

Data ONTAP™ 7G: FlexVol™ and FlexClone™ for Microsoft® SQL Server 2000

Network Appliance | June 2005 | TR-3410

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Abstract

With the release of Data ONTAP 7G, Network Appliance brings new and innovative data management solutions to the marketplace with introduction of new powerful virtualized volume structures called flexible volumes or FlexVol volumes. A FlexClone volume is a complimentary technology that provides space-efficient, near instantaneous writable copies of an entire FlexVol volume. FlexClone volumes enable SQL Server DBAs and storage administrators to quickly generate dynamic replicas of their databases for a variety of practical uses. This report discusses the uses, steps, and procedures required when using flexible volumes and flexible clones with SQL Server 2000 and SnapManager® 1.5. The overall benefits, convenience, and ROI advantages associated with leveraging Data ONTAP 7G FlexVol and FlexClone in the SQL Server database environment will also be demonstrated.

Table of Contents

Abstract.....	1
Table of Contents	2
Executive Overview	3
Introduction	3
Assumptions	3
List of Figures Used and Supplemental Information	4
A Brief Overview of FlexVol and FlexClone	4
What Are Flexible Volumes?.....	4
Figure 1) Flexible volumes, clones, and aggregates.	5
What is a FlexClone Volume?.....	5
FlexVol Volumes in the SQL Server Environment.....	6
Figure 2) Flexible volumes for SQL Server.	6
FlexClone in the SQL Server Environment	7
Figure 3) Tape, mirroring, and FlexClone over the time continuum.	7
Microsoft SQL Server Database Lifecycle.....	7
Figure 4) The SQL Server database lifecycle.	8
Creating a Flexible Volume, FlexClone Volume with Data ONTAP 7G, SnapDrive, and SnapManager 1.5	8
Environment	9
1. Provisioning	9
1.1 Creating the Aggregate	9
1.2 Creating Three FlexVol Volumes	10
1.3 Create Three CIFS shares.....	10
1.4 Create Database NTFS LUNs	10
1.5 Migrating Your SQL Server User Databases to Flexible Volumes with SMSQL 1.5.....	11
2. Tune and Maintain	11
2.1 Increasing Spindle Count On-the-fly: Adding Another RAID Group to the Aggregate	11
2.2 Grow LUN with SnapDrive On-the-Fly.....	12
3. Change Management with FlexClone, SnapDrive, and Snapshot	13
3.1 Generating SMSQL Snapshot Copies	13
3.2 FlexClone SQL Server Database from SMSQL Snapshots	13
3.3 Set Up the CIFS Share at the Flexible Volume Level.....	14
3.4 Connect a LUN to Each FlexClone Volume with SnapDrive Back to the Server SQL2K.....	14
3.5 Attaching SQL Server Database Clone to SQL Server Instance	16
3.6 Splitting the Cloned Volume While Running Live Transactions	17
Conclusions	18

Executive Overview

As consumption for data increases at an astonishing rate, so do the demands placed on today's IT and management professionals. Database administrators have a unique set of challenges, particularly around sizing for growth and throughput of heavily loaded online transaction processing (OLTP) and decision support (DSS) systems. Forecasting space requirements for enterprise database applications is often more of an art than a science. With traditional storage environments, careful planning to correctly size throughput and space utilization is needed prior to the deployment of an application. Frequently, despite meticulous and careful planning, projected sizing proves to be either grossly over or under specified for the real world space, growth, and access patterns of the databases that backend today's critical ERP, CRM, and eCommerce applications. To compound the problem, making subsequent changes to the database storage layout is cumbersome and comes at the price of reducing system and application availability.

Challenges

- Low storage asset utilization (wasted space and lost throughput)
- Change management: upgrades, patches and change of database lifecycle

NetApp Value Proposition

- Reduced TCO per database instance
- Increased Business Agility
- More precise and directed storage utilization
- Powerful cloning for safe change validation

DBAs also encounter weighty challenges when it comes to patch, upgrade, or modifications to their database environments. Despite careful planning and significant investments in testing labs and Q/A environments, sizable risks in upgrading and modifying production environments remain. This challenge becomes greatly magnified for environments in which database transactions are tied directly to revenue generation. Downtime to make adjustments to storage layout and run unpredictable upgrades can be very disruptive to business operations.

The purpose of this TR is to demonstrate how NetApp Data ONTAP 7G empowers database and storage administrators to quickly provision, modify, and boost performance of their databases, while maximizing total storage asset utilization. We will also illustrate how DBAs can leverage FlexClone technology to optimize the development, test, upgrade, and release process that surround the entire database lifecycle. By utilizing FlexVol and FlexClone, organizations can achieve greater business agility, reduce downtime, and realize higher returns from the total storage investment. This TR is tailored to the unique requirements of Microsoft SQL Server 2000 and the Windows® family of products offered by Network Appliance including SnapDrive™ 3.X and SnapManager for SQL 1.5.

Introduction

In this report, we will provide an examination and description of the features and benefits of FlexVol and FlexClone and proceed to show the SQL Server DBA and developer how these technologies can greatly enhance the productivity and flexibility of nearly all phases of the database lifecycle. A detailed hands-on process of the creation of a SQL Server purposed FlexVol, SnapManager SQL Snapshot™, FlexClone, and FlexClone split will be also demonstrated.

Assumptions

1. All NetApp storage systems used are running at least Data ONTAP 7G or greater.
2. Microsoft SQL Server 2000 SP3 is installed.
3. The reader should have a solid understanding of SQL Server, the Windows Server family and preferably some familiarity with WAFL®, Snapshot, SnapManager, SnapDrive, and the NetApp Data ONTAP operating system.
4. The DBA must have the ability and rights to run `RSH` commands to the filer on which their SQL databases reside. If this is not the case, the SQL administrator should collaborate with his or her storage administrator as needed.
5. SnapManager for SQL 1.5 and SnapDrive 3.X for Windows 2000/2003 is installed and configured on the database server.
6. A blocks-based protocol is used to connect the SQL Server to the filer—either FCP or iSCSI. In this TR we will use iSCSI. For more information about iSCSI, visit the Network Appliance iSCSI Technology Center at <http://www.netapp.com/solutions/iscsi/>.

List of Figures Used and Supplemental Information

SnapManager for SQL Server	http://www.netapp.com/ftp/sm-sql.pdf
SnapDrive for Windows	http://www.netapp.com/ftp/snapdrive.pdf
Snapshot™	A feature that creates an online, read-only copy of the entire file system—called a “Snapshot copy”—that protects against accidental deletions or modifications of files without duplicating file contents. Also used by FlexClone volumes for a point-in-time view of the parent.
RAID-DP™	A new RAID level introduced in Data ONTAP 7G that stripes parity both horizontally and diagonally, allowing for double disk failure in a RAID group. Two parity disks are used to manage double parity data. For more information, see http://www.netapp.com/tech_library/3298.html .
LUN	Logical Unit Number, often referred to as a logical unit of storage.
WAFL®	Write Anywhere File Layout. A patented file system designed exclusively for NetApp appliances to optimize write performance. Developed by Network Appliance.
IOPS	Inputs/outputs per second

A Brief Overview of FlexVol and FlexClone

What Are Flexible Volumes?

Database and storage administrators today are confronted with numerous challenges when attempting to “right-size” their database storage for effective space and throughput utilization. Many organizations add numerous RAID devices to their environments, creating many “islands of storage” with utilization rates estimated by industry analysts at roughly 30% to 40%. In this environment, storage is fragmented and management costs increase significantly with the addition of every new database server. Considerable space and throughput horsepower is also lost, greatly reducing the return of the overall storage investment. Traditional SANs have helped to consolidate RAID devices into a central location leveraged by many servers. Still, even in many SAN environments, adding more disks to boost IOPS and more effectively distribute database workload remains inseparable from adding more dedicated space to a given database environment, even when not required.

NetApp ONTAP 7G has revolutionized the landscape for customers looking to leverage the maximum spindle count on their SAN environments without forgoing control of space management. A flexible volume (also called a FlexVol volume) is a logical storage entity residing within what is termed an aggregate. An aggregate is logical entity containing a pool of one or more RAID or RAID-DP groups. Flexible volumes are created within these aggregates and can be sized appropriately anytime for the unique requirements of each SQL Server database instance. As the name implies, flexible volumes are highly flexible in that they can be grown and shrunk on-the-fly with no impact to system and database availability.

The airline industry has maximized returns of flight schedules by overbooking to compensate for the natural passenger rescheduling and defection. Much in the same way, flexible volumes enable administrators to orchestrate a similar concept called “thin provisioning” of free space across all systems to further maximize storage investments.

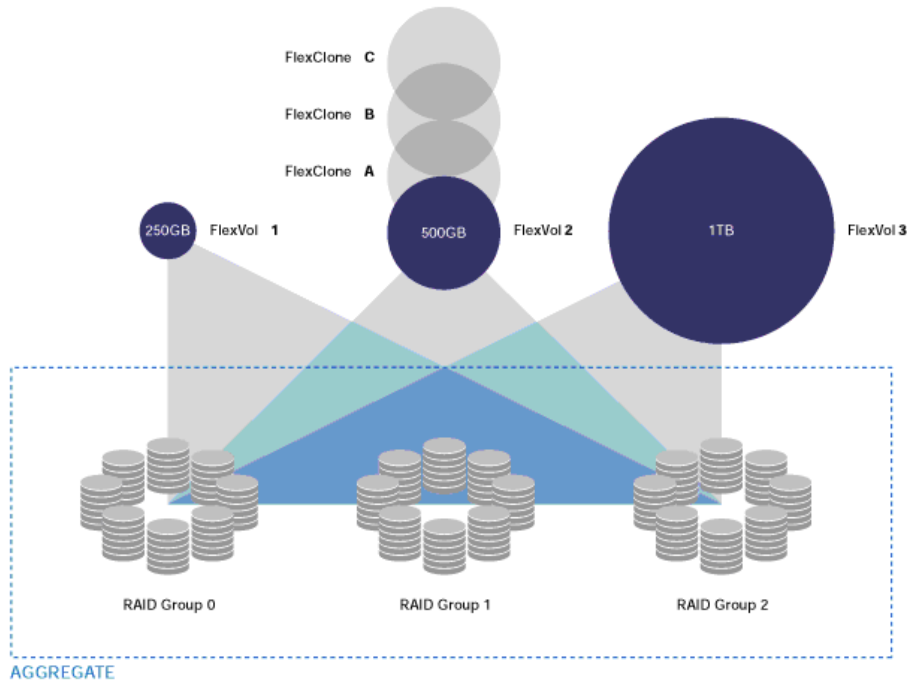


Figure 1) Flexible volumes, clones, and aggregates.

Figure 1 illustrates an example aggregate consisting of a composite of three separate RAID groups. Within this example aggregate, three flexible volumes reside (1, 2, and 3 respectively). By pooling three RAID groups together, applications accessing the FlexVol volumes residing within an aggregate get the benefit of a combined 24-disk throughput rather than a dedicated eight-disk volume. Because a FlexVol volume is a logical entity located inside the aggregate; they can be sized and resized as organizations SQL Server databases grow.

Figure 1 also depicts another layer of storage abstraction called a FlexClone volume. FlexClone is a tightly integrated and powerful cloning technology enabling storage and database administrators to effectively create instantaneous writable copies of an entire flexible volume for a variety of practical uses in the software and database lifecycle. We will now take a closer look at how FlexClone helps optimize the database lifecycle.

What is a FlexClone Volume?

A FlexClone volume is generated from a Snapshot copy of a FlexVol volume and essentially provides a transparent writable copy of its ancestor or parent. All changes to the FlexClone volume are managed on the WAFL file system and associated with the FlexClone instance. The underlying data of the cloned volume, less its changes, requires no immediate additional space because it physically points to the underlying blocks in the ancestor. As the data in the cloned volume begins to diverge from its ancestor, additional space will be occupied to hold the related changes. A cloned volume can also be “split” and become an entirely new physical copy of its ancestor, thereby creating an entirely new non-copy-bound flexible volume. One of the most powerful benefits of the cloned volume split is that it can occur while the clone is mounted and being written to by a DBMS, like SQL Server. In the case of SQL Server, an administrator may have several attached LUNs that reside in one or more flexible volumes and be actively servicing thousands of transactions while a split procedure is taking place, *without losing data*. Unlike traditional environments in which databases have to complete a physical copy before they can be mounted again, database administrators can quickly clone a given database environment, mount it for service, and then trigger a split to relocate the databases underlying blocks to another volume, all while servicing requests from applications and users. Split processes can also be scheduled and executed during off hours to minimize the resource utilization impacts of copying data to an entirely new volume. A cloned volume split will later be demonstrated in a SQL Server 2000 environment in the section “Splitting the Cloned Volume While Running Live Transactions.”

For an even more detailed description of how FlexVol and FlexClone technologies function, please see supplementary TRs listed below:

Data ONTAP 7G—The Ideal Platform for Database Applications	http://www.netapp.com/tech_library/ftp/3373.pdf
--	---

FlexVol Volumes in the SQL Server Environment

Today's SQL Server administrator is confronted with a variety of challenges around several functional areas:

- Space management and predictive growth
- I/O sizing for performance
- Managing database upgrades and patches
- Managing data-centric application upgrades like SAP, Siebel, and Remedy

A flexible volume can be grown or shrunk at will, lending a greater level of granularity in managing the inevitable changes in database space requirements. If volumes are too large for one database, the unused space can quickly be reallocated to another server as needed. By contrast, particularly in direct-attached storage environments, space allocated to a SQL Server remains dedicated regardless of whether space or throughput is utilized. As more databases are added to an environment in this manner, storage asset utilization decreases.

Flexible volumes allow administrators to aggregate massive spindle counts across all their SQL Server databases, effectively boosting performance and throughput to unprecedented levels without sacrificing control over space management. Consider Figure 2, in which six SQL Server databases have consolidated all their volumes into 12 flexible volumes contained within one large 100+ disk aggregate.

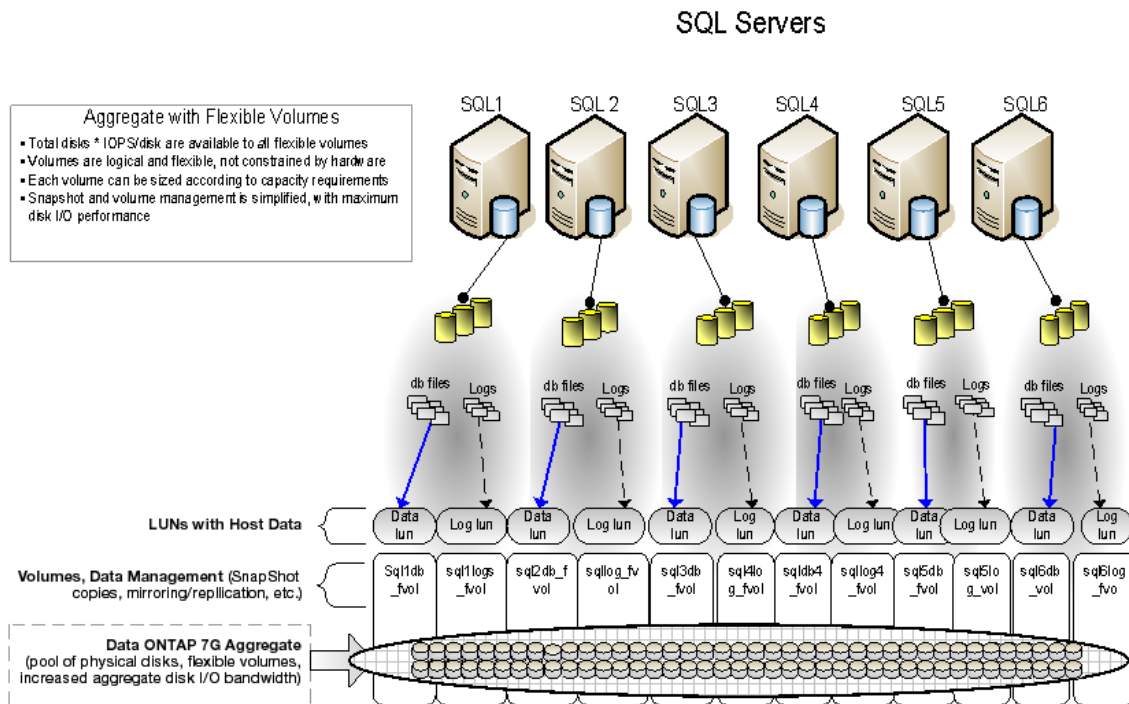


Figure 2) Flexible volumes for SQL Server.

Each database leverages the power and reliability of over 100 disk spindles and multiple RAID groups, transcending many of the inherent performance and reliability challenges of direct-attached storage (DAS) or even traditional SAN environments. If, for example, the databases on the servers SQL1 through SQL6 required additional space for any of their databases, their respective LUNs can easily be grown on-the-fly, using SnapDrive, with no impact to database availability. The previous challenges of storage inflexibility and low utilization associated with dedicated physical volumes have been virtually alleviated. Hence, flexible volumes allow administrators to quickly and effectively scale their infrastructure *out and up*, with virtually no impact to system availability. Because space and IOPS reside in a shared environment, storage asset utilization quickly surpasses that of traditional volume environments.

FlexClone in the SQL Server Environment

Descending from a flexible volume Snapshot copy, a cloned volume provides another layer of flexibility by enabling administrators to create instantaneous database clones, providing a replica of production data in a risk-free designated "sandbox." A FlexClone volume is created in seconds, whether the clone command is executed on a small 10MB volume or an enormous 16TB volume. Alternatively, to realize the benefits of a similar functionality, a DBA would be forced to recover an entire backup from tapes or nearline devices into a test environment. When databases grow to several hundred gigabytes or even terabytes of data, an organization is forced to create an entirely new instance of equal or greater storage capacity in a test environment to effectively test upgrades and modifications before they go to production. The conventional method is time consuming, more costly, and less efficient than using a cloned volume because by the time a recovery environment is recreated, the delta between production and testing environments may have diverged significantly, thereby resulting in lower accuracy of testing. Figure 3 compares tape, mirroring, and FlexClone technologies over the time continuum.

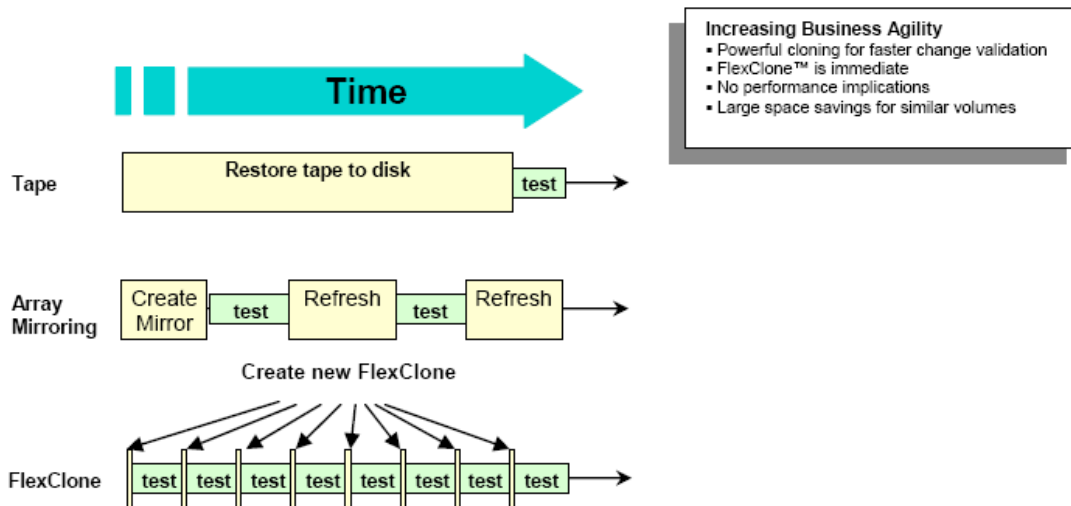


Figure 3) Tape, mirroring, and FlexClone over the time continuum.

In the time it takes to create one tape-generated test environment, eight clone environments were generated and tested using FlexClone. While array mirroring provides a slightly better recovery rate than that of tape, it is still less efficient than using FlexClone for testing purposes.

FlexClone volumes increase ROI of the overall storage investment by providing fast and dynamic instances of production data without awkward, expensive, and time-consuming recovery processes associated with conventional data recovery. Database upgrades, schema modifications, and patch testing can all be quickly tested against a FlexClone replica environment. Such testing enables the entire IT organization to quickly investigate the repercussions of making changes, without adversely affecting production environments. Unpredictable upgrades, patches, and modifications are quickly and economically validated, thereby demystifying the outcome and risks of such changes. Flexible clones enable DBAs and administrators to play out many "what if" scenarios against production data until they produce expected or acceptable results.

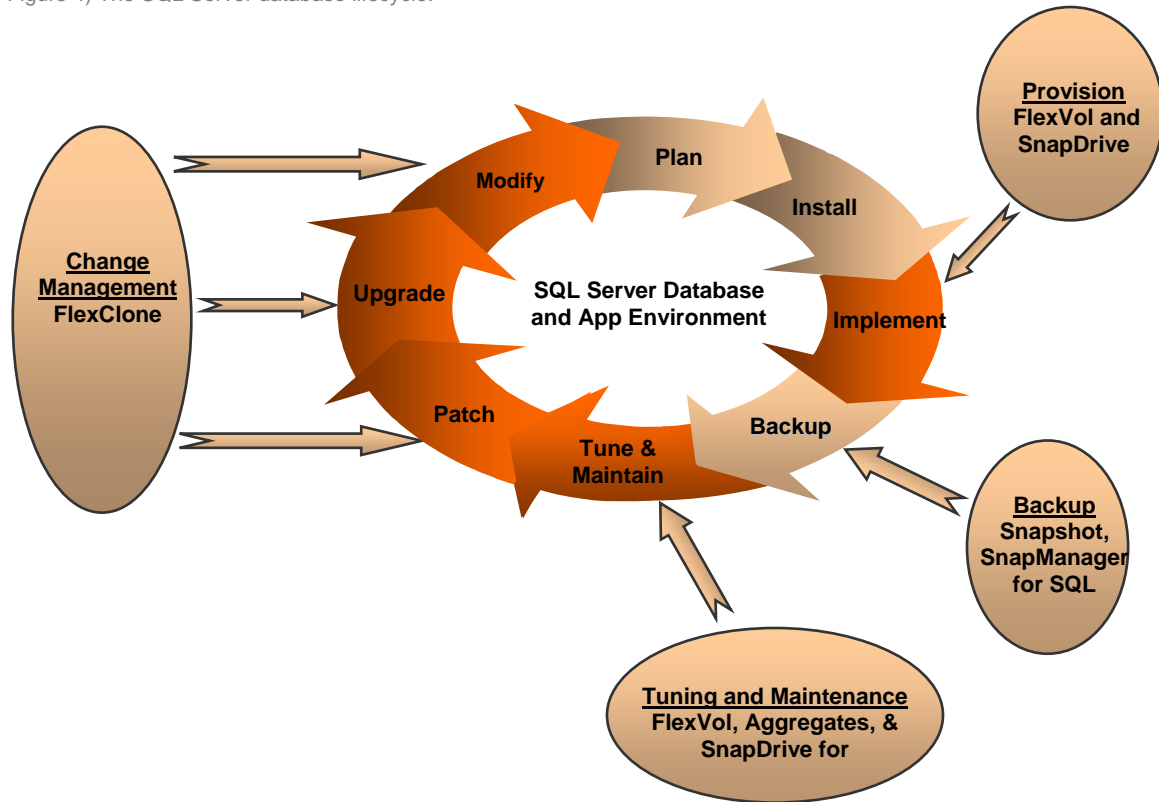
To better understand the overall context in which FlexClone and FlexVol technology adds value, let us now take a closer look at the lifecycle management of SQL Server databases.

Microsoft SQL Server Database Lifecycle

The database lifecycle consists of an ongoing and cyclical series of phases, shown in Figure 4. SQL Server administrators and developers viewing the categories identified below, can probably easily identify in what phase each of their databases currently resides. Network Appliance has taken holistic a view and provided enterprises with innovations such as Snapshot, FlexClone, and FlexVol to help improve the flexibility of the entire lifecycle of the database and software environment.

Figure 4 identifies the key areas in which using aggregates, FlexClone, FlexVol, and Snapshot technology empowers the SQL Server administrator to increase the speed and efficiency with which databases are orchestrated throughout the database lifecycle.

Figure 4) The SQL Server database lifecycle.



The list below briefly sites some examples for which SQL Server DBAs benefit from the use of these technologies:

Provision

- Quickly provision aggregates, flexible volumes, and LUNs at will

Backup

- Use SnapManager for instantaneous SQL database snapshots. Integrate with SnapMirror for offsite data recovery.

Tune & Maintain

- **Tune**—Increment spindle count to optimize database aggregates
- **Maintain**—Grow LUNs on-the-fly with no impact to system downtime

Change Management

- **Patch**—Apply Service Pack 4 to a SQL Server 2000 server, W32/SQLSlammer.worm patch and others in on a cloned SQL Sever instance for safe and accurate validation
- **Upgrade**—Go from SQL Server 2000 to SQL Server 2005; quickly clone and validate databases that provide the backend data store for mission critical applications such as SAP, Remedy, and Siebel
- **Modify**—make code-level modifications to database objects (stored procedures, triggers, and schema)

These are just a few examples of how SQL Server database administrators can use FlexClone and FlexVol in nearly every phase of the database lifecycle. By providing instantaneous clones of production environments without increasing storage costs, overall productivity is greatly enhanced. Now we will take a more detailed, hands-on look at how to use the functionality described thus far.

Creating a Flexible Volume, FlexClone Volume with Data ONTAP 7G, SnapDrive, and SnapManager 1.5

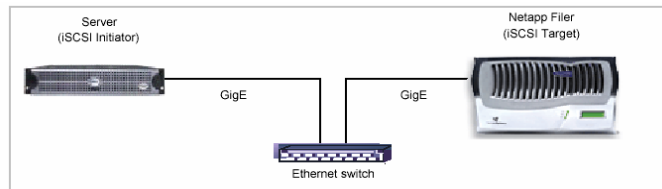
This section will show a method of creating and using flexible volumes and FlexClone volumes with SnapManager for SQL and SnapDrive. We will also demonstrate the overall business context with simulated real-world examples

Network Appliance Inc.

of several phases of the database lifecycle. Be advised that all filer-based steps demonstrated can be accomplished using any of the following methods: (1) Telnet session, (2) rsh (remote shell) or (3) FilerView®—the filer management Web interface. We will use RSH, SnapDrive GUI, and the command line interface for SnapManager for SQL called `smsqlbi.exe`.

Environment

We are using a Windows 2000 server SP4 with SQL Server 2000 SP3 installed called “SQL2K” connected to a filer called “FAS980” running over iSCSI through a Cisco GB Ethernet switch. The diagram below gives a simple overview of how the devices are connected.



1. Provisioning

Provisioning with Aggregates, FlexVol Volumes, and SnapDrive	
1.1	Create an aggregate on our filer called “AggrSQL” consisting of 3 16-disk RAID groups.
1.2	Create three FlexVol volumes on the aggregate “AggrSQL” of various sizes: A. “SQLData” B. “SQLLogs” C. “SnapInfo”
1.3	Create CIFS shares for each volume. (Note: The CIFS share is the protocol used not to attach to the database but just to broadcast the share name for mounting purposes. iSCSI will be used for connecting to the databases as mentioned earlier.)
1.4	Create three NTFS formatted LUNs on each flexible volume using SnapDrive: A. One for the database files → <code>dbfile.lun</code> B. One for logs → <code>logs.lun</code> C. One for SnapInfo for SnapManager for SQL 1.5. → <code>snapinfo.lun</code> Mount and assign drive letters to all three LUNs using SnapDrive on our Windows 2000 SQL server.
1.5	Migrate the core user database “Northwind” data and log files from its current direct-attached storage to each corresponding LUN and flexible volume with SnapManager for SQL

1.1 Creating the Aggregate

Using the `rsh` command from the Windows command prompt, run the following command to generate the 48-disk aggregate consisting of three 16-disk RAID-DP groups aggregate. (You may deviate here as needed by using different raid group sizes and disk counts.)

```

Command:
rsh fas980 -lroot aggr create AggrSQL -r 16 48
  
```

```

Output:

Creation of an aggregate with 48 disks has been initiated. The disks need
to be zeroed before addition to the aggregate. The process has been initiated
and you will be notified via the system log as disks are added.
  
```

If your disks are not yet zeroed, your filer will need to complete this task first, before it will proceed to building the aggregate. While waiting for your aggregate to generate, you can run another command to check the status of the process.

```

Command:
rsh fas980 -lroot aggr status AggrSQL
  
```

Output:

```
Aggr State Status Options
AggrSQL creating raid_dp, aggr raidsize=16,
initializing snapshot_autodelete=off,
lost_write_protect=off Volumes: <none>
Plex /aggtest/plex0: offline, empty, active
```

In this example, you can see the FAS980 is still generating the aggregate. When the aggregate has been generated, running the status command will yield the results shown below:

Command:

```
rsh fas980 -lroot aggr status AggrSQL
```

Output:

```
Aggr State Status Options
AggrSQL online raid_dp, aggr raidsize=16
Volumes: <none>

Plex /AggrSQL/plex0: online, normal, active
RAID group /AggrSQL /plex0/rg0: normal
RAID group /AggrSQL /plex0/rg1: normal
RAID group /AggrSQL /plex0/rg2: normal
```

You can see that our aggregate has been created and is now online, consisting of three RAID groups.

1.2 Creating Three FlexVol Volumes

Now we are ready to create our FlexVol volumes on AggrSQL starting with a 100G volume for our database file.

Command:

```
rsh fas980 -lroot vol create sqldata AggrSQL 100G
```

Output:

```
Creation of volume 'sqldata' with size 100g on containing aggregate
'AggrSQL' has completed.
```

Now create the log volume:

Command:

```
rsh fas980 -lroot vol create sqllog AggrSQL 30G
```

Output:

```
Creation of volume 'sqllog' with size 30g on containing aggregate
'AggrSQL' has completed.
```

Finally, create a snapinfo volume:

Command:

```
rsh fas980 -lroot vol create snapinfo AggrSQL 10G
```

Output:

```
Creation of volume 'snapinfo' with size 10g on containing aggregate
'AggrSQL' has completed.
```

1.3 Create Three CIFS shares

You will notice that each flexible volume is generated in fractions of a second. Next, set up your CIFS shares for your new FlexVol volumes.

Command:

```
rsh fas980 -lroot cifs shares -add sqldata /vol/sqldata
rsh fas980 -lroot cifs shares -add sqllog /vol/sqllog
rsh fas980 -lroot cifs shares -add snapinfo /vol/snapinfo
```

1.4 Create Database NTFS LUNs

Now that our flexible volumes are created and visible, we can generate our LUNs for our SQL Server instance using SnapDrive 3.X For more information about installation and configuration of both SnapManager and SnapDrive products,

refer to the product documentation located on the NOW™ support site: <http://now.netapp.com> (requires login information). For the purposes of this demonstration, we will assume SnapDrive 3.X and SnapManager for SQL 1.5 are both installed and configured. If you have not yet installed SnapManager and SnapDrive and would like to walk through the steps outlined in this TR, complete the installation of both products before proceeding to the next section.

Using SnapDrive, create three NTFS formatted LUNs on each flexible volume using SnapDrive:

- A. One for the database files → dbfile.lun on "/fas80/sqldata"
- B. One for logs → logs.lun on "/fas80/sqllog"
- C. One for SnapInfo for SnapManager for SQL 1.5. → snapinfo.lun on "/fas980/snapinfo"

1.5 Migrating Your SQL Server User Databases to Flexible Volumes with SMSQL 1.5

For migration instructions, refer to your SnapManager 1.5 Installation and Administration guide, page 87. Move all database data files for your user database to the *sql/data* volume and all log files to the *sql/logs* flexible volume.

2. Tune and Maintain

Tune and Maintain with Aggregates and SnapDrive	
2.1	Add an additional 16 spindles RAID-DP group to our aggregate, making the total spindle count 64 disks.
2.2	Grow LUN with SnapDrive on-the-fly.

2.1 Increasing Spindle Count On-the-fly: Adding Another RAID Group to the Aggregate

After deployment, SQL DBAs and storage administrators often find themselves in the position of having to make changes and tune their storage environments to more effectively match their unique database requirements.

After we have created our flexible volumes and aggregates, we realized we should have really added a total of four 16-disk RAID-DP groups rather than just three. Things like this can happen in the Tune phase of the database lifecycle. With aggregates and flexible volumes it's very easy to add additional RAID groups and disks as needed over time. In this example, we will add another 16-disk RAID group to our current aggregation of three 16-disk groups. From the `rsh` command prompt, we conducted the following steps:

```

Command:
rsh fas980 -lroot aggr add AggrSQL 16

Output:
Continue? ([y]es, [n]o, or [p]review RAID layout) y
Addition of 16 disks to the aggregate has been initiated. The disks need
to be zeroed before addition to the aggregate. The process has been initiated
and you will be notified via the system log as disks are added.
```

You may watch the status of the RAID group add by executing the following command:

```

Command:
rsh fas980 -lroot aggr add AggrSQL 16

Output:
Continue? ([y]es, [n]o, or [p]review RAID layout) y
Addition of 16 disks to the aggregate has been initiated. The disks need
to be zeroed before addition to the aggregate. The process has been initiated
and you will be notified via the system log as disks are added.

Command:
rsh fas980 -lroot aggr status AggrSQL

Output:
Aggr State Status Options
aggctest online raid_dp, aggr raidsize=8
growing
Volumes: sqldata, sqllog, snapinfo
```

While the growth process is taking place, the fourth RAID plex is not yet online.

```

Plex /aggctest/plex0: online, normal, active
RAID group /aggctest/plex0/rg0: normal
RAID group /aggctest/plex0/rg1: normal
RAID group /aggctest/plex0/rg2: normal
```

When it is complete, the status command will yield the following results:

```

Command:
rsh fas980 -lroot aggr status AggrSQL

Output:

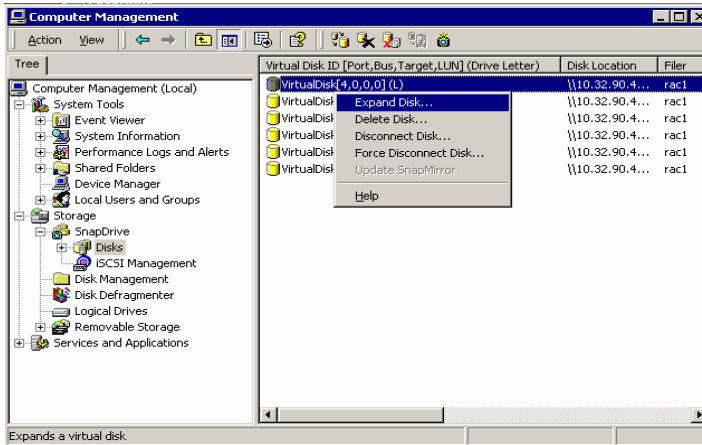
Aggr State Status Options
AggrSQL online raid_dp, aggr raidsize=16
Volumes: <none>

Plex /AggrSQL/plex0: online, normal, active
RAID group /AggrSQL /plex0/rg0: normal
RAID group /AggrSQL /plex0/rg1: normal
RAID group /AggrSQL /plex0/rg2: normal
RAID group /AggrSQL /plex0/rg3: normal
    
```

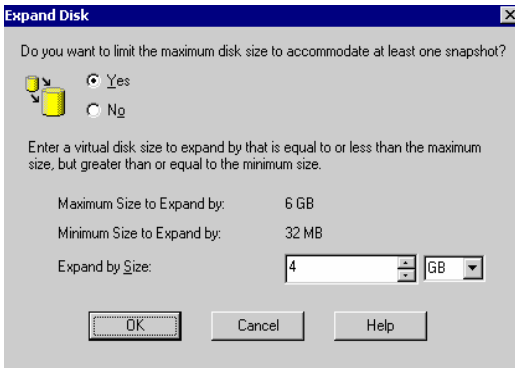
2.2 Grow LUN with SnapDrive On-the-Fly

NetApp storage-based LUNs are relinquished from traditional physical disk dependencies and can be grown on-the-fly, with no impact to database availability.

1. Go to your Computer Management console.
2. Expand the Storage section, SnapDrive, and then Disks to view the Disks section.



3. Right-click and choose Expand Disk.



4. Click OK.
5. In mere seconds the LUN will complete its growth, showing the message below.



3. Change Management with FlexClone, SnapDrive, and Snapshot

Change Management with SnapManager snapshot backups, SnapDrive and FlexClone	
3.1	Generate two SMSQL snapshots of both the <i>sqldata</i> and <i>sqllog</i> flexible volumes.
3.2	Clone the SQL Server Databases using SMSQL Snapshot copies.
3.3	Set up new CIFS shares.
3.4	Connect a LUN to each FlexClone™ volume with SnapDrive back to the server SQL2K.
3.5	Attach the FlexClone copy of Northwind to our SQL Server under an alias.
3.6	Simulate load on the database with a T-SQL script and execute a FlexClone split.

3.1 Generating SMSQL Snapshot Copies

Creating a flexible clone of your SQL Server database LUNs involves two steps: (1) Creating a clean Snapshot copy of the SQL Server databases with the SnapManager 1.5 command line interface or through the SnapManager GUI and (2) generating a clone of each resulting FlexVol Snapshot copy from the SnapManager Snapshot copy. SnapManager for SQL integrates with the SQL Server VDI (virtual device interface) API to effectively quiesce the database and flush all related NTFS buffers before taking a Snapshot copy. This ensures that the database is in a consistent state when the Snapshot copy is taken.

From your server command prompt, navigate to the *C:\Program Files\NetApp\SnapManager for SQL Server* directory and execute the `smsqlbi` command and replace your server name with the server "sql2k" written below:

```
Command:
smsqlbi -H sql2k -S sql2k -C 0
```

Wait a few moments for Snapshot to quiesce and copy the SQL Server databases. This will generate a Snapshot copy called `sqlsnap__sql2k__recent` of all flexible volumes that contain database files.

3.2 FlexClone SQL Server Database from SMSQL Snapshots

Run the next command to clone each flexible volume that contains your database and log files. For the purposes of this example, we will clone the volumes *sqldata* and *sqllog*. You may also follow the SMSQL wizard to back up your databases.

```
Command:
rsh fas980 -lroot vol clone create sqllogclone -b sqllog sqlsnap__sql2k__recent
rsh fas980 -lroot vol clone create sqldataclone -b sqldata sqlsnap__sql2k__recent
```

```
Output (NOTE: Only output from the log clone output shown here)
Mon May 9 19:42:33 PDT [waf1.snaprestore.revert:notice]: Reverting volume sqllogclone to a
previous snapshot.
Mon May 9 19:42:33 PDT [waf1.vol.guarantee.clone:info]: Space for clone volume
'sqllogclone' is not guaranteed.
Creation of clone volume 'sqllogclone' has completed.
Mon May 9 19:42:33 PDT [lun.newLocation.offline:warning]: LUN /vol/sqllogaclone/sqldata.lun
has been taken offline to prevent map conflicts after a copy or move operation.
```

```
Command:
rsh fas980 -lroot vol clone create sqllogclone -b sqllog sqlsnap__sql2k__recent
```

```
Output
Mon May 9 19:50:00 PDT [waf1.snaprestore.revert:notice]: Reverting volume sqllo
gclone to a previous snapshot.
Mon May 9 19:50:00 PDT [waf1.vol.guarantee.clone:info]: Space for clone volume
'sqllogclone' is not guaranteed.
Creation of clone volume 'sqllogclone' has completed.
```

```
Mon May 9 19:50:00 PDT [lun.newLocation.offline:warning]: LUN /vol/sqllo
gclone/logs.lun has been taken offline to prevent map conflicts after a copy or
move operation./sql.lun has been taken offline to prevent map conflicts after a copy or move
operation.
```

3.3 Set Up the CIFS Share at the Flexible Volume Level

Command:

```
rsh fas980 -lroot cifs shares -add sqllogsclone /vol/sqllogsclone/
```

Output:

```
The share name 'sqllogsclone' will not be accessible by some MS-DOS workstations
Are you sure you want to use this share name? [n]:y
```

You will need to unmap the LUNs from their respective iGroups so SnapDrive can add them back. In the case of our cloned LUNs, they are already mapped via iSCSI to `viaRPC.iqn.1991-05.com.microsoft:sql2k.dblab.netapp.com`

Command:

```
rsh fas980 -lroot lun unmap /vol/sqllogsclone/sqllog.lun viaRPC.iqn.1991-05.com.microsoft:sql2
k.dblab.netapp.com
```

Output:

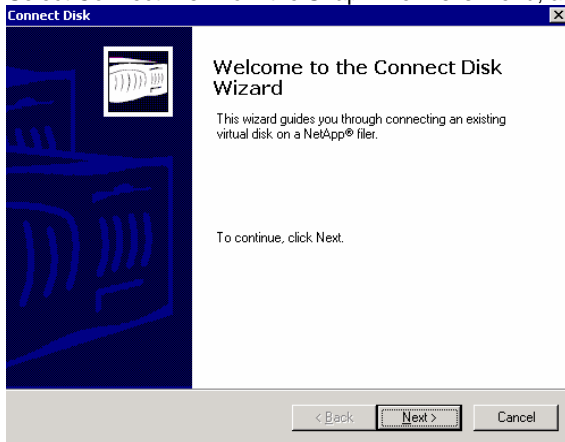
```
Wed May 4 12:23:50 PDT [lun.map.unmap:info]: LUN /vol/sqllogsclone/sql.lun unma
pped from initiator group viaRPC.iqn.1991-05.com.microsoft:sql2k.dblab.netapp.com
```

Repeat the same steps for the `sqldataclone` clone.

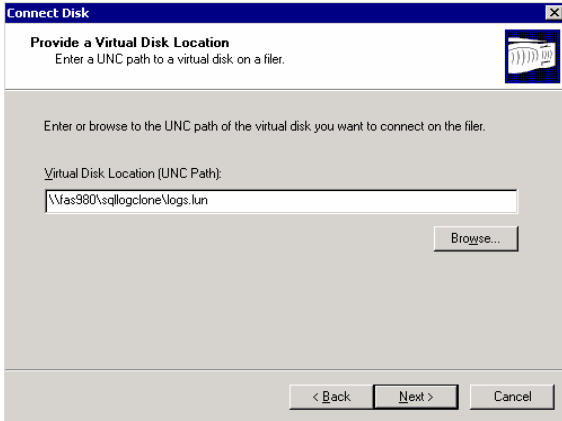
3.4 Connect a LUN to Each FlexClone Volume with SnapDrive Back to the Server SQL2K

Next, using SnapDrive, we will attach each LUN in the FlexClone LUNs to our server. **Note:** You may also attach the database to any other SQL Server.

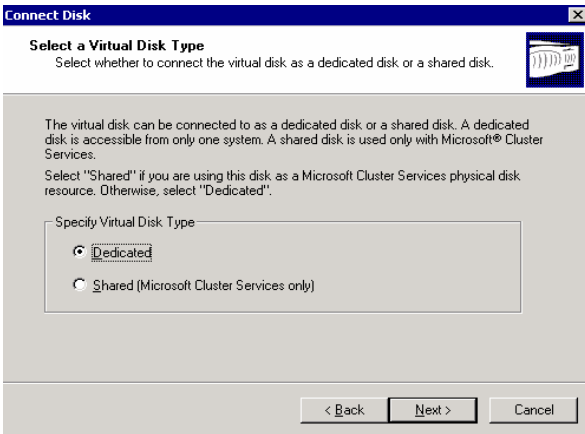
Select Connect Disk from the SnapDrive Disks menu, and click Next.



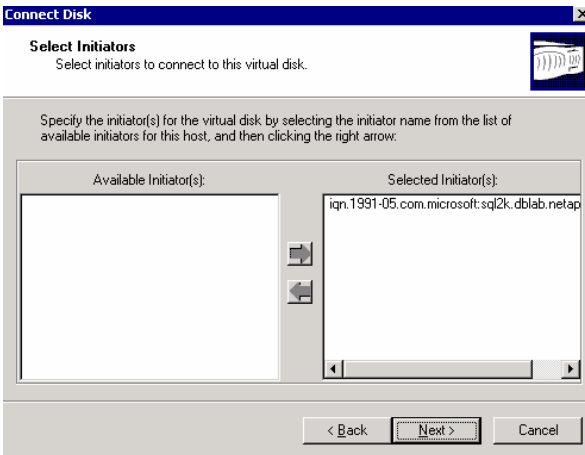
Provide the UNC to the path of the cloned LUN, and click Next.



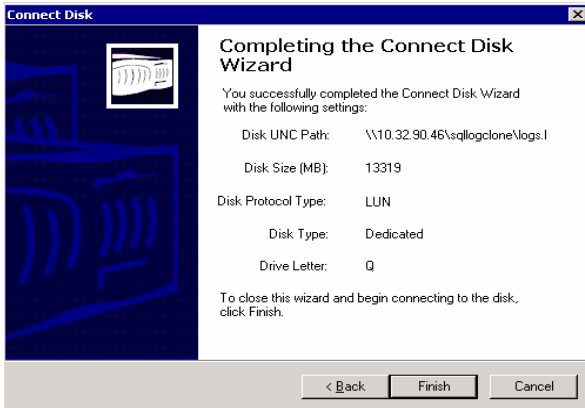
We are not in an MSCS environment, so select Dedicated and click Next.



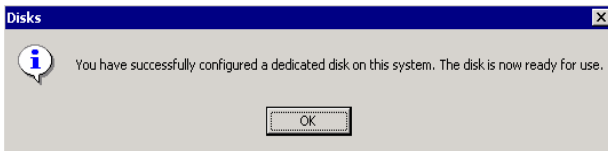
Choose a drive letter, and click Next.



Choose your initiator, and click Next



Click Finish and SnapDrive will connect your cloned LUN.

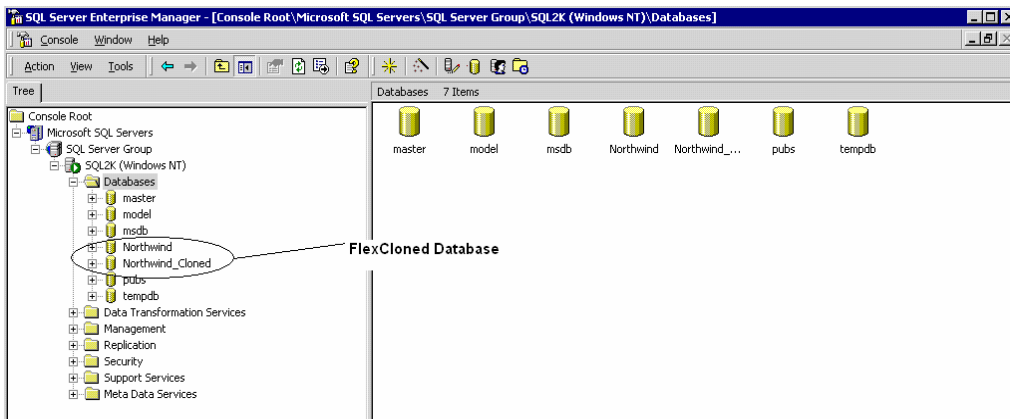


Repeat these steps for all flexible volumes and LUNs associated with the database you would like to attach. In this example, we will assume all volumes have completed the steps describe above and all database LUNs are attached to our SQL Server. We have assigned drive letter X to the *sqldata* LUN and H to the *sqllog* LUN.

3.5 Attaching SQL Server Database Clone to SQL Server Instance

Now that our two LUNs are mounted on our SQL Server with SnapDrive, we can attach our cloned database— under the alias "Northwind_Cloned" — to the same SQL Server instance. Using the system stored procedure *sp_attach_db*, we will run our SQL Query Analyzer and execute the *attach* command. You may also recover the database to another SQL Instance or an entirely different SQL Server.

```
sp_attach_db 'Northwind_Cloned',
'x:\program files\Microsoft SQL Server\MSSQL\data\northwnd.mdf',
'h:\program files\Microsoft SQL Server\MSSQL\data\northwnd.ldf'
```



3.6 Splitting the Cloned Volume While Running Live Transactions

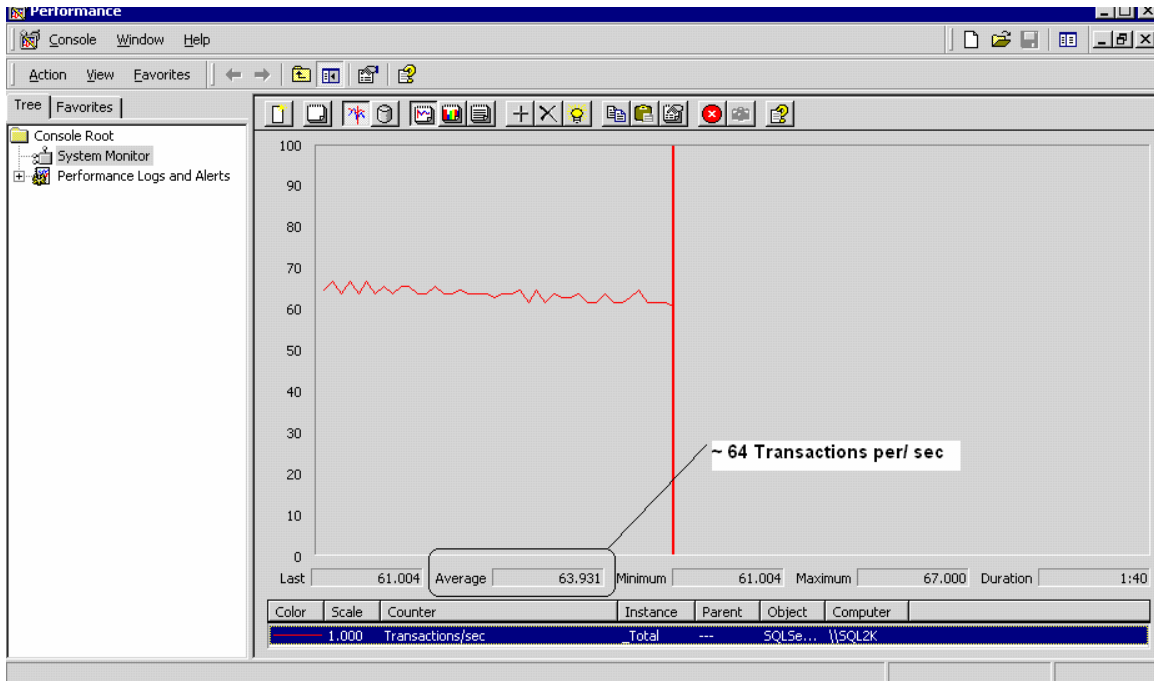
Now we will execute a SQL script to put a load on the FlexClone copy and then execute a FlexClone split. Here is a script, set up to run for five minutes, during which we will run the split.

```

/*First we will fire off a simple T-SQL script that loops for five minutes and generates inserts and
selects
create test tables */
CREATE TABLE TEST (col1 int, col2 int, col3 int, col4 int)
CREATE TABLE TEST2 (col1 int, col2 int, col3 int, col4 int)
CREATE CLUSTERED INDEX col1_ind ON TEST (col1) WITH FILLFACTOR = 10
CREATE CLUSTERED INDEX col3_ind ON TEST2 (col3) WITH FILLFACTOR = 10
GO

/*5 minute loop to insert a bunch of rows and select min and max of random values */
DECLARE @n int
DECLARE @t datetime
set @t=getdate()
set @n = 0
WHILE datediff(ss,@t,getdate())<301 BEGIN
    select @n = @n +1
    insert into TEST VALUES(@n, @n+1, @n+2, @n+3)
    insert into TEST2 VALUES(@n, @n+1, @n+2, @n+3)
    select min(col2) from TEST
    select max(col3) from TEST2
END
GO
    
```

As you can see, we have used `perfmon` to demonstrate the transactions per second, showing roughly 64 transactions per second or roughly 3900 transactions per minute (TPM).



While this load is running, we have executed a `clone split` command to the filer, to split the volumes on which the Northwind_Clone resides.

Command:

```
rsh fas980 -lroot vol clone split start sqldataclone
```

Output:

```
Wed May 4 17:01:06 PDT [waf1.scan.start:info]: Starting volume sqldataclone.
Clone volume 'sqldataclone' will be split from its parent.
```

Monitor system log or use 'vol clone split status' for progress.

Command:

```
rsh fas980 -lroot vol clone split start sqllogclone
```

Output:

```
Wed May 4 17:01:06 PDT [waf1.scan.start:info]: Starting volume sqllogclone.
Clone volume ' sqllogclone ' will be split from its parent.
Monitor system log or use 'vol clone split status' for progress
```

To monitor the split, we ran the status command every few moments to monitor the process.

Command:

```
rsh fas980 -lroot vol clone split status
```

Output:

```
Volume 'sqldataclone', 28036 of 3640875 inodes processed (0%)
7261347 blocks scanned. 212923 blocks updated.
Volume 'sqllogclone', 27932 of 1820427 inodes processed (1%)
3501326 blocks scanned. 114290 blocks updated.
-----
Volume 'sqldataclone', 753802 of 3640875 inodes processed (20%)
7261347 blocks scanned. 212923 blocks updated.
Volume 'sqllogclone', 753698 of 1820427 inodes processed (41%)
3501326 blocks scanned. 114290 blocks updated.
-----
Volume 'sqldataclone', 1515465 of 3640875 inodes processed (41%)
7261347 blocks scanned. 212923 blocks updated.
Volume 'sqllogclone', 1515361 of 1820427 inodes processed (83%)
3501326 blocks scanned. 114290 blocks updated.
-----
No clone status
-----
```

The flexible volume split took place even in the OLTP climate of nearly 3900 TPM running against the database. To validate the total rows affected by the T-SQL script, we have run a few counts on the TEST and TEST2 tables shown below.

```
select count(*) from TEST
GO
select count(*) from TEST2
GO
-----
24562
(1 row(s) affected)

-----
24562
(1 row(s) affected)
```

All tasks demonstrated in the above steps, with the exception of those conducted in Microsoft SQL Query Analyzer, can also be completed entirely through GUI tools such as SnapManager for SQL and Network Appliance™ FilerView. Please refer to your SnapManager 1.5, SnapDrive 3.X and DATA ONTAP 7G documentation for task-specific information.

Conclusions

By aggregating spindles normally sliced up across many stove-pipe database storage instances, FlexVol volumes unleash the performance potential of hundreds of spindles working as a cohesive leveraged storage grid. SQL Server DBAs and storage administrators can carve out appropriate slices of the storage tailored to the unique requirements of their databases. As time progresses, database storage can be grown on-the-fly, with no impact to system availability. We have also seen how FlexClone multipurposes storage environments by capturing instantaneous, space- efficient database clones to help more effectively drive nearly all phases of the entire database lifecycle. For more information about any of the solutions or products covered in this TR, please contact Network Appliance.

© 2005 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp, the Network Appliance logo, FilerView, SnapManager, and WAFL are registered trademarks and Network Appliance, Data ONTAP, FlexClone, FlexVol, NOW, RAID-DP, SnapDrive, and Snapshot are trademarks of Network

Appliance, Inc. in the U.S. and other countries. Microsoft and Windows are registered trademarks of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.



Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089
www.netapp.com

© 2005 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp, NetCache, and the Network Appliance logo are registered trademarks and Network Appliance, DataFabric, and The evolution of storage are trademarks of Network Appliance, Inc., in the U.S. and other countries. Oracle is a registered trademark of Oracle Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.