



Dell Compellent Storage Center and Microsoft SQL Server

Dell Compellent Technical Solutions Group
May 2014

Revisions

Date	Description
October 2007	Initial release
December 2007	Technical review
June 2008	Technical review
July 2010	Technical review
June 2012	Technical review
May 2014	Technical review. Included some of the new features of SQL 2014 related to I/O.

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2013 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

PRODUCT WARRANTIES APPLICABLE TO THE DELL PRODUCTS DESCRIBED IN THIS DOCUMENT MAY BE FOUND AT:

<http://www.dell.com/learn/us/en/19/terms-of-sale-commercial-and-public-sector> Performance of network reference architectures discussed in this document may vary with differing deployment conditions, network loads, and the like. Third party products may be included in reference architectures for the convenience of the reader. Inclusion of such third party products does not necessarily constitute Dell's recommendation of those products. Please consult your Dell representative for additional information.

Trademarks used in this text:

Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Other Dell trademarks may be used in this document. Cisco Nexus®, Cisco MDS®, Cisco NX-OS®, and other Cisco Catalyst® are registered trademarks of Cisco System Inc. EMC VNX®, and EMC Unisphere® are registered trademarks of EMC Corporation. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter® and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation. Broadcom® and NetXtreme® are registered trademarks of Broadcom Corporation. Qlogic is a registered trademark of QLogic Corporation. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and/or names or their products and are the property of their respective owners. Dell disclaims proprietary interest in the marks and names of others.



Table of contents

Revisions.....	2
1 Introduction.....	5
1.1 Audience.....	5
1.2 Customer Support	5
2 SQL Server on Dell Compellent storage.....	6
2.1 Manageability.....	6
2.2 High availability.....	6
2.3 Performance	6
3 Sizing for SQL Server.....	7
3.1 Key I/O performance metrics	7
3.1.1 IOPS.....	7
3.1.2 Throughput	7
3.1.3 Latency.....	7
3.2 The write penalty.....	7
3.3 Random I/O versus sequential I/O.....	8
3.4 OLTP workloads.....	9
3.5 OLAP/DSS workloads	9
3.6 Sizing tiers	9
3.7 Using SSDs.....	10
3.7.1 Traditional SSD configurations	10
3.7.2 Flash optimized configurations	10
4 Validating the storage design	11
4.1 Validating the I/O path	11
4.2 Validating the disks	12
5 Storage setup and configuration	13
5.1 Single versus multiple disk pools.....	13
5.2 Creating volumes.....	13
5.2.1 Performance considerations.....	13
5.2.2 Flexibility versus manageability.....	14
5.3 Storage profiles	14
5.3.1 User database data file considerations	14



5.3.2	User database transaction log file considerations	14
5.3.3	Tempdb database file considerations	15
5.3.4	System database file considerations.....	15
5.3.5	SSD considerations	15
5.4	Data Instant Replay	15
6	Windows Setup and configuration	17
6.1	Allocation unit size.....	17
6.2	MPIO.....	17
6.3	Partition alignment	17
7	Server setup and configuration	18
7.1	HBA considerations	18
7.2	iSCSI considerations	18
8	Reducing SQL Server I/O.....	19
8.1	Memory.....	19
8.2	Buffer pool extension	19
8.3	Database compression.....	19
8.4	Instant File Initialization.....	20
8.5	Resource Governor	20
8.6	Database design considerations.....	20
8.6.1	Database design	21
8.6.2	Query design.....	21
8.6.3	Application design.....	21
8.6.4	Maintenance	21
A	Additional resources.....	22
A.1	Dell TechCenter	22
A.2	Online resources.....	22



1 Introduction

This document describes the best practices for running Microsoft SQL Server (versions 2005 and later) with a Dell Compellent Storage Center. SQL Server performance tuning and profiling are beyond the scope of this paper. Due to dependencies on application design and requirements, environments and their recommendations can differ greatly. Please visit Microsoft TechNet for more in depth information on tuning SQL Server databases at <http://technet.microsoft.com/>.

1.1 Audience

The audience for this document is database administrators, system administrators, storage administrators, and architects who analyze, design, and maintain robust database and storage systems. Readers should be familiar with Microsoft SQL Server, Microsoft Windows Server, and Dell Compellent Storage Center.

1.2 Customer Support

Dell Compellent provides live support at 1-866-EZSTORE (866.397.8673), 24 hours a day, 7 days a week, 365 days a year. For additional support, email Dell Compellent at support@compellent.com. Dell Compellent responds to emails during normal business hours.



2 SQL Server on Dell Compellent storage

A Dell Compellent Storage Center provides a flexible, easy to manage storage platform that helps meet the demanding uptime and performance requirements common in SQL Server environments.

2.1 Manageability

SQL Server environments are often very dynamic. A Dell Compellent Storage Center makes it easy to adapt the storage configuration to meet changing database requirements. Volumes can be created or expanded with very little effort. Data can be easily moved between tiers or RAID levels. Disks can be added to increase performance or capacity on the fly. New storage technology can be integrated when it becomes available. All of this can be done while SQL Server remains online and available for applications.

2.2 High availability

The uptime of storage systems is more important than ever, as organizations increasingly grow dependent on SQL Server databases. The Dell Compellent Storage Center was designed with high availability and fault tolerance in mind. Redundant hardware and RAID technology protect against an unplanned outage due to the failure of a component. Maintenance tasks such as controller firmware upgrades, hardware expansions or hardware replacements can all be done without taking databases offline. The result is a storage system that rarely needs to be taken down, providing a solid foundation for SQL Server databases.

2.3 Performance

SQL Server is an I/O intensive application that requires high performance storage. A Dell Compellent Storage Center makes it easy to leverage a large number of spindles to support SQL Server workloads. Solid state disks (SSD) can also be integrated into a Storage Center for those environments that require extremely low latency. A variety of connectivity options exist, including 16Gbps fibre channel and 10Gbps iSCSI, to ensure that data can be moved quickly and efficiently between the database server and the Storage Center. Increasing performance by adding additional spindles or new storage technology can be done dynamically to ensure good performance in the future.



3 Sizing for SQL Server

The I/O subsystem is a critical component of any SQL Server environment. Sizing and configuring a storage system without understanding the I/O requirements can have disastrous consequences. Monitoring performance in an existing environment using a tool like the Dell Performance Analysis Collection Kit (DPACK) can help define the I/O requirements. For best results, capture performance statistics for a time period of at least 24 hours that includes the system peak workload.

3.1 Key I/O performance metrics

A good understanding of the key metrics used to describe I/O performance is required to successfully define I/O requirements and size storage systems.

3.1.1 IOPS

I/O operations per second, or IOPS, describe the number of reads and writes occurring each second. This is the primary metric used to determine the number of disks required in a storage system. The number of IOPS is reported by Windows Performance Monitor using the Disk Reads/sec, Disk Writes/sec and Disk Transfers/sec physical disk counters.

3.1.2 Throughput

Typically expressed in megabytes per second (MB/s), throughput describes the amount of data transferred between the server and the storage. Throughput can be calculated using the number of IOPS and the average I/O size ($\text{Throughput} = \text{IOPS} * \text{Average I/O size}$). This is the primary metric used to design the path between the server(s) and the storage. Throughput is reported by Windows Performance Monitor using the Disk Read Bytes/sec, Disk Write Bytes/sec and Disk Bytes/sec physical disk counters.

3.1.3 Latency

Typically expressed in milliseconds (ms), latency describes the amount of time an I/O operation takes to complete. When a server experiences high latency, this generally indicates an I/O bottleneck in the system. Latency is reported by Windows Performance Monitor using the Avg. Disk sec/Read, Avg. Disk sec/Write and Avg. Disk sec/Transfer physical disk counters.

3.2 The write penalty

RAID technology is used to protect data on a Dell Compellent Storage Center. For each write sent to a Storage Center volume, additional I/O is performed on the physical disks to provide the redundancy needed to protect the incoming data. The amount of additional I/O is determined by the RAID configuration for the volume. The following table shows the number of I/Os required when performing a write on a Dell Compellent Storage Center. The number of I/Os required to complete a write is referred to as the Write Penalty.



Table 1 Write penalty by RAID type

RAID Type	Write Penalty	I/O Description
RAID 10	2	2 writes
RAID 10 DM	3	3 writes
RAID 5	4	2 reads, 2 writes
RAID 6	6	3 reads, 3 writes

By default, writes to Storage Center volumes use RAID 10. Since RAID 10 has the lowest write penalty, it is strongly recommended to always use RAID 10 for writes. All storage profiles shipped with the Storage Center will write using RAID 10.

When sizing a Storage Center, it is important to factor in the write penalty when determining the number of disks required. The number of IOPS reported by the server when gathering I/O statistics does not include the write penalty. If the IOPS target for the Storage Center design does not include the write penalty, it will be undersized. To adjust the number of IOPS to include the write penalty, use the following formula:

$$\text{Number of IOPS} + (\text{Number of IOPS} * \text{Write Percentage} * (\text{Write Penalty} - 1))$$

Where:

Number of IOPS = Number of IOPS generated by the server

Write Percentage = Percent of the I/O attributed to writes

Write Penalty = Number of I/Os required to complete a write

As an example, consider a server that is performing a workload of 10,000 IOPS with a mix of 70% reads and 30% writes. Since the default storage profile is used, writes use RAID 10, which has a write penalty of 2. For this workload, the Storage Center needs enough disks to support 13,000 IOPS ($10,000 + (10,000 * .30 * (2-1)) = 13,000$).

3.3 Random I/O versus sequential I/O

Spinning disks can perform more sequential I/O than random I/O. This is because sequential I/O requires less head movement on the disk. While I/O statistics show how much I/O is occurring, they do not show how much is sequential I/O versus random I/O. The virtualized storage architecture of a Dell Compellent Storage Center may cause sequential I/O requests from the server to become large block random I/O on the physical disks. Performance will still be good, as large block random I/O provides comparable performance to sequential I/O. For best results when sizing a Dell Compellent Storage Center, assume all of the I/O is random. This will ensure good performance for both random and sequential operations.



3.4 OLTP workloads

While every environment is unique, an Online Transaction Processing (OLTP) workload typically consists of small, random reads and writes. A storage system servicing this type of workload is primarily sized based on the number of IOPS required. Different drive types have different performance capabilities. Faster spinning drives are able to provide more IOPS with lower latency than slower spinning drives. For best performance, 15K drives are recommended for OLTP workloads. For environments with very low latency requirements, consider using SSDs (see Section 3.7, "Using SSDs").

3.5 OLAP/DSS workloads

An Online Analytic Processing (OLAP) or Decision Support System (DSS) workload is typically dominated by large, sequential reads. A storage system servicing this type of workload is primarily sized based on throughput. When designing for throughput, the performance of the entire path between the server and the Storage Center needs to be considered. For best throughput, consider using 16Gbps Fiber Channel (FC) or 10 Gbps iSCSI connectivity to the Storage Center, and 6 Gbps SAS connectivity from the controllers to the disk enclosures. To meet high throughput requirements, multiple HBAs may be required in the server, the Storage Center or both. While 15K drives offer the best performance, 10K drives may be sufficient to meet throughput requirements and offer more capacity per spindle.

3.6 Sizing tiers

When sizing a Storage Center with two tiers, start by sizing Tier 1 for all of the IOPS and at least 30% of the capacity, and Tier 3 for 70% of the capacity and at least 30% of the IOPS. In a two tier system, the fastest disks will be Tier 1 and the slower disks will be Tier 3. When sizing each tier for SQL Server, there are a couple of things to consider.

- Accessible pages on a volume are not eligible to move to a lower tier until they are at least 12 days old.

In environments where indexes are rebuilt frequently, accessible pages on the database volume may never be older than 12 days. All accessible pages will always be in Tier 1. In those environments, Tier 1 will need to be sized for 100% of the capacity required by SQL Server.

- The automated tiering feature of a Dell Compellent Storage Center automatically moves infrequently accessed data to slower, cheaper storage (Tier 3), reducing storage costs without impacting performance.

For environments where the access pattern is even across the entire data set, all data may need to reside on Tier 1 to ensure good performance. In those environments, size Tier 1 for 100% of the capacity required by SQL Server.

3.7 Using SSDs

Solid state disks (SSD) offer significantly better performance than traditional hard drives for most I/O patterns. From a performance perspective, SSDs are suitable for storing any type of SQL Server file. While SSDs can support a large number of IOPS, it is easy to exceed throughput limits of the drive or other components in the path with the large I/Os that can be generated by SQL Server. It is important to understand the performance characteristics of SSDs as well as the I/O requirements of the SQL Server environment before implementing SSDs.

Both single-level cell (SLC) and multi-level cell (MLC) SSDs can be used in a Dell Compellent Storage Center. Although they provide lower capacity than MLC drives, SLC drives can perform a larger number of write cycles, making them better suited for write-intensive workloads. MLC drives offer a much higher capacity than SLC drives, but cannot endure as many write cycles, making them better suited for read-intensive workloads. Traditional SSD configurations contain only SLC drives used as a tier in the same way as spinning drives. Storage Center 6.4 introduced flash optimized configurations that use both SLC and MLC drives to provide a large amount of high performance storage at a much lower price point.

3.7.1 Traditional SSD configurations

SLC SSDs can be used as the high performance tier in a Dell Compellent Storage Center. When using SLC SSDs as a tier that participates in automated tiering, there needs to be enough drives in the tier to handle 100% of the IOPS and at least 30% of the capacity of the volumes that use the SSD tier. Since SLC SSDs can perform a large number of IOPS, a small number of drives may meet the IOPS requirement. However, the capacity requirement may need many drives. It is very important to have enough SLC SSDs to meet both the IOPS and capacity requirements. If the SSD tier becomes full, performance will degrade significantly.

Given the relatively high cost of SLC SSDs, it may be more cost effective to place a subset of the SQL Server files on SSDs, instead of using SSDs with automated tiering. Storage profiles allow an administrator to easily control which volumes reside on SSDs. While transaction log files and tempdb files require high performance and tend to be small enough to fit on a small number of SSDs, they are not always the best choice for SSDs. In some cases, it may be better to place data files on SSDs. Analyze the access patterns of the database files to determine the best fit. If a data file is too large to fit on the available SLC SSD drives, the portion of the data requiring high performance can be moved to a separate file group and placed on SSDs.

3.7.2 Flash optimized configurations

By leveraging the characteristics of both types of SSDs, a flash optimized configuration can provide both high capacity and high performance at a much more affordable price point, making it practical to store a large SQL Server data set solely on SSD drives. When sizing a flash optimized configuration, it's important to use enough SLC drives to handle the volume of writes and enough MLCs to provide sufficient capacity. When working with traditional SSD configurations, the number of drives was determined primarily by capacity requirements that were often reduced due to cost considerations. Using MLC drives makes it cost effective to provide a large amount of capacity. However, additional SLC and/or MLC drives may need to be added to the configuration to keep from exceeding the throughput limit of the drives.



4 Validating the storage design

Once the I/O requirements have been defined, it is easy to get a feel for whether the hardware can provide the desired performance by running some simple tests. Both SQLIO and IOMeter can simulate I/O patterns generated by SQL Server. SQLIO is available from Microsoft as a free download. IOMeter is available via Open Source and can be found using an Internet search engine.

4.1 Validating the I/O path

The first thing to test on a new configuration is the path between the server and the Storage Center. Running a large block sequential read test using small files will saturate the path between the server and storage. This test verifies that all paths are fully functional and can be used for I/O traffic. This test should be run on a server and Storage Center that is dedicated to this test because running it on a live system could cause significant performance issues.

To validate the I/O path using SQLIO, perform the following steps:

1. Download the SQLIO utility and install it on the server.
2. On the Storage Center, create volumes and map them to the server. If testing a single controller system, create one volume. If testing a dual controller system, create two volumes, with a volume owned by each controller.
3. On the server, format each volume using a 64KB allocation unit.
4. Create a SQLIO parameter file called "paramPath.txt". This file will contain the number of files to use, along with the number of threads for each file and the name of the file. Use one 10MB file per volume. Use a total of 8 threads.

An example of the parameter file for a single controller test:

```
L:\testfile.dat 8 0x0 10
```

An example of the parameter file for a dual controller test:

```
L:\testfile.dat 4 0x0 10  
M:\testfile.dat 4 0x0 10
```

5. Run a sequential read test. Use a block size of 512KB with 32 outstanding I/Os. This SQLIO command will run the test for two minutes:

```
sqlio -kR -s120 -fsequential -o32 -b512 -BH -LS -FparamPath.txt
```

If the displayed throughput matches the expected throughput for the number of HBA ports in the server, the paths between the server and Storage Center are set up correctly.



4.2 Validating the disks

Once the I/O path has been validated, the next step is to test the disks. For best results when testing disks on a Storage Center, use the following guidelines when configuring the test.

- In a dual controller system, create at least one volume per controller. This ensures that I/O will be distributed across both controllers. Using both controllers more closely simulates real world activity. For best results, create the same number of volumes on each controller.
- When performing I/O tests on any storage platform, it's important to use files that are larger than the controller cache. For more accurate results, use a file size that matches the amount of data being stored. In an environment where that is not practical due to a large data set, use a file size of at least 100GB.
- Some I/O test tools, including SQLIO, generate files full of zeros. By default, Storage Center tracks pages full of zeroes in the metadata, but not store them on disk. This behavior, known as thin writing, causes inaccurate results when testing with files containing only zeros. Pre-allocate the volumes being used for I/O testing to prevent thin writing from skewing the test results. Discard the test volumes and create new volumes once testing is complete.

The purpose of this type of testing is to validate that the storage design will provide the required throughput and IOPS with acceptable latency. It's important that the test does not exceed the designed capacity of the Storage Center. For example, a Storage Center designed for a workload of 5,000 IOPS is likely to perform poorly with a workload of 10,000 IOPS. If a test is generating a workload higher than the designed capacity, adjust the workload being generated by reducing the number of threads and/or outstanding I/Os.

The results of the DPACK analysis provide an I/O target to simulate using these tests. To get an idea of the performance capabilities of the Storage Center, run I/O tests with a range of I/O sizes commonly seen with SQL Server. When testing random I/O, test with an I/O size of 8KB and 64KB. When testing sequential I/O, start with I/O sizes of 8KB and 64KB. Since processes like read ahead scans and backups can issue much larger sequential I/O, it is a good idea to also test block sizes up to 1024KB.



5 Storage setup and configuration

5.1 Single versus multiple disk pools

Dell Compellent recommends using a single virtual disk pool when implementing SQL Server. This provides better performance by leveraging the aggregate I/O bandwidth of all disks to service I/O requests from SQL Server. A single disk pool is also easier to manage, allowing an administrator to quickly and easily adapt the storage system to satisfy the ever changing workloads that are common in SQL Server environments.

5.2 Creating volumes

A Dell Compellent Storage Center virtualizes its storage to take advantage of all spindles in the disk pool. Characteristics such as RAID level and storage tier can have a big impact on performance and are configured at the volume level. Compellent Data Instant Replay is also configured at the volume level. There are many different types of files that are part of a SQL Server instance. Those different types of data often have different performance and Replay requirements. Dell Compellent recommends creating at least five volumes for an instance of SQL Server as shown in the following table.

Table 2 Volume provisioning recommendations

File type	Number of volumes	Typical performance requirements	Typical Replay requirements
User DB data	At least 1 per instance	Lower performance may be acceptable	Frequent Replays, same schedule as log volume
User DB transaction log	At least 1 per instance	High performance required	Frequent Replays, same schedule as data volume(s)
Data root directory (includes system DBs)	1 per instance	Lower performance may be acceptable	Infrequent Replays, independent schedule
Tempdb data and transaction log	1 per instance	High performance may be required	No Replays
Native SQL Server backup	1 per instance	Lower performance may be acceptable	Replays optional, independent schedule

5.2.1 Performance considerations

If there is a group of databases that require high performance and another group that does not, consider creating a set of volumes for each group of databases. Even if there is only one tier of storage, this strategy will make it easier to adjust the storage configuration in the future. For example, having the high performance databases on separate volumes will make it very easy to move them to a higher performance tier in the future.



Having multiple volumes has an additional benefit in dual controller systems. All I/O requests for a given volume are processed by the controller that owns the volume. While a volume can be owned by either controller, a volume is only owned by one controller at a time. Having many volumes makes it easier to distribute the I/O load evenly across both controllers. For databases that have very high performance requirements, the database can be spread across two or more data files on separate volumes to leverage resources on both controllers.

5.2.2 Flexibility versus manageability

For ultimate flexibility, create a volume for each user database file. This provides the ability to independently optimize the storage and Replay configuration for each individual database. With thin provisioning, there is no space penalty for creating a lot of volumes. However, a large number of volumes can be difficult to manage. It is up to the DBA and/or storage administrator to find the right balance between flexibility and maintainability when determining the number of volumes to create. Understanding your database I/O patterns is critical to making the best decisions.

5.3 Storage profiles

Storage profiles define the RAID level used to protect data on a volume and the tiers where that data is stored. The information in the profile is used by Compellent Data Progression when moving existing pages as well as by the Storage Center when creating new pages. In most environments, using the recommended (and default) storage profile provides good I/O performance for all types of database volumes. It is strongly recommended to use the this profile first and evaluate its suitability before attempting to change the storage profiles.

5.3.1 User database data file considerations

On volumes storing the data files, the recommended profile keeps the highly active parts of the database on Tier 1 and the less active parts on lower tiers. In most cases, this provides the best performance. However, there are some environments where the entire dataset is highly active and needs the performance of Tier 1. Before changing the storage profile to force the data volume to live only on Tier 1, consider the impact that the frequency of both index rebuilds and Replays has on Data Progression. Using the recommended profile allows inaccessible Replay pages to be moved to Tier 3, freeing up Tier 1 resources for active pages. Moving inaccessible Replay pages to Tier 3 helps provide better performance on Tier 1.

5.3.2 User database transaction log file considerations

The transaction log requires good performance and should reside on Tier 1. This is the most important storage component for OLTP latency. Writes to the transaction log start at the beginning of the file, go to the end of the file and then start again at the beginning of the file. The recommended profile allows inaccessible Replay pages to be moved to Tier 3, freeing up Tier 1 resources for active pages.



5.3.3 Tempdb database file considerations

Tempdb files can require high performance and generally should reside on Tier 1. Some applications use tempdb heavily and others hardly use it at all. Since Replays should not be enabled on tempdb volumes, all pages will stay on Tier 1 using the recommended profile in a two tier system. In a three tier system, use a storage profile to ensure the tempdb volume stays on the desired tier. In most cases, this will be Tier 1. If Tier 1 is an SSD tier, some environments are better off with tempdb on the tier containing 15K drives instead of SSD drives.

5.3.4 System database file considerations

The system databases (master, model and msdb) generally do not have special performance requirements. Use the recommended profile for the system database volumes.

5.3.5 SSD considerations

If there are not enough SSDs to function as Tier 1 storage for all volumes in the Storage Center, custom storage profiles can be used to control which volumes will use the SSD Tier, and which volumes will be restricted to spinning drives. The type of files that benefit the most from SSDs varies from environment to environment. Monitor the performance of the database volumes to determine the best fit for SSDs.

5.4 Data Instant Replay

Data Instant Replay is very powerful when combined with SQL Server. By leveraging Replays, database backup and recovery is very fast and space efficient, even for large databases.

Dell Compellent strongly recommends using Replay Manager to take Replays of database volumes while SQL Server is online. Using Replay Manager is the only way to ensure transactional consistency, since writes are frozen inside of SQL Server while Replays are taken. Replays of active database volumes taken without Replay Manager may not be usable for database recovery. Replay Manager also allows Replays to be used as the starting point of a restore chain that includes transaction log backups, as databases can be restored without recovery. In addition, backups taken by Replay Manager are fully logged within SQL Server providing complete backup integration.

Even though Replay Manager allows specific databases to be selected, Replays are taken on the entire volume. If all databases on a given set of volumes are not backed up together, disk space could be wasted on the Storage Center. Consider placing the databases for each backup set on a separate set of volumes.

Placing a large database on its own set of volumes will allow additional recovery flexibility when using Replay Manager. If a database is on its own volumes and in its own backup set, the database can be recovered very quickly using the **Resync-RMRestorePoint** PowerShell cmdlet without affecting other databases. This can be a big time saver when recovering a large database with Replays and transaction log backups.

When taking Replays of database volumes without Replay Manager, it is recommended to use a consistency group. This will increase the odds of a successful database recovery using those Replays. Using a consistency group on the Storage Center allows Replays to be taken on multiple volumes at the



same point in time. For information on how to setup a consistency group, see the "Creating a Consistent Replay Profile" section in the *Storage Center System Manager Administrator's Guide*.

Dell Compellent recommends taking a Replay of each user database volume at least once per day, with a retention period of 25 hours or longer. Data Progression moves Replay data (such as read-only data) according to settings in the storage profile. If the recommended (default) storage profile is used, Replay data is moved to RAID 5, while writes will still occur at RAID 10. This allows Tier 1 space to be used more efficiently. For more information on how Data Progression works with Replays, see the document titled, *Data Progression Configuration for Replay Enabled Volumes*.

Tasks ideally suited for using Replays with SQL Server include:

- Restoring a copy of a database to quickly recover data or objects after errant updates or deletes.
- Restoring a copy of a database on the original server or another server for development, testing or reporting.
- Replicating databases to a DR location. During a DR test, databases can be recovered without interrupting replication.

For more information about using Replays with SQL Server, see the *Replay Manager Administrator's Guide*.



6 Windows Setup and configuration

6.1 Allocation unit size

Use a 64KB allocation unit size when formatting volumes that will contain database files (transaction log and data) or database backups.

6.2 MPIO

Set the MPIO policy to **Round Robin** for all database volumes. This is the default for Windows 2008 and newer. It allows all paths to be used, enabling higher throughput between the server and the Storage Center. This setting works best for most environments as it is easy to manage and performs very well.

For database servers with a large number of I/O ports, the overhead of the **Round Robin** MPIO policy can reduce the maximum throughput to the Storage Center. To maximize throughput, create a volume for each port on the server and use the **Fail Over Only** MPIO policy to define a single, unique, active path for each volume. All other paths should be defined as standby paths. By using a data file of the same size on each volume, the data will be evenly distributed across the volumes. This strategy is more complex to setup and maintain. It should only be used in environments that require maximum throughput.

6.3 Partition alignment

An offset of 64KB, or a multiple of 64KB, is recommended for volumes created on Windows 2003 and earlier. New volumes created on Windows Server 2008 or newer should already be aligned since 1024KB is the default offset.



7 Server setup and configuration

7.1 HBA considerations

Be sure that the HBA firmware and drivers are up to date. As with any update, it is important to verify functionality in a test or QA environment before implementing in production.

SQL Server is a very I/O intensive application. It is important to configure the queue depth of the HBA properly to optimize performance. See the "HBA Server Settings" section in the *Storage Center System Manager Administrator's Guide*.

7.2 iSCSI considerations

Use at least 1-Gbit Ethernet to connect to the Storage Center. For best performance, use 10-Gbit Ethernet.

To provide consistent performance, create a separate network for iSCSI traffic.

If using the Microsoft iSCSI Initiator, Dell Compellent recommends the following:

- Make the SQL Server service dependent on the Microsoft iSCSI Initiator Service.
- Use dedicated network interface cards (NIC) for iSCSI connections to the Storage Center.
- On each of the dedicated NICs, disable all features, protocols and clients except for TCP/IPv4.
- Explicitly create each of the iSCSI connections, rather than using the **Quick Connect** button.



8 Reducing SQL Server I/O

8.1 Memory

Unnecessary I/O can be avoided and performance can be increased by allocating the proper amount of memory to SQL Server. SQL Server performs all I/O through the buffer pool (cache) and therefore uses a large portion of its memory allocation for the buffer pool. Ideally, when SQL Server performs I/O, the data is already in the buffer pool and it does not need to go to disk. This type of I/O is referred to as logical I/O and is the most desirable because it results in the best performance. If the data SQL Server needs does not reside in the buffer pool, it will need to access disk resulting in physical I/O.

Proper memory allocation is critical to SQL Server performance and can improve storage performance as well. In many cases, SQL Server and storage performance can be further improved by adding additional memory. Generally speaking, the more memory the better but there is a point of diminishing returns that is unique to each environment.

8.2 Buffer pool extension

Starting with SQL Server 2014, the buffer pool can be extended to a file on the file system to provide additional space to cache data or index pages. Only pages that have not been modified can be stored in the buffer pool extension. While the extension file can reside on any type of storage, it's best to store it on local solid state devices. Using this feature can provide significant performance benefits without adding memory to the database server. By caching more pages on the server, the I/O load on the Storage Center is reduced.

While the SSD tier on the Storage Center can be used for the buffer pool extension, the SSD tier is typically more effective when used as primary storage. In cases where there is not enough SSD capacity to be effective for primary storage of the database volumes, the SSD tier can be considered for the buffer pool extension. When placing the buffer pool extension on the Storage Center, use the following guidelines:

- Create a separate volume for the buffer pool extension
- Assign a storage profile that is configured to store all data on the SSD tier (tier 1)
- Do not take Replays of the buffer pool extension volume

8.3 Database compression

The overall I/O workload can be reduced by enabling database compression in SQL Server. While there is a tradeoff in terms of CPU utilization on the database server, it is still a viable option to consider and test in an environment. Database compression reduces I/O by reducing the amount of data that needs to be stored. The SQL Server data pages are compressed in memory before being written to disk resulting in less data stored and therefore less I/O.



8.4 Instant File Initialization

By default, SQL Server writes zeros to the data file during the allocation process. The process of zeroing out the data files consumes I/O and acquires locks as the SQL Server data pages are written. This activity can occur for minutes or even hours depending on the file size. The Thin Writing feature of the Compellent Storage Center will save space by not physically storing all of the zero data. It will simply mark a bit in the metadata indicating these pages are filled with zeros. Although Thin Writing will mitigate the space concern, there is still processing that occurs on the controller to receive and inspect the data. While this may seem minor, writing zeros to these files can occur at critical periods when time and performance are critical such as database auto growth, expanding a full data file, replication, or restoring a database as part of a disaster recovery event.

When Instant File Initialization is enabled, SQL Server will skip the process of zeroing out its data files when allocating space. Dell Compellent recommends enabling Instant File Initialization.

8.5 Resource Governor

The Resource Governor was added in SQL Server 2008 to allow database administrators to limit the CPU and memory resources a query is able to consume. This feature was enhanced in SQL Server 2014 to allow I/O resources to be limited as well. For example, the Resource Governor can be used to reduce the impact of a user running an I/O intensive report by limiting the maximum number of IOPS that user can perform. While a query throttled by the Resource Governor will take more time to complete, overall database performance will be better.

8.6 Database design considerations

Reducing SQL Server I/O requires a holistic approach. Many of the items in this section will require involvement from the whole team responsible for the SQL Server applications including the business owner, architect, developer, database administrator, system administrator, and everyone other that are involved. Decisions at the design level have a multiplied downstream impact as data is written and read multiple times and duplicated in various types of database copies including databases copied for other uses such as testing and reporting, replicated databases, replicated storage and backups. There are more copies of databases that people retain in various forms than they realize.

One of the most challenging aspects of SQL Server is that the I/O pattern and the amount of I/O that is generated can vary greatly depending on the application, even if those applications have databases of the same size. This is because the design of both the database and the data access code control SQL Server I/O.

Database tuning can be one of the most cost effective ways to reduce I/O and improve scalability. Database tuning is a very large topic and is beyond the scope of this document. However, there are numerous resources available in both online and print form. At a high level, the following areas should be considered when tuning a database to reduce I/O.



8.6.1 Database design

The foundation of the entire database and the schema for how data will be stored and ultimately accessed is determined by the database design. The database design should support both usability and efficient data access. This includes efficient table design and data types as well as indexes, partitioning, and other features that can improve efficiency. It is common for database design to only be focused on usability.

8.6.2 Query design

How a query is written can greatly affect the amount of I/O SQL Server needs to perform when executing the query. Queries should return only the required amount of data in the most efficient manner possible. There are many great tools and resources available to help design efficient queries.

8.6.3 Application design

Consider how applications are using the data and the manner in which it is requested. Sometimes code and component reuse can result in the same data being unnecessarily retrieved over and over again. All data access should be purposeful.

8.6.4 Maintenance

SQL Server uses a cost based optimizer to generate query plans for data access. These plans are based on the statistics regarding how data is distributed in the tables. If the statistics are inaccurate, bad query plans may result and unnecessary I/O will be performed. Proper database maintenance includes ensuring that statistics are up to date.

Frequent data modifications can also lead to fragmentation within SQL Server data files, producing unnecessary I/O. Fragmentation can be addressed through index reorganization or rebuilds as part of regular database maintenance.

The database maintenance process itself can also have a large I/O impact. Typically, every table and index does not need to be rebuilt or reorganized every time maintenance is run. In addition, table partitioning strategies can also be leveraged to make the maintenance process more selective. Consider implementing maintenance scripts to perform maintenance on an as-needed basis.

For mission critical databases these maintenance activities need to be considered as part of the overall design. If maintenance is not considered as part of the overall process, issues can arise, such as unmanageable sizes and feature incompatibilities, that limit available options and strategies.



A Additional resources

A.1 Dell TechCenter

Use the best practice guides, solution guides and reference architectures on Dell TechCenter to help get the most out of the Dell Compellent Storage Center. That content can be accessed at <http://en.community.dell.com/techcenter/storage/w/wiki/5018.compellent-technical-content.aspx>. Dell TechCenter also has videos demonstrating many of the Dell Compellent Storage Center features and application integrations. The Dell Compellent video content can be accessed at <http://en.community.dell.com/techcenter/storage/w/wiki/5089.compellent-video-content.aspx>.

A.2 Online resources

There is a tremendous amount of SQL Server information available online. While not a complete list, the table below contains several useful links. Additional information can be found by using any Internet search engine.

Table 3 Useful links

Description	Link
SQL Server I/O Basics	http://technet.microsoft.com/en-us/library/cc966500.aspx
SQL Server I/O Basics, Chapter 2	http://technet.microsoft.com/en-us/library/cc917726.aspx
Pre-deployment I/O Best Practices	http://technet.microsoft.com/en-us/library/cc966412.aspx
Analyzing I/O Characteristics and Sizing Storage Systems for SQL Server Database Applications	http://msdn.microsoft.com/en-us/library/ee410782(v=SQL.100).aspx
Dell SQL Server Solutions	http://www.dell.com/sql
Microsoft SQL Server homepage	http://www.microsoft.com/sql
SQL Server Customer Advisory Team	http://sqlcat.com
Microsoft online SQL Server forum	http://social.msdn.microsoft.com/Forums/en-US/category/sqlserver
Professional Association for SQL Server	http://sqlpass.org
Online SQL Server resources	http://www.sqlservercentral.com http://www.sqlteam.com http://www.sql-server-performance.com

