Brick XtremIO cluster. Each of OLTP-like workload simulates a stock trading application, and generates I/O activities of a typical SQL Server online transaction workload of 90% read and 10% write. The majority of data access is 8K random. The study measures IOPS and latency as the number of SQL Server workload increases. As indicated in Figure 3, as the number of SQL Server instances increases from 1, 2, 4, and 8, the total aggregated IOPS increases from 22K, 45K, 95K, and 182K respectively. The scalability of IOPS is nearly linear as the number of SQL Server workloads increases. The average disk latency is observed as the SQL Server application is maintained at about 500µs. For further details on test configuration and methodologies, refer to Appendix II: Performance Testing Configurations and Methodologies.

Figure 3. XtremIO Performance with Consolidation of Up To 8 SQL Servers

The ability to maintain consistent sub-millisecond latency under high load, and to scale linearly in a shared environment, uniquely differentiates XtremIO from the traditional hybrids and other all-flash arrays. XtremIO Storage Array offers the predictability and consistency that are absolutely critical for running any enterprise SQL Server applications on a SAN.
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 include thin provisioning, inline data deduplication, and inline data compression.

THIN PROVISIONING

Legacy storage systems use thick instead of thin provision. Thick provision typically requires a projection of potential application storage usage for three years, but pay for cost of the storage at purchase time. In many cases, the storage is overprovisioned, and consequently a large amount of space ends up wasted for a long period of time. Thin provisioning was designed to increase storage utilization and maximize organization storage investment.

XtremIO storage is natively thin provisioned and supports 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 provisioned unused capacity.

SQL SERVER PRE-ALLOCATED SPACE AND THIN PROVISIONING

Typical SQL Server deployment pre-allocates 10% to 30% of space to anticipate data growth and to avoid data file auto-grow during run time. The portion of pre-allocated space is zero initialized but not yet written by SQL Server.

With XtremIO, the portion of pre-allocated space is considered as white spaces that will not be stored physically. In this study, a 1.6TB database was created, with a large amount of pre-allocated space (about 61%) in order to observe the physical storage footprint on XtremIO. As indicated in Figure 4 and Figure 5, the XtremIO Dashboard's Storage pane shows the amount of space provisioned to the volume as 2TB, and the actual consumption of space of about 610GB. The amount of pre-allocated space is thin provisioned.

Figure 4 shows a database with about 1.6TB of space reserved, with 610GB of actual space used.

![Figure 4. SQL Server Database Space Used](image)
Figure 5 shows the volume capacity that is considered as used by XtremIO. As indicated in the figure, the pre-allocated SQL Server space is considered as thin provisioned.

<table>
<thead>
<tr>
<th>Storage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Efficiency</td>
<td>5.3:1</td>
</tr>
<tr>
<td>Data Reduction Ratio</td>
<td>1.6:1</td>
</tr>
<tr>
<td>Deduplication</td>
<td>1.0:1</td>
</tr>
<tr>
<td>Compression</td>
<td>1.6:1</td>
</tr>
<tr>
<td>Thin Provisioning Saving</td>
<td>70%</td>
</tr>
<tr>
<td>Volume Capacity</td>
<td>2TB</td>
</tr>
<tr>
<td>Physical Capacity</td>
<td>30.489TB</td>
</tr>
</tbody>
</table>

**Figure 5. XtremIO Dashboard Storage Pane**

XtremIO’s native thin provisioning provides the following 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 use the full provisioned capacity. The ability to consume capacity on demand allows IT administrators to more efficiently maximize the physical capacity of the storage.

Thin provisioning is an integral part of XtremIO’s content-aware architecture. Unlike some disk-oriented architectures, XtremIO’s thin provisioning does not impact the performance or cause fragmentation issues.

Thin provisioning provides many benefits. However, it requires IT administrators to implement measures to monitor the percentage of the available physical capability. In order to avoid running into issues with SQL Server, 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

XtremIO stores data according to its content. When data stream enters the system, it is divided into 8KB data blocks, and fingerprinted based on its content. Each data block is automatically checked for duplication globally across the entire XtremIO cluster. Only unique data that does not already exist on the XtremIO All Flash Storage Array is written onto the SSDs.

XtremIO’s unique architecture of using content fingerprints not only enables inline data deduplication, but also provides flexibility for even 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 hotspots 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’s inline data deduplication and intelligent data storage process ensure:

- Balanced usage of the system's 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 8KB data unit. Figure 6 depicts the structure of an 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 databases due to the unique page header on each data page. However, there are many occasions where more than one copy of databases are required due to issues like reporting, test and development.

Figure 6. Structure of an SQL Server Data Page
MULTIPLE COPIES OF DATABASE

Figure 7 shows the effects of inline data deduplication when deploying multiple copies of the same database. The graph displays the deployment of 1, 2, 4, and 8 copies of a database. While the provisioned capacity (space reserved by SQL Server) and space used by SQL Server multiplies as the number of copies increase, XtremIO only consumes the physical space for a single database. When copies are initially created, no additional space is required. Space consumption only increases afterwards, when there are changes to the content of the database.

![Figure 7. Storage Efficiency with Database Copies](image-url)
INLINE 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 only performed on unique data blocks. XtremIO uses an algorithm that is based on Lempel-Ziv-Welch, 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 the storage footprint by storing data blocks in the most efficient manner. XtremIO can effectively compress unique data by as much as 4:1. The overall compressibility of the data is determined by the nature of the data.

SQL SERVER DATA COMPRESSION
SQL Server database administrators are familiar with the SQL Server native data compression features, namely row compression and page compression. Row compression stores fixed-length data types in variable-length format, and removes storage needs 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 8KB page. Redundant column data is stored only once on a page, and is referenced by multiple columns.

Implementing row compression and page compression effectively reduces the space used by SQL Server. However, there are a few challenges that hinder wide adoption of this process. They include:

- Compression and decompression consumes host CPU cycles. Especially for page compression, the CPU overhead could be substantial, and may even show a negative impact on query performance if the workload is running 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
- Due to the potential performance overhead, turning on row or page compression needs to be carefully evaluated on table, partition, or even index level.
- Turning on compression requires moving data to a new record format, and removing duplications. This process is time and resource intensive. The initial cost for turning on compression could take hours or days, depending on the size of the database. That could be a big burden when managing a large number of databases.
- Space saving from compression cannot be reclaimed until the database or data files are shrunk. This shrink operation is very intrusive, and 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.

This study examined an uncompressed database, a row-compressed database, and a page-compressed database, on both the XtremIO Storage Array and on a local disk without storage compression. The SQL Server space used data was captured from the SQL Server's Management Studio Disk Usage report to determine the actual database space that is used with the different compression methods. XtremIO's physical used capacity counter was used to measure the physical storage consumption on XtremIO.

As Figure 8 shows, the compression savings for row compression, page compression, and XtremIO compression are 16%, 48%, and 37%, respectively. Storing uncompressed database on XtremIO reduces the storage footprint from 610GB to 383GB. That is a 37% saving on space without requiring the use of lengthy index rebuilds. In addition, there is no impact to performance.
The study showed that page compression provided the highest space saving of 48%. In comparison, row compression had the least compression saving of 16%. Both page and row compressions are subject to the above caveats of potential performance overhead, operational complexity, and the space cannot be reclaimed until after the expensive shrink operation. While page compression is worth considering for less active or archival data, XtremIO's inline compression appears to provide a better alternative than row compression for any active data.

The advantages of using the XtremIO's inline compression include:

- Good compression saving compared to SQL native data compression options
- Automatic application to all SQL Server databases without the need to turn on any additional knobs
- Zero impact on the existing query performance and no host CPU overhead
- No additional licensing fees

For databases that are already compressed with row or page compression, XtremIO's inline compression can further maximize the compression savings. Figure 9 shows the physical capacity consumed by databases that are row or page compressed. XtremIO's inline compression is able to deliver an additional 38% of saving for a row compressed database, and an additional 7% of saving on a page-compressed database.

Compression savings may vary depending on data compressibility.

**Figure 9. Using SQL Server Native Data Compression with XtremIO Array Compression**

<table>
<thead>
<tr>
<th></th>
<th>SQL Native Compression</th>
<th>XtremIO + SQL Native Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>row compressed</td>
<td>510</td>
<td>316</td>
</tr>
<tr>
<td>page compressed</td>
<td>318</td>
<td>297</td>
</tr>
</tbody>
</table>

**Best Practices Tips:**

- Use XtremIO's inline compression instead of SQL Server native row compression to achieve automatic compression of active data. Use SQL Server native page compression to maximize compression savings on less active or archival data.
- Use SQL Server native row or page compression if the database is encrypted with SQL Server transparent data encryption. A transparent data encrypted database does not gain compression savings with XtremIO's inline compression. An alternative to resolve the compatibility issue is to consider XtremIO's Data at Rest Encryption.
SQL SERVER AND XTREMIO COPY DATA SERVICES

During an SQL Server application lifecycle, database administrators often have the need to create copies of databases (for example, for downstream data service and data protection). Traditional methods for creating copies of databases typically involve the following methods:

- **Backup/restore** — Backing up the database on the source, and restoring on the target.
- **Copy/attach** — By temporarily pausing the SQL Server service. For example, in a side-by-side upgrade scenario, it is possible to make a copy of the SQL files from the source, and attach the copy to a target SQL Server.

Both methods require reading the data from the storage, transferring data from source to target via the host network, and writing the data back to the storage. The process is not just time consuming. It also demands a large amount of resources from the host CPU, memory, and network bandwidth.

XtremIO offers a new alternative for deploying copies of the databases with copy services. XtremIO's copy data services are based on its redirect-on-unique-write technology, where you are able to instantly create a fully writable copy of the database. It is 100% space efficient and does not take any resources from the host side.

In the lab, a study was performed to compare deployment time for a 1TB database using:

- Traditional backup/restore
- Copy/attach
- XtremIO copy data services

Figure 10 shows the results of the study.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Deployment Time</th>
<th>Data Reduction for Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup / Restore</td>
<td>Backup 28 minutes</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Restore 33 minutes</td>
<td></td>
</tr>
<tr>
<td>Copy / Attach</td>
<td>Copy 29 minutes</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Attach Instantaneous</td>
<td></td>
</tr>
<tr>
<td>Copy Services</td>
<td>Snapshot Instantaneous</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Mount / Attach &lt; 5 seconds</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 10. Comparisons of Database Deployment Time**

The backup/restore or copy/attach method of creating database copies automatically acquires the benefit of XtremIO's inline data deduplication feature. Database copies do not take up any additional storage space. However, both methods still require transmitting data blocks from the storage to host and back to the storage. The deployment time is bound to the available host resources, network bandwidth, and the read/write performance of the storage. In the study conducted, deploying a copy of a 1TB database using the backup/restore method on an idle host, over a 10GB network sitting in XtremIO storage, took over 60 minutes. The copy/attach method uses the more efficient attach method for recovering the database, but still took 30 minutes to deploy.

With the new XtremIO copy data services alternative, creating a copy of the 1TB database was instantaneous, while mounting and recovering the database took less than 5 seconds.

XtremIO copy data services is based on its unique snapshot technology. Snapshots can be writable or read only. XtremIO's snapshot technology is implemented by leveraging the array's content-addressing capabilities along with in-memory metadata and the system's dual stage metadata (which results in Inline Data Reduction).
As depicted in Figure 11, when a snapshot is created, the metadata for the source volume becomes an "ancestor" entity that is shared between the production source volume and the snapshot. New empty containers are created for subsequent changes to the production source volume and the snapshot volume. Thus, the act of creating a snapshot is extremely efficient and only requires metadata operation in memory. The operation of creating a snapshot on a database is instantaneous whether it consists of a few megabytes or hundreds of terabytes.

![Figure 11. XtremIO Copy Data Services Based on Snapshot Technology](image)

As XtremIO volumes are thinly provisioned, the new empty containers that are created for the snapshot creation process do not take up any space. Thus, deploying copies of databases using the copy data services is 100% space-efficient. Snapshot capacity consumption occurs only if a change requires writing a new block.

**Best Practice Tip:**

- Use XtremIO copy data services for rapid deployment of database copies with storage efficiency.
STORAGE PLANNING AND OPERATIONAL CONSIDERATIONS

PROVISIONS FOR SQL SERVER
SQL Server maps a database over a set of files on disk. Data and transaction log information are stored in separate files. In order to maximize performance and operational efficiency of the SQL Server, optimizing database file layout has been the focal point of discussion for many years. In this section, we examine the common considerations and recommendations, and discuss the 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 recommendations on RAID, when considering aspects like performance, data protection, and cost considerations:

- Use RAID 10 for user data and log files for best performance and availability. However, when cost is a concern, data can be allocated on RAID 5 or equivalent.
- For write intensive TempDB, use RAID 10. RAID 0 is an option if cost is an issue. However, 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 considerations are no longer relevant, for the following reasons:

- The XtremIO system has a built-in "self-healing" double-parity RAID as part of its architecture.
- The XtremIO Data Protection (XDP) is designed to take advantages of flash-media-specific properties and XtremIO's 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

- 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 origin for separation of files is due to SQL Server I/O characteristics and optimization 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 patterns helps to minimize disk head movements, and thus optimizes storage performance.

This guideline is no longer applicable when deploying SQL Server on XtremIO Storage Array, because:

- XtremIO uses all solid state disks (SSDs). SSDs have no moving parts and it does not matter where the physical data block is located. There is no disk head movement for read/write access.
- With XtremIO's unique architecture, data is intelligently and evenly distributed across all SSDs by its content-addressable storage engine. This maximizes 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 perspective, 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 an SQL Server instance. It is a temporary work space that is recreated each time an SQL Server instance starts. When using the XtremIO snapshot technology to create a backup, deploy copies of a database, or replicate a 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 reasons are required for the separation of files.
- 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 across multiple databases, do not share LUN with multiple databases. A LUN/volume is the smallest unit for snapshot operation.

**NUMERO LUNS**

SQL Server allows database objects and files to be grouped in filegroups. Filegroups provide the means to separate user objects from one another into different LUNs or physical disks. Because a filegroup cannot be spanned with any one file, you can never have data from objects assigned to different filegroups residing in the same physical file.

SQL Server storage best practices recommend the following:

- Use filegroups for administration requirements, such as backup/restore and partial database availability.
- Use data files to "stripe" the database across your specific I/O configuration (for example, for physical disks and LUNs).
- Unless you are very familiar with the application, avoid over optimizing the I/O by selectively placing objects on separate spindles.

This guideline can be significantly simplified for the majority of SQL Server deployments:

- In general, a single LUN is capable of driving an XtremIO storage system to its maximum capability if the application is driving the amount of I/O activities.
- Multiple filegroups should continue to be used for enhancing database availability and managing table partitioning.
- There is no need to over optimize the I/Os by selectively placing objects on separate spindles. A single LUN per database provides the performance needs for the majority of SQL Server deployments. You can keep LUN configuration simple and standard by leveraging XtremIO’s capability of automatically balancing data and workload.

**Best Practices Tips:**

- A single LUN per database suffices for the majority of deployment scenarios. XtremIO automatically takes care of even distribution of data, and ensures maximum parallel processing.
- For large scale I/O intensive SQL Server deployment, consider the following tuning options:
  - Increase LUN queue depth. For details, see Queue Depth.
  - If I/Os exceed the maximum allowable queue depth, spread the database files across multiple LUNs. This increases 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. This takes full advantage of parallel processing powers of the multiple Storage Controllers in the cluster and maximizes performance.
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 are the same size (as recommended by SQL Server) the allocation is round-robin.

Having multiple data files adds 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 improvements of workloads with heavy insert activities.

Best Practice Tip:
- Follow the SQL Server recommendations on creating multiple data files for a user database and creating TempDB. For SQL Server recommendations, refer to https://support.microsoft.com/en-us/kb/2154845.

When deploying SQL Server on XtremIO, be aware that many of the existing SQL Server storage considerations or recommendations are centered around the traditional spinning hard disk media. In addition, the XtremIO Storage Array is built with the more advanced SSD media, which are optimized for both random and sequential access. The system’s intelligent content engine and unique scale-out architecture also significantly simplify the complexity of provisioning storage for an SQL Server. With XtremIO, database administrators should be freed from worrying about RAID, spindle counts, or any of the underlined storage configurations.

SERVICING DOWNSTREAM DATA NEEDS
Throughout the SQL Server application lifecycle, database administrators may need to deploy copies of the production database for downstream data consumption. Servicing test and development environments and near real-time reporting data needs are two major use cases for this scenario. Database administrators can deploy copies of a database on XtremIO using traditional methodologies and enjoy the storage efficiency that XtremIO offers. A more efficient way to service downstream data needs is achieved through deployment with XtremIO copy data services.

TEST AND DEVELOPMENT
While application features continue to be developed, tested, and rolled out to production, copies of the existing production database are needed to support the development efforts, and to perform user acceptance tests. As the production database size continues to increase, database administrators are facing some serious challenges, some of which are listed below:

- How to deploy and refresh test and development environments efficiently? Traditional backup/restore methods 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 that the test and development copies are running on storage with performance similar to the production?
XtremIO copy data services 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 or a sanitization process) to be prepared as a golden image for development or testing purposes. Multiple copies can then be created from each master copy and presented to various development teams. Provisioning copies is an easy and instantaneous process. Checkpoints to monitor the stages of development can be created by using multi-level snapshots.

Figure 12. Deploy Test and Development Environments with XtremIO Copy Data Services

XtremIO copy data services enables database copies to be created based on the demand for maximum business efficiency, rather than based on storage capacity or performance limitations. Copies of a database can be created instantly, without consuming additional storage. With abundant IOPS and consistent low latency XtremIO offers, test and development can enjoy the same performance as production.

**Best Practices Tips:**

- Use copy services to deploy copies of the database for test and development purposes.
- Use the multi-level snapshot capability to cleanse sensitive data before presenting the database for test or development use.
- Use the multi-level snapshot capability to create checkpoints or branches for development.

**NEAR REAL-TIME REPORTING**

One of the primary purposes of a database is to service reporting. Reporting queries are typically long running and require a large amount of CPU and memory resource (as it is heavy on disk I/Os). Mixing reporting with online user activities on the same compute unit may cause resource contention, locking, and deadlocking issues. The ability to offload reporting and to separate reporting activities from primary online user workloads helps to reduce load, and improve performance on the production environment. XtremIO copy data services enable you to easily build a solution that scales out a database to provide near real-time copies of the production data for reporting purposes. The solution also supports periodic refresh of the production data through rolling updates of reporting databases.


The solution for using XtremIO copy data services for scale-out reporting enhances SQL SSD with rapid snapshot cloning and refreshing capabilities that are native to XtremIO. The following table shows detailed steps for the proposed solution.
<table>
<thead>
<tr>
<th>TIME</th>
<th>PHASE</th>
<th>FIGURE</th>
</tr>
</thead>
</table>
| T0   | This phase defines the build process for creating the initial reporting databases. | ![Network Load Balancer](image)

1. Create master snapshot of database volume(s) at T0.
2. Create second level snapshots against the master snapshot.
3. Mount second level snapshots to all reporting servers.

*Figure 13. Near Real-Time Reporting Initial Phase*
<table>
<thead>
<tr>
<th>TIME</th>
<th>PHASE</th>
<th>FIGURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>This phase defines the process for periodic refresh of the reporting databases. In this phase, half of the reporting databases were refreshed at a time and the rolling update approach was used to maximize availability of the reporting service.</td>
<td></td>
</tr>
</tbody>
</table>

1. Refresh master snapshot with current product data at T1.

2. Take one half of the reporting databases offline. Reports can be serviced without interruption with the remaining half.

3.Unmount disks owned by the offline reporting databases.

4. Use the snapshot refresh capability to refresh the second level snapshots, using the master snapshot.

5. Mount disks for all offline reporting databases.

6. Bring the databases online.

7. Repeat Steps 2 through 6 to update the second half of the reporting database.

Figure 14. Near Real-Time Reporting Refresh: Phase I

Figure 15. Near Real-Time Reporting Refresh: Phase II

Figure 16. Near Real-Time Reporting Refresh: Phase III
REFRESHING DOWNSTREAM DATA WITH SNAPSHOT REFRESH

Whether you are using XtremIO copy data services to provide data services for test, development or reporting, the snapshot will eventually become outdated and lose its usefulness. There should be a process in place to periodically update the reporting data, or provide new versions of the database for test or development usage. The XtremIO snapshot refresh feature simplifies the process for updating a stale snapshot.

The process for refreshing a downstream database requires completing the following five steps:

1. Detach the stale database
2. Dismount the database volume(s)
3. Run snapshot refresh on XtremIO
4. Mount the volumes
5. Attach the refreshed database

For detailed scripts of each step, refer to Appendix III: Snapshot Refresh.

Best Practices Tips:

- Use copy data services to build readable secondary replicas for near real-time reporting, without implementing alwayson availability groups.
- Use multi-level snapshots to synchronize data on all reporting instances.

DATABASE MAINTENANCE

Throughout the SQL Server application lifecycle, databases need to be maintained to ensure that there are no data integrity issues. While database maintenance tasks help to keep SQL Server running in its best condition, running the tasks can use up a large amount of host resources. This affects the performance of a production SQL Server. XtremIO copy data services enable rapid deployment of database copies with storage efficiency. Resource-heavy database maintenance tasks can be offloaded to a secondary compute unit to minimize impacts to the production workload.

Figure 17. Offloading Database Maintenance Tasks
QUERY TUNING
As applications continue to develop with additional new features, the existing indexes on the database may not be sufficient for supporting user activities any more. Index tuning is part of the daily life for any SQL Server database administrator. Effective query tuning requires running on a production-like database, and may take many trials and errors. Trying to tune on a production environment is not practical. However, it would be challenging trying to get a copy of the terabyte size production database for query tuning, both from a storage capacity requirement perspective, and from the deployment time perspective. This is a dilemma many SQL Server database administrators face with traditional storage.

XtremIO copy data services can create a fully writable snapshot of a database instantly, without taking any additional physical storage space. When attaching the snapshot to a secondary SQL Server instance, you can have an environment dedicated for tuning purposes without impacts to the production workload.

DBCC
DBCC CHECKDB is a maintenance task that checks for the logical and physical integrity of all objects in the database. It is an important task to perform, as it ensures the health of the SQL Server database. However, DBCC CHECKDB can be very disruptive to a production SQL Server. SQL Server’s buffer pool may be completely trashed as CHECKDB reads all of 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. However, others may not even have the luxury of running DBCC CHECKDB, and thus incur more risks of data corruption.

With XtremIO copy data services, database administrators can easily attach a snapshot copy of the production database to a secondary SQL Server instance to run DBCC CHECKDB. In addition, there is no concern about trashing the production SQL Server buffer pool. With XtremIO snapshot refresh feature, database integrity can be checked periodically on an updated copy of the snapshot to ensure the ongoing health of the database.

Best Practice Tip:
• Offload resource-intensive database maintenance tasks, such as query tuning and DBCC CHECKDB to a secondary compute unit, using copies of databases, created with copy data services.

BACKUP/RESTORE
Backup is essential for providing data protection to an SQL Server database. Backing up the SQL Server database, running on a traditional storage, can be time consuming, resource-intensive, and requires a large amount of storage. A common backup storage strategy involves first backing up the local SAN, then copying the backup to a network file share, and then copying the backups to an archive system on-site plus creating copies to ship off-site.

XtremIO copy data services can significantly enhance SQL Server database backup/restore to the local SAN in many ways:
• Rapid backup/restore with snapshot technology
• Offloading resource-intensive backup processes to secondary compute unit
• Storing multiple backup images efficiently with inline data deduplication technology

Restoring from a snapshot backup is effortless. Database administrators can easily perform full recovery of the database, or mount the snapshot with a different database name to partially restore table(s), or row(s) of a database.
SNAPSHOT BACKUP/RESTORE WITH EMC APPSYNC

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.

In this study, a comparison was carried out to compare 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 without any needs to copy or move any physical data, the larger the database, the bigger difference in backup time.

AppSync supports 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 NORECOVERY state to allow additional transaction logs to be restored. This option supports point-in-time recovery with transaction logs.

A copy-only backup 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 a full database backup using AppSync application-consistent snapshot capability is an effective way of obtaining a fast recoverable copy of the database on local storage. To further protect databases in the events of a catastrophic disaster, or physical storage failure that is beyond recovery, the snapshot volume(s) should be copied to the network file share, onsite archive system, and offsite storage for complete protection.

Best Practices Tips:

- Offload backups to a secondary compute unit, using copy data services.
- Create rapid and storage efficient full database backups, using AppSync application-consistent snapshot. Copy snapshot volume(s) to secondary storages for disaster recovery, and protect data from unrecoverable physical storage failure.
- Use AppSync flexible recovery type to restore a database into NORECOVERY state for point-in-time recovery, using transaction log backups.

UPGRADE

As applications and SQL Server continue to mature, more features are released. Environments need to be upgraded to take advantages of the new features. XtremIO copy data services can enhance the upgrade experiences by simplifying the upgrade process, minimizing downtime, and reducing risks of upgrade.

With the capability to create an instant writable copy of the database without using any additional storage space, use XtremIO copy data services to:

- Test the upgrade on a copy of the database in order to resolve any potential issues.
- Simulate the upgrade process using a copy of the database in order to get a real sense of how long the upgrade will take.
- Protect the database by getting a copy of it prior to the upgrade. If the upgrade fails for any reason, simply point the SQL Server instance at the copy and recover.

Best Practices Tips:

- Run an upgrade test on a snapshot copy to identify potential upgrade issues and measure upgrade time.
- Create a snapshot backup before the upgrade, in case there is a need to failback.
DEPLOYMENT CONSIDERATIONS

MULTIPATHING
Multipathing facilitates the routing of I/Os 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 like SQL Server can continue to service I/Os without any interruption. In addition to protecting applications from hardware path failure, multipathing also enhances application performance by load balancing I/Os across all available paths to optimize resource, maximize throughput, and reduce I/O latency.

XtremIO supports the following 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 presented from XtremIO. Using 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:
  - Set the native round-robin path selection policy on XtremIO volumes presented to the ESX host.
  - Set the vSphere NMP round-robin path switching frequency to XtremIO volumes from the default value (1000 I/O packets) to 1.
- XtremIO supports the EMC PowerPath. For details on installing and configuring EMC 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 distribution and availability of load between I/O paths to the XtremIO storage.

For detailed steps on enabling and configuring multipathing for different host platforms, refer to XtremIO Host Configuration Guide at https://support.emc.com/docu56210_XtremIO-2.2.x----4.0-Host-Configuration-Guide.pdf?language=en_US.

**Best Practices Tips:**

- For Windows 2008 and above, use the Least Queue Depth policy with Microsoft Native Microsoft Multipath I/O (MPIO).
- For SQL Server on VMware vSphere, use the VMware native round-robin path selection policy, and set switching frequency to one.
- For best practices with PowerPath, refer to the 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 that 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. As the SQL Server I/O latency increases, throughput can suffer. If the queue depth value is too high, and the storage system cannot keep up with the I/O processing, the storage system can be overwhelmed, thus impacting on I/O performance for all applications running on the storage system.

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

- 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. A typical host bus adapter (HBA) vendor pre-configured queue depth value is set to 32. The value may vary by vendor or in some virtualization implementations. Each initiator port is capable of supporting a 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 microseconds. The system's unique scale-out architecture also means that it could be capable of processing millions of concurrent I/O requests. In addition, given XtremIO's design simplicity, the number of LUNs can potentially be reduced down to a single LUN. The default queue depth may become the limiting factor to hinder the SQL Server performance.

For optimal operation with XtremIO storage, consider the followings:

- Set the queue depth to 256. Queue depth setting is a global setting. If you have a mixed environment with a host connected to multiple storage platforms, check for recommendations on all storage platforms to avoid performance issues.
- For deployment on VMware vSphere environment, queue depth is further controlled by limits set on vSCSI adapter, and VMKernel scheduler.
- Queue depth configuration should be set on all layers, including physical HBA, vSCSI adapter, and VMKernel scheduler to ensure proper configuration.
- 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](http://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=1267)
  We recommend setting Disk.SchedNumReqOutstanding to the maximum supported value of 256 for XtremIO volumes.
- For detailed steps on setting queue depth with different HBA vendor implementations, refer to XtremIO Host Configuration Guide at: [https://support.emc.com/docu56210_XtremIO-2.2.x---4.0-Host-Configuration-Guide.pdf?language=en_US](https://support.emc.com/docu56210_XtremIO-2.2.x---4.0-Host-Configuration-Guide.pdf?language=en_US)

**Best Practices Tips:**

- Set the LUN queue depth to 256.
- 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. 4KB PHYSICAL SECTOR SIZE

XtremIO can present either 512B or 4KB 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 have been written based on 512B disk sectors size. Microsoft started native support for 4KB sector size with release of Windows 2012. For more details on support, refer to:

- 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 the 4KB sector size enables an SQL Server data page to occupy two physical sectors instead of 16 physical sectors with the 512B sector size. 4KB sector size aligns efficiently with SQL Server data page 8KB boundary and reduces the amount of metadata overhead. The effect of the metadata improvement is very visible with an ODX copy that is metadata operation-heavy. With an ODX copy of a set of SQL Server files with the total size of 1TB, the average bandwidth for copying from 512B to 512B volume is 6GB/s. The average bandwidth for copying from 4KB to 4KB volume is 12GB/s.

While 4KB sector size offers better storage capacity with less metadata overhead, the following are a few caveats that you should be aware of before presenting 4KB 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 a multiple of the physical sector size, with up to 60KB. With a 4KB sector size, applications that perform a large number of small write operations with frequent commits may see increased log space usage.

- 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. For more details, refer to: 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. Therefore, it is not possible to set legacy SQL Server versions on a 4KB sector volume, as it will fail.

- When placing TempDB and user database on different volumes, be aware that database and TempDB creations use the sector size that is 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 files or TempDB files are located in volumes with a different physical sector size. Variance of sector size in I/O paths should be avoided to prevent any suboptimal data access.

- If you run SQL Server in a virtual environment, be aware of that 4KB physical sector size is not yet supported by all Hypervisor vendors. For example, VMware vSphere does not currently support 4KB sector size. For more details on VMware vSphere and Windows Hyper-V support, refer to:
  - https://support.microsoft.com/en-us/kb/2515143

Before deploying an SQL Server database, always check on the physical sector size of the volumes provisioned for SQL Server. This is performed by running FSUTIL. Figure 18 shows sample outputs of executing FSUTIL against a 512B and a 4KB sector volumes.
Figure 18. Check Volume Sector Size Using FSUTIL
SQL Server database administrators can also run a `dbcc fileheader` to check on the sector size of the database (which is determined during database creation), and the physical sector size of the volume on which the database currently resides. Figure 19 shows sample outputs of a `dbcc fileheader` for a database created on 4KB sector size, and a database created on 512B sector size. The `SectorSize` column shows the sector size the database was created with. The `ActualSectorSize` column shows the sector size of the volume the database is currently on. For best performance, the `SectorSize` value should be equal to that of the `ActualSectorSize`.

**Figure 19. Sample dbcc Fileheader**

**DETERMINING WHICH SECTOR SIZE TO USE**

Using 4KB sector size is becoming the more popular technology of choice within the software and storage industry, because it enhances SQL Server data page storage and reduces metadata overhead. For any new database development, 4KB sector size would be a good choice as the new standard. However, despite the advantages that 4KB sector size offers, the 512B sector size remains to be the widely supported sector size that is less likely to cause any application or tools support issues. Be sure to check support on software platform and storage integrated tools when deploying on 4KB sector size.

**Best Practices Tips:**

- Use 4KB 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.
- Use 512B sector size for SQL Server on VMware deployments.
- When moving an 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 like SQL Server. The allocation unit size is the smallest unit of storage that any individual file can take up.

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

- SQL Server files are typically much bigger than 64KB.
- SQL Server allocation is defined by extent. Each extent in SQL Server has eight 8KB pages, which is 64KB.

![Figure 20. Configure Allocation Unit Size]

**Best Practice Tip:**
- Use 64KB Allocation Unit Size for SQL Server data and log volumes.
APPLICATION-CONSISTENT VS. CRASH-CONSISTENT SNAPSHOT

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 the snapshot technology. It supports both crash-consistent snapshots and application-consistent snapshot. So what is the difference between crash-consistent snapshot and application-consistent snapshot? When do you need which?

CRASH-CONSISTENT SNAPSHOT

Crash-consistent snapshots capture 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 running the SQL Server instance. A crash-consistent snapshot of an SQL Server database is equivalent to or consistent with the state of a running database during a power failure. In this state, there may be data pages in memory not yet flushed to the disk (open transactions). SQL Server uses write ahead logging (WAL), where all committed transactions are logged in SQL Server's transaction log on disk. This SQL Server can recover from a crash using information in the transaction log. This is where data pages which are not yet flushed to disk, can be recreated, and in-flight transactions are 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 coordinates between SQL Server (via SQL Server Virtual Device Interface) and XtremIO to take a snapshot. XtremIO Ver. 4.0 also ships with a native VSS Provider that enables developers or third party vendors to write their own "AppSync-like" utilities to take application-consistent snapshots.

Before taking an application-consistent snapshot, the SQL Server is notified that it is about to become backed up, so it can be prepared. The SQL Server can then:

- Either commit or roll back any in-flight transactions.
- Run a checkpoint to flush dirty pages to disk and make a note of the log sequence number. This is done to ensure that the data files and the log files are synchronized and thus minimize the amount of work during a restore process.
- Freeze I/Os.
- Take a backup of the metadata.

Application-consistent snapshots are considered as a full backup for SQL Server. With AppSync, an application-consistent snapshot can be mounted and restored into various recovery states, i.e. recovery, NORECOVERY, or standby, as shown in Figure 21. With restore with NORECOVERY, the database is left in a non-operational state and does not roll back the uncommitted transactions. As a consequence, additional transaction logs can be restored.

![Figure 21. AppSync Supported Database Recovery Types](image)
DETERMINING WHICH SNAPSHOT TO USE

The key differences between application-consistent and crash-consistent snapshot are:

- **The amount of work needed during recovery of the database:**
  
  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, application-consistent snapshot can offer better RTO.

- **If the SQL Server freezes I/Os during snapshot operation:**
  
  SQL Server allows up to 10 seconds for a snapshot to be taken. Application may experiences a performance drop during the time the I/Os are frozen.

- **If the 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 NORECOVERY mode to continue rolling forward log backups.

To determine which type of snapshot to use, you should consider the availability requirement, frequency of the snapshot, its impacts to performance, and whether there is need to support log backups.

Considerations for using application-consistent snapshots vs. crash-consistent snapshots:

- **Application-Consistent Snapshot**
  
  - Database backup
  
  - Creating always-on availability groups secondary replica, database mirroring mirror server, or log ship secondary

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

**Best Practices Tips:**

- Use application-consistent snapshots for backup, and for 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.
CONCLUSION

The EMC XtremIO All Flash Storage Array is engineered to transform SQL Server through uniquely consistent scale-out performance architecture, with in-memory always-on compression, deduplication and space-efficient copy data services, enabling application acceleration, consolidation, and agility.

The XtremIO All Flash Storage Array not only provides exceptional performance to any SQL Server deployment, it has also refined efficiency and simplicity for storage and database administrators managing the SQL Server application lifecycle. It enhances the SQL Server workload and resolves SQL Server database administrators’ challenges in many ways, as listed below:

- **Performance** – The XtremIO Storage Array 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 storage capacity can increase linearly on-demand without interruption of services with XtremIO’s unique scale-out architecture and non-disruptive expansion capability.

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

- **Agility** – The XtremIO copy data services refines agility for SQL Server lifecycle workflow management. The XtremIO copy data services instantly create full size, full performance, yet entirely space-efficient volume copies that can be repurposed for downstream data services. Additionally, it can be used to simplify database maintenance, backup, or upgrade workflows. This results in dramatic improvements of database administrator productivity, and increased workflow agility.

- **Simplicity** – No more complex capacity planning for SQL Server workloads. Database administrators no longer need to worry about RAID, spindle counts, or any of the underlined 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 Storage 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.
### APPENDIX I: BEST PRACTICES SUMMARY

The table below summarizes the best practices recommendations as discussed in this paper.

<table>
<thead>
<tr>
<th>CATEGORIES</th>
<th>BEST PRACTICES RECOMMENDATIONS</th>
</tr>
</thead>
</table>
| **GENERAL** | • Use XtremIO’s inline compression instead of SQL Server native row compression to achieve automatic compression of active data. Use SQL Server native page compression to maximize compression savings on less active or archival data.  
• Use SQL Server native row or page compression if the database is encrypted with SQL Server transparent data encryption. A transparent data encrypted database does not gain compression savings with XtremIO’s inline compression. An alternative to resolve the compatibility issue is to consider XtremIO’s Data at Rest Encryption.  
• Use XtremIO copy data services for rapid deployment of database copies with storage efficiency. |
| **PROVISIONING AND SQL SERVER FILE LAYOUT** | • No RAID configuration is needed. A flash optimized RAID (XDP) is built in and pre-configured on XtremIO.  
• No performance reason is required for separation of files.  
• For more efficient use of copy data services or storage replication, consider the following:  
  o Separate TempDB from user data files and log.  
  o Unless your business transaction spans across multiple databases, do not share LUN with multiple databases. A LUN/volume is the smallest unit for snapshot operation.  
• A single LUN per database suffices for the majority of deployment scenarios. XtremIO automatically takes care of even distribution of data, and ensures maximum parallel processing.  
• For extreme large scale I/O intensive SQL Server deployment, consider the following tuning options:  
  o Increase LUN queue depth. Refer to the Queue Depth Section.  
  o If I/Os exceed 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. This takes full advantage of parallel processing powers of the multiple Storage Controllers in the cluster and maximizes performance.  
• Follow the SQL Server recommendations on creating multiple data files for a user database and creating TempDB. For SQL Server recommendations, refer to [https://support.microsoft.com/en-us/kb/2154845](https://support.microsoft.com/en-us/kb/2154845). |
| **SQL SERVER LIFECYCLE MANAGEMENT** | • Use copy services to deploy copies of the database for test and development purposes.  
• Use the multi-level snapshot capability to cleanse sensitive data before presenting the database for test or development use.  
• Use the multi-level snapshot capability to create checkpoints or branches for development.  
• Use copy data services to build readable secondary replicas for near real-time reporting, without implementing always-on availability groups.  
• Use multi-level snapshot to synchronize data on all reporting instances.  
• Offload resource-intensive database maintenance tasks, such as query tuning, DBCC CHECKDB to secondary compute unit, using copies of database created with copy data services.  
• Offload backups to secondary compute unit, using copy data services.  
• Create rapid and storage efficient full database backups, using AppSync application-consistent snapshot. Copy snapshot volume(s) to secondary storages for disaster recovery, and protect data from unrecoverable physical storage failure.  
• Use AppSync flexible recovery type to restore a database into NORECOVERY state for point-in-time recovery, using transaction log backups.  
• Run an upgrade test on a snapshot copy to identify potential upgrade issues and measure upgrade time.  
• Create a snapshot backup before the upgrade in case there is a need to failback. |
| **MULTIPATHING** | • For Windows 2008 and above, use the Least Queue Depth policy with Microsoft Native Microsoft Multipath I/O (MPIO).  
• For SQL Server on VMware vSphere, use the VMware native round-robin path selection policy, and set switching frequency to one.  
• For best practices with PowerPath, refer to the EMC PowerPath on Windows Installation and Administration Guide or EMC PowerPath on VMware vSphere Installation and Administration Guide. |
<table>
<thead>
<tr>
<th>CATEGORIES</th>
<th>BEST PRACTICES RECOMMENDATIONS</th>
</tr>
</thead>
</table>
| QUEUE DEPTH      | • Set the LUN queue depth to 256.  
• For SQL Server deployment on VMware vSphere:  
  o Set queue depth on all layers, including physical HBA, vSCSI adapter, and VMKernel scheduler when configuring queue for SQL Server running on VMware vSphere.  
  o Set Disk.SchedNumReqOutstanding to 256.                                                                                                                                                                           |
| PHYSICAL SECTOR SIZE | • Use 4KB sector size for physical Windows and Hyper-V deployments with standalone XtremIO deployment.  
• Use 512B sector size if deployment includes products such as VPLEX and RecoverPoint for replication.  
• Use 512B sector size for SQL Server on VMware deployments.  
• When moving an existing database to the XtremIO platform, check the sector size on the existing database, using `dbcc fileheader`. Align the XtremIO volume sector size with the sector size of the existing database. |
| ALLOCATION UNIT SIZE | • Use 64KB Allocation Unit Size for SQL Server data and log volumes.                                                                                                                                                                   |
| APPLICATION-CONSISTENT VS. CRASH-CONSISTENT SNAPSHOT | • Use application-consistent snapshot for backup, and for HA that requires 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 II: PERFORMANCE TESTING CONFIGURATIONS AND METHODOLOGIES**

This section describes the configurations and methodologies used in the performance and scalability study of scaling up to eight SQL Server workloads on XtremIO Storage Array.

**TEST METHODOLOGIES**

The purpose of this study is to examine XtremIO storage system performance, in terms of IOPS and latency when the number of SQL Server instances scale-up in a consolidated environment. The goal is not to demonstrate the maximum performance capability of the system.

This study scales the number of SQL Server instances on a dual X-Brick XtremIO cluster. Initial testing started with a scenario of one SQL Server instance. Each subsequent test doubled the number of SQL Server instances of that in the previous test, until the number of the run SQL Server instances reached eight. Performance metrics were captured for 30 minutes during the steady state for each test scenario.

Each SQL Server instance ran a 1TB OLTP-like workload, generated by Microsoft BenchCraft TPC-E Toolkit. The Microsoft toolkit simulates user activities of a brokerage firm. The workload portrays a typical online transactional application access profile, with 90% read and 10% write, and majority 8KB random IOPS.

**TEST CONFIGURATIONS**

The test environment consists of the following hardware and software components.

**Hardware Resources**

<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 20TB X-Brick</td>
</tr>
<tr>
<td>Hosts</td>
<td>4</td>
<td>Dell PowerEdge R620</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 x 8 cores @2.40GHz, 380 GB RAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 x 1GbE NICs (management network)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 x 10 GBE NICs (data network)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 x 8 GB FC HBAs</td>
</tr>
<tr>
<td>Virtual Machines (VMs)</td>
<td>8</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</td>
</tr>
<tr>
<td>SAN Switches</td>
<td>1</td>
<td>16GB Fibre 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>v3.0</td>
<td>Storage OS</td>
</tr>
<tr>
<td>VMware vSphere</td>
<td>v5.5</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>Microsoft BenchCraft TPC-E Toolkit</td>
<td>1.12.0-1026</td>
<td>Workload generator and load driver</td>
</tr>
<tr>
<td>VOLUME NAME</td>
<td>VOLUME SIZE</td>
<td>TEMPDB/USER DATABASE</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| Infra       | 4TB         | 250GB x 8 TempDB      | • Shared datastore for all 8 OS/SQL binary vmdks  
|             |             |                       | • Shared datastore for all TempDBs |
| Sql1        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 1  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql2        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 2  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql3        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 3  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql4        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 4  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql5        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 5  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql6        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 6  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql7        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 7  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
| Sql8        | 2TB         | 1TB user database     | • SQL Server user data files and log files for SQL Server 8  
|             |             |                       | • Dedicated datastore with 1:1 datastore to vmdk mapping |
APPENDIX III: SNAPSHOT REFRESH

The following steps describe the detailed Transact-SQL, Windows PowerShell, and XtremIO CLI scripts and processes to refresh data of an SQL Server database using XtremIO snapshot refresh feature:

Step 1: Detach the stale database.

Execute the following SQL statement:

```
sp_detach_db @dbname='<database name>'
```

Step 2: Dismount the database volume(s).

In PowerShell, first list all XtremIO disks, then offline the specific database disk(s), verify disk is offline.

```
Get-disk -FriendlyName XtremIO*
Set-Disk -Number 6 -IsOffline $True
Get-Disk | Where-Object IsOffline -Eq $True
```

![Sample PowerShell Scripts on Dismounting a Volume](image1.png)

**Figure 22. Sample PowerShell Scripts on Dismounting a Volume**

Step 3: Run snapshot refresh on XtremIO.

**Figure 23** shows the parent-child relationship for the production source volume, the master snapshot, and the second level snapshots for the databases.

![XMS Volume Snapshot Tree View](image2.png)

**Figure 23. XMS Volume Snapshot Tree View**
Run the "create-snapshot-and-reassign" CLI command to refresh the master snapshot from the production source volume, and then run the "create-snapshot-and-reassign" on each of the reporting volumes. The "create-snapshot-and-reassign" command updates the snapshot volume with data from the source, and saves and renames the old snapshot with a new name. For example, SnapshotSet.1437607649 as shown in Figure 24 is the old version of sql1-master-snap.

Figure 24. Sample CLI on Snapshot Refresh

The "create-snapshot-and-reassign" CLI command supports snapshot refresh at volume, consistency group, or snapshot set level. See Figure 25 for details on syntax.

Figure 25. CLI Syntax for Snapshot Refresh

Step 4: Mount the volume(s).

In PowerShell, list all offline XtremIO disks, and then online the specific database disk(s), verify disk is online.

```powershell
Get-disk -FriendlyName XtremIO* | Where-Object IsOffline -Eq $True
Set-Disk -Number 6 -IsOffline $False
Get-disk -FriendlyName XtremIO*
```
Step 5: Attach the database.

Execute the "create database ... on <filespec> for attach" SQL statement to attach the database, for example:

```
USE [master]
GO
CREATE DATABASE [oltp] ON
    ( FILENAME = N'J:\MSSQLTpce_root.mdf' ),
    ( FILENAME = N'J:\TPCE_Log.ldf' ),
    ( FILENAME = N'J:\Fixed_1.ndf' ),
    ( FILENAME = N'J:\Scaling_1.ndf' ),
    ( FILENAME = N'J:\Scaling_2.ndf' ),
    ( FILENAME = N'J:\Scaling_3.ndf' ),
    ( FILENAME = N'J:\Growing_1.ndf' ),
    ( FILENAME = N'J:\Growing_2.ndf' ),
    ( FILENAME = N'J:\Growing_3.ndf' )
FOR ATTACH
GO
```
REFERENCES
This section lists links relevant to the findings described in this White Paper.

**XtremIO**


**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