Technical Report

Development and Test Solution for SQL Server Databases

Sourav Chakraborty, NetApp
September 2009 | TR-3793

TABLE OF CONTENTS

1 INTRODUCTION

Database-dependent applications such as Online Transaction Processing (OLTP), Decision Support Systems (DSS), Enterprise Resource Planning (ERP), and Customer Relationship Management (CRM) provide the foundation for today's information-driven enterprise. The development and testing of these applications demands access to multiple copies of sample databases. The data in sample databases is used to validate and test the application.

Creating copies of databases for application development and testing can put additional work on DBAs who already have to meet stricter service-level agreements for other critical and semi critical databases. Today's databases are more distributed, generate more transactions, and include new data types. This further complicates the provisioning of copies of databases for supporting requirements of database application development, even as these development environments grow increasingly costly and complex.

NetApp® solutions dramatically reduce the effort required to deploy databases for application development, testing, QA, maintenance, and migration. As a result, companies can accelerate time to deploy new application functions while enhancing application quality. Database administrators and application developers can now use simple but powerful tools to quickly and easily create local or remote database clones in seconds, using little additional storage, without the need for extensive storage skills or storage administrator support. This can drastically reduce both storage and administrative costs for supporting application development efforts within the enterprise.

In this paper we discuss the solutions that address the challenges faced by DBAs in supporting application development and testing. These solutions are based on NetApp's technologies and bring tremendous savings in time, resource utilization, and effort.

1.1 INTENDED AUDIENCE

This technical report is intended for information technology professionals, storage professionals, and SQL Server® DBAs responsible for corporate database management infrastructure. For methods and procedures mentioned in this technical report, NetApp assumes that the reader has working knowledge of the following:

- SQL Server 2005 architecture
- SQL Server storage architecture and database administration
- Service-level expertise for Microsoft® SQL Server recovery options

Working knowledge of NetApp solutions, including:

- Data ONTAP®
- SnapDrive® for Windows® (SDW)
- SnapManager® for SQL Server (SMSQL) backup and restore procedures
- SnapMirror®

2 OVERVIEW OF THE DATABASE CLONING SOLUTION

Table 1 summarizes the various combinations of SQL Server host and NetApp storage system pairings that are addressed in this report. For example, Solution 2 addresses database cloning for those cases in which the source (original) and target (clone) databases must reside on different NetApp storage systems.

	Source and Destination SQL Server Databases Reside On	
	Same Storage System	**Different Storage Systems**
Solution 1	X	
Solution 2		X
Solution 3		X (source is on non NetApp storage)

Table 1) Possible setting of source and target SQL Server hosts.

For each of the solutions we present the following stages of creating and managing database clones:

STAGE 1—Setup

This stage deals with the steps that are needed to ensure that Stage 2 can be executed properly. The steps in this stage are taken mostly one time for each source database and can be repeatedly used to create clones by following the steps in Stage 2.

NOTE: We will not cover the setup of SQL Server 2005 in this paper. NetApp assumes that SQL Server instances are appropriately installed in all the hosts that are involved. For more information refer to the following link:

How to: Install SQL Server 2005 (Setup)

STAGE 2—Clone Creation and Usage

This stage deals with ways and means to create clones of a database.

STAGE 3—Clone Destruction

In this stage we need to provide a complete cleanup of a database clone. The cleanup should free all the resources used by the database clone at the following locations:

1. SQL Server Host
   i. Unmount the database clone.
   ii. Remove any LUNs or mount points that were presented in the process of creating the clone.
2. Target NetApp Storage System
   i. Release the storage resources used by the database clone.

3 DATABASE CLONING SOLUTION #1

The first solution is restricted to cases in which the source and target databases are located on the same NetApp storage device. However, the source and target databases can be on the same or different SQL Server instances. The following figure explains the same:

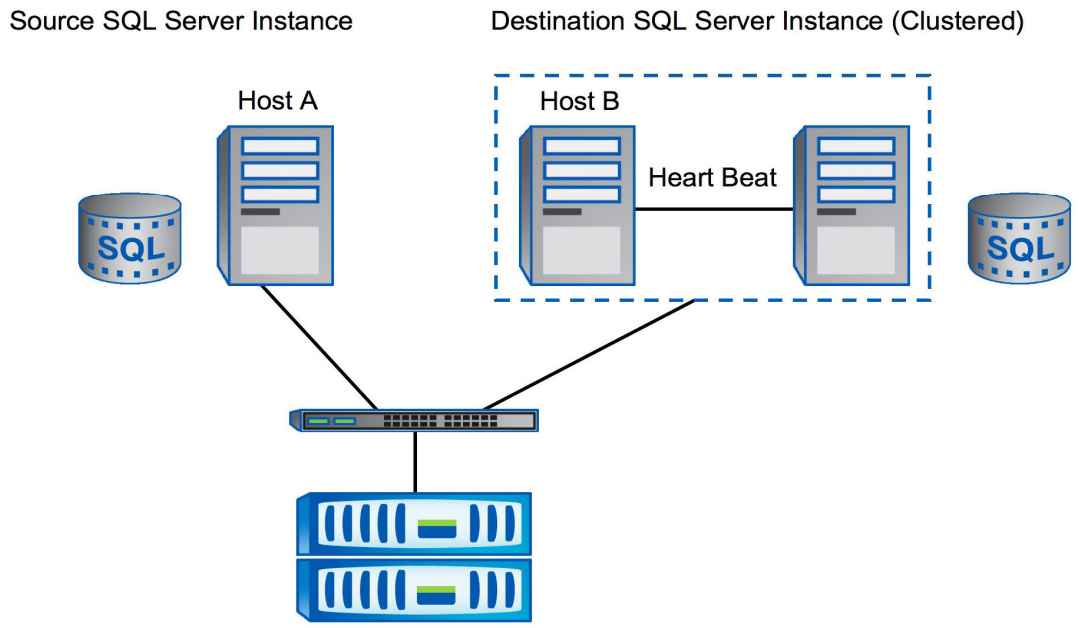

Figure 1) Environnent description for Solution #1.

3.1 SETUP

Ensure that the following setup is present prior to using this solution:

1. Both SQL Server hosts A and B must have SMSQL 5.0 installed.

3.2 CLONE CREATION—TARGET SQL SERVER INSTANCE IS STANDALONE

For database clone creation, we use the Database Cloning feature of SMSQL 5.0. One of the most important features introduced in SnapManager 5.0 for SQL Server is the ability to create clones of a database. The key advantages of SMSQL 5.0's database cloning feature are:

1. Ability to create space-efficient clones using NetApp FlexClone® technology
2. NetApp Snapshot® technology, along with the SQL Server log replay feature, allows creation of point-in-time clones of databases

SMSQL 5.0 supports the following operations with respect to database clones:

- **Clone creation.** The supported sources are:
  - Active databases
  - Database backups

- **Clone deletion.** This includes cleaning up all clone-related resources, including host-side LUN mappings.

Note: It is desirable to have FlexClone licensed on the storage system where the backup sets or active database is based. However, this is not a requirement. If FlexClone is not licensed in this manner, the LUN is completely backed by the Snapshot copy.

For information about how database cloning can be used, and for the step-by-step process of creating a database clone, see Chapter 11 of the Installation and Administration Guide. This section discusses some of the internals of how database clones are created using SMSQL.

CLONE CREATION—TARGET SQL SERVER INSTANCE IS CLUSTERED

Following are the steps to create a clone with the target as a clustered SQL Server instance:

1. If the database clone needs to be created from a historic backup set, then do the following. Otherwise, go to Step 2.
   a. Within the SMSQL GUI, go to the Restore tab.
   b. View the backup sets created for the source database.
   c. Note the backup-set name that has the appropriate timestamp. Backups sets are typically named as sqlsnap_SQL SERVER instance-name_TimeStamp. Therefore, the timestamp of a backup can be found by looking at the suffix of the backup set's name.
   d. The corresponding Snapshot copy must be used in the subsequent steps.
   e. Go to Step 3.
2. If the database clone needs to be created from the active database, then do the following:
   a. Within the SMSQL GUI, go to the Backup tab.
   b. Create a backup of the database.
   c. The most recent Snapshot copy must be used in the subsequent steps.
3. Follow the steps outlined in Appendix C.
4. The database clone is now ready for use by applications that wish to read/write to the cloned database.

3.3 CLONE DESTRUCTION

In order to destroy the clone that is created at the target SQL Server instance, use the Delete clone feature from the SMSQL 5.0 instance present on Host B. The details of this operation are presented in Chapter 11 of IAG for SMSQL 5.0.

It is important that a process be developed and implemented to provide continual cleanup of the database clones. The operation mentioned in the paragraph above must become a part of this process.

4 DATABASE CLONING SOLUTION #2

The second solution is for cases in which the source and target databases will be located on different NetApp storage systems. Typically, in such a scenario, the source and target databases are also located on different SQL Server instances that are connected to independent NetApp storage systems. The following figure represents the environment:

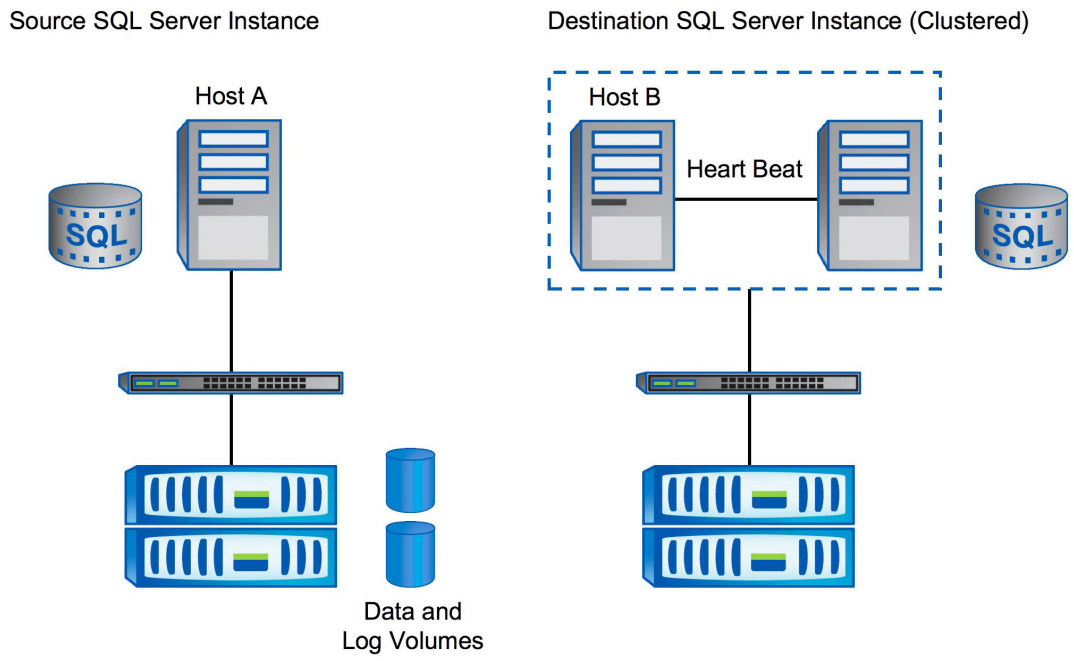

Figure 2) Environnent description for Solution #2.

4.1 SETUP

In order to create database clones of databases present on Host A, the following setup should be implemented:

1. Hosts A and B should have SMSQL 5.0 installed on them.
2. Ensure that an Asynchronous SnapMirror relationship exists between all the volumes of the source database and mirror volumes at the target NetApp storage system. Please follow the steps in Appendix B.

Figure 3 shows the above setup.

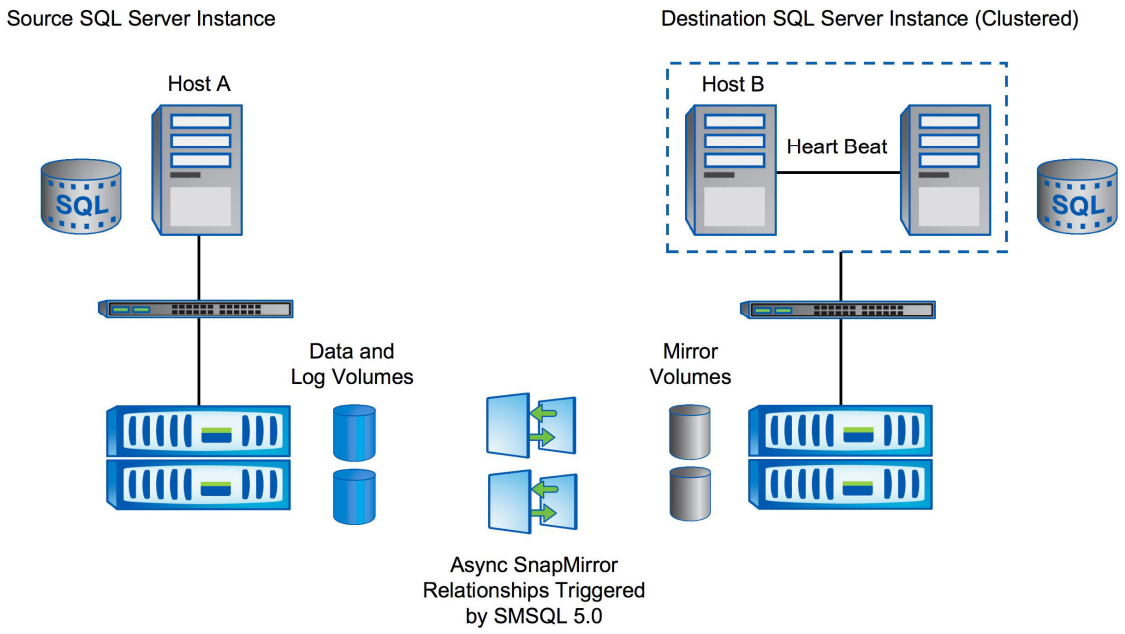

Figure 3) Setup for Solution #2.

4.2 CLONE CREATION—TARGET SQL SERVER INSTANCE IS STANDALONE

There are two sub-cases in creating clones:

Case 1: Creating a clone of a live database

Case 2: Creating a clone from historical backup sets

The steps for each of these cases are discussed in the subsections below.

STEPS FOR CASE 1

1. From Host A, initiate an SMSQL backup for the concerned database. Choose the “Update SnapMirror” option.

2. Using SDW on Host B, connect to the LUNs within the latest Snapshot copies in the SnapMirror destination volumes that were replicated to the SnapMirror destinations in Step 1.

NOTE: One must connect to the Snapshot copies in both the data and log volumes on the destination storage system.

3. Attach the database files to the SQL Server instance on Host B by providing the appropriate data file and TLOG file paths as per the mount points used in Step 2 above.

4. The database clone is now ready for use by applications that wish to read/write to the cloned database.

STEPS FOR CASE 2

1. Use SDW on Host B and check if the appropriate Snapshot copy is present at the destination.

2. If it is not present, then initiate an SMSQL backup for the concerned database. Choose the “Update SnapMirror” option.

3. Using SDW on Host B, connect to the LUNs within the latest Snapshot copies in the SnapMirror destination volumes that were replicated to the SnapMirror destinations in Step 1.

NOTE: One must connect to the Snapshot copies in both the data and log volumes on the destination storage system.

4. Attach the database files to the SQL Server instance on Host B by providing the appropriate data file and TLOG file paths as per the mount points used in Step 2 above.

5. The database clone is now ready for use by applications that wish to read/write to the cloned database.

CLONE CREATION—TARGET SQL SERVER INSTANCE IS CLUSTERED

There is a minor change in the steps mentioned in the above section for creating clones on a clustered SQL Server instance.

For Case 1: In Step 2 use the process outlined in Appendix C.

For Case 2: In Step 3 use the process outlined in Appendix C.

The rest of the steps mentioned above for cases 1 and 2 remain the same.

4.3 CLONE DESTRUCTION

In order to destroy the clone that is created at the target SQL Server instance, use the Delete clone feature from the SMSQL 5.0 instance present on Host B. The details of this operation are presented in Chapter 11 of IAG for SMSQL 5.0.

It is important that a process be developed and implemented to provide continual cleanup of the database clones. The operation mentioned in the paragraph above must become a part of this process.

5 DATABASE CLONING SOLUTION #3

The third solution is for cases in which the source and target databases are located on heterogeneous storage systems. Specifically, the source database is on a non NetApp system and the target database is on a NetApp system. The following figure represents the environment:

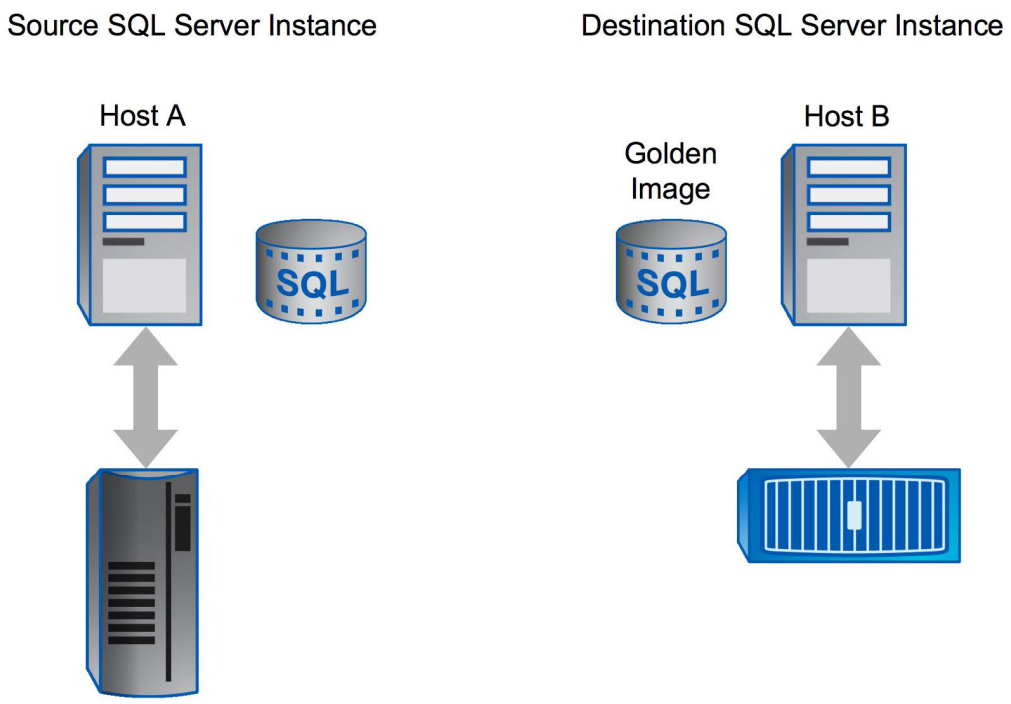

Figure 4) Environnent description for Solution #3.

5.1 SETUP

In order to create database clones of databases present on Host A, the following setup should be implemented:

1. A staging SQL Server instance needs to be created that is connected to a NetApp storage system. Such a system is depicted as Host B in the figure below.
2. A suitable replication mechanism needs to be put in place to replicate the source database from Host A to Host B. The steps to set up the replication are as follows:
    a. Provision a LUN greater than or equal to the size of the source database. Please refer to Appendix A for more information.
    b. Set up transactional replication between the source database and a database on Host B. The database files for the database on Host B must be placed on the LUN created above. This is the recommended replication mechanism to be used.

Please note that the database on Host B serves as the “Golden Image” from which all the database clones will be created on SQL Server instances connected to NetApp storage systems.

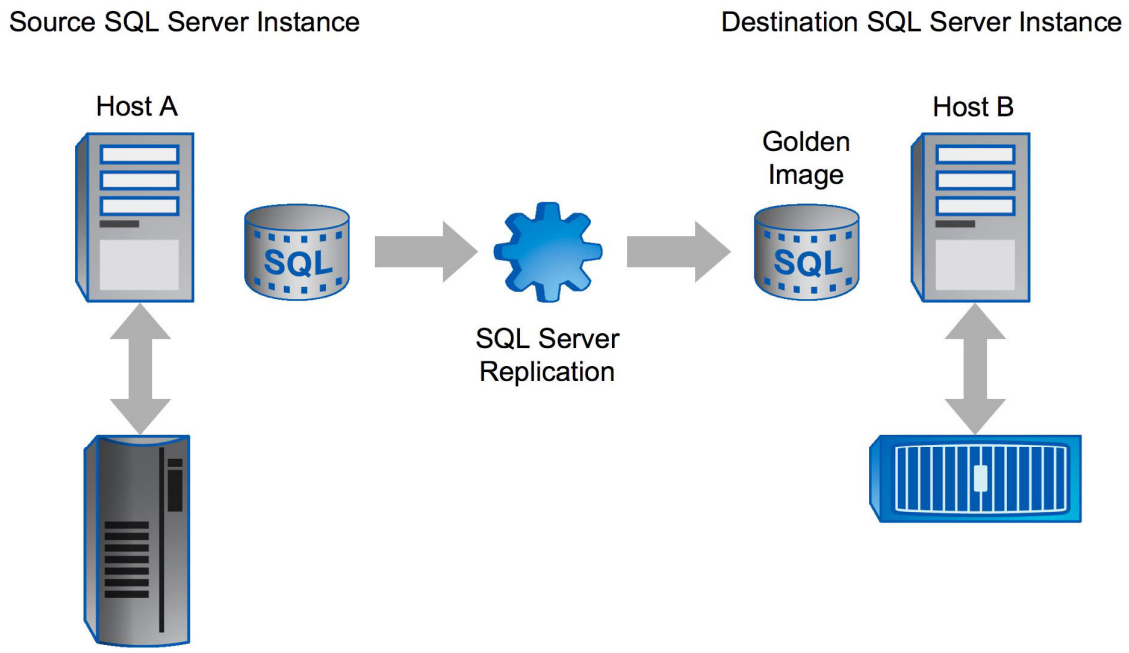

Figure 5) Setup for Solution #3.

5.2 CLONE CREATION

Follow Solution #1 or Solution #2 based on the location of the target SQL Server instance to create clones of the “Golden Image.”

5.3 CLONE DESTRUCTION

In order to destroy the clone that is created on the target SQL Server instance, use the Delete Clone feature from the SMSQL 5.0 on the appropriate host. The details of this operation are presented in Chapter 11 of IAG for SMSQL 5.0.

It is important that a process be developed and implemented to provide continual cleanup of the database clones. The operation mentioned in the paragraph above must become a part of this process.

6 DATABASE CLONING FRAMEWORK

As described in sections 3, 4, and 5, the Clone Management Framework may be presented as follows:

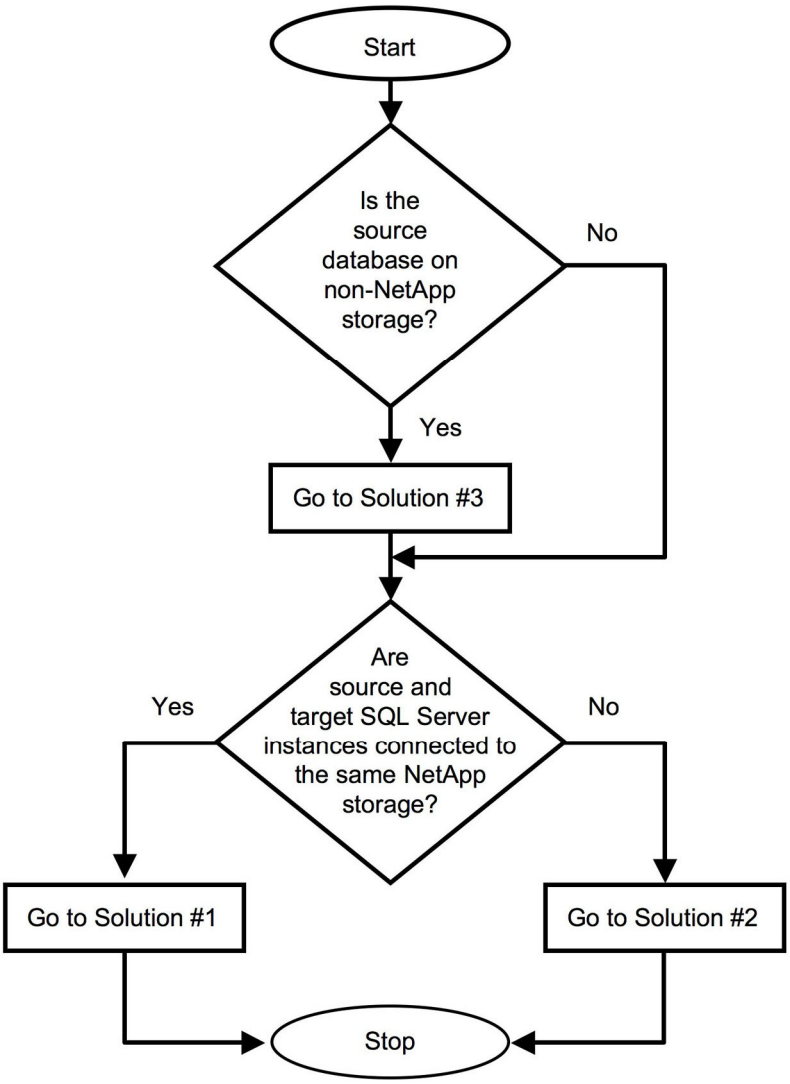

Figure 6) Clone Management Framework.

7 CONCLUSION

Database development and testing are often thought of as a series of tradeoffs between increasing development speed, improving developer service levels, and lowering costs. That's not the case if you leverage the powerful capabilities of FlexClone, SnapMirror, and SnapManager from NetApp. Many organizations today use these solutions to streamline SQL Server development, test, QA, and reporting

while saving time, dramatically improving efficiency and service levels, and reducing storage and administrative costs.

The implications of this for many Microsoft SQL Server organizations are far reaching. The unique NetApp FlexClone implementation, coupled with SnapManager for SQL Server and SnapMirror replication technology, can significantly reduce the time required to create database clones for a variety of development and operational scenarios.

Efficient database cloning with NetApp:

- Improves DBA productivity
  - Space-efficient, on-demand database cloning
  - Near-instant database recovery
- Speeds up the dev/test cycle
- Enables rapid application development models
- Supports geographically distributed teams

NetApp gives SQL Server database administrators and application developers automated and powerful tools to quickly and easily create nearly instantaneous database clones on demand, using negligible additional storage and without the need for extensive storage expertise or support.

8 APPENDIX A

In order to create a LUN for Solution #3, perform the following steps:

Step 1: Create a volume on the NetApp storage system of the following size:

VOL_SIZE = (1.1 * LUN_SIZE) + (CHANGE RATE * # of Snapshot copies to be created)

Step 2: Using SnapDrive on the appropriate host, create a LUN of the desired size.

9 APPENDIX B

This section describes the steps involved in setting up a SnapMirror relationship.

To illustrate the steps, the following are the source and destination volumes:
Source Storage: FILER1
Source Volume: Vol1

Destination Storage: FILER2
Destination Volume: Vol2

```
(A)    On both source and destination storage ensure that SnapMirror is
licensed. In case it is not, run the following command:
  license add <license key>

(B)    On FILER1 run the following command:
        options snapmirror.access host=FILER2


(C)     On both FILER1 and FILER2 run the following command:
options snapmirror.enable on

(D)     On FILER2 edit the snapmirror.conf file as follows:
```

```
(i)     Type wrfile /etc/snapmirror.conf and hit Enter.
(ii)    Type the following command:
        FILER1:Vol1 FILER2:Vol2 - - - - -
(iii)   Hit Enter followed by ctrl-c to exit.
(iv)    Verify the contents of the file by running
         rdfile /etc/snapmirror.conf

(E)     On FILER2 run the following command:
vol restrict Vol2

(F)     On FILER2 run the following command:
snapmirror initialize -S FILER1:Vol1 FILER2:Vol2

(G)     On FILER2 run the following command:
snapmirror status
```

10 APPENDIX C

In this section we present the steps to mount a LUN inside a Snapshot copy as a shared disk:

1. Create FlexClone volumes of all the database volumes based on Snapshot copies that were created at the same point in time. Normally, such Snapshot copies are created by SMSQL 5.0 as a part of a FULL backup of an SQL Server database.

   Note:
   The source database could be on a clustered SQL Server instance or a standalone SQL Server instance.

2. Create an iSCSI initiator group on the appropriate appliances on which the database volumes exist. This group should contain the iSCSI initiator names pertaining to the cluster nodes on which the clustered SQL Server instance exists.

3. Map each of the LUNs inside the FlexClone volumes to the iSCSI initiator group created above.

4. Go to each of the cluster nodes and open the Disk Management MMC.

5. Run the "Rescan Disks" to determine that the Windows operating system is aware of the LUNs that were presented to it from the previous step. If the cluster node is running on Windows 2008 Server, then the disks will appear "offline." This is a safety feature in Windows Server 2008 clusters.

6. Right-click on the new disks and make them online.

7. Go to SnapDrive for Windows MMC on each node and refresh the disk list. The new LUNs will show as disks in the SnapDrive disk list without drive letters.

8. Assign the same drive letters (through the SDW interface) to the new disks on all the nodes.

9. From the node that currently owns the virtual SQL Server resource group, add the disks created above as cluster resources (physical disks) to the appropriate virtual SQL Server cluster group using the Windows Cluster Administration MMC. In order to perform this step, the login administrator needs to have domain admin rights or should be the cluster administrator for the given Windows cluster.

10. Take the virtual SQL Server cluster group (from the previous step) offline.

11. Add the new physical disk resources from above in the dependency list of the virtual SQL Server resource.

12. Bring the cluster group online.

13. Go to the SQL Server management studio and attach the database by using the files in the disks added above.

11 REFERENCE

NETAPP MICROSOFT SQL SERVER SOLUTIONS
http://www.netapp.com/us/solutions/applications/microsoft-sql/

SNAPMANAGER FOR SQL SERVER
http://www.netapp.com/us/products/management-software/snapmanager-sql.html

FLEXCLONE
http://www.netapp.com/products/software/flexclone.html

SQL SERVER TRANSACTIONAL REPLICATION
http://msdn.microsoft.com/en-us/library/ms151247(SQL.90).aspx

www.netapp.com