Technical Report

Best Practice Guide for Microsoft SQL Server and SnapManager 7.0 for SQL Server with Data ONTAP Operating in 7-Mode

Cheryl George, NetApp

October 2013 | TR-4232

Abstract

This best practice guide is intended for storage administrators and database administrators to help them successfully deploy Microsoft® SQL Server® 2012, 2008, and 2005 on NetApp® storage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

Today's business applications are more data centric than in the past, requiring fast and reliable access to intelligent information structures that are often provided by a high-performance relational database system. Many organizations use Microsoft SQL Server as a back-end datastore for mission-critical business applications. The latest release, Microsoft SQL Server 2012, delivers performance, scalability, availability, and security.

SQL Server implementations have become more complex and require more reliability than before. Database service-level agreements (SLAs) require predictable performance, and outages are disruptive and costly for database consumers. The underlying physical storage architectures supporting SQL Server are expected to scale in order to meet the growing capacity, but frequently bottlenecks arise, and backup processes get slower as databases grow. SQL Server databases are growing significantly larger to meet the needs of today's organizations, while business requirements dictate shorter backup and restore windows.

Using the built-in SQL Server streaming backup and recovery solution, restore times require the current backup time plus additional time to play any transaction logs. The larger the database, the harder it is to meet organizational SLAs. Failure to meet the organization’s SLA can have a direct and devastating effect on revenue and customer goodwill. It is clear that conventional backup mechanisms simply do not scale, and many organizations have sought new and innovative solutions to solve increasingly stringent business requirements.

1.1 Purpose and Scope

This document describes the best practices and offers insight into design considerations when deploying Microsoft SQL Server on NetApp storage systems running Data ONTAP® operating in 7-Mode, with the goal of achieving effective and efficient storage deployment planning and end-to-end data protection and retention planning. The scope of this guide is limited to technical design guidelines based on the design principles and preferred standards that NetApp recommends for storage infrastructure when deploying aforementioned versions of Microsoft SQL Server. The end-to-end implementation is out of scope of this report.

The best practices and recommendations described in this guide enable Microsoft SQL Server architects and NetApp storage administrators to plan a highly available and easy-to-manage SQL Server environment and to meet stringent SLAs. It is assumed that the reader has working knowledge of the following:

- NetApp Data ONTAP 7-Mode operating system
- NetApp SnapDrive® for Windows® data management software
- NetApp SnapManager® for SQL Server (SMSQL)
- Microsoft SQL Server architecture and administration
- Microsoft SQL Server

For configuration compatibility across the NetApp stack, refer to the NetApp Interoperability Matrix Tool (IMT).

1.2 Intended Audience

This report is intended for experienced SQL Server administrators, IT managers, and storage administrators who have reviewed the following relevant product documents:

- NetApp SnapDrive for Windows (SDW)
- SnapManager for Microsoft SQL Server (SMSQL)

- Data ONTAP Operating in 7-Mode Documentation

Readers should ideally have a good understanding of SQL Server storage architecture and administration as well as SQL Server backup and restore concepts. For more information about Microsoft SQL Server architecture, refer to SQL Server Books Online (BOL).

1.3 Caveats

- Applications utilizing SQL Server as a back end might have specific requirements dictated by the design characteristics of the application and are beyond the scope of this technical report. For application-specific guidelines and recommendations relating to physical and logical database layout, contact your application provider.
- Best practices for managing SQL Server environments focus exclusively on the latest NetApp storage operating system, the Data ONTAP architecture.

2 Database and Storage: Logical and Physical Layout

The combination of NetApp storage solutions and Microsoft SQL Server enables the creation of enterprise-level database storage designs that can meet today's most demanding application requirements.

To optimize both technologies, it is vital to understand the Microsoft SQL Server relational engine storage architecture. A well-designed layout of a SQL Server database is necessary for proper performance and management of the SQL Server infrastructure. A good database storage design effectively supports the business requirements defined for the database. This enables the initial deployment to be successful and the environment to grow smoothly over time as the business grows. This technical report discusses the SQL Server relational database storage features that can be used to help achieve that goal. When determining which features to implement in your designs, remember to keep the design as simple as possible while utilizing the appropriate features.

2.1 Aggregate

Aggregates are the lowest level in the storage stack. They are containers of physical disks from which volumes are carved out. Fewer large aggregates will maximize performance; however, they might not meet the data availability requirements set forth in the SLA. In SQL Server environments with multiple database copies, Microsoft no longer requires separating database and transaction log files to separate sets of disks. This means that database and transaction log volumes can be placed in the same aggregate.

NetApp recommends using one large aggregate for all SQL Server databases.

There are two reasons for this recommendation:

- One aggregate makes the I/O abilities of all spindles available to all files.
- One aggregate enables the most efficient use of disk space.

NetApp has performed various tests using both approaches, testing shared and dedicated aggregates with data and log separated as well as workload type separation (decision support system [DSS] and online transaction processing [OLTP]). The conclusion is that one large aggregate yields significant performance benefits and is easier for administrators to manage. Also, as physical disks become larger and larger, efficient space management using multiple aggregates becomes even more challenging. For example, significant disk space is wasted in an aggregate containing high-capacity drives dedicated just to a high-utilization log file.

The prevailing reason for using more than one aggregate is high availability, for example, one for data and another for logs. With more than one aggregate, if one of the aggregates fails (which is highly unlikely), the odds are increased that there will be less data loss because the other aggregate is still available.

For environments requiring extremely high availability, NetApp SyncMirror® software can be used to create and maintain a local mirror of the complete aggregate.

When creating and sizing aggregates, take the following into consideration:

- The total size of all the databases using the aggregate
- The I/O load generated by all users accessing all the databases that will share the same aggregate
- The projected storage space growth
- The projected user count growth
- Plans for adding new databases to the aggregate
- Any other nondatabase files that may use the same aggregate

When creating aggregates, let the NetApp storage system select which physical disks will be in each aggregate.

Best Practices

- Make sure your aggregates containing sensitive Microsoft SQL Server data are not larger than they need to be, which in turn affects the time, disk space, and bandwidth required for disk sanitization.
- NetApp recommends having at least 10% free space available in an aggregate hosting SQL Server data for optimal storage performance.
- While performing aggr move or vol move or cloning:
  - Make sure that the source and destination NetApp storage systems are running the same version of Data ONTAP.
  - Be sure to check the system health and SQL Server load before you initiate these move operations.
  - To prevent interference with cutover phase, do not schedule long-running data management or data protection operations. Also, LUN provisioning, backup and restore operations, and Snapshot™ copy management operations in SnapDrive for Windows (SDW) will be unsuccessful during this time.
  - Avoid starting any conflicting operations such as changing volume attribute settings or LUN attribute settings in the source volume.
- If you are creating a FlexClone® volume from a SnapMirror® destination volume and you are expanding the aggregates containing the source and destination volumes, expand both source and destination, and use a base Snapshot copy that was created after the source volume was expanded.

2.2 Volumes

NetApp FlexVol® volumes are created and reside inside aggregates. Many volumes can be created in a single aggregate, and each volume can be expanded or shrunk or moved between aggregates.

Volume Design Considerations

Before a database volume design can be created, the backup and recovery requirements must be defined. They provide the “specs” needed for the volume design process. Following are best practice considerations to apply during the volume design process:

- Place the SQL Server system databases on a dedicated volume to provide separation from the user databases.

- Place tempdb on a dedicated volume since it is write intensive. However, after careful planning, it can be consolidated into fewer volumes and stored in the same volume as other system databases in large environments where volume count will be a challenge.
- When using SnapManager for SQL Server, place the SnapInfo directory on a dedicated volume.
- Make sure the database residing on LUN within a volume is separate from that used by the SnapInfo volume to avoid stream-based backup.
- All the volumes created will contain a single qtree, which will contain a single LUN.
- It is common to take transaction log backups more frequently than database backups, so place the transaction log on a separate volume from the data files so independent backup schedules can be created for each. This also separates the random I/O of the data files from the sequential I/O to the log files and can improve SQL Server performance.
- If each database has a unique backup requirement:
  - Either create separate FlexVol volumes for each database and transaction log, or
  - Place databases with similar backup and recovery requirements on the same FlexVol volume. This can be a good option in cases of many small databases in a SQL Server instance.
- Storage administration overhead can increase as more volumes are used.

Create the SnapManager share to store log files in an availability group environment on dedicated FlexVol volumes. For complete details, refer to the SQL Server Books Online and the SnapManager 7.0 for SQL Server Installation and Administration Guide.

Snapshot Copies

SMSQL uses SnapDrive for Windows to take Snapshot copies at the volume level. With volume-level Snapshot copies, all the data in the volume is included in the Snapshot copy, even when only some of the data in the volume is pertinent to the specific SQL Server database being backed up. If a database uses multiple volumes, then Snapshot copies are made sequentially. However, if it uses multiple LUNs on the same volume, then all Snapshot copies of these LUNs are made simultaneously, because Snapshot copies are volume-based.

Best Practice
• When using availability groups, NetApp recommends placing each database in a separate volume with copies of the same database isolated in separate aggregates.

SnapMirror

SnapMirror also operates at the volume level as well as at the LUN/qtree level. All the data in a source volume is mirrored to the target volume. Using one volume per database provides the most granular control over SnapMirror frequency and also provides the most efficient use of bandwidth between the SnapMirror source and the destination volumes. When addressing HA implementations, it is more common to have one volume for logs and a separate volume for all the data LUNs.

Windows Volume Mount Points

NetApp storage solutions and Microsoft SQL Server 2005 onward support mount points. Mount points are directories on a volume that can be used to “mount” a different volume. Mounted volumes can be accessed by referencing the path of the mount point. Mount points eliminate the Windows 26-drive-letter limit and offer greater application transparency when moving data between LUNs, moving LUNs between hosts, and unmounting and mounting LUNs on the same host. This is because the underlying volumes can be moved around without changing the mount point path name.

NetApp SnapManager for SQL Server supports the use of mount points. For additional information, refer to the SnapManager 7.0 for SQL Server Installation and Administration Guide.

The following factors need to be considered for NetApp flexible volume capacity for SQL Server 2012:

- Individual database size
- Database workload type (OLTP, data warehouse, and so on)
- Database maintenance

Also, calculate disk space requirements to accommodate data growth, Snapshot copies, and space reservations.

Best Practices

- Use FlexVol volumes to store SQL Server database files and avoid sharing volumes between hosts.
- Disable opportunistic locking (oplocks) on volume qtree where SQL Server data is stored to avoid corruption due to caching.
- To make sure performance improvement for SQL Server:
  - Disable minimum read ahead (minra) on the volume where SQL Server data is stored.
  - Enable no updates of access times on inodes when a file is read (no_atime_update).
- Enable checking of NVRAM at controller boot (nvfail) to alert administrators to shut down databases if there is a problem with the NVRAM. This setting helps prevent SQL Server data corruption.
- Use NTFS mount points instead of drive letters to surpass the 26-drive-letter limitation in Windows.
- When using volume mount points, a general recommendation is to give the volume label the same name as the mount point name.
- Configure volume autosize policy, where appropriate, to help avoid out-of-space conditions.
- Enable read reallocation on the volume where the SQL Server database I/O profile consists of random writes and large sequential reads (read_realloc).
- Set Snapshot reserve in the volume to zero for ease of monitoring from an operation perspective:
  - Set the fractional reserve of the volume to zero to improve storage efficiency. Use this setting with volume autosize policies.
  - Disable storage Snapshot schedules and retention policies. Use external scripts to coordinate the Snapshot copy of the SQL Server data on the volumes.
- While creating volumes, we need to use "NTFS" security style for the volumes.

2.3 LUNs

SnapInfo Directory

The SnapInfo directory is the main repository of the SnapManager for SQL Server software. All the metadata related to the SMSQL instance, such SnapManager backup set metadata, streaming system database backups, and transaction log backups, installed on a host is kept in SnapInfo along with all transaction log backups taken through SMSQL. For a SQL Server host, there can be one or more SnapInfo directories. This is because SMSQL allows one or more SQL Server instances on a host to share a SnapInfo directory as well as each instance to have its own SnapInfo directory. This can be configured at any time by running the configuration wizard.

Note: For large production systems, either of two approaches can be followed to allocate SnapInfo directories:

- Give each highly active instance that sees a lot of backups and transactional activity and has greater criticality to the business operations a SnapInfo LUN and preferably volume of its own.

- For SQL Server instances in a production host that do not see too much activity and house noncritical databases, divide these instances into groups of two or three and give them a SnapInfo LUN of their own. Note that you should not group together instances that have databases with large TLOGs.

You can back up volumes containing LUNs to a backup vault. You can restore all LUNs in a volume, or you can restore a single LUN from the vault. You can also access the latest Snapshot copy of a LUN directly from the backup vault. This direct access is read-only.

Best Practices

- Ensure the snapinfo LUN does not host any other type of data.
- Use SnapDrive for Windows to create LUNs that are properly aligned on disk for the specific Windows file system that is used. Also, creating LUNs with SnapDrive for Windows guarantees that LUNs are of the correct type and with the correct disk offset, which is crucial for guaranteeing the best disk performance.
- NetApp does not recommend creating LUNs on the root volume of the Vserver.
- Do not place user databases or system databases on a LUN that hosts mount points.
- Each SQL Server instance must have its own SnapInfo LUN.
- The size of the SnapInfo LUN must be able to accommodate all of the SMSQL metadata, system database streaming backup files, and transaction log backup files generated over the SMSQL backup retention period. Additional space is also required to allow for any problems with SMSQL deleting old backups or an unplanned increase in transaction log backup data.
- For clustered instances of SQL Server:
  - The SnapInfo LUN must be a cluster disk resource in the same cluster group as the SQL Server instance being backed up by SnapManager.
  - Place user databases onto shared LUNs that are physical disk cluster resources assigned to the cluster group associated with the SQL Server instance.
- Separate the system and user databases to LUNs on different volumes to be able to leverage NetApp Snapshot technology during backup, rather than doing a stream-based copy of this data, which normally takes more time.
- Do not place user databases on the same LUN in which the SQL Server error logs are stored.
- Make sure the database and SnapInfo LUNs are on separate volumes to avoid retention policy from overwriting Snapshot copies when used with SnapVault® technology.
- Place one database per LUN for faster restores.
- Make sure the SQL Server databases reside on LUNs separate from those LUNs that have nondatabase files such as FullText search-related files.
- Placing database secondary files (as part of a file group) on separate LUNs might increase the performance of the SQL Server database. This is only valid if the databases' MDF file does not share its LUN with any other MDF files.
- If any database file groups share the same LUNs, the layout and number of secondary files must be identical in all databases.
- Make sure the name resolution of the storage system is set up correctly for SnapDrive for Windows to allow creating and displaying LUNs and SMSQL to identify these LUNs. Be sure to leave automatic Snapshot scheduling off as configured by SnapDrive for Windows.
- Each database copy of the same database must be placed in a separate LUN when using SQL Server 2012 availability groups.

Snapshot Writable LUNs or SIS Clone

Snapshot backups of SQL Server databases are read-only, point-in-time images. This protects the integrity of the Snapshot backups. In certain situations, SMSQL restores a LUN from a Snapshot copy for temporary read/write access (restored to an alternative location using a writable Snapshot copy[1]). Writable Snapshot copies are used during a verification process, when a DBA restores a database to an alternative location for recovering from operational mistakes, or in some cases when external archive backups take place. A restored writable Snapshot copy is for short-term use only.

Best Practice
Avoid creating Snapshot copies on volumes with active Snapshot writable[2] LUNs.

NetApp does not recommend creating a Snapshot copy of a volume that contains an active Snapshot copy because this creates a "busy Snapshot"[3] issue. Avoid scheduling SMSQL Snapshot copies when:

- Database verifications are running
- Archiving a LUN backed up by Snapshot to tape or other media

SMSQL now supports the verification of backup sets at SnapMirror and SnapVault destinations without breaking the relationships.

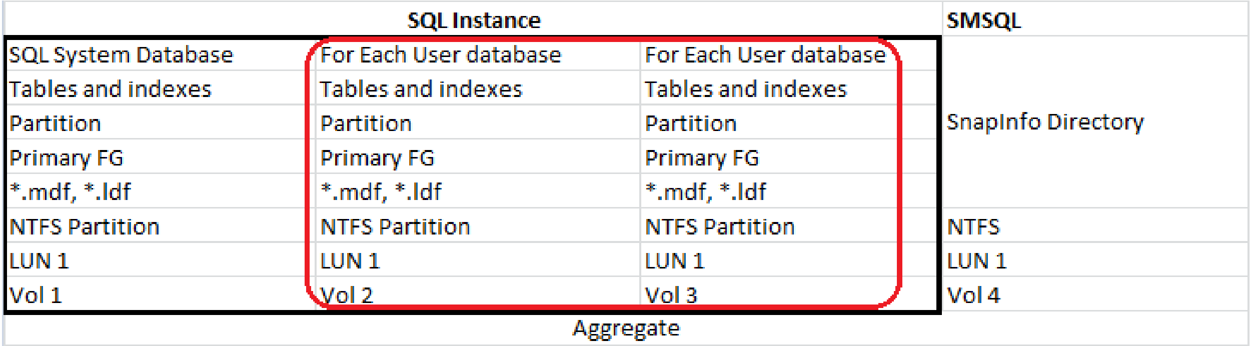
Database Storage Designs

Following are a few examples of SQL Server designs for NetApp storage and considerations of environments that use SnapManager for SQL Server.

Design Example 1: Basic

Figure 1 provides information about the basic SQL Server database design for a NetApp storage system.

Figure 1) Basic SQL Server database design for NetApp storage system.



	SQL Instance		SMSQL
SQL System Database	For Each User database	For Each User database	SnapInfo Directory
Tables and indexes	Tables and indexes	Tables and indexes	
Partition	Partition	Partition	
Primary FG	Primary FG	Primary FG	
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf, *.ldf	
NTFS Partition	NTFS Partition	NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1	LUN 1
Vol 1	Vol 2	Vol 3	Vol 4
Aggregate			

- It does not use SQL Server partitions beyond the default configuration.

[1] A writable Snapshot copy or Snapshot writable LUN is a LUN that has been restored from a Snapshot copy and enabled for writes. A file with .rws extension holds all writes to the Snapshot copy.

[2] There are two ways to determine whether you have a busy Snapshot copy: (1) View your Snapshot copies in FilerView®. (2) Use the following Data ONTAP command to list the busy Snapshot copies:

```
snap list usage VolumeName BusySnapshotName
```

[3] A Snapshot copy is in a busy state if there are any LUNs backed by data in that Snapshot copy. The Snapshot copy contains data that is used by the LUN. These LUNs can exist either in the active file system or in another Snapshot copy. For more information about how a Snapshot copy becomes busy, see the Block Access Management Guide for your version of Data ONTAP.

- There is one aggregate for the SQL Server instance.
- It uses a dedicated vol/LUN for the SQL Server system databases, including tempdb.
- It uses a dedicated vol/LUN for each user database.
- It uses a dedicated vol/LUN for the SMSQL SnapInfo directory.

This configuration can be scaled for multiple user-defined databases per instance by replicating the circled area. It can also be scaled for multiple SQL Server instances on the same server by replicating the highlighted box.

The same aggregate can be used as the user databases and the SQL Server instances are scaled out. In fact, the same aggregate can be used for all the SQL Server hosts connected to the storage system.

For SnapManager for SQL Server, there can be one SnapInfo directory for all the databases, which is implied in this design, or one per database. The SnapManager 7.0 for Microsoft SQL Server Installation and Administration Guide provides the information needed to determine what is appropriate for your specific design.

Design Example 2: Separate tempdb

The design example in Figure 2 provides the information about the SQL Server database design for NetApp storage systems: tempdb on dedicated volume.

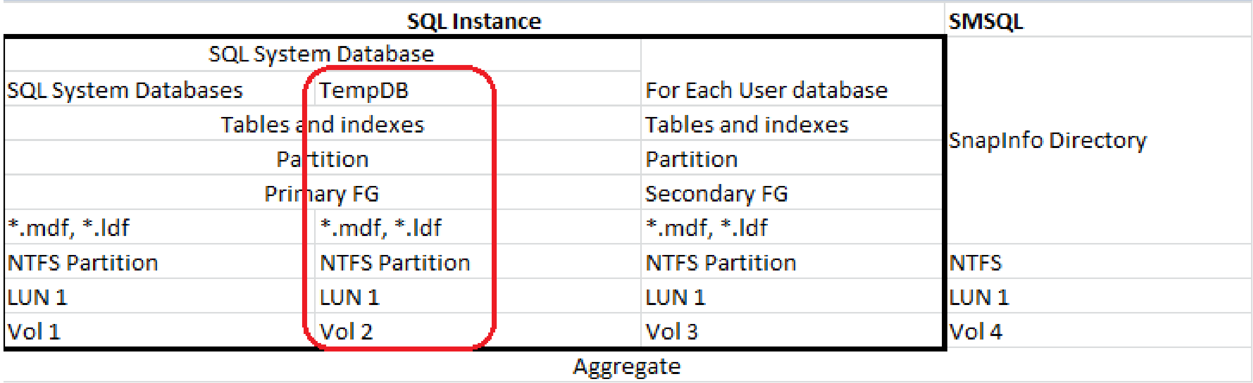
Figure 2 is identical to design example 1 in Figure 1 provides information about the basic SQL Server database design for a NetApp storage system.

Figure 1 except that tempdb has been placed on its own volume. Isolating tempdb onto its own volume makes it possible to keep it out of Snapshot copies. It also provides more granular control of which disks it resides on and of how many LUNs it is composed. Spreading tempdb across multiple LUNs can help improve its ability to handle higher I/O requirements.

After the storage is configured, tempdb can be moved to its own LUNs using the NetApp SnapManager for SQL Server configuration wizard or Microsoft SQL Server Management Studio.

Figure 2 provides the information about the SQL Server database design for NetApp storage systems: tempdb on dedicated volume.

Figure 2) SQL Server database design for NetApp storage systems: tempdb on dedicated volume.



SQL Instance			SMSQL
SQL System Database			SnapInfo Directory
SQL System Databases	TempDB	For Each User database	
Tables and indexes		Tables and indexes	
Partition		Partition	
Primary FG		Secondary FG	
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf, *.ldf	
NTFS Partition	NTFS Partition	NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1	LUN 1
Vol 1	Vol 2	Vol 3	Vol 4
Aggregate			

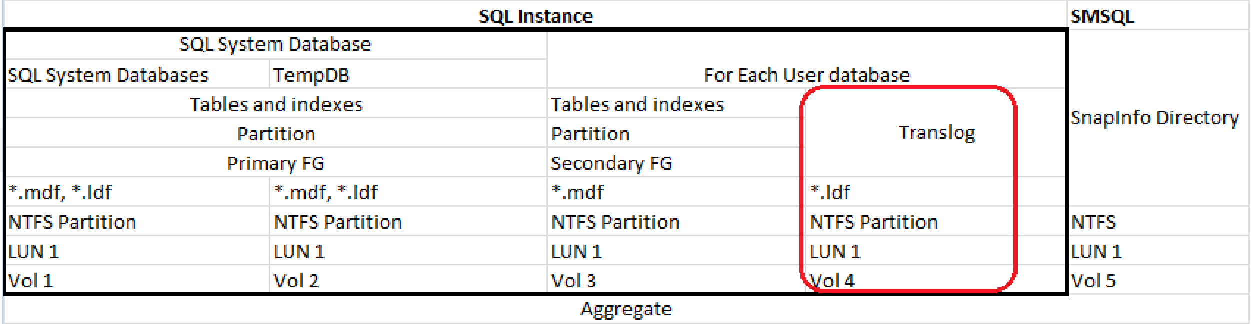
Design Example 3: Separate Transaction Log

The design shown in Figure 3 builds on the previous two designs. In this design, the user database log has been separated into its own volume. This provides more granular control of which disks it resides on and how many LUNs are included in it. Spreading the log across multiple LUNs can help improve its ability to handle higher I/O requirements.

This also allows the log file to be managed independently of the data files. Remember that Snapshot copies occur at the volume level. Using SnapManager for SQL Server, you schedule can create database Snapshot backups every hour, and a different schedule can create log backups every 10 minutes.

Figure 3 provides information about the SQL Server database design for NetApp storage systems: transaction logs on dedicated volume.

Figure 3) SQL Server database design for NetApp storage systems: transaction logs on dedicated volume.



SQL Instance				SMSQL
SQL System Database		For Each User database		SnapInfo Directory
SQL System Databases	TempDB			
Tables and indexes		Tables and indexes	Translog	
Partition		Partition		
Primary FG		Secondary FG		
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf	*.ldf	
NTFS Partition	NTFS Partition	NTFS Partition	NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1	LUN 1	LUN 1
Vol 1	Vol 2	Vol 3	Vol 4	Vol 5
Aggregate				

the least amount of space with little or no impact on the overall performance of the system. Storage efficiency goes beyond just data deduplication; it is a combination of RAID, provisioning (overall layout and utilization), mirroring, and other data protection technologies.

NetApp Technology for Storage Efficiency

The following NetApp technologies implement storage efficiency and reap its cost-savings benefits by optimizing existing storage in the infrastructure as well as deferring or avoiding future storage expenditures. The more these technologies are used in conjunction, the larger the savings.

3.1 Flash Accel

Flash Accel™ is a NetApp server caching software solution. Flash Accel is fully supported software technology that can turn a server-based PCI-e flash card or SSD drive into a server cache for Data ONTAP.

Best Practice
• Consistent Snapshot copies cannot be created with writeback caching enabled without flushing the cache contents to disk first. If you do not flush the contents to disk first, you will likely have an invalid Snapshot copy or backup set. Hence, make sure writeback caching is disabled when using it with SnapManager for SQL Server.

3.2 Flash Cache

To help improve the storage efficiency and read I/O performance and latency of SATA-based deployments, Flash Cache™ should be used. Flash Cache is a read cache that can be installed on certain models of NetApp storage controllers. Flash Cache enables fewer SATA disks to be used in SATA-based deployments, because a percentage of the SQL Server database dataset is cached in Flash Cache, thus greatly reducing the amount of read I/O on the SATA disk. NetApp recommends Flash Cache and SATA for deployments when SATA-based designs are bound by performance instead of capacity. Also, use Flash Cache with SQL Server workloads that are not write intensive and with random I/O.

3.3 Snapshot

NetApp Snapshot technology provides low-cost, fast-backup, point-in-time copies of the volume for an LUN that hosts the SQL Server databases, by preserving Data ONTAP architecture WAFL® consistency points.

There is no performance penalty for creating Snapshot copies, because data is never moved, as it is with other copy-out technologies. The cost for Snapshot copies is only at the rate of block-level changes, not 100% for each backup as with mirror copies. It also allows the database clones to be created near instantaneously, because no data is actually copied. It can result in savings in storage costs for backup and restore purposes and opens up a number of efficient data management possibilities.

Snapshot copies of the production SQL Server databases can be created with no downtime to the production database, due to the integration of NetApp SnapManager with Microsoft Volume Shadow Copy Service (VSS) technology, which is built into Windows Server.

3.4 Storage Thin Provisioning

NetApp thin provisioning deployed with NetApp FlexVol technology allows you to allocate storage on demand as data is written to disk, instead of preallocating all of the capacity ahead of time, thereby optimizing utilization of available storage. Thin provisioning eliminates almost all whitespace, which helps avoid poor utilization rates. FlexVol volumes (flexible volumes) are the enabling technology behind NetApp thin provisioning that can be thought of as the virtualization layer of Data ONTAP.

When storage consumption is unpredictable or highly volatile, it is best to reduce the level of storage overcommitment so that storage is available for any growth spikes. Consider limiting storage commitment to 100%—no overcommitment—and using the trending functionality to determine how much overcommitment is acceptable, if any. Overcommitment of storage must be carefully considered and managed for mission-critical applications in which even a minimal outage is not tolerable. In such a case, it is best to monitor storage consumption trends to determine how much overcommitment is acceptable, if any.

If the time required to procure new storage is very long, storage overcommitment thresholds should be adjusted accordingly. The overcommitment threshold should alert administrators early enough to allow new storage to be procured and installed.

The potential risk when configuring the SQL Server environment for thin provisioning is a LUN going offline when there is not enough space to write further data.

NetApp thin provisioning allows the storage to be purchased as the application grows, preventing the need to purchase multiple years of future storage up front. As the application's storage needs to grow, physical disk space can be added without taking the application offline or adversely affecting performance.

Best Practices

- When you enable NetApp thin-provisioned LUNs, NetApp recommends deploying these LUNs in FlexVol volumes that are also thin provisioned with a capacity that is 2x the size of the LUN. When a LUN is deployed in this manner, the FlexVol volume acts merely as quota. The storage consumed by the LUN is reported in FlexVol and its containing aggregate.
- NetApp recommends using thin-provisioned LUNs for maximum storage efficiency. However, when enabling NetApp thin provisioning, administrators should also configure storage management policies on the volumes that contain the thin-provisioned LUNs. The use of these policies aids in providing the thin-provisioned LUNs with storage capacity as they require it. The policies include automatic sizing of a volume, automatic NetApp Snapshot copy deletion, and LUN fractional reserve.
- Thin provisioning is recommended for both SQL Server database and transaction log volumes.
- When additional space is required, you can add more disks to the aggregates and provision storage to the user. For more efficient use of disk space in a SnapMirror configuration, use thin provisioning to overcommit aggregates, because SnapMirror requires the destination volume to be the same size as or greater than the source volume.

3.5 Space Guarantee

Space guarantee enables thin provisioning. The space guarantee option can be set at the volume level, depending on the requirements of SQL Server. If the space guarantee at the volume level is set to “volume,” the amount of space required by the FlexVol volume is always available from its aggregate. This means the space is subtracted, or reserved, from the aggregate’s available space at volume creation time. This is the default setting for FlexVol volumes.

If the space guarantee for the volume is set to “none,” the volume reserves no space from the aggregate during volume creation. Space is first taken from the aggregate when data is actually written to the volume. Write operations to space-reserved LUNs in a volume with guarantee=none will fail if the containing aggregate does not have enough available space. LUN reservation makes sure that the LUN has space in the volume, but setting “guarantee=none” does not make sure that the volume has space in the aggregate.

When the space guarantee for the volume is set to “file,” the aggregate makes sure that space is always available for overwrites to space-reserved LUNs.

3.6 Space Reclamation

Space reclamation integrated into NetApp SnapDrive for Windows helps you conserve and reuse disk space already allocated to LUNs. Use the Invoke-NAHostVolumeSpaceReclaim from the Data ONTAP PowerShell Toolkit for this purpose of space reclamation.

Best Practices

- It is a good practice to run space reclamation before creating a Snapshot copy. Otherwise, blocks that should be available for freeing will be locked in the Snapshot copy and not be able to be freed.
- Because space reclamation initially consumes cycles on the host, it should be run during periods of low activity.
- Normal data traffic to the LUN can continue while the space reclamation process runs. However, certain operations cannot be performed during the space reclamation process:
  - Creating or restoring a Snapshot copy stops space reclamation.
  - The LUN may not be deleted, disconnected, or expanded.
  - The mount point cannot be changed.
  - Running Windows defragmentation is not recommended.

3.7 Autodelete

This volume setting allows Data ONTAP to automatically delete Snapshot copies when a threshold/trigger such as the following is met:

- **Volume.** The volume is nearly full, reported in the first line for each volume by the `df` command. Note that the volume can be full even though there might still be space in the snap_reserve areas.
- **Snap_reserve.** The snap_reserve space is nearly full.
- **Space_reserve**. The overwrite reserved space is full. This is the space determined by the LUNs with space reservations enabled and the fractional_reserve option. The reserve space will never be filled until both the volume and the snap_reserve areas are full.

Best Practices

- When autodelete is used, NetApp recommends that the retention functionality in SMSQL is used rather than the Data ONTAP autodelete feature. If not, SMSQL will not delete the Snapshot copies based on the retention defined within the backup wizard of SMSQL.
- Autodelete works at the volume level and not on individual LUNs. This means that LUNs will not automatically grow and must be handled separately using NetApp SnapDrive for Windows (SDW).
- NetApp recommends not using the autodelete option in NAS environments. Keeping a certain amount of space for Snapshot copies for file versioning or file restores is part of the SLAs defined for file services.
- NetApp recommends setting snap reserve to 0 for SQL Server in SAN environments (where the SQL Server databases resides on LUNs) because it simplifies space management, allowing maximum usable volume space by either the LUNs or the Snapshot copies within the volume. It is advised not to keep snap reserve to the default value of 20% because user writes are already limited by the LUN size.
- When using SnapMirror or SnapVault technology to replicate a SQL Server database, NetApp recommends not using the “disrupt” option for commitment, because SnapMirror baseline Snapshot copies can be destroyed by autodelete. In many configurations, deleting the last SnapMirror Snapshot copy is not desirable because a new full baseline copy is required to resume mirroring operations. For example, if the source and destination are at different sites, recreating this baseline can be a time-consuming and costly process.

3.8 Autosize

This volume setting for FlexVol volumes defines whether a volume should automatically grow to avoid filling up to capacity. It is possible to define how quickly the volume should grow by using the “-i” option.

The default growth increment is 5% of the volume size at creation. It is also possible to define how large the volume is allowed to grow by using the -m option. If volume autosize is enabled, the default maximum size to grow to is 120% of the original volume size.

Best Practices
• Implement NetApp Data ONTAP volume autosize policies as appropriate to the environment to help make sure of efficient use of storage capacity. • NetApp recommends planning for additional buffer space in the aggregate containing the volume to allow for the expansion of the volume used for storing SQL Server data when implementing volume autosize policies. • AutoSize works at the volume level and not on individual LUNs. This means that LUNs will not automatically grow and must be handled using NetApp SnapDrive for Windows (SDW). • NetApp recommends prioritizing autosize over autodelete because deletions occur at the Data ONTAP level, and it is possible to have a backup set of a transaction log and database where one of the Snapshot copies has been automatically deleted or orphaned. • If volume autosize and/or Snapshot autodelete policies are not implemented on the volume that stores the SQL Server data, it is recommended to keep 20% free space in the volume to help avoid an out-of-space condition.

3.9 Fractional_Reserve

Fractional_reserve is a volume option that specifies how much space Data ONTAP reserves for Snapshot overwrite after all other space in the volume is used, can be set between 0% and 100%.

Best Practices
• Exercise caution when changing the fractional reserve value because when space is fully consumed, the write operations will fail and disrupt the SQL Server environment. • Do not modify fractional reserve: – Unless there is a mechanism to monitor fractional reserve or volume and aggregate available space. SnapDrive for Windows does not provide this functionality. – If there are multiple LUNs in a volume and each LUN has a different rate of change, estimation must be made of the overall volume size and the combined fractional reserve setting based on the average rate of change of all the LUNs. • The threshold value for the fractional space policy “deletion of backup sets” should always be less than that of “dismount databases.” • When a LUN is fully space reserved (fractional space reservation set to 100%), write operations to that LUN are protected against failure caused by an out-of-space condition due to Snapshot copy disk space consumption. If you do require a fractional space reservation policy, work with your NetApp support representative. The representative can help implement a fractional space reservation policy that fits your environment.

Best Practice Configurations When Using Thin Provisioning for SQL Server Environments

There are many ways to configure the NetApp storage appliance for LUN thin provisioning; each has advantages and disadvantages. It should be noted that it is possible to have thinly provisioned volumes and non–thinly provisioned volumes on the same storage system or even the same aggregate. The following are considered to be best practice configurations when using thin provisioning for Microsoft SQL Server.

Option 1: Volume Guarantee Set to “None”

Volume guarantee	= none
LUN reservation	= enabled
fractional_reserve	= 0%
snap_reserve	= 0%
autodelete	= volume / oldest_first
autosize	= off
try_first	= snap_delete

This configuration has the advantage of the free space in the aggregate being used as a shared pool of free space. The disadvantages of this configuration are the high level of dependency between volumes and that the level of thin provisioning cannot easily be tuned on an individual volume basis. When using this configuration, the total size of the volumes is greater than the actual storage available in the host aggregate. With this configuration, storage administrators can generally size the volume so that they only need to manage and monitor the used space in the aggregate. This option does not affect the space for hosting the live data, but rather allows the backup space to dynamically change.

Option 2: Using Autogrow/Autodelete

Volume guarantee	= volume
LUN reservation	= disabled
fractional_reserve	= 0%
snap_reserve	= 0%
autodelete	= volume / oldest_first
autosize	= on
try_first	= autogrow

This configuration allows the administrator to finely tune the level of thin provisioning for Microsoft SQL Server environments. With this configuration, the volume size defines or guarantees an amount of space that is only available to LUNs within that volume. The aggregate provides a shared storage pool of available space for all the volumes contained within it. If the LUNs or Snapshot copies require more space than available in the volume, the volumes will automatically grow, taking more space from the containing aggregate. Additionally, the advantage of having the LUN space reservation disabled is that Snapshot copies can then use the space that is not needed by the LUNs. The LUNs themselves are also not in danger of running out of space because the autodelete feature will remove the Snapshot copies consuming space.

Note: Snapshot copies used for creating FlexClone volumes will not be deleted by the autodelete option.

Best Practice

NetApp recommends using autogrow for most common deployment configurations.

Monitoring

When using NetApp efficiency features, the volumes should be appropriately sized so that autosize and/or autodelete policies are not triggered unless there is an abnormal rate of change or a problem with Snapshot copy retention. NetApp OnCommand® Unified Manager Core Package management software that includes Operations Manager is the recommended tool to monitor SQL Server volumes for these events and to send notifications to the storage administration team to follow up further with the SQL Server administration team. SNMP can also be used to monitor these events.

After a notification for a volume autogrow or Snapshot autodelete event has been received by the storage administration team, the recommended action is for the storage administration team to examine the affected storage controllers and then follow up with the SQL Server administration team for further administrative actions.

A typical cause of volume autosize events is that the rate of change greatly surpassed the rate of change assumption used in sizing the volume. Typically, OLTP workloads experience increased data change rates. Another cause for volume autosize events is that older Snapshot copies created by SnapManager for SQL Server are not being deleted. As Snapshot copies age, they can grow in size and consume more capacity than originally allocated in the volume.

Reasons for SnapManager for SQL Server not deleting backups are:

- SMSQL backups are failing.
- SMSQL backup retention policies are not being enforced correctly because Snapshot copies were manually removed outside of SnapManager for SQL Server on the controller itself.

Monitoring the health of SnapManager for SQL Server can be done by monitoring for SnapManager for SQL Server event IDs. To monitor the health of SnapManager retention, use Windows PowerShell® commands from the Data ONTAP PowerShell Toolkit and native Windows PowerShell commands.

3.10 NetApp FlexClone

A FlexClone volume is a writable point-in-time Snapshot copy of a FlexVol volume or another FlexClone volume. FlexClone uses space very efficiently, leveraging the Data ONTAP architecture to store only data that changes between the parent and the clone. FlexClone volumes are great for any situation in which testing or development occurs, any situation in which progress is made by locking in incremental improvements, and any situation in which there is a desire to distribute data in changeable form without endangering the integrity of the original. A common scenario is to use FlexClone in an environment before committing a Microsoft SQL Server rollup or hotfix into production.

FlexClone technology can be leveraged both at the primary storage system and at the SnapMirror destinations for effective utilization of resources. FlexClone can also be used for disaster recovery testing without affecting the operational continuity of the Microsoft SQL Server environment.

For detailed information on how FlexClone works and on command line references, refer to the Storage Management Guide.

Best Practice
Use SnapDrive to connect to the required Snapshot copy. This will automatically execute the FlexClone operation.

3.11 NetApp Deduplication

NetApp deduplication is a data compression technique for eliminating coarse-grained redundant data, typically to improve storage utilization. NetApp deduplication combines the benefits of granularity, performance, and resiliency with a significant advantage in the race to improve storage utilization demands. The deduplication process stores only unique blocks of data in the volume and creates additional metadata in the process.

When deduplication runs for the first time on a FlexVol volume with existing data, it scans the blocks in the volume and creates a fingerprint database, which contains a sorted list of all fingerprints for used blocks in the volume. Each 4k block in the storage system has a digital fingerprint, which is compared to other fingerprints in the volume. If two fingerprints are found to be the same, a byte-for-byte comparison is done of all bytes in the block. If they are an exact match, the duplicate block is discarded, and the space is reclaimed. The core enabling technology of deduplication is fingerprints.

Deduplication consumes system resources and can alter the data layout on disk. Due to the application I/O pattern and the effect of deduplication on the data layout, the read and write I/O performance can vary.

Note: Deduplication is transparent to SQL Server that does not recognize the block changes, so the SQL Server database remains unchanged in size from the host, even though there are capacity savings at the volume level. Data compression and deduplication can provide significant space savings, but proper testing should be done to determine the savings for your environment.

Note: Adequate space must be available on the FlexVol volume for the `sis on` command to complete successfully. If the `sis on` command is attempted on a FlexVol volume that already has data and is completely full, it fails because there is no space to create the required metadata.

There is a limit on the maximum size of the volume for deduplication, because deduplication depends primarily on the amount of system memory, which varies based on the storage platform. While considering using deduplication in SQL Server environments, be sure to consider this factor when sizing the volume layout.

Best Practices
• Run deduplication before creating new Snapshot copies. • Schedule deduplication only after significant new data has been written to the volume. • Configure appropriate reserve space for the Snapshot copies.

4 NetApp Solution for Microsoft SQL Server

NetApp storage software and tools:

- **NetApp Windows Host Utilities Kit.** Installation of the Host Utilities Kit sets timeout and other operating system–specific values to their recommended defaults. It includes utilities for examining LUNs provided by NetApp storage. This kit should be used in Windows Server 2008 R2 (both physical and virtual environments) because it helps to align the master boot record for the Microsoft VHD file layout, preventing it from getting out of alignment with the underlying NetApp LUN. This is very important for optimal I/O performance. However, this is also no longer the case in Windows Server 2012. The new VHDX format stays aligned with the 4k blocks in WAFL, even when it is a dynamically expanding disk. This kit is not necessary on Windows Server 2012 when Data ONTAP DSM is installed.

Note: SMSQL does not support SQL databases on Hyper-V virtual disks.

- **Microsoft Windows and native multipath I/O (MPIO).** Microsoft MPIO is a framework provided by Microsoft for developing multipath solutions that contain hardware-specific information required to enhance connectivity for storage arrays. To operate as intended, Data ONTAP requires MPIO and ALUA.

  When using the iSCSI protocol, it is necessary to configure multipath support on iSCSI devices in the MPIO Properties administrative application. Navigate to the Discover Multi-Paths pane, select the “Add support for iSCSI devices” checkbox, and click Add. It is also necessary to create multiple sessions from the host initiators to the target iSCSI Interfaces on the Data ONTAP system operating in 7-Mode. This can be accomplished using the native iSCSI initiator: Select the “Enable multi-path” checkbox when logging on to a target.

Sessions can also be managed by using the NetApp SnapDrive iSCSI management pane. This is the preferred method, because SnapDrive remembers which target logical interfaces already have an established session and preselects an unused target portal.

- **Data ONTAP DSM.** The Data ONTAP DSM supports attaching to Data ONTAP operating in 7-Mode beginning with version 3.5. Consult the Interoperability Matrix Tool for current information on supported configurations.

  Use the Data ONTAP DSM over the native MPIO implementation by Windows 2008 and Windows 2012 to achieve optimal performance and advanced path decisions in the NetApp DSMs and also provide for autoconfiguration, heuristics for specific storage arrays, statistical analysis, and integrated management.

  During installation, the Data ONTAP DSM sets a number of Windows registry values to optimize performance and provide correct behavior during the failover scenarios.

5 Back Up and Restore Databases Using SnapManager 7.0 for SQL Server

This release of SnapManager 7.0 for SQL Server (SMSQL 7.0) aligns with Data ONTAP 8.2. The main launch themes are:

- **Restore to alternate location.** This facilitates NetApp backups to be restored to a different path and file name.
- **Sub-LUN clone.** When multiple databases share the same LUN, a file-based or streamed restore will occur.

SnapManager for SQL Server leverages NetApp Snapshot technology as a means to protect data, which greatly reduces the time it takes to back up large SQL Server instances and does not adversely influence database performance with optimum use of the storage system by tracking only the changed blocks of data compared to traditional backup with SQL Server BACKUP statements. SnapManager backup performs backups at the volume level.

Use the SMSQL configuration wizard to make sure the SQL Server databases are migrated to a NetApp LUN to be backed up using SnapManager for SQL Server.

When planning the backup and restore process, the methodology or considerations are different from considerations when backup is done by Enterprise Manager or SQL Server Management Studio.

The two main considerations are:

- **Recovery point objective (RPO).** To what point in time must the data be recovered?
- **Recovery time objective (RTO).** How long will it take to get the database back online and rolled forward or backward to the RPO?

5.1 Backup

During SMSQL backup:

- Snapshot copies are created of all volumes used by the SQL Server databases being backed up.
- User databases placed in a LUN that also contains system databases are backed up using stream-based backup.
- Backup of transaction logs always uses stream-based backup to provide point-in-time restores. Transaction log backup is functionally the same using SMSQL as the transaction log backup done using BACKUP log statements and is streamed out to the SnapInfo directory for safekeeping. The length of time required to back up a transaction log depends mainly on the active section of the transaction log that has to be extracted and copied to the dump directory.

How to back up and restore SQL Server databases is thoroughly discussed in the SnapManager 7.0 for SQL Server (SMSQL) Installation and Administration Guide.

Backup Considerations

There are several considerations to keep in mind regarding the SMSQL backup process:

Note that there are several factors that affect the time needed to complete the Snapshot copy operation.

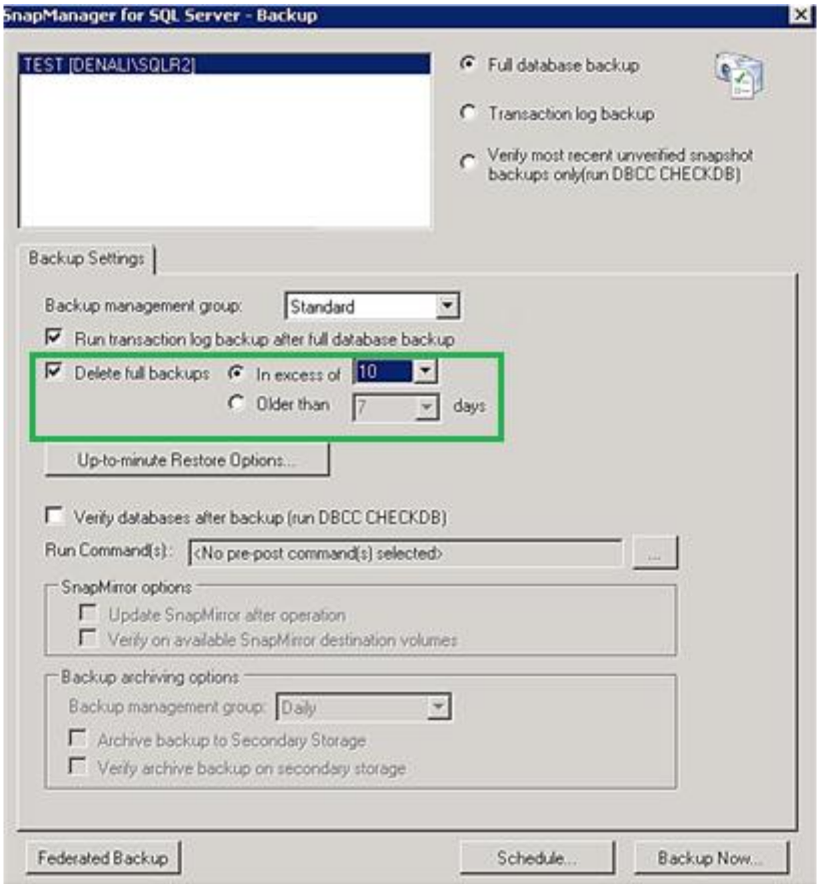
- Size of the volumes on which the database LUNs reside and the rate of change in them. Note that SMSQL Snapshot copies volumes serially, so if a database is spread across a large number of volumes, the copying takes more time with multiple Snapshot copies.
- Number of databases sharing the same volume, even though the databases might be in different LUNs on the same volume. This is because NetApp Snapshot copying is volume wide and so also is the Snapshot copy granularity from a VDI standpoint.
- How busy the storage controller is in terms of CPU usage, IOPS bandwidth, and so on.
- Network delays. These could cause delays in the acknowledgement from the storage controller reaching SMSQL and then SQL Server. Until the acknowledgement occurs, SQL Server will keep the databases frozen.
- When a Snapshot copy of a LUN is taken, or VMDK for a SnapManager backup, the entire volume is captured in that Snapshot copy. However, that backup is valid only for that server. If data from other servers resides on the same volume, it is not restorable from that Snapshot copy.
- In a Microsoft SQL Server environment, you should perform backups using only the SnapManager application. Making Snapshot copies of the storage system from the storage console directly is not supported and results in an inconsistent Snapshot copy image of online databases. However, you can use SnapDrive for Windows to make Snapshot copies of SQL Server databases, although you cannot restore these Snapshot copies using SMSQL.
- SMSQL supports both synchronous and asynchronous mirroring, which means that you can mirror the database volumes in any manner without any effect on SMSQL backups. If the database volumes are synchronously mirrored, then there is no need to select the Update SnapMirror option for the backup plan. Refer to the section "How SnapManager Uses SnapMirror" of the SnapManager 7.0 for SQL Server (SMSQL) Installation and Administration Guide.

Best Practices

- When cloning FlexVol volumes, make sure there is sufficient space for the clone metadata depending on the technology used (FlexClone or SISClone).
- Tempdb should not be included in a backup because the data it contains is temporary. Place tempdb on a LUN that is in a storage system volume where Snapshot copies will not be created; otherwise, large amounts of valuable Snapshot space could be consumed.
- The database and transaction log files for a database must be placed in the same CIFS volume for a valid backup of the database to be created.
- While doing a federated backup, make sure both SAN and NAS databases from same host are not included in a single federated group. However, SAN and NAS databases from different hosts are supported in a single federated group. Also, SAN and NAS in different federated groups from the same host are supported.

Backup Retention Management

This is an efficient way to manage the retention of transaction log backups, because we need to limit the number of old transaction logs we save. When the new backup runs, it marks the previous backups as being point-in-time only, then deletes the logs as well as the Snapshot copies to reduce log traffic.



SnapManager for SQL Server - Backup

TEST (DENALI\SQLR2)

● Full database backup

○ Transaction log backup

○ Verify most recent unverified snapshot backups only(run DBCC CHECKDB)

Backup Settings

Backup management group: Standard

☑ Run transaction log backup after full database backup

☑ Delete full backups ● In excess of 10

○ Older than 7 days

☐ Verify databases after backup (run DBCC CHECKDB)

Run Command(s): <No pre-post command(s) selected>

SnapMirror options

☐ Update SnapMirror after operation

☐ Verify on available SnapMirror destination volumes

Backup archiving options

Backup management group: Daily

☐ Archive backup to Secondary Storage

☐ Verify archive backup on secondary storage

Backup Grouping Performance

This feature groups databases that share storage into larger groups to be processed as a single backup set (or subjob) of the entire backup job. The one rule of backup that must be understood is that Snapshot copies are currently always made at the storage volume level, and the volume can have many qtrees, LUNs, and, of course, file system objects.

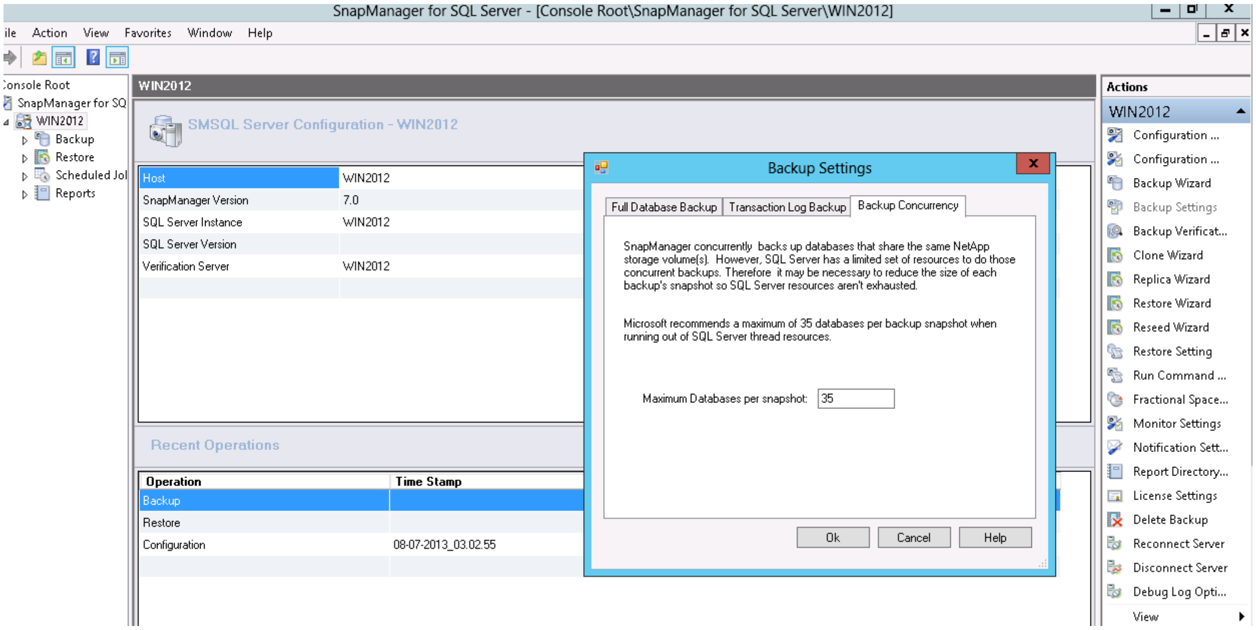
The advantages of the grouping:

- This new grouping by default reduces the per-group overhead (of VDI, NetApp, Windows, and so on) for each backup subjob.
- This also allows more NetApp volumes to be copied by Snapshot concurrently with increased parallel processing.

Default Limits

- Maximum databases per backup set = 35.

This feature groups databases into a minimum of 35 databases in a backup set. If each database is located on a separate NetApp volume, then after 35 databases it will commit the backup set to execute. If the combined number of databases exceeds the maximum, this feature will commit the set as a backup set for the overall current backup.



Host	WIN2012
SnapManager Version	7.0
SQL Server Instance	WIN2012
SQL Server Version	
Verification Server	WIN2012

Operation	Time Stamp
Backup	
Restore	
Configuration	08-07-2013_03.02.55

SnapManager concurrently backs up databases that share the same NetApp storage volume(s). However, SQL Server has a limited set of resources to do those concurrent backups. Therefore it may be necessary to reduce the size of each backup's snapshot so SQL Server resources aren't exhausted.

Microsoft recommends a maximum of 35 databases per backup snapshot when running out of SQL Server thread resources.

Maximum Databases per snapshot: 35

Remote Backup Verification with SMSQL

SMSQL has an important feature with respect to backup verification. It now supports the verification of backup sets at SnapMirror and SnapVault destinations without breaking the relationships.

Best Practices

- The architecture and version of SQL Server on the database verification server must match the architecture and version of SQL Server being backed up by SMSQL.
- The verification server must be connected to the NetApp storage system where the backup Snapshot copies are located. The connectivity can be Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), or iSCSI. The connectivity of the verification server to the NetApp storage does not have to match the connectivity of the production SQL Server.
- If there are OLTP databases on the host in a production environment, NetApp recommends setting a remote server for verification to offload the resource impact of verification.
- To use a remote verification, configure the transport protocol setting in SDW on that remote server to be able to understand on which controller the LUN reside.
- If the SMSQL backup set scheduled for verification is stored on primary storage, schedule SMSQL backup verification during times when the NetApp storage utilization is low. Scheduling verification during off-peak times reduces the impact of verification on other SQL Server workloads on the NetApp storage system.
- License NetApp FlexClone on the primary NetApp storage system in order to avoid busy Snapshot copies during verification.
- License FlexClone on the secondary NetApp storage system (SnapMirror destination or SnapVault destination) to allow verification to be performed on the secondary NetApp storage system. Performing verification on the secondary NetApp storage system allows the verification workload to be offloaded from the primary NetApp storage system.
- In order to mount the backup for verification, SMSQL first asks SDW to create FlexClone clones of the SnapMirror destinations and mount the LUNs inside them to the verification server. Hence, make sure the aggregate on the storage system has sufficient space.
- The remote verification server should be a nonproduction SQL Server host. For environments with large databases, it is preferable to have one dedicated SQL Server host as the verification server for each SMSQL instance.
- It is best to use a shared mount point LUN to set verification for a SQL Server failover cluster instance.

5.2 Restore

In the earlier SMSQL releases, a restore operation could only be performed to the same database. The user could opt for a different database name but could not perform restore to a different path and different file names.

SMSQL 7.0 can now restore a database to a new location (a LUN) using the quick dialog and restore wizard. Restore to alternate path feature in SMSQL provides the new way to restore database to different path and file names. User is able to select different storage LUNs and different file names for convenience. Restore to alternate path allows user to restore database with different name. This restore feature validates existing configuration and allows user to perform backup on restored database.

Performing Table-Level Restores with SMSQL

Restoring particular tables from within backup sets is a common requirement in many database environments. In the case of SQL Server, this requirement becomes more challenging because of the lack of support for table-level restores from within SQL Server irrespective of the version. However, SMSQL provides an effective solution in this space with the use of the database cloning feature.

Best Practices

- When databases are on their own LUN, restore tends to be faster in cases when the entire database is being restored. This is because SMSQL can take advantage of LUN clone split restore (LCSR) for much faster restores.
- When multiple databases share the same LUN, a file-based or streamed restore will occur, which normally takes more time. For this reason, NetApp recommends keeping the size of the databases that share the same LUN to ~50GB–80GB.
- When restoring to alternate location (either LUN), make sure there is sufficient disk space on the destination volume to be able to successfully restore these SQL Server databases. If the operation fails and the database does not get attached to SQL Server, then the user has to explicitly clean the partially copied database files.
- "Restore to other location" will work only to the same instance as the backup and not to a different SQL Server instance. Also, this restore to other location feature is not supported for availability groups.
- Always run the reseed wizard on the secondary replica where there is a failure to recover the failed or corrupted database of the availability group and resync with primary databases.

5.3 Clone

Database cloning is the process of creating a point-in-time copy of a production database or its backup set.

Note: NetApp FlexClone technology needs to be licensed to be able to create a database clone.

Note: If a database resides on a virtual machine with a VMDK disk, it is not possible to clone the database to a physical server.

Best Practices

- Make sure either FlexClone or sis clone is licensed; this is mandatory.
- Use the clone wizard because it provides you with a complete set of cloning options, unlike using cloning using the Actions pane in SnapManager restore, which gives you quick cloning with fewer options than the clone wizard.

5.4 Reseed Availability Group

SQL Server 2012 availability groups are a logical grouping of databases that can fail over as a single unit to secondary copies (synchronous and async). SMSQL helps restore and reseed functionalities to facilitate faster recovery of availability group databases and to restore them to a synchronized state. In an AlwaysOn environment, AG failover can occur whenever there is a failure or corruption of one or more AG databases. This failover can be automatic or manual, based on the availability replica's configuration. In such cases, the primary replica switches to the secondary role, and one of the secondary replicas becomes the primary.

Best Practices: Reseed Wizard

- Always run the reseed wizard on the secondary replica where there is a failure after failover.
- For the reseed wizard to be successful, a SnapManager share should be configured for all the replicas that should have read/write permission for SnapManager for SQL Server to store the log files backups.

6 Virtualization

6.1 Hyper-V

The improvements in Windows Server 2012 Hyper-V and its expanded virtual machine capabilities have eliminated most of the limits related to performance in a virtualized environment. Windows Server 2012 Hyper-V provides better consolidation of workloads that are traditionally more complex and that tend to saturate resources and contend for other system resources and storage. Microsoft SQL Server 2012 and Windows Server 2012 provide a host of new features that can be used to effectively virtualize demanding complex database workloads such as online transaction processing/analysis (OLTP/OLTA), data warehousing (DW), and business intelligence (BI), which previously were not considered for virtualization. The documentation for support for SQL Server 2012 in virtualized environments can be found in the Microsoft TechNet.

Here is a high-level list of some important considerations:

- Carefully consider the intended SQL Server 2012 workloads and their requirements when planning for hardware resources to make sure of reasonable response time or I/O usage for highly intensive workloads.
- Run Windows Server 2012 Hyper-V on processors that support second-level address translation (SLAT) technologies, which add a second level of paging functionality under the paging tables of x86/x64 processors. They provide an indirection layer that maps virtual machine memory addresses to physical memory addresses, which reduces load on the hypervisor for address translation. This offers the additional benefits of improved performance of demanding SQL Server workloads, more virtual machine density per host machine, and reduced overhead.
- When configuring the Hyper-V "root" partition, dedicate the "root" (or parent) partition only to Hyper-V. No other roles or applications should be installed on this partition. The root partition must have sufficient memory to provide services such as input/output (I/O) virtualization, virtual machine Snapshot, and management to support the child partitions. Hyper-V calculates an amount of memory known as the root reserve that is guaranteed to be available to the root partition. This memory is never assigned to virtual machines. Root reserve is calculated automatically, based on the host's physical memory and system architecture.
- The recommendation is to have one SQL Server instance per VM. However, if multiple instances of SQL Server are installed on a single VM, consider setting SQL Server processor Affinity Mask in some cases.
- Virtual machines with multiple virtual processors have additional overhead related to synchronization costs in guest operating systems. Therefore, if a virtual machine will never have CPU-intensive SQL Server loads, even at peak hours, configure it to use only one virtual processor. Multiple virtual processors should only be configured in cases where the virtual machine requires more processing power under peak loads.
- Windows Server 2012 Hyper-V also provides the weights and reserves feature, which can be a great tuning mechanism when used properly. If there is higher demand for CPU than is physically available, Hyper-V makes sure that a virtual machine needing CPU resources gets at least its CPU reserve when there is contention.
  - Weights are assigned to a virtual processor to grant it a larger or smaller share of CPU cycles than the average cycle share.
  - Reserves are set for a virtual processor to make sure that it gets at least a specified percentage of the total possible CPU usage of a virtual machine when there is contention for CPU resources.

6.2 VMware

VMware® ESX® supports three types of configuration when connecting to shared storage arrays: raw device mapping (RDM), NFS datastore, and VMFS datastore. In virtualized SQL Server deployments using VMDK/NFS configurations, you must have a dedicated management ESX or ESXi™ server with Virtual Storage Console (VSC) and the appropriate vCenter™ software installed on it. VSC provides the integration between VMware and NetApp Snapshot copies to provide the ability to create backups of SQL Server data stored in a VMware virtual disk (vmdk) when used with SnapManager for SQL Server. Each ESX server should have a service console port defined on the vSwitch that transmits public virtual machine traffic and on the vSwitch configured for IP storage traffic.

RDM

Best Practice

Do not enter ESX or VC credentials unless you are using RDM LUNs on the virtual machine.

NFS Datastore

ESX 5.0 or higher version supports up to 256 numbers of NFS datastores. The default value is 8 and can be increased to a maximum specific to the version of ESX or ESXi.

Best Practices

- Each NFS export to be used as a NFS datastore should reside in its own volume.
- The best practice is to have the following layout for NFS datastore:
  - One NFS datastore for system database VMDK
  - One NFS datastore for user database/user log or separate the user database and user log on separate NFS datastore
- Do not define a default gateway for the NFS storage network.
- Each NFS datastore should be connected only once from each ESX or ESXi server using the same NetApp target IP address on each ESX or ESXi server.

VMFS Datastore

Best Practices

- Have separate VMDKs for primary (mdf) and log (ldf) files for user databases. However, ensure these VMDKs reside in a datastore on a separate volume from system databases and the operating system VMDKs.
- Have separate VMDKs for system databases (master, model, and msdb). Ensure these VMDKs reside in a datastore on a separate volume from user databases and the operating system VMDKs.
- Have separate VMDKs for temp db (can be in separate datastore also).
- Do not mix or keep system databases and user databases in the same VMDK. Also do not keep system databases and user databases on separate VMDKs in the same datastore and volume.
- Use NetApp Virtual Storage Console (VSC) to provision vmdk to install SQL Server binaries.
- User databases should be created directly on the NetApp VSC provisioned VMDKs.
- Data files (tables and indexes) are the primary files that are used by the SQL Server storage engine. Each database might have multiple files and be spread across multiple vmdks.
- Avoid sharing volumes/datastores between different Windows host machines.

To make sure of VMware configuration compatibility across the NetApp stack, refer to the Interoperability Matrix Tool (IMT).

7 High Availability and Disaster Recovery

The demand for higher availability and disaster recovery for database systems is creating a need to have geographically dispersed databases. This means that it is imperative not only to understand where all the critical and sensitive information resides, but also to make sure it is backed up consistently and securely for a timely recovery.

Prior to SQL Server 2012, SQL Server had several high-availability and disaster recovery solutions for an enterprise's mission-critical databases such as failover clustering, database mirroring, log shipping, or combinations of these. Each solution typically has a major limitation. In the case of failover clustering, for example, its configuration is very tedious and complex, and you arguably have single shared storage or a single point of failure. Database mirroring is relatively easy to configure in comparison with failover clustering, but you can have only one database in a single mirroring setup, and you cannot read from the mirrored database. Log shipping does not provide automatic failover (higher availability); though it can be used for disaster recovery with some expected data loss.

HA and DR Features Supported by Editions for Each SQL Server Version

Not every edition of SQL Server supports all of the solutions that are mentioned in the preceding section. It is important to understand which edition of each SQL Server version supports the solution that you are going to use.

Table 1 lists the SQL Server version and features supported by edition links.

Table 1) SQL Server version and features supported by edition links.

SQL Server Versions	Features Supported by Editions Links
2005	http://msdn.microsoft.com/en-us/library/ms143761(v=sql.90).aspx
2008	http://technet.microsoft.com/en-us/library/cc645993(v=sql.100).aspx
2008 R2	http://technet.microsoft.com/en-us/library/cc645993(v=sql.105).aspx
2012	http://technet.microsoft.com/en-us/library/cc645993.aspx

Table 2 lists the advantages and disadvantages of each solution.

Table 2) Advantages and disadvantages of each solution.

Solution	Advantages	Disadvantages
Backup and restore feature	• You can back the database up to removable media to help protect against disk failures. • You do not have to depend on the network as you do when you use failover clustering or log shipping. • Full and log backup can be performed quickly with SMSQL and transported to other data center with SnapMirror.	• When you back up the database, you cannot perform operations such as table creation, index creation, database shrinking, or nonlogged operations. • If a failure occurs, you might lose your most recent data. • If a disaster occurs, you must manually restore the database.

Solution	Advantages	Disadvantages
Log shipping	• You can recover all database activities. The recovery includes any objects that were created such as tables and views. It also includes security changes such as the new users who were created and any permission changes. • You can restore the database more quickly. The restoration of the database and the transaction log is based on low-level page formats. Therefore, log shipping speeds up the restoration process and results in the fast recovery of data. • SMSQL can perform both full and log backup from the primary server.	• The database is unusable during the restoration process because the database is in exclusive mode on the standby server. • There is a lack of granularity. During the restoration process, all the changes in the primary server are applied at the standby server. You cannot use log shipping to apply changes to a few tables and to reject the remaining changes. There is no automatic failover of applications. When the primary server fails because of a disaster, the standby server does not fail over automatically. Therefore, you must explicitly redirect the applications that connect to the primary server to the standby (failover) server.
Transaction replication	• You can read data on a subscriber while you apply changes. • Changes are applied with less latency. • SMSQL works with transaction replication and can be performed in the publisher instance.	• Schema changes or security changes that are performed at the publisher after establishing replication will not be available at the subscriber. • The distributor in transactional replication uses open database connectivity (ODBC) connection or an OLE database (OLEDB) connection to distribute data. However, log shipping uses the RESTORE TRANSACTION low-level transact-SQL statement to distribute the transaction logs. A RESTORE TRANSACTION statement is much faster than an ODBC connection or an OLEDB connection. • Typically, switching servers erases replication configurations. Therefore, you have to configure replication two times: – When you switch to the subscriber. – When you switch back to the publisher. • If a disaster occurs, you must manually switch servers by redirecting all the applications to the subscriber.

Solution	Advantages	Disadvantages
Peer-to-peer transactional replication	• Read performance is improved because you can spread activity across all nodes. • Aggregate update performance, insert performance, and delete performance for the topology resemble the performance of a single node because all changes are propagated to all nodes. • SMSQL can be used to back up for all the nodes that participate in peer-to-peer replication.	• Peer-to-peer replication is available only in SQL Server 2005 Enterprise Edition. • All participating databases must contain identical schemas and data. • We recommend that each node use its own distribution database. This configuration eliminates the potential for SQL Server 2005 to have a single point of failure. • You cannot include tables and other objects in multiple peer-to-peer publications within a single publication database. • You must have a publication enabled for peer-to-peer replication before you create any subscriptions. • You must initialize subscriptions by using a backup or by setting the value of the subscription synchronization type to replication support only. • Peer-to-peer transactional replication does not provide conflict detection or conflict resolution. • We recommend that you do not use identity columns.

Solution	Advantages	Disadvantages
Failover clustering	• You have high server availability. Failover clustering automatically occurs if the primary server fails. • SDW and SMSQL recognize the server is participating in failover cluster. The databases can be backed up with less time and space using SMSQL.	• You incur a greater expense. The maintenance of two servers is two times the cost of maintaining a single server. Because you have to maintain two servers at the same time, it is more expensive to install and maintain clustered nodes. • Servers should be in the same location. If the branches of the organization are around the world and the active-active clusters must be implemented in the branches, the networking and the storage infrastructure that you have to use are very different from a standard quorum device server cluster. Therefore, although it is possible, it is best not to use geographically distant servers. • You have no protection against a disk array failure. • Failover clustering does not allow you to create failover clusters at the database level or at the database object level, such as the table level.
Database mirroring	• Database mirroring increases data protection. • Database mirroring increases availability of a database. • Database mirroring improves the availability of the production database during upgrades. • SMSQL can be used to back up databases in the principal server with less time and space.	• The mirror database should be identical to the principal database. For example, all objects, logins, and permissions should be identical. • Database mirroring involves the transfer of information from one computer to another computer over a network. Therefore, the security of the information that SQL Server transfers is very important.
AlwaysOn availability groups	• Supports up to five availability replicas. An availability replica is an instantiation of an availability group that is hosted by a specific instance of SQL Server and maintains a local copy of each availability database that belongs to the availability group. Each availability group supports one primary replica and up to four secondary replicas. • Supports alternative availability modes, as follows: – Asynchronous-commit mode – Synchronous-commit mode	• Can take time to set up with large size database since the process of setting AlwaysOn availability groups uses setup and restore features.

Solution	Advantages	Disadvantages
	• Supports several forms of availability group failover: automatic failover, planned manual failover (generally referred as simply "manual failover"), and forced manual failover (generally referred as simply "forced failover"). • Enables you to configure a given availability replica to support either or both of the following active-secondary capabilities: – Read-only connection access – Performing backup operations • Supports an availability group listener for each availability group. • Supports a flexible failover policy for greater control over availability group failover. • Supports automatic page repair for protection against page corruption. • Supports encryption and compression, which provide a secure, high-performing transport. • Provides an integrated set of tools to simplify deployment and management of availability groups, including: – Transact-SQL DDL statements – The AlwaysOn dashboard monitors AlwaysOn availability groups – Windows PowerShell cmdlets • Using SDW, SMSQL, and SnapMirror to accelerate building availability groups.	

Table 3 lists comparisons for HA and DR options.

Table 3) Comparisons for HA and DR options.

Feature	Back up and Restore	Log Shipping	Transaction and Peer-to-Peer Replications	Failover Clustering	Database Mirroring	AlwaysOn Availability Groups
Automatic failover	No	No	No	Yes	Yes	Yes
Ease of configuration	Easy	Easy	Medium to hard	Hard	Medium	Medium

Feature	Backup and Restore	Log Shipping	Transaction and Peer-to-Peer Replications	Failover Clustering	Database Mirroring	AlwaysOn Availability Groups
Granularity of recovery	Database	Database	Database object	SQL Server instance	Database	Databases
RPO	No data loss	Possible data loss	Some data loss is possible	No data loss	No data loss	No data loss
RTO	Depends on the size of the database and backup method	Time taken for database recovery	Might run into minutes	20–30 seconds plus time taken to recover databases	<3 seconds	<3 seconds
Administrative overhead	Minimal	Minimal	Might get involved in case of complex publisher-subscriber scenarios	Maintaining cluster hardware	Checking mirror status	AlwaysOn availability groups dashboard

SMSQL 7.0 can be used to integrate and support all of Microsoft SQL Server HA/DR solutions listed in Table 4. SMSQL can even help accelerate building AlwaysOn availability groups (http://media.netapp.com/documents/tr-4106.pdf).

7.1 NetApp HA/DR Solution for SQL Server

The key advantages of using NetApp solutions to create NA and DR plan for SQL Server databases include:

- **Ease of configuration.** The most user-friendly aspect of NetApp solutions is the ease with which you can deploy the previously discussed HA and DR plan. Using the SMSQL GUI and a few Data ONTAP commands, you can arrive at a robust HA and DR solution. This reduces administrative overhead in complex database environments.
- **Speed and performance.** Since SMSQL backups are based on volume Snapshot technology (Data ONTAP), the duration for which the database being backed up remains frozen is minimized. This means that for very large OLTP databases, minimal interference with transaction processing occurs. Sync SnapMirror updates are also reasonably fast and provide healthy RPO and RTO figures.
- **Database restoration options.** SMSQL provides two types of database restoration when it comes to restoring a database using transaction log backups: point-in-time restore and up-to-the-minute restore. These modes provide varying degrees of recovery in between full backups in the event of sudden loss of the primary site.

- **Simplified SQL Server dispersion of DR location.** Note that using the simplified SMSQL GUI, an administrator can opt for per-database or system-wide SQL Server DR plans. No special or extra steps must be taken for controlling the databases to which DR capability must be granted.

Table 4 provides a quick overview of some business problems and how they are addressed on the primary site to provide a resilient architecture.

Table 4) Business problems and NetApp solutions.

Business Problem	Addressed?	How	Description
Single point of failure	✓	Windows cluster + NetApp storage	Windows cluster addressing server resilience and NetApp storage cluster addressing resilience on the storage, providing no single point of failure for applications, server hardware, or storage
Fast backup/recovery	✓	SMSQL	SMSQL automating the complex and manual backup process by creating fast and space-efficient Snapshot copies and providing faster recovery
Disaster recovery	✓	SMSQL + SDW + SnapMirror	SnapMirror replicating the database and log files, and SMSQL providing faster backups and rapid restores
Near-zero-minute RPO	✓	SnapMirror	Scheduled full backups of the SQL Server database replicated every four hours and the T-Log volume replicated synchronously using SnapMirror
Less RTO	✓	SDW/SMSQL	Volume SnapRestore® providing an instantaneous restore

7.2 NetApp SnapMirror for DR

SnapMirror is an integral part of DR plans. If critical data is replicated to a different physical location, a serious disaster does not have to result in extended periods of unavailable data. Clients can access replicated data across the network until the damage caused by the disaster is repaired. Application servers at the recovery site can access replicated data to restore operations for business-critical applications for as long as necessary to recover the production site. Recovery might include recovery from corruption, natural disaster at the production site, accidental deletion, and so on.

SMSQL has built-in integration with SnapMirror. You can elect to have SnapMirror update and verify on an available SnapMirror destination volume after the Snapshot copies have been taken from the SMSQL GUI interface.

Figure 4 illustrates how SMSQL creates Snapshot copies and integrates with SnapMirror to transfer Snapshot copies out of the primary data center.

Figure 4) SMSQL creates Snapshot copy and integrates with SnapMirror to transfer Snapshot copy out of primary data center.

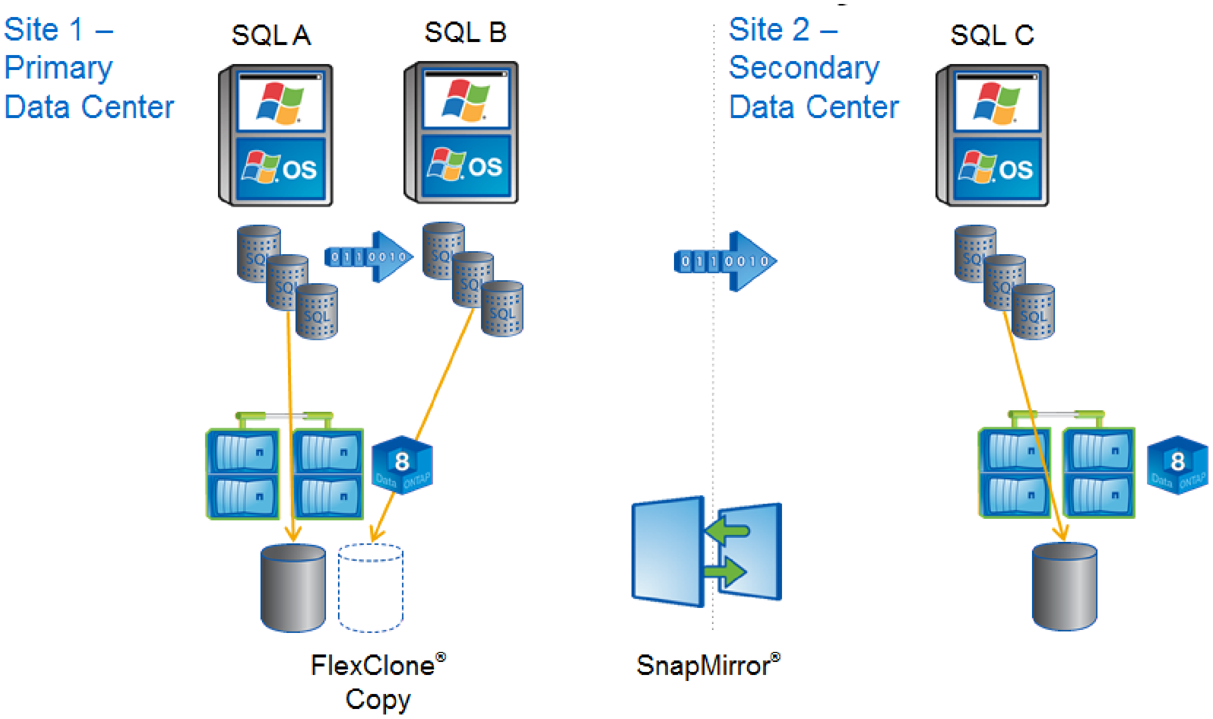
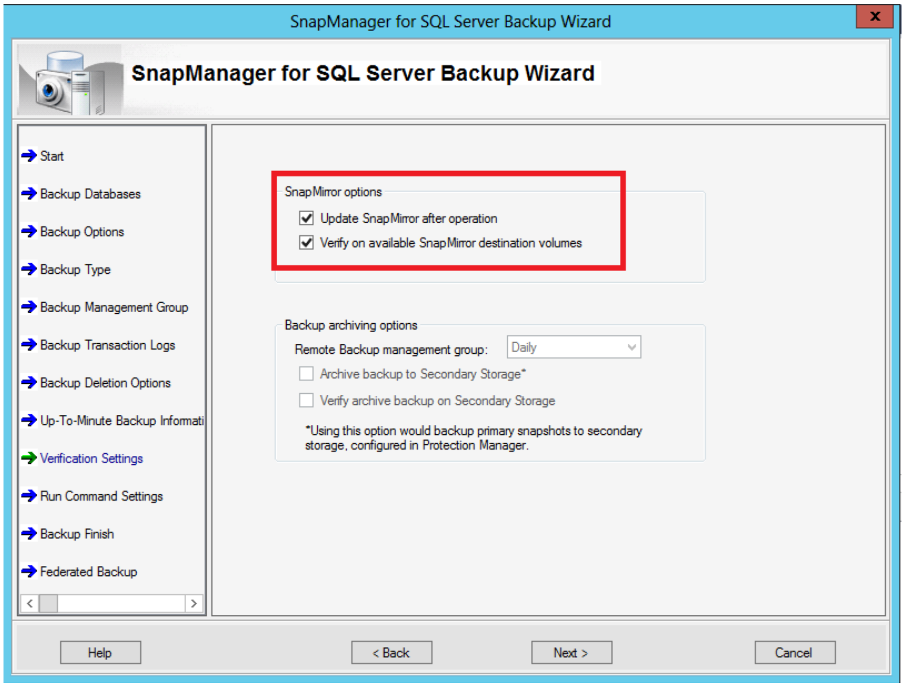


Figure 5 illustrates the display option of SnapMirror in SMSQL.

Figure 5) Display option of SnapMirror in SMSQL.



SnapManager for SQL Server Backup Wizard

SnapManager for SQL Server Backup Wizard

SnapMirror options

☑ Update SnapMirror after operation

☑ Verify on available SnapMirror destination volumes

Backup archiving options

Remote Backup management group: Daily

□ Archive backup to Secondary Storage*

□ Verify archive backup on Secondary Storage

*Using this option would backup primary snapshots to secondary storage, configured in Protection Manager.

The use of SnapMirror elevates the ability of businesses to recover from major data center disasters and outages. The time to look at SnapMirror, both synchronous and asynchronous, is when you need geographical DR. The best practice for SnapMirror with SQL Server is defined in TR-3604: NetApp Disaster Recovery Solution for Microsoft SQL Server.

Many organizations have requirements for mirroring SQL Server databases to geographically dispersed locations for data protection and business continuance purposes. Like Snapshot technology, SnapMirror works at the volume level, so proper grouping of databases by the high-availability index can be particularly useful. Locating one database per FlexVol volume provides the most granular control over SnapMirror frequency as well as the most efficient use of bandwidth between the SnapMirror source and destination. This is because each database is sent as an independent unit without the additional baggage of other files, such as tempdb, or databases that might not share the same requirements. While sharing LUNs and volumes across databases is sometimes required because of drive letter limitations, every effort should be made to group databases of like requirements when specific mirroring and backup policies are implemented.

NetApp SnapMirror technology mirrors data to one or more NetApp storage volumes over a local area network (LAN) or wide area network (WAN) to another NetApp storage system. Once source and destination relationships are established, a SnapMirror baseline transfer initializes the mirror to create a replica of the source on the destination. SnapMirror maintains the synchronization of the replica through efficient, incremental updates that use bandwidth intelligently to distribute deltas across a WAN.

Mirroring Frequency

The frequency of SnapMirror update events (in most environments) is most often determined by the frequency of SMSQL backups. SnapDrive triggers SnapMirror updates upon completion of a SnapManager backup procedure. A supplemental type of Snapshot copy optimized for use with SnapMirror, called “rolling Snapshot copies,” can be used to augment standard SQL Server backups. Rolling Snapshot copies are controlled outside of SMSQL by using the scripting executable `sdcli.exe` for SnapDrive. Be mindful of the following rolling Snapshot technology best practices.

Best Practice: When Using Supplemental Rolling Snapshot Copies
Be sure to configure rolling Snapshot copies to occur only between SMSQL backup and SnapMirror processes. This prevents overlapping Snapshot schedules.

Key SnapMirror Concepts and Tips

- SnapMirror relationships are based on sources and destinations.
- A destination updates its mirrored copy or a replica of its source by "pulling" data across a LAN or WAN when an update event is triggered by the source. The pull events are triggered and controlled by SnapDrive.
- Consistent with the pulling nature of SnapMirror, relationships are defined by specifying sources from which the destination NetApp storage systems synchronize their replicas.
- Destination systems are identified on source systems by assigning destination privileges by using the snapmirror.access protocol access option or by inclusion in the SnapMirror `.allow` file.

As discussed earlier, SnapManager is integrated with SnapDrive, which interfaces directly with a NetApp storage system by using either iSCSI, or FCP access protocols. SnapManager relies on SnapDrive for disk management, controlling Snapshot copies, and SnapMirror transfers.

8 SnapVault Integration Using OnCommand Unified Manager

SnapVault is a disk-to-disk backup and recovery solution that protects data on a SnapVault primary system (formerly known as a SnapVault client) by maintaining a number of read-only versions of that data on a SnapVault secondary system (formerly known as a SnapVault server) and on the SnapVault primary. It leverages the efficiencies of Snapshot copies and protects data at the block level. Once the initial full backup is complete, only changed blocks are replicated to the secondary storage system.

SnapVault integration. SnapVault support in SMSQL requires database (SAN/NAS) backups to be copied using SnapVault to secondary storage (archived backup). OnCommand Unified Manager is used to SnapVault dataset.

Best Practices
• Make sure the database and SnapInfo LUNs are on separate volume to avoid SnapVault retention policy from overwriting Snapshot copies. • For availability group restore, clone, and reseed from secondary, it is recommended not to use the `-vaultsec` parameter. For additional information, refer to the SnapManager 7.0 for SQL Server (SMSQL) Installation and Administration Guide.

9 Best Practices for Ongoing Management and Monitoring

SMSQL gives you a report with overall status of operations such as backup, verification, and clone resync. When using NetApp efficiency features, the volumes should be appropriately sized so that autosize and/or autodelete policies are not triggered unless there is an abnormal rate of change or a problem with Snapshot copy retention. NetApp OnCommand management software that includes Operations Manager is the recommended tool to monitor SQL Server volumes for these events and to send notifications to the storage administration team to follow up further with the SQL Server administration team. SNMP can also be used to monitor these events.

After a notification for a volume autogrow or Snapshot autodelete event has been received by the storage administration team, the recommended action is for the storage administration team to examine the affected storage controllers and then follow up with the SQL Server administration team for further administrative actions. A typical cause of volume autosize events is that the rate of change greatly surpassed the rate of change assumption used in sizing the volume. Another cause for volume autosize events is that older Snapshot copies created by SnapManager for SQL Server are not being deleted. As Snapshot copies age, they can grow in size and consume more capacity than originally allocated in the volume. A typical cause of SnapManager for SQL Server not deleting backups is that SnapManager for SQL Server backups are failing. By default, SnapManager for SQL Server does not delete Snapshot copies of older SnapManager for SQL Server backup sets if the backup fails. Another cause for SnapManager for SQL Server not deleting backups is that the SnapManager for SQL Server backup retention policies are not being enforced correctly because Snapshot copies were manually removed outside of SnapManager for SQL Server on the controller itself. Monitoring the health of SnapManager for SQL Server can be done by monitoring for SnapManager for SQL Server event IDs and the enhanced monitoring functionality in SMSQL.

SnapManager operational reports are automatically created for SnapManager configuration, backup, restore, backup set deletion, and other miscellaneous operations. Each report is a log file that includes step-by-step details of the operation, the final status of the operation, and any error messages encountered during the operation.

SnapManager also writes to the Windows event application log. Backup and verification operations that are incomplete or did not run are logged in the Windows event application log as errors. For cloning operations, all events are logged as informational.

The SnapManager report directory can be local or be configured to use a Windows file share network resource.

SnapManager also supports the generation of e-mails for backup, verification, and cloning operations. The e-mail can be configured to send a brief summary as well as be configured to include the operational report as an attachment.

References

This section lists useful resources to assist in planning and managing your SQL Server storage environment.

- NetApp Storage Systems
- Data ONTAP Documentation

Additional Documentation Available from NetApp Support Site (Formerly NOW)

- NetApp SnapDrive for Windows
- SnapManager for Microsoft SQL Server (SMSQL)
-
- Microsoft SQL Server Storage Engine Blog
- MSDN: Product documentation including SQL Server Books Online
- SQL Server Best Practices
- SQL Server Hardware and Software Requirements

Microsoft SQL Server

- Description of disaster recovery options for Microsoft SQL Server
- INF: Disaster Recovery Articles for Microsoft SQL Server
- Disaster Recovery
- Introduction to Backup and Restore Strategies in SQL Server
- You cannot restore system database backups to a different build of SQL Server

NetApp SnapMirror

- SnapMirror How-To Guide
- SnapMirror Best Practices Guide
- SMSQL Best Practices Guide

Version History

Version	Date	Document Version History
Version 1.0	October 2013	Initial release

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

TR- 4232-1013

NetApp®

www.netapp.com