EMC²
where information lives®

White Paper

EMC CLARiiON Data Replication Options for Microsoft SQL Server Deployments

Technology Concepts and Business Considerations

Abstract

This technical white paper describes features, implications, and considerations of EMC® CLARiiON® replication options used for Microsoft SQL Server deployments.

February 2008

Copyright © 2008 EMC Corporation. All rights reserved.

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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

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

All other trademarks used herein are the property of their respective owners.

Part Number H4184

Table of Contents

Executive summary

Microsoft SQL Server is a premier database used by many financial, manufacturing, and ecommerce applications. For many of these mission-critical deployments, the SQL Server database and the applications have to run nonstop to ensure uninterrupted business operation. In the event of production failure, the database must be brought online quickly on alternate servers at local or remote sites with little or no loss of data. This typically implies the need for copies of data replicated locally or remotely to achieve the desired rapid recovery. EMC provides a full range of storage-based as well as appliance-based replication software for replication and disaster recovery. These storage-based and out-of-band appliance-based solutions are extremely efficient in replicating large amounts of data in SQL Server databases stored in EMC® CLARiiON® and certified third-party storage systems without utilizing host resources. The CLARiiON replication software SnapView™ and RecoverPoint CDP provide local replication while MirrorView/S, MirrorView/A, SAN Copy™, and RecoverPoint CRR provide remote replication.

Introduction

This white paper describes how EMC RecoverPoint and CLARiiON storage-based replication technology can be utilized for designing replication and disaster recovery solutions suitable for Microsoft SQL Server deployments. Various EMC replication software options including SnapView, SAN Copy, MirrorView® and RecoverPoint are discussed and general guidelines for applying these features in designing SQL Server disaster recovery solutions are provided.

Audience

This white paper is intended for customers, EMC partners, SQL Server database administrators, storage architects, and storage administrators who require information about how CLARiiON replication products work and how they can provide a replication solution satisfying the need in most SQL Server environments. It is assumed that the reader has a general knowledge of the SQL Server database as well as CLARiiON storage systems and software.

Terminology

CLARiiON array replication software: Optional software set offered for CLARiiON systems including SnapView, MirrorView and SAN Copy, supporting efficient replication of data stored in CLARiiON systems.

CLARiiON consistency support: The logical concept of designating (and enforcing) a group of CLARiiON storage LUNs to be replicated by the replication software as a coherent set. The replication action is performed automatically for all the members in the set. Either all members will be replicated, or none will be replicated.

Consistency group: A set of mirrors that are managed as a single entity and whose secondary images always remain in a consistent and restartable state with respect to their primary image and each other.

Fracture: A condition in which I/O is not mirrored to the secondary image; this can happen when you initiate the fracture (Admin Fracture) or when the system determines that the secondary image is not reachable (System Fracture).

Logical unit number (LUN): A unique identifier that is used to distinguish among logical devices that share the same bus.

Primary image: The LUN containing the production data, the contents of which are replicated to the secondary image.

Promote: The operation by which the administrator changes an image's role from secondary to primary. As part of this operation, the previous image becomes a secondary image.

Secondary image: The LUN that contains a mirror of a primary image LUN. This LUN must reside on a different CLARiiON storage system from the primary image.

Replication and recovery operations

Local replication

Local replication involves replicating SQL Server database LUNs using SnapView snapshots, SnapView clones, or RecoverPoint CDP snapshots. These point-in-time replicas can be quickly reverted to production in the event of a production corruption due to application or user errors. They can be mounted on secondary hosts for rapid database service recovery, to offload database backup and to facilitate other repurposing needs such as product testing. Locally replicated data typically coexists with the source data at the same geographical location, and frequently in the same storage unit. As such, local replication typically does not protect against unanticipated site or storage system physical failures.

Remote replication

Remote replication replicates SQL Server database LUNs to a remote recovery site using SAN Copy, MirrorView/S, MirrorView/A, or RecoverPoint CRR. These replicas can be mounted on secondary hosts for disaster recovery, backup, and product testing. Since the data is replicated to a remote site on a separate physical storage unit, remotely replicated data offers operational protection against site and physical storage system failures.

Synchronous replication

Synchronous replication ensures that data replicated at a secondary site is an identical copy of the primary site. In synchronous replication the data update won't be acknowledged to the host as completed until the data update is successfully completed at both the primary and secondary sites. This guarantees that the data recovered from the secondary server is a completely up-to-date copy of the primary site data. The main advantage of synchronous replication is that it provides zero data loss in the event of a failure of the primary site. Synchronous replication can impact the application performance in a high-latency or low-bandwidth network environment. It is therefore used primarily for local or short-distance replication.

Asynchronous replication

During asynchronous replication the application does not wait for the data update to be acknowledged from the secondary site. Because of this the application does not suffer from the response time degradation caused by each update incurring the cost of a network round trip. Instead application updates are written at the primary site, and data is also queued in order and forwarded to the secondary site as network bandwidth allows. Asynchronous replication can cause data lag in the secondary site, resulting in data loss in the event of a failure of the primary site. Asynchronous replication is primarily used for long-distance replication.

Failover

Failover is the process of switching the production processing to another host in the same site or to another site in the event of unanticipated production site failure or for a scheduled maintenance of the production site. The transitioning of operation can be achieved through either restart or recovery approaches.

Restart

Restart involves replicating the production database to a secondary site continually with all database changes to create a consistent restartable database copy using consistency technology. In the event of a failure of a production site or database, production is restarted at the secondary site by mounting the restartable database copy replicas on a host. Note that a restartable database copy is not suitable for backup or recovery operation using transaction logs.

Recovery

Recovery operation requires maintaining a recoverable copy of the production database at the recovery site. A recoverable database copy is one that includes a stable set of business transaction content from some particular point in time. Transaction logs can be applied against this database to roll the database content forward to a point in time after the database copy was created. The database transaction log file backup files are periodically shipped to the recovery site. SAN Copy is an ideal option to support shipping of database logs. In case of production failure, transaction log backups can be applied to the recoverable database to bring the production database to an operational state at the recovery site and mitigate data loss.

Failback

Failback is the process of switching production operations back to the production site from the recovery site when the production site is back in service. This involves shipping the production database back to the production site using SAN Copy or the failback operation of RecoverPoint CRR.

CLARiiON consistency group

Most SQL Server databases have data that is spread across multiple CLARiiON LUNs. These LUNs can be write-order dependent because the content stored in the different LUNs collectively represents a particular transactional state of the database as recognized by SQL Server. When these LUNs are replicated it is crucial that they must be replicated as a content coherent set to ensure the resulting database is in a restartable state. The CLARiiON consistency group is an array-based consistency feature that enables such a set of content-related LUNs to be replicated as a single entity for maintaining data consistency and write-order fidelity across all LUNs in the group. Write-order fidelity is the ability to keep data at the recovery site in the same order as it was in the production site. When replicating with the consistency group feature, a single replication request replicates all LUNs specified in the group as one logical action. This ensures replicated LUNs of database are guaranteed to be identical point-in-time, dependent write consistent restartable replicas of their source LUNs. The CLARiiON SnapView, MirrorView, and RecoverPoint CDP solutions provide consistency for LUNs on the same CLARiiON array. RecoverPoint CRR can provide constancy among LUNs in a heterogeneous storage environment that consists of multiple CLARiiON systems and multi-vendor storage systems.

SQL Server VDI mode

A SQL Server database replica used for backup or disaster recovery using transaction logs must be a consistent copy of the production database. Such a consistent database replica can be created using SQL Server Virtual Device Interface (VDI) mode. VDI flushes the SQL cache to storage and freezes I/O to the database to put the database into a quiescent state. Both Replication Manager and the RecoverPoint *kutils* utility invoke VDI to create a consistent replica of the SQL Server database, which can be mounted on recovery or backup hosts.

CLARiiON replication vs. SQL Server native replication

SQL Server provides native replication solutions to support database recovery in the event of a disaster. However, total disaster protection solutions for SQL Server deployments frequently require more than just the SQL Server database. An example may be deployments involving federated databases. Federated database environments consist of a collection of distinct databases and other related application data that must maintain data content relationship coherency for the enterprise operation to be supported correctly. In the event of a disaster, while it may be possible to individually recover the databases replicated through the native solutions, it may be very difficult, if not impossible, to operationally verify that the recovered databases are coherently related to each other to allow operation to resume properly.

When these federated data sources are stored and replicated as a consistent group of data LUNs using EMC consistency group technology, the consistency of the recovered data is guaranteed. Another advantage of CLARiiON replication is that unlike host-based SQL Server replication, CLARiiON storage-based and out-

of-band replication solutions do not consume host processing cycles, thus avoiding performance impact on the production database. Additionally, the RecoverPoint policy-based replication and built-in network bandwidth reduction technology features are well suited for efficient SQL Server replication in low-bandwidth environments.

SQL Server local replication and recovery

The SQL Server local replication and recovery can be done using EMC CLARiiON SnapView or RecoverPoint CDP replication products.

SnapView

EMC SnapView is a storage-based software product option of the CX3 UltraScale™ series CLARiiON product lines that can create point-in-time snapshots or full copy clones of CLARiiON storage LUNs without using host processor cycles. These SnapView snapshots or clones can be used for capturing a copy of all relevant SQL Server LUNs with minimal impact to the production database. These snapshots and clones can be presented to secondary servers as independently usable database copies for different functions such as debugging, product testing, and data migration while the production host continues processing. The database can be restarted from clones if the primary productions LUNs become unavailable or corrupted. SnapView persistent mode ensures the session can be used in a rollback operation and that the session will survive storage processor failures. SnapView in combination with Replication Manager can be used for creating consistent database snapshot or clone replicas for recovery or backup purposes. In this scenario Replication Manager enables SQL Server VDI mode to quiesce the database before creating a snapshot or clone.

SnapView snapshot

SnapView snapshot is a point-in-time image of a LUN. With SnapView snapshots, the entire SQL Server database can be replicated in seconds because snapshots are pointer-based. For each snapshot session, SnapView keeps track of how the source LUN looks at a particular point in time. Any writes to the source LUN results in SnapView storing a copy of the original data on a reserved LUN. This copy is referred to as copy-on-first-write (COFW) and occurs only once for each data block that is modified on the source LUN. The snapshots can be presented to a host server for read and write access as shown in Figure 1. The snapshot becomes visible to a secondary host server when a snapshot *session* is activated, and the snapshot included in a storage group projects the snapshot as a distinct LUN to the host. The main advantage of a snapshot is that it can be created almost instantaneously, and consumes only a small fraction of the total size of the source LUN, sufficient to track all original source LUN data that may have been changed in the source LUN since the snapshot session was started. The size of the storage may vary, and is typically 10 to 20 percent of the source LUN depending on how much data has been changed. The snapshot storage is allocated from the reserved LUN pool. Because a snapshot contains pointers to the source LUN, heavy concurrent usage of the source LUN by the production host and the corresponding snapshot by the repurposing host may result in undesired performance impact. SnapView can roll back a snapshot session to the source LUN in the event the LUN is corrupted due to application error or user error. SnapView can manage up to eight snapshot copies of a single source LUN. A snapshot is most useful for providing point in time view of a SQL database that can be used for performing short duration functions such as online backup and production testing. The snapshot should not be used as permanent operational backup storage for a SQL Server database, because losing the source LUN makes the snapshot unusable.

Local replication and recovery using SnapView snapshots

The following steps are used for replication and rollback of SQL Server database LUNs using SnapView snapshots:

1. Using SnapView, create snapshots of SQL Server database LUNs.

2. Mount the snapshots on secondary hosts for product testing or backup.

3. Roll back snapshots to the production LUNs in case of production database corruption.

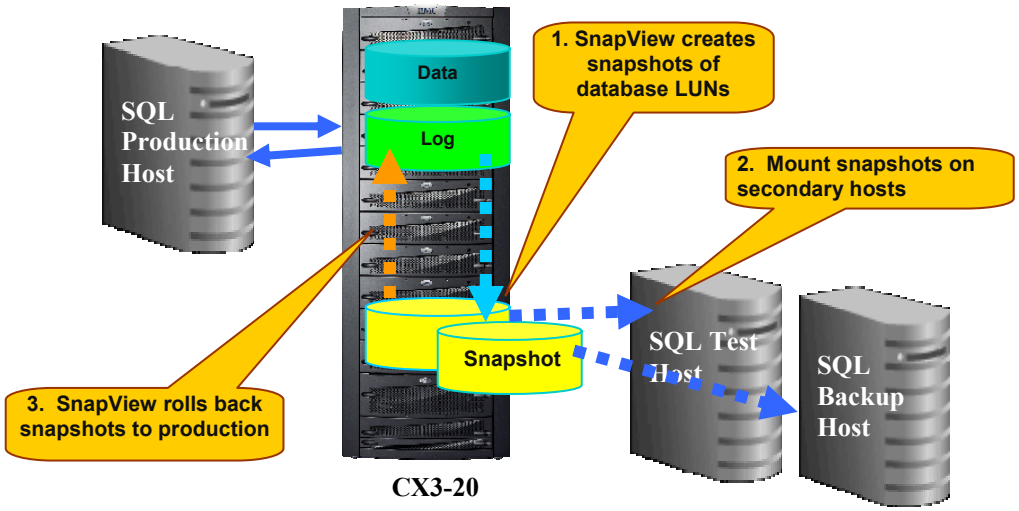

Figure 1. Local replication using SnapView snapshots

SnapView clone

SnapView clone is a point-in-time physically separate and complete copy of the production LUN, rather than a copy-on-first-write model of snapshots. Clones provide better data availability and performance than snapshots do. Unlike a snapshot, which is unusable if the production LUN is damaged, a fully populated clone is usable even if the production storage spindles or storage processor fail. However, clones take more storage space than snapshots since they are exact replicas of their source LUNs. Clones are usually created from physical space separate from those used for holding the source data.

Once the clone is created, it needs to be synchronized, which is the process of copying the data from the source LUN to the clone. While the clone is being synchronized, the source LUN can still be accessed by the production host. Initial synchronization normally takes longer to complete, depending on the size of the source LUN. After the synchronization, it can be fractured from the source LUN. A change tracking map is maintained to keep track of any changes that may have been made to the source but have not yet been reflected in the clone. The map content is persistently stored in a special area in the reserved LUN pool, so the change delta information will always be accurate even should a storage system failure. The tracking map is used to drive bitwise resynchronization when needed.

Like a snapshot, a clone can be presented to a host server for read and write access when it is in the fractured state. Because the clone typically occupies space on disks separate from where the source LUN data resides, concurrent use of both source and clone will not interfere with each other. A clone's *reverse-synchronization* feature can be used for replacing the contents of the production LUN with the contents of the clone in case the production LUN becomes corrupted due to user or application errors or an earlier point-in-time data is desired. Figure 2 shows how clones can be used for replicating and reverse synchronizing SQL Server database LUNs. A clone can be used for mirroring, but it has the additional overhead of mirroring writes when unfractured. SnapView can create up to eight clones of a production LUN. Snapshots can also be created from clones, allowing a maximum of eight snapshots of each clone.

Local replication and recovery using SnapView clones

The following steps are used for replicating and reverse-synchronizing SQL Server database LUNs using SnapView clones:

1. Using SnapView, create clones of SQL Server database LUNs.
2. Mount clones on a secondary server in case of production failure.
3. Reverse-synchronize clones back to the production LUNs in case of production database corruption.

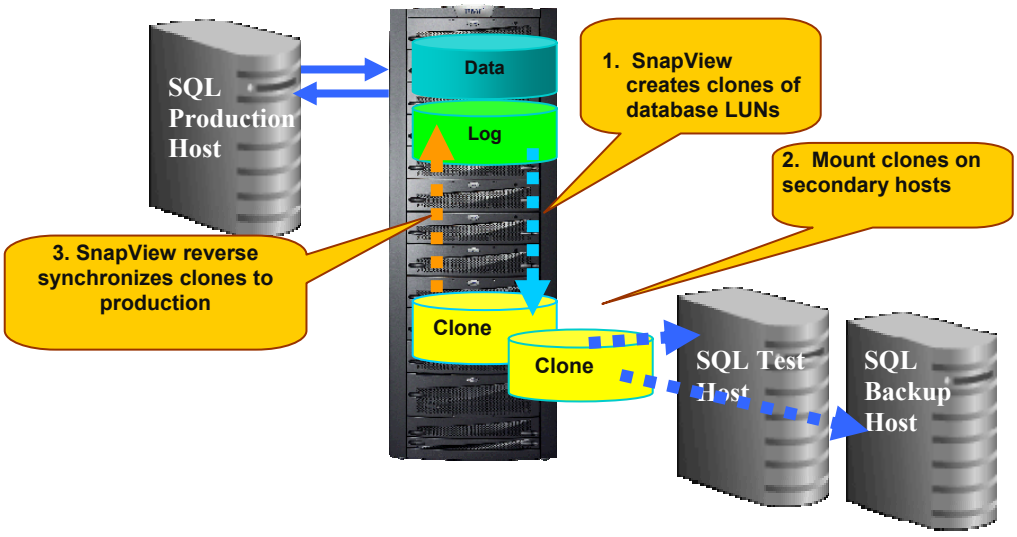

Figure 2. Local replication using SnapView clones

Creating consistent replicas using SnapView and Replication Manager

EMC Replication Manager is a host-based software product that simplifies the creation and management of snapshots and clones for recovery operations without the need to work directly with CLARiiON replication commands and primitives. It can be integrated with SnapView to handle all the necessary steps associated with creating, mounting, and dismounting of SnapView snapshots and clones including the scheduling required for managing replication and recovery operations.

A database replica used for backup or recovery operation using transaction logs must be a consistent copy of the production database. Consistent database copies can be created only when the database is in a quiescent state with the aid of SQL Server VDI mode. Replication Manager can invoke SQL Server VDI mode to generate consistent copies of SQL Server database replicas that can be used for backup and recovery operations as shown in Figure 3. VDI mode puts SQL Server briefly into a quiescent state while Replication Manager in conjunction with SnapView creates consistent replicas of the database. Replication Manager disables VDI mode once the replicas are made. These consistent replicas can be used as backup copies by mounting on a backup server or used for database recovery using transaction logs.

The following steps are used for creating backup copies of SQL Server database replicas using SnapView in conjunction with Replication Manger:

1. Using Replication Manager, invoke SQL Server VDI mode to quiesce the database.
2. Create consistent snapshots of production database LUNs.
3. Disable VDI mode.
4. Mount consistent snapshots on backup or recovery hosts.

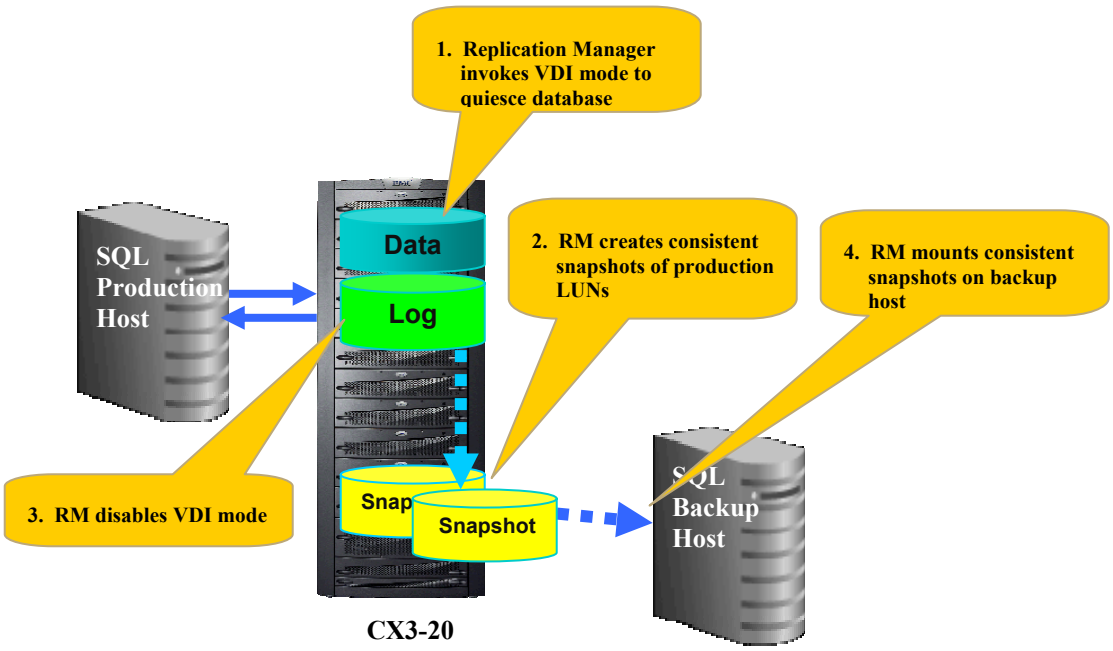

Figure 3. Creating consistent database replicas using SnapView and Replication Manager

RecoverPoint and RecoverPoint/SE

EMC RecoverPoint and RecoverPoint/SE are out-of-band appliance-based replication software offerings. The full-featured version is known as RecoverPoint, and the CLARiiON-specific version is known as RecoverPoint/SE. Each version offers two complementary replication products:

- RecoverPoint Continuous Data Protection (CDP) used for local replication (see next section)
- RecoverPoint Continuous Remote Replication (CRR) used for remote replication (see the "RecoverPoint CRR" section)

RecoverPoint CDP

RecoverPoint CDP allows the replication of SQL Server production data to replication volumes that reside on the same CLARiiON array as the production database. RecoverPoint CDP is capable of creating snapshots of SQL Server database images at different points in time through the journaling mechanism as shown in Figure 4. In the event of a SQL production storage corruption, RecoverPoint CDP can recover any point-in-time snapshot maintained through its journal back to a recovery host, and the database service can be restarted from that image. Data loss due to user or application errors or rolling disaster can also be quickly reversed by going back to a version before the error was committed. The user and application specific bookmark feature further allows easily identifiable operational database points to be registered in the Recover Point CDP. The bookmarks make it extremely easy to revert to a desired database state when needed using bookmarked CDP snapshots.

Local replication using RecoverPoint CDP

The following steps are used for replication and recovery using RecoverPoint CDP:

1. Create consistent snapshot images of the production database.
2. Replicate the snapshot to the Journal volume.

3. Recover the bookmarked image and mount it on the recovery host.

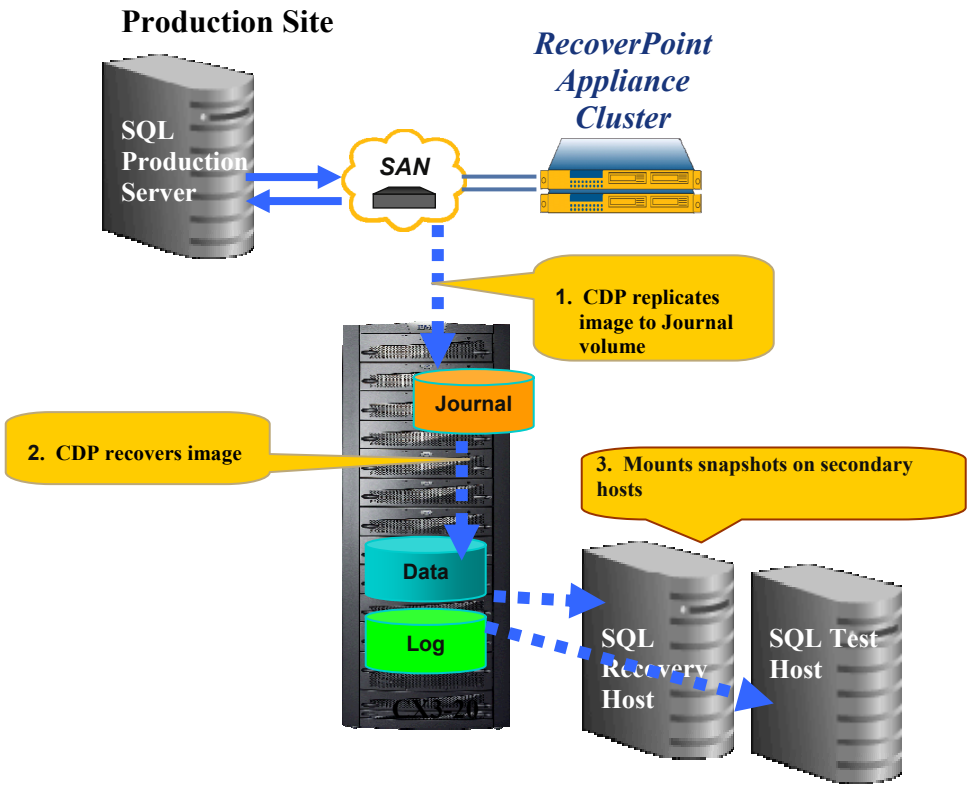

Figure 4. Local replication using RecoverPoint CDP

SQL Server remote replication and recovery

Microsoft SQL Server remote replication and recovery can be provided using EMC SAN Copy, MirrorView/S, MirrorView/A, or RecoverPoint CRR.

SAN Copy

SAN Copy is a storage-based software product that can be used for copying content of the source LUN to destination LUN in local storage or remote storage systems without host intervention. A key feature of SAN Copy is that it can copy data between CLARiiON and multi-vendor storage systems (including IBM, HP, Hitachi Data Systems, and Sun storage arrays). The source LUN for a SAN Copy session can be the production LUN or SnapView clone of the production LUN. SAN Copy supports full and incremental copies of a source LUN. The initial copy synchronization requires a full copy replication of the source LUN. After the initial full copy of the source LUN, the incremental SAN Copy feature can be used to perform a bitwise-refresh of the destination LUN(s) with just the blocks that have changed since the last copy of a source LUN. This significantly reduces the time needed to refresh the destination LUN. The incremental SAN Copy feature requires sufficient storage space in the reserved LUN pool on the source array. Applications where performance is a critical factor should consider clones of the source LUNs to drive the incremental SAN Copy sessions instead of the source LUN. SAN Copy gives its best performance if the source and destination arrays are connected through multiple front-end sources to destination pairs. SAN Copy typically utilizes Fibre Channel between primary and secondary arrays. As of FLARE® release 26, SAN Copy data replications can also be provided by using the built-in iSCSI ports on CX3 FC/iSCSI systems.

Creating a consistent SQL Server database copy at the recovery site using SAN Copy

Consistent replicas of SQL Server database replicas suitable for backup or disaster recovery can be created at the recovery site using SAN Copy in conjunction with SnapView and Replication Manager as shown in Figure 5. Replication Manager quiesces the database with the aid of SQL Server VDI mode. Replication Manager in combination with SnapView creates the production database clones. Clones are then replicated to the recovery site using SAN Copy. The consistent replicas can be mounted on backup servers. The following steps are used for creating a consistent replica of a SQL Server database at the recovery site.

1. Create a target LUN for each source LUN on the recovery site.
2. Using Replication Manager, invoke SQL Server VDI mode to quiesce the database.
3. Using RM, create production database clones.
4. Using RM, disable VDI mode.
5. Using RM, replicate clones to the recovery site using SAN Copy.
6. Mount consistent clone images on the backup host.

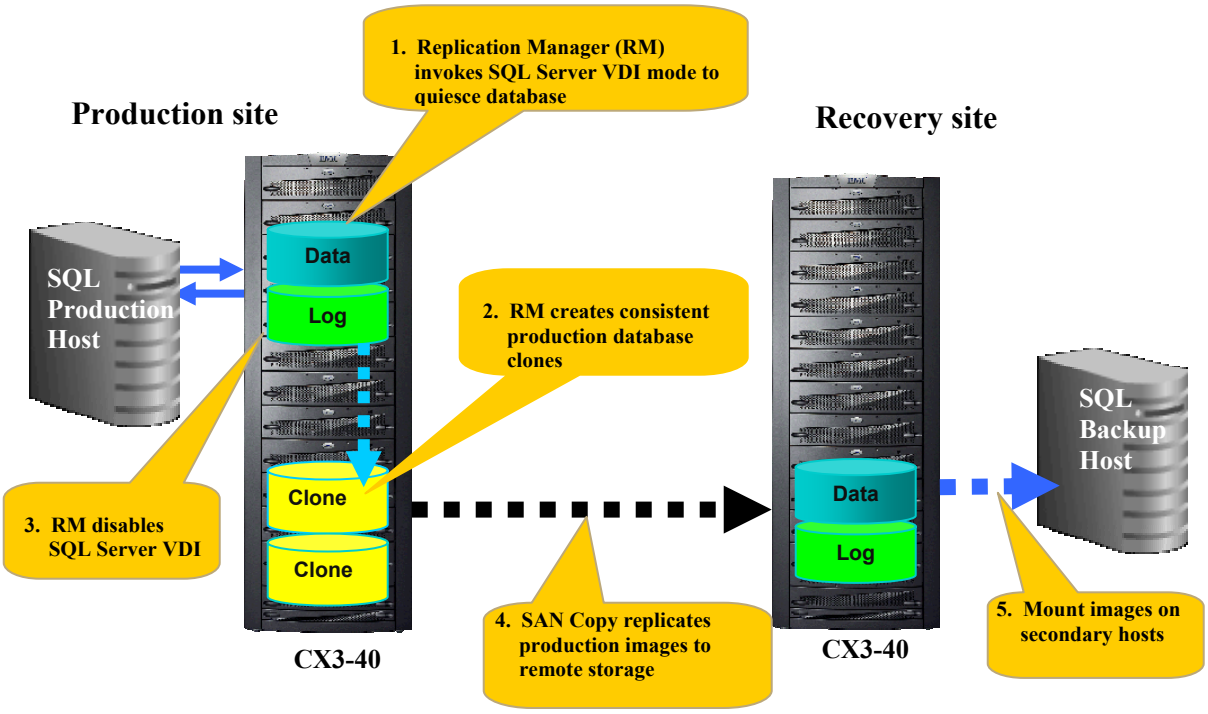

Figure 5. Creating a consistent SQL Server backup copy at the recovery site

SQL Server remote replication and recovery using log shipping

SQL Server database recovery can be performed at the recovery site using production database transaction log files. This involves maintaining a standby database at the recovery site and applying incremental database logs to the standby database to restore production at recovery. In this scenario, SnapView creates clones of database log LUNs and they are replicated to the recovery site using SAN Copy as shown in Figure 6. Thereafter, production database log LUNs are periodically shipped to the recovery site through SAN Copy. In case of production failure, the incremental database log backups are applied to the standby database to restore production to a specific point in time at the recovery site. The following steps are used

for replication and recovery of the SQL Server database at the recovery site using SnapView and SAN Copy:

1. Create a target LUN for each source LUN on the recovery site.
2. Using SnapView, create production database clones.
3. Fracture clones from source LUNs
4. Create a SAN Copy session for each clone.
5. Unmount destination LUNs from the host before starting the SAN Copy sessions.
6. Start each session, and then monitor and complete the sessions.
7. Mount log replicas on the recovery host.
8. Apply incremental logs to the database using the SQL Server RESTORE DATABASE command with the norecovery option.

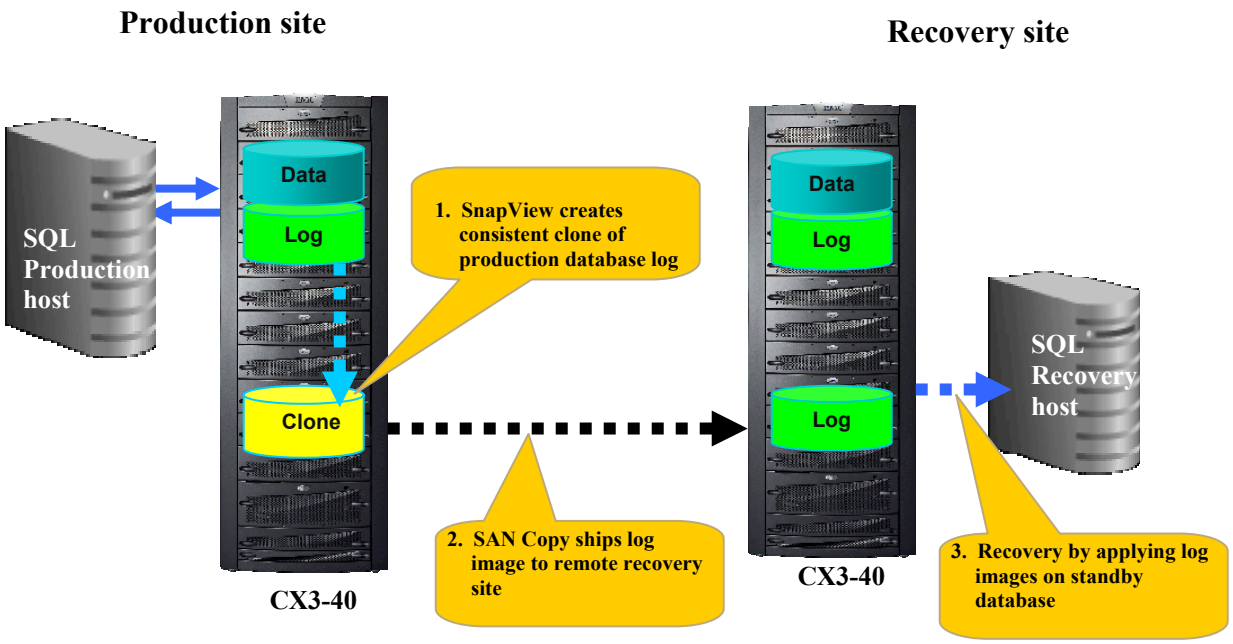

Figure 6. SQL Server log shipping using SAN Copy

MirrorView

MirrorView is a storage-based replication software product that can replicate SQL Server database LUNs to remote locations for disaster recovery. MirrorView replication is transparent to the host Remote replication facilitates failover by restarting the secondary mirror image on the recovery site in case the production host or the production storage system fails. This involves fracturing and promoting a secondary mirror image to a recovery site host as shown in Figure 7. Fracturing stops MirrorView replication from the primary image to the secondary image. Promoting the image involves an administrator changing an image's role from secondary to primary in order to mount the image to the secondary host. MirrorView can integrate with SnapView to create SnapView snapshots of a secondary image that can be used for product testing and data verification at the remote site. Clones of the secondary image can also be created to maintain a usable copy (gold copy) of the primary image at the secondary site. As of FLARE release 26, MirrorView data replications can also be done by using the built-in iSCSI ports on CX3 FC/iSCSI systems.

MirrorView allows bidirectional replication that facilitates failback of production from the secondary to production site once the production site is back online. The MirrorView consistency group feature can maintain write ordering across LUNs at the remote recovery site in the event of a failure of the write-order dependent LUNs on the source, or the mirroring connections. MirrorView software offers two complementary mirroring products: MirrorView/S and MirrorView/A.

MirrorView/S

MirrorView/S is a replication software product that can mirror data images of a production host LUN synchronously in real time to secondary storage at a remote site. Synchronous mirroring offers zero data loss in the event of the failure of a production site. The mirror image of the production data at remote sites can be used to restart production at the recovery site in the case of primary site failure. Bandwidth and latency of the MirrorView/S interconnect are critical to successful deployments with SQL Server replication. Sufficient bandwidth must exist to initialize the remote mirror and sustain active mirroring operations. The greater the distance is between the production site and the remote protection site, the greater the propagation latency. Latency must remain low enough to prevent database and application bottlenecks that could ripple through to the end user. High bandwidth and low latency requirements restrict the use of MirrorView/S for long-distance replication. MirrorView/S is primarily targeted for short-distance (metropolitan/campus) replication.

MirrorView/A

MirrorView/A is an asynchronous replication software that offers long-distance replication based on a periodic incremental-update model. It periodically updates the remote copy of the data with all the changes that occurred on the local copy since the last update. This can result in minutes of data loss in case of the failure of the primary site. The performance of MirrorView/A depends on:

- The distance between production and recovery sites
- Latency and bandwidth characteristics of the connecting link
- The amount of data transfer
- Duration of the update

Because of these performance factors, user must properly size the replication requirements before deploying MirrorView/A for SQL Server disaster recovery. MirrorView/A is generally suited for replicating images with lower rates of data changes over long distance on lower bandwidth links.

MirrorView/A remote replication and failover

The replication involves mirroring SQL Server database LUNs to the recovery site. Failover is initiated by fracturing the mirror to create a restartable database replica, which is then mounted on the recovery host to restart the database at the recovery site. The following steps are used for replication and failover using MirrorView/A:

1. Establish remote mirrors.
2. Create a consistency group.
3. Include SQL Server production database LUNs to be replicated in the consistency group.
4. Periodically replicate the consistency group to remote mirrors.
5. In the event of the failure of the production database, fracture mirrors at the recovery site.
6. Promote mirrors to primary state.
7. Mount mirrors on a recovery host to restart the database.

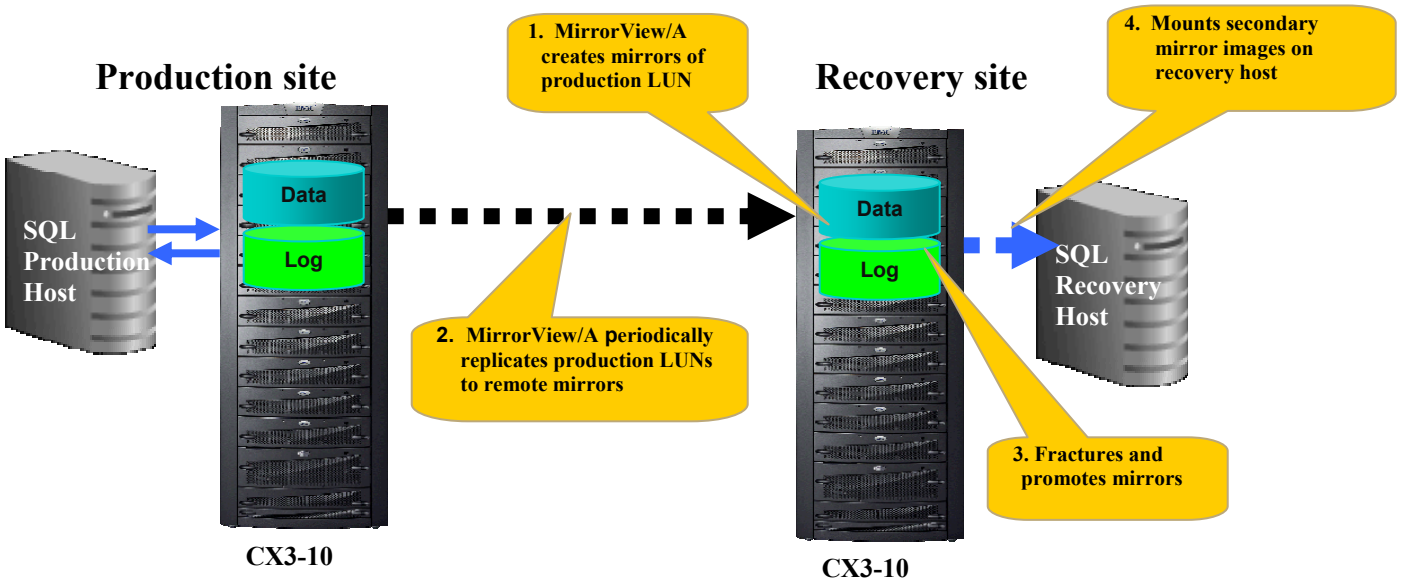

Figure 7. SQL Server remote replication using MirrorView/A

Creating a gold copy of a SQL Server database using clones of mirrors

Starting with FLARE 24, MirrorView offers a *clones of mirror* feature that allows SnapView clones of a mirror to be used as a secondary image. Clones of mirrors can be used for creating a full copy of a production database at a remote recovery site. Performance-sensitive SQL Server applications should consider using clones of mirrors instead of snapshots for restartable copy at the secondary site because unlike snapshots, clones once fractured do not put load on the primary site. Also clones being independent copy of the mirrors provide greater data protection than snapshots.

The following steps are used for replication and failover using clones or mirrors:

1. Establish remote mirrors of production database LUNs.
2. Create a consistency group for the LUNs.
3. Include LUNs to be replicated in the consistency group.
4. Using MirrorView/A, periodically replicate production LUNs to target mirrors.
5. Using SnapView, create clones of mirrors at the recovery site.
6. Using SnapView, fracture clones of mirrors to create a gold copy of the production database.
7. Mount full copy images on the recovery host in case of a production site failure.

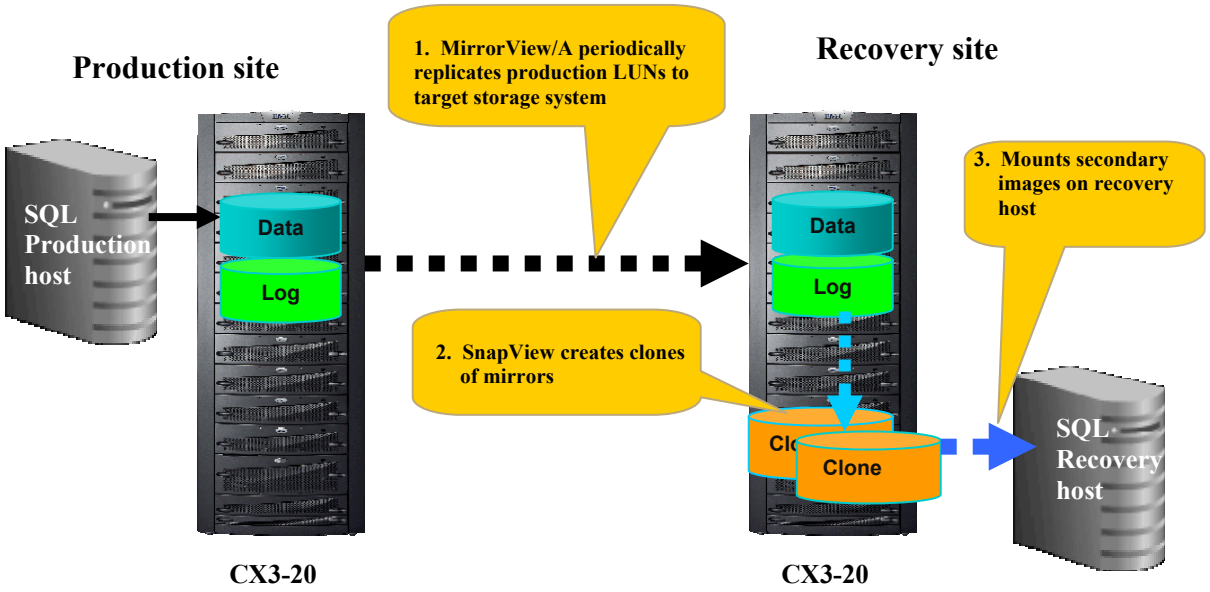

Figure 8. Creating a gold copy of a production database using clones of mirors

RecoverPoint CRR

EMC RecoverPoint CRR is an enterprise-class replication and disaster recovery solution that provides bidirectional data replication over standard IP networks across any distance. RecoverPoint CRR can replicate volumes synchronously or asynchronously to a remote site and is capable of automatically switching between these modes based on available bandwidth, application load, and replication policy. It compresses the data before transmitting to the remote site in order to conserve WAN bandwidth. RecoverPoint CRR replicates incoming writes asynchronously to the remote recovery site as snapshot replica images in a journal volume. Each snapshot contains only the changed data between the current and previous snapshot. Even if corrupted data is replicated to the journal during a rolling disaster, the user can select a known point-in-time snapshot image before the data corruption occurred from the journal volume to restart the production database. The main drawback of snapshot replication is that it creates a copy that is not updated. RecoverPoint overcomes this shortcoming by replicating small aperture snapshots frequently in order to reduce the lag between data written to the production site and the same data replicated to the recovery site.

SQL Server LUNs can spread over single or multiple heterogeneous storage systems and hosts in a federated database environment. Maintaining data consistency across replicated LUNs using consistency groups is a key feature of RecoverPoint. The consistency group feature allows multiple interrelated SQL Server application volume pairs from the primary site to be grouped together to operate as a single unit. This assures write-order fidelity across all LUNs assigned to a consistency group, guaranteeing a consistent restartable copy of the database on the recovery site in the event of a production site failure. Write-order fidelity is the ability to keep data at the recovery site in the same order as it was in the production site.

RecoverPoint CRR supports policy-based replication using bookmarks for application-aware recovery. In a SQL Server environment, it can leverage Microsoft VDI for creating a consistent image of the database LUNs. This image can be bookmarked and later it can be identified and accessed at the recovery site for backup or recovery of production during a disaster recovery operation.

Creating a consistent replica using the RecoverPoint kutils utility

RecoverPoint provides a kutils utility that can be used for creating consistent database replicas used for backup and recovery. The kutils *sqlsnap* command puts SQL Server into a quiescent mode with the aid of

SQL Server VDI mode while taking the snapshot. The VDI-enabled snapshot is then bookmarked by kutil. The bookmarked snapshot is then replicated to a remote site as shown in Figure 10 and it can be restored using the kutil *sqlRestore* command. RecoverPoint can mount the snapshot on a backup host or apply to a recovery host

Long-distance replication using RecoverPoint CRR

The following steps are used for long-distance replication and failover using RecoverPoint CRR:

1. Using Kutils, quiesce the database and bookmark the VDI snapshots.
2. Using the RecoverPoint appliance, compress data and replicate to the recovery site.
3. Using the recovery site appliance, decompress data and store the snapshot in the Journal volume.
4. Using RecoverPoint, recover the snapshot from the Journal volume and mount on the recovery host.

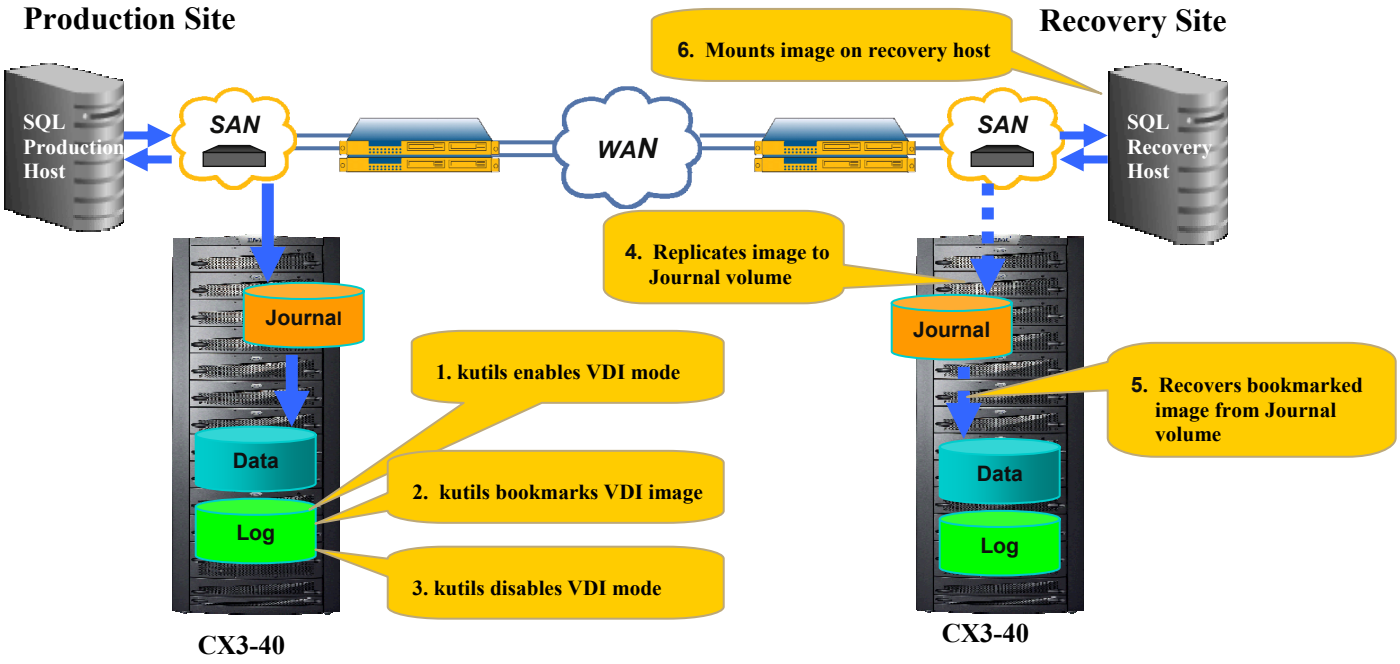

Figure 10. VDI-aided long-distance SQL Server replication

Heterogeneous storage replication

RecoverPoint CRR allows heterogeneous replication over multiple vendor storage systems. For example, it can replicate between a CLARiiON and Symmetrix® or other vendor storage systems from vendors such as IBM, HP, and Hitachi. RecoverPoint consistency group is used for replicating storage replicas from each storage system to remote storage systems. RecoverPoint heterogeneous replication provides disaster recovery support for a tiered storage infrastructure as well as federated database environments.

Federated database replication and recovery

A major advantage of using RecoverPoint CRR is that it allows federated database replication and restart. The federated database environment typically consists of multiple databases, storage platforms, and operating systems from one or more vendors. Replication and disaster recovery of federated database environments are more challenging because they require replication and restart of all member databases and applications to the same dependent-write consistent point of time. This guarantees to maintain data relationships across applications after the restart during a disaster recovery operation. The RecoverPoint

CRR with embedded consistency group technology is well suited for implementing replication and restart in such heterogeneous database and storage system environments. Consistency among databases and related applications is enforced during replication by including associated data LUNs in a common consistency group as illustrated in Figure 12. Each application LUN image is then bookmarked for identifying the correct application image during recovery time.

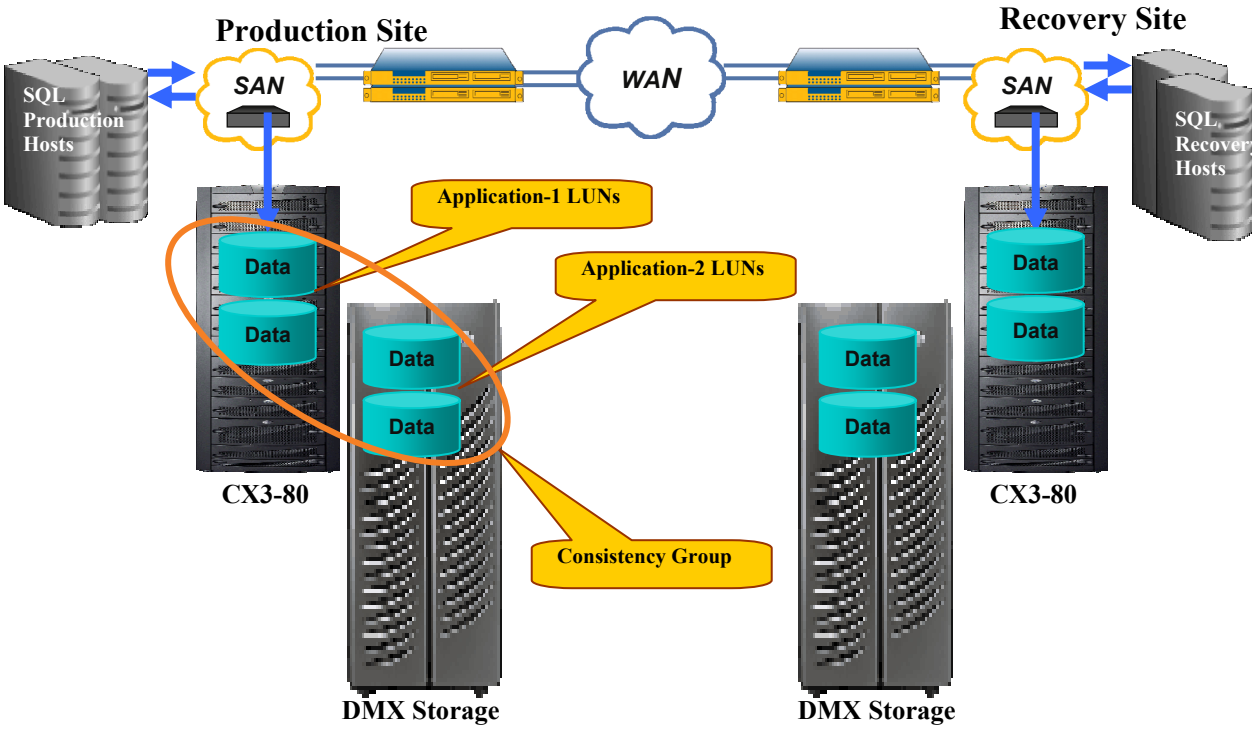

Figure 12. Federated database replication using RecoverPoint CRR

Conclusion

EMC CLARiiON customers are able to leverage CLARiiON array-based or out-of-band replication solutions for their replication requirements. EMC RecoverPoint and CLARiiON storage-system-based replication software such as SnapView, SAN Copy, and MirrorView are powerful value-added technologies to implement efficient and cost-effective disaster recovery solutions for SQL Server deployments. RecoverPoint, with its heterogeneous support and appliance-based replication that offloads work from storage systems, is typically used for large-size data replication in the midrange to high-end models of the CLARiiON platform such as the CX3-80 and CX3-40. Storage-based replication software like MirrorView/A can be an effective fit when deployed in the low-end models of CLARiiON (such as the CX3-10 and AX-4) where cost-efficiency is critical, yet database performance during replication is not as major a factor. Many factors such as available software, distance between production and recovery sites, bandwidth characteristics of the link, and latency need to be taken into consideration when designing replication and disaster recovery solutions.

References

The following online resources can provide more information:

- EMC CLARiiON Family page on EMC.com: http://www.EMC.com/products/systems/clariion.jsp
- Microsoft SQL Server page on Microsoft.com: http://www.microsoft.com/sql/default.mspx

Related documents

The following documents provide additional information on using CLARiiON storage arrays:

- *CLARiiON SnapView Snapshots and Snap Sessions Knowledgebook - Applied Technology* white paper
- *EMC CLARiiON SnapView Clones – A Detailed Review* white paper
- *EMC SnapView for Navisphere Administrator's Guide*
- *Using SAN Copy to Migrate Data to CLARiiON Storage Systems – Applied Technology* white paper
- *EMC CLARiiON SAN Copy – A Detailed Review* white paper
- *MirrorView Knowledgebook: FLARE 26 – Applied Technology* white paper
- *EMC MirrorView/Synchronous for Navisphere Administrator's Guide*
- *EMC RecoverPoint Family Overview – A Detailed Review* white paper
- *Introduction to EMC RecoverPoint 2.4: New Features and Functions - Applied Technology* white paper
- *EMC RecoverPoint 2.4 Deploying with Microsoft SQL Server Technical Notes*