HP OpenView

Storage Mirroring application notes

Guidelines for using Storage Mirroring with SQL server reporting services

Part number: T2558-88076
First edition: June 2006

Storage Mirroring Guidelines for using Storage Mirroring with SQL server reporting services application notes

Document overview

This document is a Storage Mirroring application note. An application note provides guidelines on the use of Storage Mirroring in a specific environment.

This document contains:

- **Document overview**—Explains what an application note contains, how it should be used, what you need to know before trying to use the application note, and where you can go for more information.
- **Solution overview**—Explains how the application works with Storage Mirroring. Review this section to make sure that you understand the theory involved with using Storage Mirroring and your application.
- **Implementation guidelines**—Describes all of the considerations that you must weigh when implementing your Storage Mirroring solution. Includes both basics, such as system requirements, as well as configuration and environment-specific topics, such as interactions with specific clients or special considerations for WAN (Wide Area Network) environments. Pay special attention to those topics that are directly related to your environment.
- **Sample implementation**—Describes specific examples of how to use Storage Mirroring for this solution. This includes information about the specific hardware and software used in the sample implementation. Use these procedures as a guideline for creating your own implementation. Because no two environments or configurations are exactly the same, you will probably need to implement additional or different steps than what is documented here in order to make the solution work in your environment.

Audience

This document is written for network and application administrators who have a working understanding of the applications and environments where the Storage Mirroring solution is to be deployed. You many need to expand on the documented information in order to customize the solution to fit your environment.

Before you use this application note, you should have an understanding of:

- Storage Mirroring
- SQL Server Reporting Services
- The database server that you will be using (such as SQL 2000)
- The snapshot utility that you will be using (Volume Shadow Copy Service (VSS), or another third-party snapshot utility)

Expectations

Application notes are intended to provide a framework for configuring a Storage Mirroring solution in a specific environment and to draw attention to decisions you will need to make when configuring your solution.

Because there are an infinite number of possible configuration, network, and environment scenarios, application notes contain general configuration guidelines as well as an example configuration procedure that has been tested for a specific environment.

This document assumes that you are comfortable working with your operating system, Storage Mirroring, and SQL Server Reporting Services.

Related documentation

Before you begin to configure your solution, make sure that you have complete documentation for your operating system, application, and Storage Mirroring. This application note does not provide step-by-step instructions for using standard operating system, application, and Storage Mirroring functionality.

The following documents contain additional information that you may need while setting up this solution:

- *HP OpenView Storage Mirroring user's guide* or online documentation
- Reference guides or documentation for SQL Server Reporting Services
- Reference guides or documentation for your database server (for example, SQL 2000)

- Reference guides or documentation for Volume Shadow Copy Service (or for your snapshot utility, if you are not using VSS)
- Hewlett-Packard application notes for your database server and snapshot utility:
  - *HP OpenView Storage Mirroring High availability for Microsoft SQL Server 2000 application notes*
  - *HP OpenView Storage Mirroring High availability for Microsoft SQL Server 2005 application notes*
  - *HP OpenView Storage Mirroring Data protection for Volume Shadow Copy Service (VSS) application notes*

Getting help

Hewlett-Packard has application notes that describe how to configure Storage Mirroring with a variety of popular third-party applications. These application notes are available on the Storage Mirroring web site: http://h18006.www1.hp.com/products/storage/software/sm/index.html.

For help using Storage Mirroring, refer to the Storage Mirroring online manual or online help

Solution overview

This document describes how to use Storage Mirroring with SQL Reporting Services to create reports, either on-demand or from cached snapshots of SQL data, without negatively impacting performance on the production database.

In a typical SQL Reporting Services configuration, SQL Reporting Services can be used either to query current data or to create a cached copy of data for a report. Because running a live query against production data negatively impacts the performance of the production server, cached copies are frequently used to reduce this impact. However, the process of creating cached copies will also impact the production server, so the cached copies typically need to be created during non-peak hours. The choice is either to wait until non-peak times to create a report from cached data that may be out-of-date, or to query live, current data and impact the performance of the production server.

By using Storage Mirroring to replicate production SQL Server data to a target server, you can generate live reports from a target copy of the SQL Server database. This allows for reports to be created using more current data without impacting the performance of the production database.

Application note scope

The solution documented in this application note describes how to use SQL Reporting Services with SQL Server data sources (including clusters serving as the SQL data source). While the same principles should apply to other database servers supported by Storage Mirroring (such as Oracle), *specific* implementation procedures for using other database servers will likely be different from those documented here. In addition, this solution has not been tested with other database servers and they are not supported in this document.

This application note does NOT include the following information:

- How to install and configure SQL Reporting Services
- How to design and publish SQL Reporting Services reports
- How to provide high availability for your SQL server. If you want to provide high availability for your SQL data, you can configure an additional target server for high availability. For more information, see the *Storage Mirroring High availability for Microsoft SQL Server application notes* for your version of SQL Server.

If you need more information about working with SQL Reporting Services, see the reference guides or documentation for SQL Server Reporting Services.

The following configuration(s) are not supported by this application note:

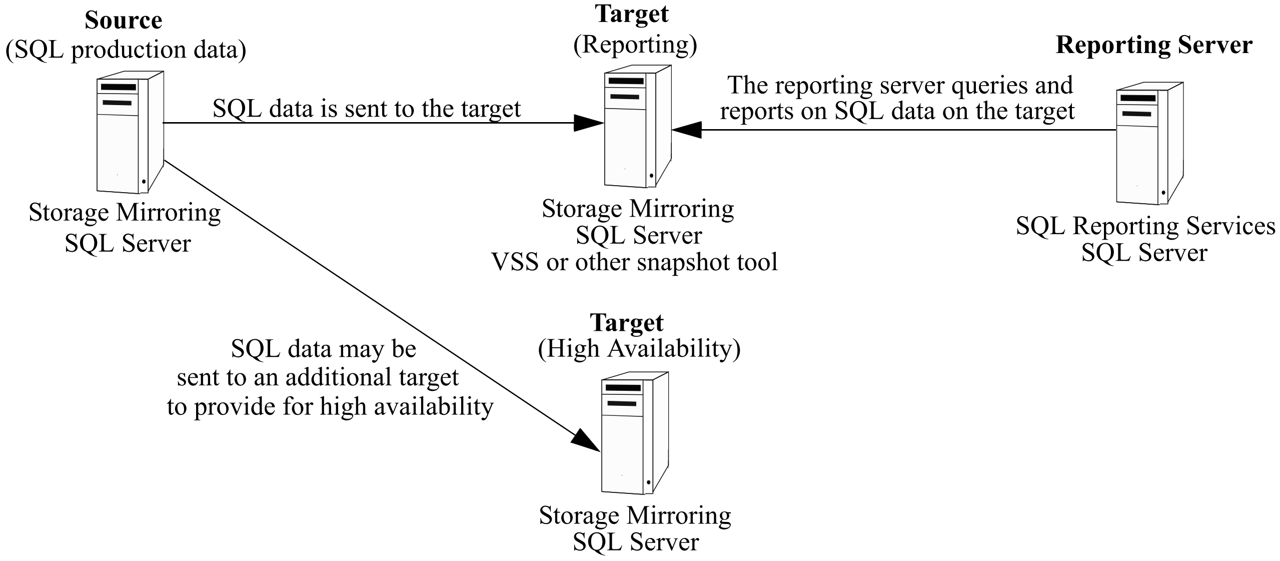
- SQL Reporting Services components deployed on separate physical machines. All Reporting Services components (Report Manager, Report Server, and Report Server database) should be installed on the same SQL server.

How this solution works

Server configuration

In this solution, SQL data is replicated from the production database (source server) to a database used for reporting (target server). A separate SQL reporting server receives requests from client servers for queries and reports, then generates reports from the data on the target server.

In addition to the servers used in this reporting solution, the following diagram also shows an additional target used for high availability. Details about configuring high availability for your SQL Server can be found in the *Storage Mirroring High availability for Microsoft SQL Server application notes* for your version of SQL Server.



NOTE: You may use a cluster as the SQL source. If you want to use a cluster for your source, first configure the source as described in the Cluster to Stand-alone section in the *Storage Mirroring High availability for Microsoft SQL Server application notes* for your version of SQL, then follow the guidelines in this application note to configure the reporting target server.

General process

Prior to running a report against the target server, the connection from the source to the target is paused, then a snapshot is taken of the target. The reporting server queries and reports against the target server. After reporting is complete, the snapshot is restored on the target. Finally, data in the target queue (collected while the connection was paused) is applied to the target. By taking a snapshot of the data prior to reporting, then restoring the data to its pre-reporting state, you can ensure that any changes made to the target server during reporting are cleared. While you may not have explicitly changed anything on the target server, the process of starting SQL Server alters the data slightly. After you restore the snapshot and the data from the queue has been applied, your target data will again be identical to the source.

By creating reports from the data on the target, there are no negative performance impacts to the production server when reports are run. In addition, you will not be required to perform a re-mirror.

Reporting options

When a report is published using SQL Reporting Services, it can be configured in one of two ways:

- To use a cached snapshot of the query results for the report (snapshot-based reporting)
- To retrieve real-time query results for the report from SQL (on-demand reporting)

If you want to run scheduled Reporting Services queries at one or more times during the day, you would configure Reporting Services to use the reporting target as its data source for all reports. These reports would be configured to create their cached copy of the query data at a specified time every day. This would allow for work to be off-loaded from the production SQL server and to provide faster reporting for end users. In this solution, replication is paused on the Storage Mirroring target and the SQL services are started directly before the scheduled reporting window. After all reports are run, the SQL data can be reverted and replication can be resumed.

If you want to run on-demand Reporting Services reports during normal business hours, you would configure Reporting Services to use the reporting target as its data source for all reports. This would allow for work to be off-loaded from the production SQL server and to provide faster reporting for end users. In this solution, replication is paused from the Storage Mirroring high availability target to the Storage Mirroring Reporting Services target, and the SQL services are started while reports are run using Reporting Services. At a scheduled interval, the SQL data can be reverted and replication can be resumed.

Application interactions

Microsoft SQL Server Reporting Services allows you to create, execute, and publish on-demand reports based on data gained from one or more data sources.

Reports may either query live, current data residing on the SQL server, they may pull data from a cached snapshot of the data gained from the data source at a specified time of day (typically during non-peak hours), or they may pull data from a cached version of the target data at any time (even during peak hours). After the snapshot has been created, reports can query the cached data residing on the report server without impacting the production database at all. With cached versions of reports, the system could generate the data set once, then reuse that data set for other users requesting the same data. SQL Reporting Services' scheduling capabilities could be configured to run all reports at a pre-determined time on a recurring basis.

To use SQL Server Reporting Services with Storage Mirroring, the SQL reporting server will have a series of sample reports configured with the Storage Mirroring target server as their data source.

Implementation guidelines

This section describes what you need to consider in order to successfully implement this solution.

Requirements

Each server must meet the following requirements:

- A licensed copy of Microsoft Windows Server 2003 with service pack 1 or later

NOTE:

- Microsoft Windows Server 2000 may be used in place of Windows Server 2003; however, you will also need a third-party snapshot utility.
- The source and target do not have to be running the same version of Windows Server.

- A licensed copy of Storage Mirroring version 4.4 with the most recent service pack
- A licensed copy of SQL 2000 Reporting Services or SQL 2005 Reporting Services
- A licensed copy of SQL Server 2000 or SQL Server 2005

NOTE:

- If you are using SQL 2000 Reporting Services, you will need to use SQL Server 2000
- If you are using SQL 2005 Reporting Services, you may use either SQL Server 2000 or SQL Server 2005

Installation and server configuration

Installation prerequisites

For the high availability scenario, this application note assumes that SQL Server and Storage Mirroring are already properly installed and running on the source and target servers. For more information about configuring the source and target servers, see the *Storage Mirroring High availability for Microsoft SQL Server application notes* for your version of SQL Server.

Reporting server setup

All reporting server components should be installed on the same physical server. The reporting server should be a separate SQL deployment from any other SQL instances. In addition, the SQL Reporting Services should reside on its own server (not on the source or target server).

For information about configuring the reporting server, see the documentation for SQL Server Reporting Services.

Application data location

Volume Shadow Copy Service cannot take a snapshot or revert a volume that resides on the Windows system drive. While your SQL server application may be installed on a system drive, the application data (such as databases and logs) should be on one or more other drives on the target server.

Storage Mirroring queue location

For this solution to work, the volume on the target where the application data is being sent cannot be the volume where the Storage Mirroring queue resides. To view where the Storage Mirroring queue resides, right-click on the target server name in the Management Console, and select **Properties**. Select the Queue tab and note the drive letter of the path in the Folder field.

Configuring the data source

To run existing reports from the reporting target, a new data source must be created. Use all of the same settings for this data source as you did for your existing data source for the production server, except use the hostname of the target server instead of the hostname of the production server (corresponding with the data source part of the connection string.

Determining snapshot restoration delay

In order to enter an appropriate wait time for the snapshot restoration to complete in the `resume-target.dtcl` script, you will need to perform a snapshot restoration to see how long it takes. Since every environment is different, it is impossible to recommend a delay time that will be appropriate for all situations.

You may want to add some additional time to the actual time needed for the restoration to account for changes to the data or processing speed.

Modifying the sample script files

You will need to edit the `pause-target.dtcl` and `resume-target.dtcl` files to include the correct server and login information for your environment. In addition, you will need to modify the wait time in `resume-target.dtcl`.

If you are using VSS, you will also need to edit the `start-reporting.bat` file to include the correct drive letter of the drive where the snapshot will occur.

NOTE: If your application data is stored on multiple drives, you will need either to modify the script or to create and restore the shadow copies manually.

NOTE: After you modify the sample scripts, save them with a new name to remove the `sample_` prefix. Copy the scripts to the directory where Storage Mirroring is installed.

The sample batch files provided are only examples. Because no two environments or configurations are exactly the same, you **MUST** modify the sample scripts in order to make the solution work in your environment.

The following sample script files are used in this process:

- **start-reporting.bat**—Sample batch file to run the `pause-target.dtcl` script, which will pause the target and create the snapshot.
- **pause-target.dtcl**—Sample script to pause execution of operations on the target prior to starting a snapshot.
- **end-reporting.bat**—Sample batch file to restore the snapshot and resume transmission to the target server.
- **restore-snapshot.vbs**—Get the shadow copy ID gained from the shadow copy creation and restore that shadow copy.

NOTE: The original data snapshot will need to be restored, even if nothing was explicitly changed on the target database. This is because the process of starting SQL Server alters the database slightly.

- **resume-target.dtcl**—Sample script to resume execution of operations on the target after the snapshot restore is complete. When you edit this script, enter the wait time found in ”Determining snapshot restoration delay” on page 7.

NOTE: Running the `end-reporting.bat` file will take at least two minutes because the script has a built-in delay. You must modify the delay to allow enough time for the volume to be restored to its original state.

sample_start-reporting.bat

```
rem ***SAMPLE*** batch file to run the pause-target.dtcl script and to create the snapshot

rem This sample batch file is provided as an example only. Because no two environments or
rem configurations are exactly the same, you MUST modify this script in order to make the
rem solution work in your environment.

cmd /c DTCL -f pause-target.dtcl

rem Edit the drive letter from X: below to the drive where your data resides.

vssadmin create shadow /for=X: >scid.txt
```

sample_pause-target.dtcl

```
# ***SAMPLE*** script to pause execution of operations on the target prior to starting #
# a snapshot. #

# This sample script is provided as an example only. Because no two environments or #
# configurations are exactly the same, you MUST modify this script in order to make the #
# solution work in your environment. #

# Substitute the name of your target machine, username, password, and domain in the variable #
# definitions below. #
$TheTarget = "TargetServer";
$User = "Username";
$Pass = "password";
$TheDomain = "domain.com";

login $TheTarget $User $Pass $TheDomain;
target $TheTarget;
pausetarget $TheTarget;
```

sample_end-reporting.bat

```
rem ***SAMPLE*** batch file to restore the snapshot and resume transmission to the target server

rem This sample batch file is provided as an example only. Because no two environments or
rem configurations are exactly the same, you MUST modify this script in order to make the
rem solution work in your environment.

call restore-snapshot.vbs

cmd /c DTCL -f resume-target.dtcl
```

sample_restore-snapshot.vbs

```
' ***SAMPLE*** script to get the shadow copy ID gained from the shadow copy creation and
' restore that shadow copy

' This sample script is provided as an example only. Because no two environments or
' configurations are exactly the same, you MUST modify this script in order to make the
' solution work in your environment.

' Get the shadow copy ID gained from the shadow copy creation, and restore that shadow copy
Const FOR_READING = 1
Set objFSO = CreateObject("Scripting.FileSystemObject")

' Open output file created from the creation of the shadow copy
Set objFile = objFSO.OpenTextFile(".\scid.txt", FOR_READING)
str_length=0
Do Until str_length>0
  str_line=objFile.ReadLine

  ' Correct line of output file
  if InStr(str_line, "Shadow Copy ID:") > 1 then
    first_char=InStr(str_line, "{")
    last_char=InStr(str_line, "}")
    str_length=Len(str_line)
    scID=Right(str_line, str_length-first_char+1)
  end if
Loop
Set WSShell=CreateObject("WScript.Shell")

' Execute vssadmin with the revert command using the shadow copy ID that was obtained
WSShell.run "vssadmin.exe Revert Shadow /Shadow=" & scID & " /ForceDismount /Quiet",,true
Set WSShell=Nothing
objFile.Close
```

sample_resume-target.dtcl

```
# ***SAMPLE*** script to resume execution of operations on the target after the snapshot #
# restore is complete#

# This sample script is provided as an example only. Because no two environments or #
# configurations are exactly the same, you MUST modify this script in order to make the #
# solution work in your environment. #


# Substitute the name of your target machine, username, password, and domain in the variable #
# definitions below. #
$TheTarget = "TargetServer";
$User = "Username";
$Pass = "password";
$TheDomain = "domain.com";

login $TheTarget $User $Pass $TheDomain;
target $TheTarget;

# Enter a wait time that will allow the snapshot restore to complete. To determine the time #
# needed for the restoration to complete, you should perform a test restoration, making note #
# of the time it takes for the restoration to complete. #
wait 120000;
resumetarget $TheTarget;
```

Scheduling report generation

If you want to run a report during a regular time period, you can use Windows Task Scheduler to help automate the process. You will use Task Scheduler to run the `start-report.bat` and `end-report.bat` scripts, which will take care of pausing and re-starting Storage Mirroring transmissions, taking a snapshot of the target volume at the beginning of the time period, and restoring the target to its original state.

1. First, you will need to create or verify that you have the appropriate post-failover (`postover.bat`) and pre-failback (`preback.bat`) scripts for the high-availability target for your version of SQL. Copy the scripts from your high-availability target server to the reporting target's Storage Mirroring directory, where the other scripts reside.

NOTE: You will need to modify the sample script files for your environment, if you have not done so already.

2. Next, you will need to modify the start-report.bat and end-report.bat files to call the post-failover and pre-failback scripts.

   At the end of the `start-report.bat` script file, insert the following line:

   ```
   call postover.bat
   ```

   At the beginning of the `end-report.bat` file, insert the following line:

   ```
   call preback.bat
   ```

3. To add these to Windows Task Scheduler, run the Scheduled Task Wizard on the target. Browse to the Storage Mirroring directory where these scripts reside. Select `start-report.bat` and enter schedule information so that the it will run at the beginning of the time window. Repeat to select `end-report.bat` and configure it to run at the end the time window. Be sure to enter the desired schedule information, including start time and the frequency with which the task is performed (such as daily or weekly).

Sample implementation

This section describes how this solution has been implemented to run a report from a target that is serving as a Reporting Services data source.

NOTE: Because no two environments or configurations are exactly the same, you will need to modify the sample implementation in order to make the solution work in your environment.

Test environment setup

The following equipment was used in testing this solution:

- All servers:
  - Microsoft Windows Server 2003 with Service Pack 1
  - Storage Mirroring version 4.4, Service Pack 1
- Source and target servers:
  - SQL Server 2000
  - VSS installed on the target machine
- Reporting server:
  - SQL 2000 Reporting Services
  - SQL Server 2000 (used by reporting services)

NOTE: This solution has also been tested successfully using SQL 2005 Reporting Services in conjunction with either SQL Server 2000 or SQL Server 2005.

Running a report

1. Copy the following script files to the Storage Mirroring directory on the target server:
   - `start-reporting.bat`
   - `pause-target.dtcl`
   - `end-reporting.bat`
   - `restore-snapshot.vbs`
   - `resume-target.dtcl`
2. Verify that the script files have been updated to include the appropriate drive, server, and login information.
3. Run the `start-reporting.bat` file on the reporting server target. The file will pause Storage Mirroring writing data to the target reporting server volumes and initiate a snapshot of the volumes being protected.
4. Run the SQL post-failover script to start the SQL Server services after the snapshot has been successfully created. The services that must be manually started include:
   - Distributed Transaction Coordinator
   - Message Queueing (SQL Server 2000 only)
   - MSSQLServer
   - SQLServerAgent

NOTE: You will need to modify the sample post-failover script for your environment, if you have not done so already.

5. Complete one of the following:
   - Run live reports from SQL Reporting Services
   - Create a cached copy of data from copy of SQL server on target, then run reports on the cached copy

NOTE: If you need more information about working with SQL Reporting Services, including how to design, run, and publish SQL Reporting Services reports, see the reference guides or documentation for SQL Server Reporting Services.

6. After reporting is complete, run the pre-failback script to stop the SQL Server services. This should be the same SQL services that you started in step 4.

NOTE: You will need to modify the sample pre-failback script for your environment, if you have not done so already.

7. Run the `end-reporting.bat` file on the target. This file uses VSS to revert the volume to its pre-reporting state and then to resume writing data on the target.

NOTE:

- The original data snapshot will need to be restored, even if nothing was explicitly changed on the target database. This is because the process of starting SQL Server alters the database slightly.
- Running the `end-reporting.bat` file will take at least two minutes because the script has a built-in delay. You must modify the delay to allow enough time for the volume to be restored to its original state.