

# Remote Disaster Recovery Concepts for Microsoft SharePoint Server 2010 with Storage Based Replication

## Abstract

This white paper explains the use and value of storage based replication for the purposes of disaster recovery within a SharePoint Server 2010 environment.

September 2011

Copyright © 2011 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate 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](http://EMC.com).

VMware is a registered trademark of VMware, Inc. All other trademarks used herein are the property of their respective owners.

Part Number h8290

# Table of Contents

- Executive summary..... 4**
  - Audience..... 5
- Microsoft Office SharePoint Server 2010 component overview..... 5**
  - Web Front End server..... 5
  - Application Server (Service Applications) ..... 5
  - Microsoft SQL Server ..... 7
- Microsoft Office SharePoint Server 2010 farm topology overview ..... 8**
  - Small server farm ..... 8
  - Medium server farm ..... 9
  - Large server farm..... 9
- Scenarios for configuring a DR environment ..... 9**
  - Failover farm scenario ..... 10
  - Full Farm replication scenario..... 18
  - Stretched farm scenario ..... 23
- Conclusion..... 29**

## Executive summary

Microsoft Office SharePoint Server 2010 is a suite of server-enabled application features aimed at delivering content management, enterprise search, and organizational collaboration capabilities. Due to its rich feature set, SharePoint is fast becoming a mission-critical application for many enterprises. As SharePoint becomes more mission critical to organizations, the necessity to protect the data stored within the SharePoint environment as well as protect the server infrastructure becomes equally as important in both the context of high availability (HA) and disaster recovery (DR.)

SharePoint Server 2010 and underlying technologies it depends on like Microsoft SQL Server provide several layers of high availability within the application, server and storage infrastructure, including redundant server roles, clustering and data replication. The majority of these technologies fit well for providing high availability within a datacenter.

EMC offers several SAN and storage array based remote replication options which can be used to protect a SharePoint 2010 environment in the context of disaster recovery between multiple datacenters. Some of these options include:

- EMC RecoverPoint – Provides synchronous local replication using continuous data protection (CDP), synchronous and asynchronous continuous remote replication (CRR), and concurrent local and remote (CLR) data protection across heterogeneous storage arrays and storage area networks (SAN)
- EMC Symmetrix Remote Data Facility (SRDF) – Primarily provides synchronous (SRDF/S) and asynchronous (SRDF/A) replication between Symmetrix storage arrays. SRDF can also be configured to replicate from one source to two targets in a concurrent or cascaded fashion for the purposes of providing synchronous and/or asynchronous recoverability across three sites.
- EMC MirrorView – Provides synchronous (MirrorView/S) and Asynchronous (MirrorView/A) replication between CLARiiON storage arrays.
- VPLEX Metro and Geo provide for synchronous and asynchronous replication respectively across heterogeneous storage arrays. The VPLEX enables distributed federated access to shared LUNs over distance and across datacenters.

The goal of this white paper is to discuss the use of block Storage or SAN based replication mechanisms in a SharePoint Server 2010 environment. Topics include DR environment scenarios, the SharePoint components that are required to be replicated, and how to perform the recovery at the DR site.

## Audience

This technical note is intended for EMC employees, customers and partners with a need to understand the SharePoint architecture and the possible uses of Storage or SAN based replication to protect the SharePoint environment.

## Microsoft Office SharePoint Server 2010 component overview

SharePoint Server 2010 has several components and server roles that require attention when designing a DR solution. The following is a list and description of the main server roles in a SharePoint environment. There are three main tiers of SharePoint hosts; Web Front End server, Application Server (Service Applications) and Microsoft SQL Server

### Web Front End server

The purpose of the SharePoint web front end (WFE) server is to process client requests. Client requests are processed via Hypertext Transfer Protocol (HTTP) by the Internet Information Service (IIS), included in the Windows operating system. Specifically, the SharePoint WFE will maintain the IIS virtual server instance that clients use to connect to portal content. These IIS virtual servers and related application pools are commonly referred to as web applications. It is under these web applications where SharePoint websites are created. The SharePoint WFE will also hold specific customizations, such as web pages, webparts, web solutions, and web security setting and mappings as optionally modified by an Administrator. SharePoint farms will have at least one, but typically multiple WFEs for redundancy and enhanced performance through load balancing. It is common for the WFE servers to perform additional SharePoint roles, for example to perform the query server role to help scale search capabilities for the SharePoint farm.

### Application Server (Service Applications)

With SharePoint 2010 Microsoft introduced “Service Applications,” a replacement of the Share Service Provider (SSP) architecture of SharePoint 2007. Service Applications represent specific functions to be performed within the SharePoint environment including Search, User Profile, Visio graphics, World automation, Access and Excel among others. Service Applications can potentially store data within SQL Server databases and/or within “application servers” hosting components for the service. Below is a summary of some of the more common service applications including their associated components.

### Search service application

The search service application (SSA) is used to provide enterprise search capabilities for content across SharePoint farms. The search service application can also index content which resides outside of the SharePoint environment, such as file shares, external websites and Microsoft Exchange public folders. There are three kinds of SSA that can be configured with SharePoint 2010, including the native (built-in) search SSA, the FAST query SSA and the FAST component SSA. For the initial publication of this document, only the native SharePoint SSA, commonly referred to as Microsoft SharePoint Server 2010 search, will be discussed.

The built-in search SSA has several components and related databases to provide indexing and query resolution processing for the SharePoint environment. Those components include the Administration, Crawl and Query components

### **Administration Component**

The administration component defines the server and database through which configurations changes are executed and stored for the SSA.

### **Crawl Component**

The crawl component(s) defines the server(s) which perform the crawling of content and initial index creation. The crawl component replaces the SharePoint 2007 equivalent Indexing server and provides additional features and functionality.

The server hosting the crawl component with SharePoint 2010 only stores a temporary copy of the index it creates during the crawl process. This is a change in behavior compared to the index server role with SharePoint 2007, where the index server crawled and maintained a complete copy of the index.

As the index is created, it is propagated to query servers (defined by the Query Component discussed later in this section.) Once the index is propagated the corresponding data is truncated from the crawl server, therefore the index location as defined within the crawl component is considered temporary. The crawl component also references a crawl database which is used to maintain information including content sources, crawl schedules and other data for a given SSA.

Crawl components can be scaled out to provide both improved indexing performance and high availability. Scale out is achieved by configuring multiple servers to host crawl components, which can index shared content in parallel and store information across one or many crawl databases. Processing scale out and high availability can also be achieved by having redundant servers hosting the crawl component, assigned to index the same information while pointing to the same crawl database.

### **Query Component**

The query component(s) define the servers which will hold a copy of the search index and provide search results for the SharePoint environment. The query component also defines the property database associated with crawled data. The property database maintains the metadata associated with the crawled content including properties, history and crawl queues. Another function of the query component is to allow for mirroring and partitioning of the SharePoint index. A mirrored query component allows for a redundant copy of the index for either load balancing or failover functionality. It is also possible to create multiple index partitions for the purposes of segmenting the index across multiple query component servers and property databases. Index partitions allow for a distributed index to help with performance in a scale out architecture.

### **User profile service application**

The user profile service application provides a central location for managing settings across several user based features including user profile properties, my site settings, audiences, organization browsing and profile synchronization. There are three databases within SQL Server that will maintain the data specific to a user profile service, the profile database, synchronization database and social tagging database.

### Secure store service application

The secure store service application is a replacement for the Single Sign-On (SSO) service available within SharePoint 2007. The secure store service is used to maintain and map credentials to external databases and applications for the purposes of authentication and data retrieval from these external entities. The secure store service uses a database to maintain its encrypted credentials and keys.

### Additional service applications

There are many built-in as well as external service applications that can be deployed within a SharePoint 2010 environment. For a more complete list of such applications please see the “Services architecture planning” guide available on technet (<http://technet.microsoft.com/en-us/library/cc560988.aspx>)

### Microsoft SQL Server

All SharePoint servers rely on the backend Microsoft SQL Server databases in order to function and otherwise maintain configuration and portal content information. There are several databases types that are created and used by SharePoint namely in three categories;

- The Configuration database: An essential database which defines and maintains the SharePoint farm configuration
- Content databases: Databases which store user data
- Service application Databases: Configuration and metadata databases for the service applications

The following paragraphs outline the most common database types found in a typical SharePoint 2010 farm.

### Microsoft SQL Server system databases

The system databases include the master, model, msdb and tempdb databases as installed by default and required for an instance of Microsoft SQL Server.

- Master: The master databases maintains instance information including the attached databases, user rights information and configuration settings.
- MSDB: The msdb database maintains alerts, jobs, replication and backup history information.
- Model: The model database is the template from which the settings for new databases being created for an instance are defined.

- TempDB: The tempdb database is a temporary workspace used to maintain result sets for specific kinds of transactions within SQL Server. The data maintained within the tempdb is reset whenever the instance is restarted.

### Configuration database

Each SharePoint Farm maintains one unique configuration database. The configuration database, defines the farm, the servers, security models and bindings between top-level objects such as site collections, webapps and mappings. The configuration database is required for the SharePoint farm to be operational.

### Content databases

Content databases maintain all user data for a given SharePoint site, including (but not limited to) documents, site properties, and user rights. There can be many content databases in a SharePoint environment dedicated to one or multiple web applications. As an example, a content database created by default is the Administration Content database. The Administration Content database maintains information as it relates to the Central Administration site.

### Service Application Databases

SharePoint 2010 offers a number of built-in service applications where the relevant data and settings for that application are stored within Microsoft SQL Server databases. The **Error! Reference source not found.** section goes into detail regarding some of these databases for the most common service application deployments.

## Microsoft Office SharePoint Server 2010 farm topology overview

While it is possible to install all of the necessary SharePoint components on a single server, a single (standalone) server topology is not generally recommended considering most performance and high availability (HA) requirements. When SharePoint components are running across multiple hosts, utilizing the same configuration database, it is considered a SharePoint farm. Based on the previously described server roles, there are several ways in which to design a SharePoint Server farm.

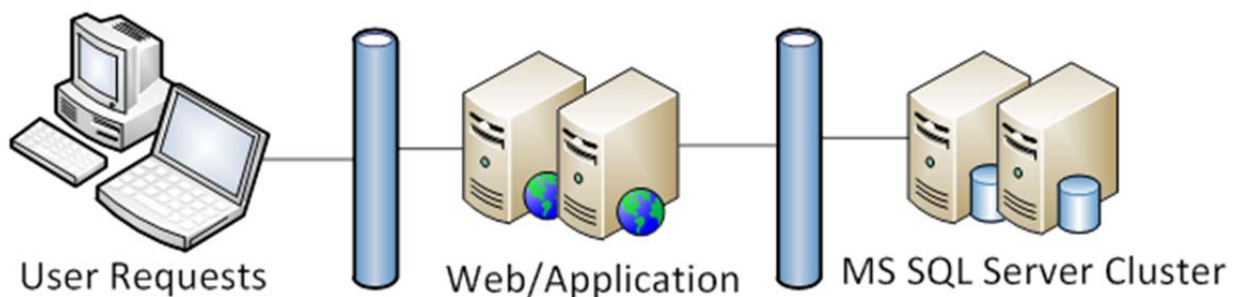
### Small server farm

A small server farm generally consists of one host running Microsoft SQL Server and one host running the web and application server roles.



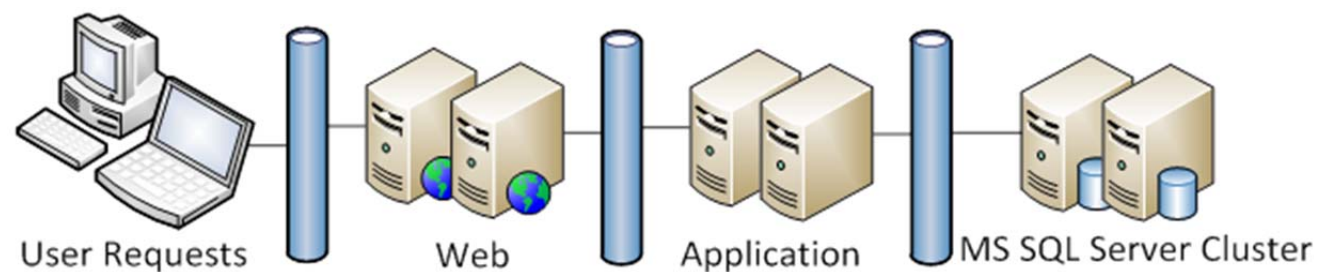
### Medium server farm

A medium server farm may consist of one or more host cluster running Microsoft SQL Server, one or more host running as an application server, and one or more hosts operating as web servers.



### Large server farm

A large server farm will consist of one or more clustered deployments of Microsoft SQL Server, multiple (possibly load balanced) web servers, and multiple application servers dedicated to a specific role.



## Scenarios for configuring a DR environment

Given the various components and topologies that comprise a SharePoint farm, there are several possible options for configuring a DR environment with storage or SAN based replication. The following sections will discuss three specific scenarios: Failover (also known as mirror) farm, full farm replication, and stretched farm. Under each scenario the replication requirements, configuration, and recovery steps will be discussed. For all three scenarios it is expected that both the production and the DR sites are operating under the same Microsoft Active Directory and DNS infrastructure, with the same user accounts and

permissions in place, and sufficient hosts to support these operations available at the DR site.

**Table 1 Brief description of possible DR configurations with SharePoint 2010 using storage replication**

Failover farm scenario	Provides for two independent farms where only content is replicated between the sites. Server roles are maintained separately between the farms.
Full Farm replication scenario	All data and server roles are replicated to a passive site for the purposes of disaster restart.
Stretched farm scenario	Allows for one farm to stretch and otherwise have active server roles enabled between multiple sites.

### Failover farm scenario

The failover farm scenario, also referred to as a mirror farm within some Microsoft documentation, involves a source SharePoint farm that would exist in a production site and a target SharePoint farm that would exist in a remote or DR site. With independent farms implemented in both sites, the web, query, service application, database, and index servers will be maintained separately in each of the farms. While it is possible to replicate several types of SharePoint databases, generally speaking, only the content databases and specific service application databases will be replicated. Any customizations to web servers or indexed data will need to be configured and maintained separately in each farm. Users can be redirected between the farms as required via DNS updates or network load balancers.

## Failover Farm

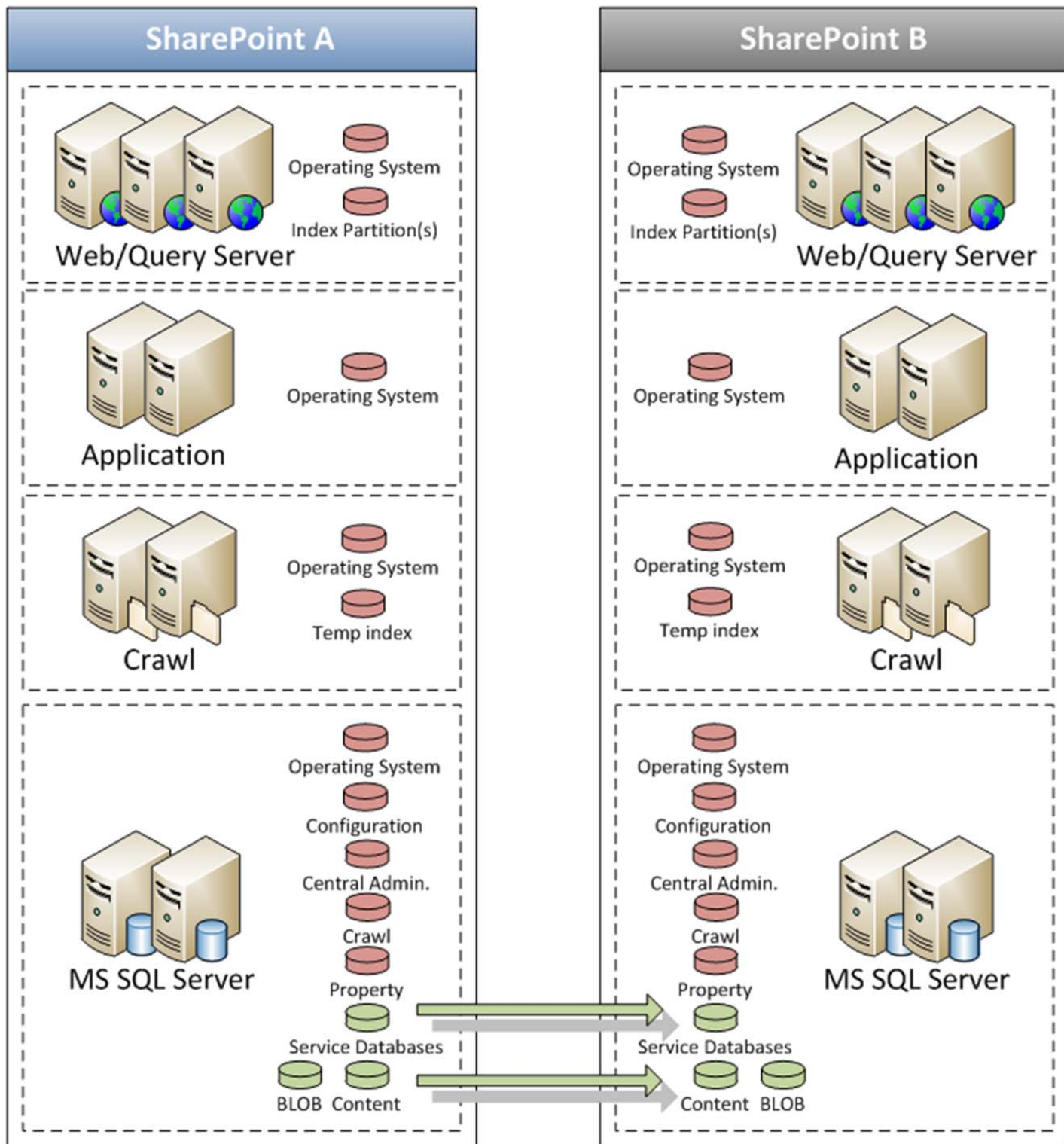


Figure 1 Failover farm scenario

Main considerations for the failover farm scenario include:

- Support for either Synchronous or Asynchronous storage replication.
- Allows for multiple active farms for the purposes of testing, disaster recovery and SharePoint patching/upgrades.
- Content changes can be incrementally refreshed between the farms.
- Content refresh requires a temporary outage to the secondary farm while incremental storage changes are replicated.
- The search service application maintained within the failover farm has the potential to perform incremental crawls across content database refreshes.

## Configuring the failover farm environment

### Web front-end server

Specifically for the *failover farm scenario*, the web front-end servers will be maintained independently between the farms, thus, only customizations will need to be maintained and applied to the web servers at the DR site. This could be done by maintaining a change control system and applying customizations to both sites as they are made. Additionally, similar to SharePoint 2007, SharePoint 2010 offers a solutions framework that enables packaging of customizations for simplified and automated deployment.

Generally speaking there is a minimal amount of data that needs to be maintained or replicated on each of the web servers. The customized data of the highest interest includes:

- Custom web parts
- Custom templates
- Custom web pages
- Add-in software
- SSL certificates

Also, as the DR SharePoint server farm configurations will be maintained separately, the design does not have to completely mirror the production site. The number and types of front-end servers, the network names of the hosts, as well as the physical hardware can all differ. It is critical that the following items be duplicated at the target site:

- Windows operating system, including installed service packs and hotfixes
- SharePoint program install, including installed service packs and hotfixes
- Active Directory replication
- IIS settings for the web front-end servers, including any customization.

For DR purposes it is recommended that the virtual server name and URL of the web application at the DR site match the production site. Then for client redirection, DNS can be modified to reference the web servers that host the appropriate URL in the failover farm.

There are also global hardware load balancer devices which can automatically re-point network names in the event of a failover, and these are seen as a necessity in enterprise solutions.

As a rule of thumb, when creating and editing pages at the production site, it is recommended to use relative paths to link items within the site. In the event that the content database is used at the failover farm under a different web application name, relative paths will ensure that links and images will continue to work.

### **Back-end SQL Server databases**

As the configuration database is being maintained independently between the farms in this scenario, it is only necessary to replicate specific databases to the DR site. Most importantly the content databases should be replicated and attached to web applications at the DR site. Certain service application databases can also be replicated and used at the DR site.

Because only the content and some service application databases are to be replicated to the DR site, there is the opportunity to maintain a search service application instance in the failover farm using local databases. The search service application in the failover farm can be used to crawl content so that search will be available for web applications at the DR site.

An important consideration when replicating content databases under this scenario is the need to periodically register new site collections within the configuration database at the DR site.

Each time a new site collection is added to a given content database, information about that site collection is also added to the configuration database. As new site collections are added to the content databases at the production site, the replicated copies of the content databases will also contain the new site collection information. The configuration database at the DR site, however, will not automatically get updated with this new site collection information. In order for the configuration database to be updated at the DR site, the content databases must either be deleted and re-added to the web application, or refreshed using the “.RefreshSitesInConfigurationDatabase()” method. The act of re-adding the database to the web application or using the aforementioned method will force the update of the site collections into the configuration database at the DR site.

More information regarding the .RefreshSitesInConfigurationDatabase() method can be found online at <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.administration.spcontentdatabase.refreshsitesinconfigurationdatabase.aspx>

### **Application servers**

A number of databases supporting application server roles can be replicated as a part of the failover farm scenario. Because the failover farm scenario similarly matches a Microsoft failover farm topology using SQL Server based log shipping or database mirroring, it is recommended to

follow the Microsoft best practices outlined for this scenario in the technet document “Plan for disaster recovery (SharePoint Server 2010)” (<http://technet.microsoft.com/en-us/library/ff628971.aspx>) Additionally, a more detailed list of databases that are supported to be replicated under the failover farm scenario can be found within “Database types and descriptions (SharePoint Server 2010)” at <http://technet.microsoft.com/en-us/library/cc678868.aspx>

### **Search Service Application considerations**

The servers hosting any service applications will be independently maintained within a secondary farm at the DR site. As indexing will be maintained separately at the failover farm in the DR site, the query and crawl servers will also be independent. Therefore, it is not necessary to replicate any data, be it index or database as it relates to the search service application. The query and crawl servers at the DR site can be designed to match the requirements of the failover farm.

Each content database as replicated from the Primary SharePoint farm maintains a “change log” which is the “EventCache” table in the content database. The change log is essentially a journal of updates against that content database and is used by the search service application to determine changes which need to be indexed since the last incremental crawl process. Entries within the change log are retained for 60 days by default. The retention period can be modified at the Web Application level by editing the “ChangeLogRetentionPeriod” property. More information about the change log can be found on MSDN (<http://msdn.microsoft.com/en-us/library/bb417456.aspx>)

The long change log retention policy allows for this list of changes from within the production farm to be available to the search service application running within the failover farm between content databases refreshes. Additionally, the content database unique identifiers or GUIDs are persisted across database detach/attach or “.RefreshSitesInConfigurationDatabase()” operations. The combination of these factors makes it possible to continually perform incremental crawl operations across content database refresh operations while using storage replication.

### **Storage replication considerations for the failover farm scenario (SRDF example)**

For the initial database server setup, a decision needs to be made regarding the type of replica to be used within the failover farm. For example, on the Symmetrix platform using SRDF, the failover farm can be configured to utilize an SRDF target copy (R2) of the replicated databases, or a TimeFinder®/Mirror, TimeFinder/Clone, or TimeFinder/Snap copy as created off of the SRDF R2.

When mounting R2 devices directly, it is important to understand that once replication is resumed, the R2 database devices become write disabled and will now be updated from a storage perspective. This will prohibit the use of these devices for SharePoint, which means the farm at the DR site may not be accessed. Only once replication is suspended, and the R2 devices are again made read/write, can the failover farm be accessed. During access of R2 devices (on a suspended link), any pending writes from production are only staged at the production Symmetrix and so are not available on the DR site, increasing the RPO). Once access is disabled and the SRDF link resumed, all pending writes are transferred to the DR site.

When utilizing a TimeFinder copy created from the R2 devices, it is possible to configure the environment such that the DR SharePoint site can remain online, between refresh cycles while maintaining remote replication. The DR site will be available for remote access as the TimeFinder copy can remain ready while the R2 devices are write disabled during replication. In order to refresh the data within the failover farm, the TimeFinder copy can be incrementally updated from the R2 devices. Any refresh of the data on the TimeFinder copies would require that the SharePoint services be stopped so that the databases can be detached and file systems unmounted before the refresh. Also in this configuration, the search service application at the DR site can be used to crawl the available content databases and therefore update the indexes for use in the failover farm.

To implement the TimeFinder scenario, it is necessary to maintain an additional (third) copy of the content databases. Usage of TimeFinder/Snap can alleviate the space concerns of a third copy, however, using snaps against an R2 device should be done carefully with performance implications in mind. In the event of a failover event, the databases will either need to be detached and redirected to the R2 LUNs or the TimeFinder copy will need to be refreshed one last time from the R2 devices, which will increase the RTO of the failover.

### **Device mounting considerations**

If volumes are not properly unmounted from a Windows host prior to being updated at the storage system level, there is the possibility that Windows will maintain cached file system information about the prior state of the volumes. This can create problems for LUN-based replication technologies, since while data is changing on the volume from the storage perspective, Windows may be maintaining its own view of the data from the server perspective. This inconsistency between the Operating System view of the LUN and the storage state may lead to corruption. It is therefore recommended that when LUN based replication mechanisms are active, the volumes on these LUNs should be unmounted or otherwise masked from their respective hosts at the DR site. Once replication is stopped, the devices should then be unmasked or otherwise mounted to the DR hosts.

An alternative to masking devices for a standalone host would be properly unmounting the volume using either `mountvol` with the `/P` switch, or using the Symmetrix Integration Utilities (SIU.) If the SQL Server host is clustered and the devices are physical disk resources, they should not be unmounted with `mountvol /P` or SIU. To properly manage volume cache for a clustered physical disk resource, the resource should be taken offline. The act of taking the disk resource offline will clear any stale volume cache. In general the disk resources should be taken offline prior to refreshing data and remain offline until replication is stopped and the LUN is read/write enabled.

### **Initial setup**

The following steps can be used to create the initial association between a replicated content database and a web application within the farm at the DR site.

1. Present the appropriate replica LUNs to the SQL Server at the DR site.

2. Mount the LUNs to the appropriate drive letters or mount points.
  - a. Keeping the same enumeration (logical file location) as production simplifies the attach procedure.
3. Attach the databases to SQL Server, by script or by SQL Server Management Studio
  - a. CREATE DATABASE Content\_DBName ON (FILENAME = 'S:\primary\_datafile.mdf') FOR ATTACH
4. Create the web application(s) that will support the replicated content database(s).
  - a. The name of the replicated content database can be entered at the time of either extending or creating the web application via Central Administration.
  - b. Use the same web application names as Production.
5. Ensure all customizations from the production farm have been added to the web servers supporting the web applications for the replicated content databases.
6. If the content database was not available at the time the web application was created or extended:
  - a. From the Central Administrative website go to **Application Management**.
  - b. Select **Manage Content Databases**.
  - c. Ensure the correct web application is selected.
  - d. Click on the content database(s) for the specified web application.
  - e. Select the **remove content database** checkbox for any default databases that may have been created.
  - f. Once unwanted databases are removed, select **Add Content Database** and enter the name of the appropriate replicated database.
  - g. Site Collections should now be available for connectivity and search crawls.

### Refreshing the failover farm content databases

Updating the data for the failover farm requires that the content databases to be refreshed must first be detached from SQL Server. There are several options that can be used to remove connections from the SQL databases for the refresh.

The first option can simply involve forcing the detach of the content databases from SQL Server by setting the databases into single user mode. Once the databases are in single user mode they can be detached so that the storage replica can be refreshed. Forcing the detach in such a way will not gracefully disconnect users from the SharePoint farm. A safer process for detaching the content databases prior to the data refresh is included in the following steps.

To stop user access to the farm and begin a replica refresh (SRDF example):

1. Quiesce the farm. The quiesce farm feature is designed to drain connections for session-aware applications, like Infopath forms. For general connections that are not session-aware, the users will still be able to connect, read and modify data for the site.

- a. `stsadm -o quiescefarm -maxduration <time in minutes>`
2. Use a site collection lock. A site collection lock can be used to truly disconnect all users from a site. The lock, however, does not release all SharePoint-related handles open against the SQL databases.
  - a. `stsadm -o setsitelock -url http://mywebsite -lock noaccess`
3. Alternatively access to a site can simply be halted by stopping the appropriate IIS website.
  - a. `appcmd stop site "[SiteName]"`
4. At this point we can detach the content database from the web application via the `stsadm deletecontentdb` command. The reason for deleting the content database from the web application is so site collections can be added to the configuration database following the refresh
  - a. `stsadm -o deletecontentdb -url http://mywebsite -databasename [dbname] -databaseserver [sqlserver]`
5. An alternative to Step 4 (deleting the content database) would be to issue the following from powershell:
  - a. `$db = get-spdatabase | where {$_.Name -eq "content_database_name"}  
$db.RefreshSitesInConfigurationDatabase()`
6. Following a site collection lock and content database deletion (or alternate configuration database refresh,) it is safe, from a user perspective, to use `single_user` mode and then detach the SQL databases. SharePoint services running on the front-end and index servers, however, may still have sessions open with the database. These sessions will be lost and may cause errors to be logged due to the unexpected detach of the SQL databases. These errors are not a major concern, and once the database is again available, the appropriate services will reconnect. To avoid the errors and more cleanly complete the quiesce process, the following can optionally be done:
  - a. Recycle the appropriate IIS application pool:
    - i. `appcmd recycle apppool "[site name]"`
  - b. Stop SharePoint Timer Service:
    - i. `net stop "SharePoint 2010 Timer"`
    - ii. Or alternatively – `'sc \\WFEHostName stop SPTimerV4'`
7. It is now safe to detach the appropriate databases:
  - a. `sp_detach_db 'Contet_DBName'`
8. Once the databases are detached, the file systems need to be either unmounted or masked as previously discussed in “Device mounting considerations.”

9. After the file systems are unmounted/masked, the appropriate SRDF or TimeFinder command can be issued to refresh the data on the R2 or TimeFinder LUN. To give an example with TimeFinder/Mirror.
  - a. Symmir -g ContentDBs est -rdf -nop
  - b. Symmir -g ContentDBs verify -rdf -synched -i 15
  - c. Symmir -g ContentDBs split -rdf -consistent
10. At this point the previous process can now be reversed, beginning with the mounting/unmasking of the appropriate LUNs.
11. Attach the databases to SQL Server:
  - a. CREATE DATABASE Content\_DBName ON (FILENAME = 'S:\primary\_datafile.mdf') FOR ATTACH
12. Start the SharePoint Timer Service:
  - a. net start "SharePoint 2010 Timer"
  - b. alternatively - 'sc \\WFEHostName start SPTimerV4'
13. If the content database was removed as a part of step 4, add the content database back to the web application. This will register any new site collections in the configuration database at the failover farm.
  - a. stsadm -o addcontentdb -url http://mywebsite -databasename [dbname] -databaseserver [sqlserver]
14. Remove the site lock:
  - a. stsadm -o setsitelock -url http://mywebsite -lock none
15. If the alternative to stop the IIS site was used, start the service:
  - a. appcmd start site "[SiteName]"
16. Unquiesce the farm:
  - a. stsadm -o unquiescefarm

### Full Farm replication scenario

Under the Full Farm replication scenario, the DR site is intended to host the production farm in its entirety. This requires that the DR site must be designed to exactly replicate the production site. This will allow index information to be replicated; however, operating system information must also be replicated.

## Full Farm Replication

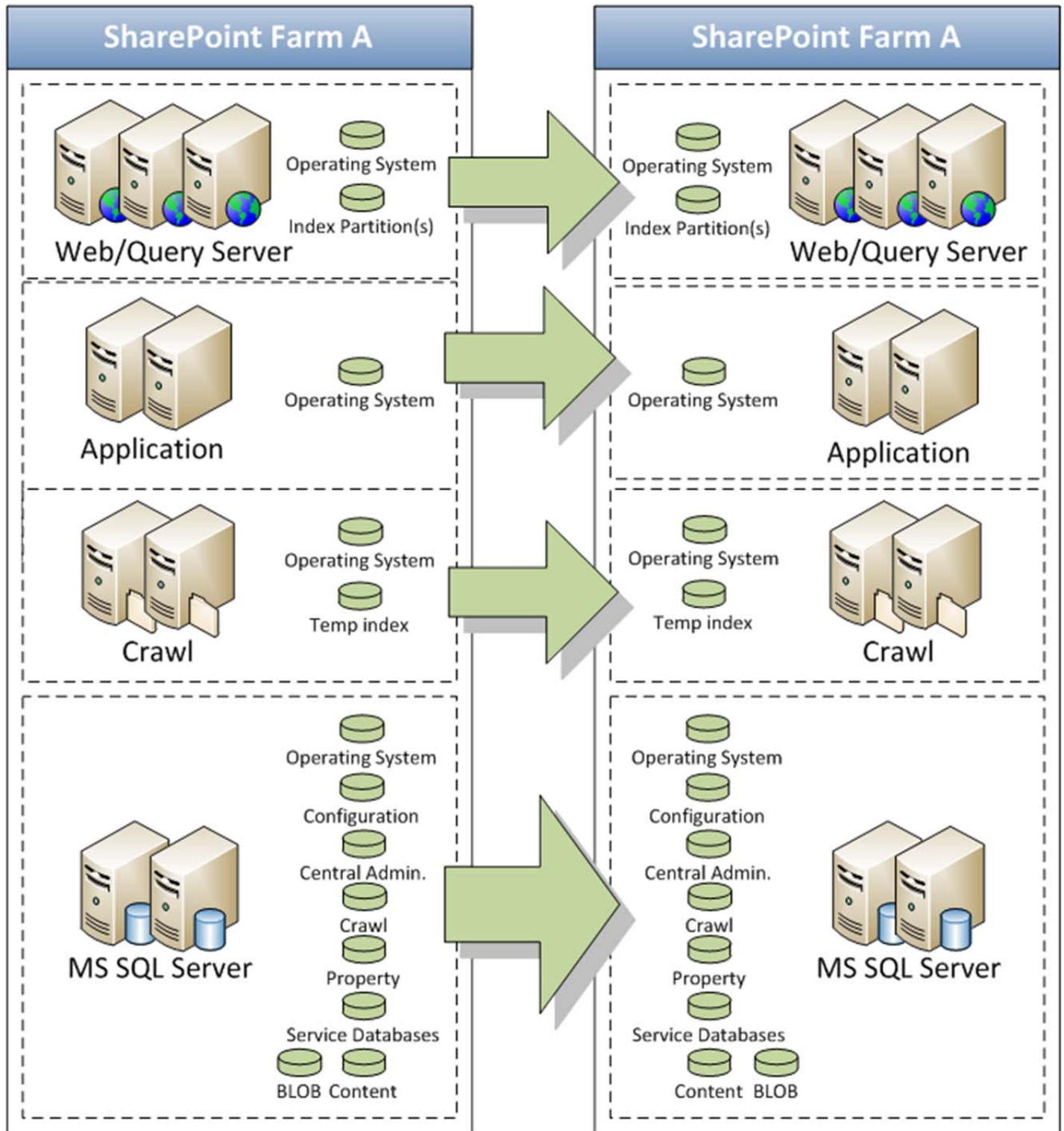


Figure 2 Complete replication scenario

Considerations for the complete replication scenario include:

- Support for either Synchronous or Asynchronous storage replication.
- Provided the SQL databases and the Search indexes are replicated within the same consistency group, a full crawl of content to rebuild the indexes will *not* be required.
- Operating System data will need to be replicated. Virtualization technology such as VMware or Microsoft Hyper-V can aid in encapsulating the operating systems for the purposes of disaster recovery across heterogeneous environments.
- If the network subnet in the DR site differs from that in the production site, IP Addresses for the replicated OS images and clustered IP resources will need to be updated to reflect the new subnet. DNS will also need to be updated to reflect the new IP assignments.

### ***Configuring the disaster recovery environment***

#### ***Web front-end server***

Generally speaking there is a minimal amount of data that needs to be maintained or replicated from the web servers. The customized data of the highest interest includes:

- Root directories for virtual servers
- Custom web parts
- Custom templates
- Custom web pages
- Add-in software
- SSL certificates

For the *full farm replication scenario* specific operating system information will be replicated along with the customizations in order to exactly re-create the server environment at the DR site. There are several methods that can be utilized to perform this task including backups that include the system state, and imaging tools, however, this document will only cover direct OS replication using LUN based replication, enabled by either virtualization technology or configuring physical hosts for SAN boot.

By booting servers from the SAN, it is possible to replicate the entire operating system using SAN or storage based replication technologies. This allows any changes made to the source site to be automatically replicated to the DR site. Booting from the SAN has specific requirements, including the need for the server hardware at the remote site to exactly match the server hardware from the production site. This is required so that the replicated operating system can be utilized without having to make driver changes to the OS.

A method to avoid this physical hardware restriction between the primary and DR site is to utilize virtualization technology. By using virtualization technology, the operating system will have the same virtual hardware configured for both the source and DR site, even though the physical hardware may differ. Also, by utilizing virtualization, the possibility for consolidating the DR environment onto less physical hardware exists.

Additionally, virtualization technologies offer integrated solutions to help automate the disaster recovery process. VMware ESX environments can leverage technologies such as Site Recovery Manager (SRM) to integrate with storage replication for the purposes of disaster recovery restart. SRM integrates with and has adapters for RecoverPoint, SRDF as well as MirrorView replication technologies.

Microsoft Hyper-V can also leverage integrated solutions for disaster restart, such as EMC Cluster Enabler. EMC Cluster Enabler creates a geographically dispersed clustering environment to enable both HA and DR across datacenters for Microsoft Failover clusters. Cluster Enabler offers modules to support RecoverPoint, SRDF and MirrorView replication technologies.

### ***Back-end SQL Server databases***

For this scenario it is possible to replicate all SQL Server databases related to SharePoint. Because RecoverPoint, SRDF and other array or SAN based technologies perform replication at the storage level, all SharePoint databases across all SQL instances can be replicated together, consistently, to the DR site, leaving a common RPO for the entire SharePoint farm. For indexing to work properly upon failover, and not potentially require a full crawl, it is recommended for the crawl database(s), property database(s), and index file system(s) as maintained on the query servers to be replicated together using consistency technology.

Additionally, the incremental relationship between a content database and the index crawl process is determined by the SharePoint change log. As items are modified within a content database, time stamped pointers to the actual modifications are inserted into the change log. The incremental crawl process can then query the change log and only index items determined to have changed since the last crawl. The change log is maintained within each content database, specifically, in the eventcache table. When considering this relationship between the content database change log and the indexing process, it is recommended that the content databases be replicated consistently with any indexed data being replicated.

As discussed in the “Failover farm scenario” each time a new site collection is added to a given content database, information about that site collection is also added to the configuration database. Given the relationship between sites within a content database and the configuration database, it is generally recommended to also replicate the configuration database consistently with all content databases. While it is not required to replicate content databases consistently with other content databases, replicating the configuration database would require that all content databases be replicated together consistently.

Should the configuration database be replicated separately from the content databases, there is the possibility for sites to exist with the content databases that might not exist within the configuration database. Should this happen, users may not be able to connect to those sites. To correct this problem, the `.RefreshSitesInConfigurationDatabase()` method will need to be executed, or the content database would have to be removed and re-added to its web application in order to register the missing sites with the configuration database.

More information regarding this process is discussed in the “Failover farm scenario” section.

Since it is possible to attach SQL Server databases, including the system databases, to a different host running SQL Server, it is not an absolute requirement to boot and replicate the host OS running SQL Server from the SAN. A good example of a database that generally should not be replicated is the tempdb. The tempdb can be active in SharePoint environments therefore it is not ideal, for performance and bandwidth reasons, to replicate the tempdb. If the SQL instance is not booted from the SAN or otherwise virtualized, the following requirements should be followed:

- The SQL instance name at the DR site should match the instance name from the production site. By matching the instance name, no configuration changes will need to be done to the farm.
  - It is supported to rename a host or virtual server (if clustered) running SQL Server. More information can be found within *SQL Server Books Online*.
  - It is also possible to utilize SQL Aliases for the purposes of redirecting applications like SharePoint 2010 to instances of SQL Server with different names.
- The correct user accounts related to SharePoint are replicated to the DR instance of SQL.
  - The master database maintains user accounts as they relate to the SQL instance. If the master database is replicated to the DR SQL instance then all user accounts, and specifically for SharePoint the service accounts, will be available. To manually migrate users between SQL instances refer to Microsoft KB article 246133: <http://support.microsoft.com/kb/246133>

### ***Application servers***

The same considerations as discussed for the web front-end servers, with respect to virtualization and boot from SAN, can be applied to the application servers as well. If the application servers are replicated to the DR site then the installation of their respective server role, would not have to take place following the failover.

Specifically for the search service application, it is possible to maintain the search indexes and incremental crawl capabilities when failing over the complete farm to the DR site. To ensure consistency between the content databases, search databases and indexed data, it is recommended to place all of these entities into a consistency group to ensure they are replicated to the same dependent write point-in-time.

### **Full farm replication example**

With complete replication, there are a number of ways to replicate and continually update the DR site. Depending on the chosen method, the setup, maintenance, and testing of the environment can vary dramatically.

The simplest way to replicate the environment under this scenario is to utilize a virtualized environment and to have all SharePoint hosts and data replicated consistently. With this scenario the virtual hosts can be configured to utilize the replicated target devices. Since

all OS data, index, content, and configuration databases are replicated, it is possible to simply Read Write enable or otherwise enable access to the remote devices and boot the virtual environment in the event of a failover. Ideally the restart of servers and services would be sequenced so that the SQL Server Instance would be started first, followed by the web servers then application servers. When virtualizing a VMware ESX environment, Site Recovery Manager could be used to help with automatically enabling access to the remote LUNs, and restarting the environment in the proper order.

The main concern with this scenario is the network setup and whether a VLAN is available at the DR site to accept the same IP address and subnet information as configured and replicated from the production site.

If a VLAN cannot be used, the replicated OS copies at the DR site will need to be modified with the appropriate network information. This can be done as a part of maintenance and testing of the environment, or done as a part of the failover procedure.

Host name resolution, including DNS, at the DR site will also have to be adjusted accordingly depending on whether VLANs can be used in the environment.

Additional details regarding full farm replication can be found within the white paper “EMC Continuous Protection for Virtualized SharePoint 2010 Farms” available on powerlink ([http://powerlink.emc.com/km/live1/en\\_US/Offering\\_Technical/White\\_Paper/h8139-protection-virtualized-sharepoint-wp.pdf](http://powerlink.emc.com/km/live1/en_US/Offering_Technical/White_Paper/h8139-protection-virtualized-sharepoint-wp.pdf)) The white paper details an example of protecting a SharePoint Server 2010 farm using VMware ESX and Site Recovery Manager with RecoverPoint CDP and CRR technologies.

### Stretched farm scenario

In the stretched farm scenario, the DR site is intended to support the production site with separate servers that exist within the same farm. With this configuration the hosts from the DR site will be within the same farm as the servers of the production site, operating with redundant roles. All SQL Server databases will be replicated, which will also allow the possibility to replicate all farm databases including the configuration, content and those supporting service applications. Web, query, crawl and application servers will be maintained independently, but within the same SharePoint farm. Figure 3 provides an example of this scenario.

## Stretched Farm

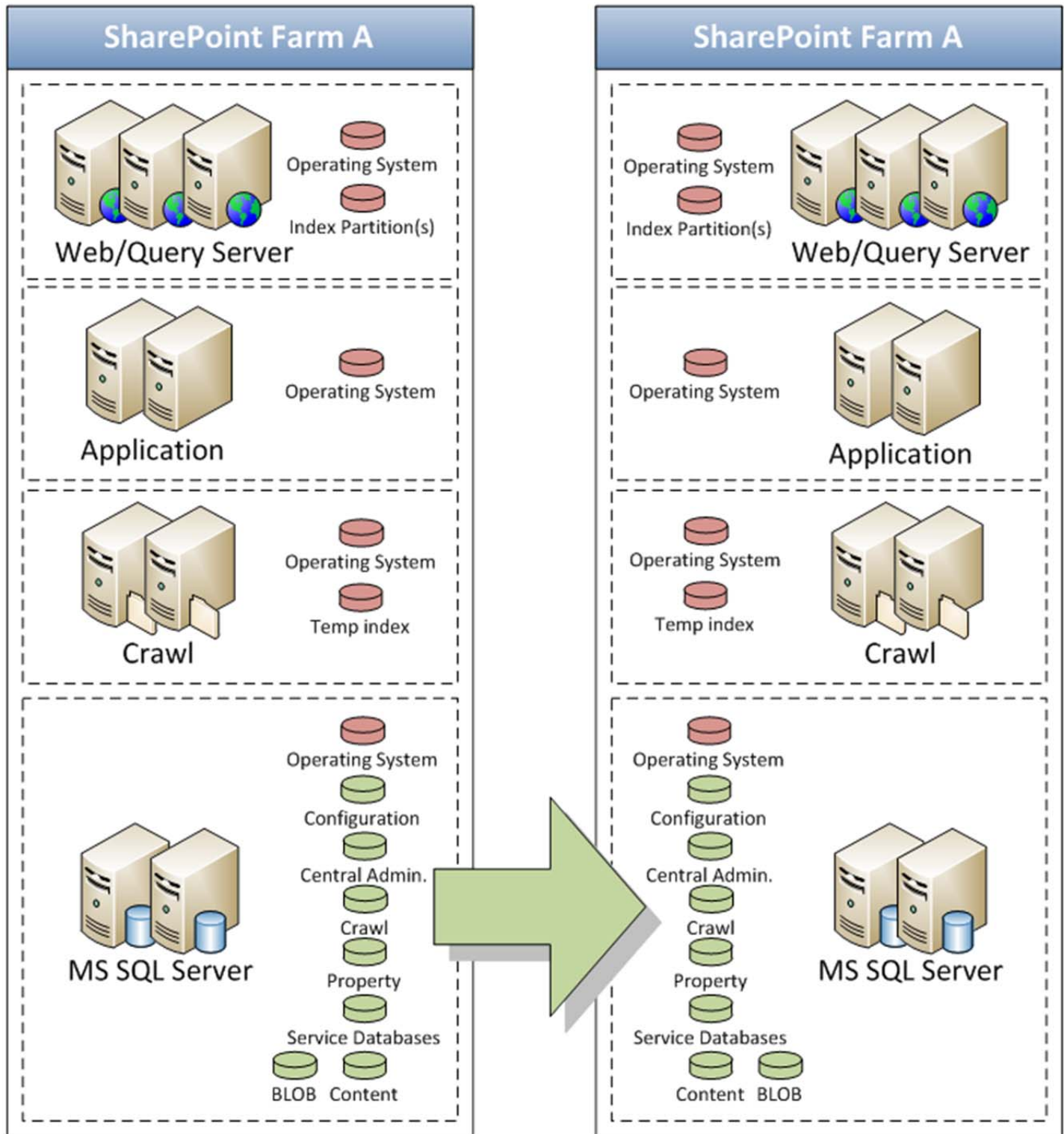


Figure 3 Stretched farm scenario

Considerations for the stretched farm scenario:

- Synchronous storage replication
  - Synchronous replication provides the potential to maintain the search service application with redundant query and crawl components at the secondary site. Redundant web front end servers can also be maintained and active depending on the round trip latency between the Primary and DR data centers.
  - Web front end servers and search service application roles maintained within the DR site can also be marked as passive until such time as the content databases are active within the DR datacenter. Access to the web front end servers can be controlled by network load balancing technologies, while access to the search query components can be managed by adding them as “failover-only” mirrors.
- Asynchronous storage replication
  - Asynchronous storage replication can be supported in a stretched farm scenario, however, it should be expected for the web front end servers and application servers maintained within the DR site to be passive where round trip network latency exceeds Microsoft recommendations.
  - Depending on the nature of the failover, the indexes maintained on the redundant query servers may not be consistent with the storage based copy of the search related databases. A full crawl may be required to resolve any conflicts should they exist.
- When considering network bandwidth and performance limitations across a WAN, Microsoft provides limited support for geographically separated servers within a farm. Some of the restrictions include:
  - Network latency of no more than 1 millisecond between a web server and the database server.
  - All server roles involved in shared services (database, index, query and excel) are located in the same data center.
  - Servers within a farm cannot cross time zones
- For additional restrictions and best practices from Microsoft, please see the “Optimizing Office SharePoint Server for WAN environments” article available on TechNet.

## Configuring the disaster recovery environment

### *Web front-end server*

Specifically for the *stretched farm scenario*, as the web front-end servers will be maintained within the same farm, only customizations will need to be applied to the web servers at the DR site. This could be done by maintaining a change control system and applying customizations to both sites as they are made. Additionally, similar to SharePoint

2007, SharePoint 2010 offers a solutions framework that enables packaging of customizations for simplified and automated deployment.

Generally speaking there is a minimal amount of data that needs to be maintained or replicated on each of the web servers. The customized data of the highest interest includes:

- Custom web parts
- Custom templates
- Custom web pages
- Add-in software
- SSL certificates

Depending on the desired configuration, distances between the sites, networking infrastructure available (VLAN for instance) as well as performance requirements, the web servers at the DR site can be active within a Network Load Balancing (NLB) infrastructure so that they support the same web applications as the production site.

An alternative would be to leave the DR servers inactive and left out of a NLB infrastructure until a failover needs to occur. DNS can then be modified to redirect users to the web servers at the DR site

Also, in order to have redundancy for the Central Administration site it will need to be installed on a web server at the DR site. Installing the central administration web application can be done by going into the “advanced settings” within the SharePoint Product and Technologies Configuration Wizard and selecting the radio button to host the Central Administration web application. Alternate access mappings for the central administration web application can be used to help manage connectivity to the appropriate server.

### ***Back-end SQL Server databases***

For this scenario it is possible to replicate all SQL Server databases related to SharePoint. Because RecoverPoint, SRDF and other array or SAN based technologies perform replication at the storage level, all SharePoint databases across all SQL instances can be replicated together, consistently to the DR site, leaving a common RPO for the entire SharePoint farm.

As discussed in the section “Failover farm scenario,” each time a new site collection is added to a given content database, information about that site collection is also added to the configuration database. Given the relationship between sites within a content database and the configuration database, it is generally recommended to also replicate the configuration database consistently with all content databases. While it is not required to replicate content databases consistently with other content databases, replicating the configuration database would require that all content databases be replicated together consistently.

Should the configuration database be replicated separately from the content databases, there is the possibility for sites to exist with the content databases that might not exist within the configuration database. Should this happen, users may not be able to connect to those sites. To correct this problem, the `.RefreshSitesInConfigurationDatabase()` method will need to be executed, or the content database would have to be removed and re-added to its web application in order to register the missing sites with the configuration database. More information regarding this process is discussed in the “Failover farm scenario” section.

Since it is possible to attach SQL Server databases, including the system databases, to a different host running SQL Server, it is not a requirement to boot and replicate the host OS running SQL Server from the SAN. A good example of a database that generally should not be replicated is tempdb. The tempdb can be very active in SharePoint environments; therefore it is not ideal, for performance and bandwidth reasons, to replicate the tempdb. If the SQL instance is not booted from the SAN or otherwise virtualized, the following requirements must be followed:

- The SQL instance name at the DR site should match the instance name from the production site. By matching the instance name, no configuration changes will need to be done to the farm.
  - It is supported to rename a host or virtual server (if clustered) running SQL Server. More information can be found within SQL Server Books Online
  - It is also possible to utilize SQL Aliases for the purposes of redirecting applications like SharePoint 2010 to instances of SQL Server with different names.
- The correct user accounts related to SharePoint are replicated to the DR instance of SQL
  - The master database maintains user accounts as they relate to the SQL instance. If the master database is replicated to the DR SQL instance then all user accounts, and specifically for SharePoint the service accounts, will be available. To manually migrate users between SQL instances refer to Microsoft KB article 246133: <http://support.microsoft.com/kb/246133>

It is also possible to utilize geographically dispersed clustering technologies, like EMC’s Cluster Enabler to create SQL instances that can span between the Primary and DR sites.

### ***Application servers***

Similar to the web server considerations in this scenario, it is necessary to decide whether application servers are hosted within the DR site to process data during normal operation. In the case of the search service application, SharePoint 2010 offers the ability to have both redundant Query and Crawl components that can be configured and installed at the DR site. The benefit of having the application servers available beyond serving queries is the fact that index propagation can occur at all times. Then in the event of a failover to the DR site, search can continue to work without the need to reindex the data.

While there are benefits, it may be problematic to host the query servers at the DR site. When considering index propagation and user queries, which may very well reference data from an index server and databases at the production site, it is possible for considerable bandwidth to be used across the WAN. Microsoft also provides limited support for geographically separated servers in a farm. Microsoft recommends that all servers that host a shared services component exist in the same data center as the database server. The distance between sites, network bandwidth, and performance requirements should be taken into account when designing the solution.

### **EMC Cluster Enabler considerations**

A possible configuration to help automate the replication and failover of SharePoint resources is to utilize EMC's Cluster Enabler for Microsoft Failover Clusters (Cluster Enabler or CE for short). CE can be used in conjunction with synchronous or asynchronous replication to help automate the failover of SQL instances or highly available Hyper-V virtual machines to a DR site.

CE, however, does have some limitations with respect to replicating data outside of its configured Microsoft Failover Clustering group. Microsoft Cluster Service utilizes resource groups as the granularity for moving resources between hosts. CE subsequently organizes the physical disk resources within the Failover Clustering resource groups into device groups. To use SRDF as an example, because SRDF/A devices must replicate and therefore be managed together, all devices within an SRDF/A RA group must be added to the same device group. SRDF/CE will then add all physical disk resources as defined within the device group into a single MSCS resource group.

When considering a distributed environment like a SharePoint farm, it will not be possible to place all of the necessary devices that require consistency into the same MSCS group for Cluster Enabler to manage. This is problematic in the *complete replication* and *stretched farm* scenarios as there may be a requirement to replicate multiple SQL Server instances and/or the index server data consistently in order to fail over search seamlessly to the DR site.

This is not to say that Cluster Enabler cannot be used in a SharePoint environment. CE can still be used to replicate and help automate the failover for all of the SQL Server databases or Hyper-V virtual machines to the DR site. It should be expected, however, to have to configure consistency groups outside of Cluster Enabler's control where consistency across multiple SQL instances or service application servers is required.

Another consideration with respect to CE is tempdb replication. SQL Server under MSCS requires that all LUNs that support SQL exist within the cluster. This requirement forces CE to also include the tempdb LUNs within its device groups. This means that the tempdb traffic will be replicated while under CE control, which may have performance or bandwidth ramifications in some environments.

## Conclusion

Considering the various options available for protecting a SharePoint environment, it is important to match the business requirements with the most appropriate DR scenario. Some of the business requirements may include the RTO for specific components of the SharePoint environment. For instance, it may be a requirement for content to be available almost immediately following a DR event; however, having search available may not be required for several days. The RPO may also impact requirements. For example, it may be required to have a zero data loss synchronous solution, or perhaps a solution that can replicate the entire SharePoint farm to the same consistent point in time.

Other considerations may include cost, in the form of network bandwidth provisioning and remote hardware availability, as well as the complexity of managing the DR solution. An excellent example regarding network bandwidth is the decision to replicate indexing traffic to the DR site. The obvious benefit is the ability to quickly resume search following a DR event, however, the additional bandwidth requirements for replicating the index from the query servers and the associated search service application databases should be considered.

As SharePoint is rarely the only application in an IT environment, considerations on how to perform DR for all of the various business applications should also be taken into account. Since storage or SAN based replication provides a host-independent method for replicating data, other applications, such as external content sources (like file shares or non-SharePoint sites) that can be crawled within search, can also be replicated using storage technologies.

SAN replication not only offloads the process of replication from already-burdened SharePoint and SQL hosts, but also enables enterprise scalability in protecting SharePoint farms across extended distances. Ultimately, storage based replication should provide the flexibility necessary to match the desired DR configuration for virtually any enterprise level SharePoint implementation.

Copyright © 2011 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.