



# Block Management with Data ONTAP™ 7G: FlexVol™, FlexClone™, and Space Guarantees

Jim Lanson | Network Appliance | October 2004 | TR 3348

## 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.

## Table of Contents

<b>1) Abstract</b>	<b>3</b>
<b>2) Audience</b>	<b>3</b>
<b>3) Overview</b>	<b>3</b>
Storage Architecture	3
NetApp Space Guarantees	4
Fractional Reserve Option	5
<b>4) Volumes and LUNs</b>	<b>6</b>
<a href="#">Traditional or Flexible Volumes?</a>	
Volume Sizing and LUN Creation	6
Best Practices	8
<b>5) LUN Management</b>	<b>8</b>
Snapshot Copies	8
LUN Reallocation Scans	8
Best Practices	13
Snap Delta	14
Snap Reclaimable	15
<b>6) Cloning and Snapshot Copies</b>	<b>15</b>
LUN Cloning	15
Volume Cloning	16
Clone a Volume or Clone a LUN?	17
<b>7) Summary</b>	<b>18</b>
<b>8) Additional Resources</b>	<b>18</b>

## 1) Abstract

The purpose of this document is to discuss the impact of Data ONTAP 7G with the new FlexVol and FlexClone creation and management of LUNs. In addition, in this document, the differences between creating LUNs on traditional volumes versus FlexVol are discussed. The treatment of LUNs described in this document applies to LUNs created for use with either the Fibre Channel or iSCSI protocol. Recommendations regarding best practices will be provided as necessary.

## 2) Audience

This paper assumes the reader has a basic familiarity with Data ONTAP administration and block management. While reviews of some functionality existing prior to Data ONTAP 7G are provided, the primary focus of this paper is on the new functionality introduced in Data ONTAP 7G and how it relates to LUN management.

## 3) Overview

### Storage Architecture

Before exploring the impact on block management of Data ONTAP 7G, a review of the Data ONTAP 7G storage architecture is in order.

The following are the basic units of storage, their purpose, and how they relate to one another (Figure 1):

**RAID groups** have the same functionality as in previous Data ONTAP releases. They can be constructed using either NetApp RAID4 or RAID-DP™. The difference between the two is described in other publications. RAID groups consist of a number of disk spindles allocated for data, parity, or diagonal parity. Before Data ONTAP 7G, a single volume was built using one or more of these RAID groups.

**Aggregates**, introduced in Data ONTAP 7G, are built using one or more RAID groups. These RAID groups can be grouped into plexes within an aggregate to facilitate synchronous mirroring. Multiple volumes can be built upon these aggregates.

**Volumes** can be designated either traditional or flexible. A traditional volume is directly tied to the aggregate upon which it is built. This is the model provided in previous releases of Data ONTAP. A flexible volume, or FlexVol volume, is more loosely tied to its aggregate. This allows multiple volumes to be created across the same underlying disk infrastructure, thus maximizing flexibility and scalability. The choice of either type of volume is specified during volume creation. Volumes that have been migrated from previous Data ONTAP releases will be brought forward as traditional volumes.

**Qtrees, files, and logical unit numbers (LUNs)** are built on top of volumes of either type. A qtree is a subdirectory of the root directory of a volume. Files for the network-attached storage environment or LUNs for the SAN environment can be created at either the volume or the qtree level.

Further information can be obtained from Chapter 1, “Introduction to NetApp Storage Architecture,” in the *Data ONTAP System Administration Storage Management Guide*.

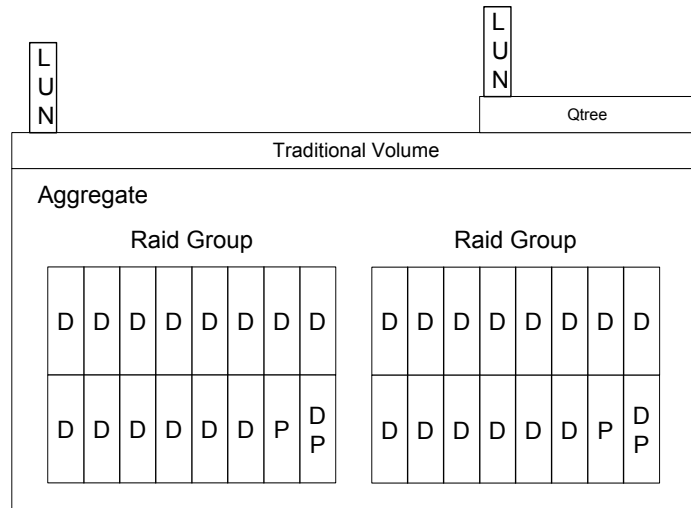


Figure 1a) Traditional volume.

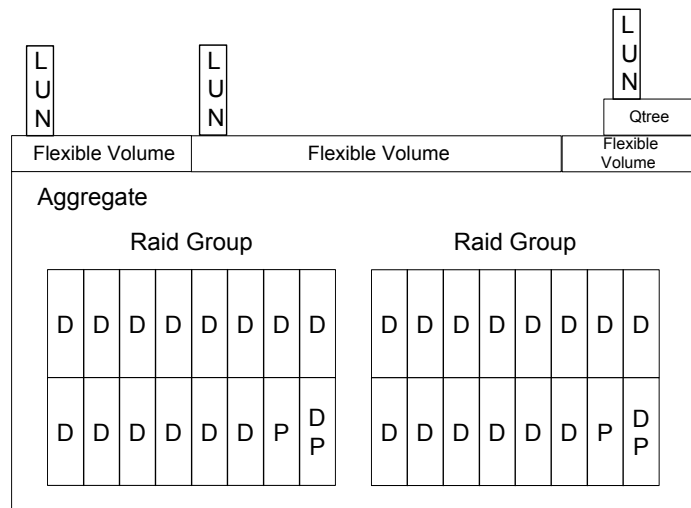


Figure 1b) Flexible volume.

### NetApp Space Guarantees

Space guarantees are used to guarantee that space is available for completing writes to a LUN or for overwriting data in a LUN.

LUNs appear as local disks attached to the host. Because of this, LUNs must act as though they are locally attached disks. This means that when a LUN of a certain size is created, the application storing data on the LUN must be guaranteed that amount of space.

When you create a LUN of a specified size, Data ONTAP turns the space guarantees on by default and reserves the number of blocks (not the physical blocks) in the WAFL<sup>®</sup> volume that are required to complete writes to the LUN. This ensures that writes to the LUN do not fail due to a lack of disk space in the volume (Figure 2).

## Fractional Reserve Option

Fractional reserve controls the amount of space a volume reserves for overwrites to space-reserved LUNs. When you create a volume, the `fractional_reserve` option is by default set to 100. This means that Data ONTAP automatically reserves 100% of the total LUN size for overwrites.

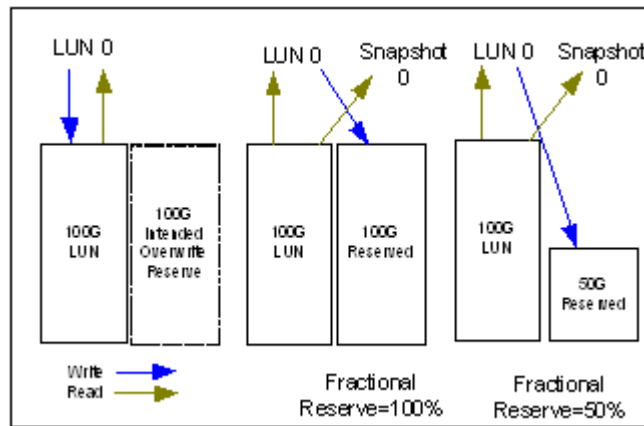


Figure 2)

For example, if you create a 100GB LUN, Data ONTAP by default ensures that the host-side application storing data in the LUN always has access to 100GB of space.

You use the following command to set `fractional_reserve`:

```
vol options vol-name fractional_reserve pct
```

*pct* is the percentage of the LUN you want to reserve for overwrites. You can specify a value from 0 to 100.

You can reduce the amount of space reserved for overwrites to a percentage of the total LUN size when you create LUNs in the following types of volumes:

Traditional volumes

Flexible volumes that have the `guarantee` option set to `volume`

The amount of space reserved for overwrites is based on the total size of all space-reserved LUNs in a volume. For example, if there are two 200GB LUNs in a volume, and the `fractional_reserve` option is set to 50%, then Data ONTAP guarantees that the volume has 200GB available for overwrites to those LUNs.

**It is extremely important to keep in mind that a change in fractional reserve percentage might result in the failure of a write operation should the change in that LUN exceed that percentage (frequent monitoring of space is recommended).** Therefore, one must be sure of the change rate of the data in the LUN before changing this option (see description of `snap delta` later). As a best practice, it would be best to leave it at 100% for a time, check the change rate, and then lower the percentage accordingly. In addition, keep in mind that when

multiple LUNs exist on the same volume, this percentage should be set to accommodate that LUN with the highest change rate.

## 4) Volumes and LUNs

### Traditional or Flexible Volumes?

One of the first questions that may arise with the release of Data ONTAP 7G is whether it is better to have LUNs reside on traditional volumes or FlexVol volumes. In order to make this decision, it may be best to look at the challenges placed on LUNs by traditional volumes.

In traditional volumes, there is a direct relationship of a volume to one or more RAID groups of disks. If a volume is built on a RAID group with many spindles for performance reasons, then any LUN needing the same level of performance has to exist on that volume and share a Snapshot™ copy schedule or on another volume that uses up more disks. With flexible volumes, each LUN could reside on a different volume (up to the limit of the number of volumes) and have a different Snapshot copy schedule.

### Volume Sizing and LUN Creation

Volume sizing for LUNs is determined as before, with the exception of the increased flexibility provided by FlexVol volumes. If you do not intend to keep Snapshot copies in the volume, the size of a traditional volume must be equal to the sum of the sizes of all LUNs on that volume plus 10%. A flexible volume does not need the extra 10%. If Snapshot copies or applications that utilize Snapshot copies—for example, SnapMirror®, SnapRestore®, etc.—are being used, then these factors must also be added to the size of the volume.

The amount of space needed for Snapshot copies will depend on the change rate of the data, the fractional\_reserve setting, and the type of space reservation guarantee in place (volume, file, none).

#### Example 1

Our application needs two 50GB disks, so we will create two 50GB LUNs:

We are using *volume guarantees* in which the aggregate will guarantee there will be enough space for the volume.

We have determined that the change rate of the data is never more than 30%. This allows us to lower the fractional\_reserve setting from 100% to 30%, although 50% may be better just in case. Conservative estimates are better to provide a buffer for extra growth. They also want to keep the last three Snapshot copies. Now our volume size calculations become:

Total size of LUNs = size of LUN0 + size of LUN1

Fractional\_reserve percent = 50%

Space reservation = total size of LUNs \* fractional reserve percent

Change rate = % of the total size of LUNs on that volume that changes between creation of Snapshot copies (the new snap delta command assists in this determination)

Snapshot space = number of Snapshot copies \* (change rate \* total size of LUNs)

Traditional volume size = (total size of LUNs + space reservation + Snapshot space) \* 1.10

Flexible volume size = total size of LUNs + space reservation + Snapshot space

Adding the numbers to the above formula provides the following:

Total size of LUNs =	50GB + 50GB =	100GB
Fractional_reserve = 50%		
Space reservation =	100GB * 50% =	50GB
Change rate = 30% of the total of the LUNs change between creation of Snapshot copies		
Snapshot space =	3 * (30% * 100GB) =	90GB
Traditional volume size =	(100GB + 50GB + 90GB) * 1.10 =	264GB
Flexible volume size =	(100GB + 50GB + 90GB) =	240GB

Using a fractional\_reserve setting of less than 100% allows use of more space on the volume, but care should be taken to ensure a comfortable setting based on the true change rate of data. **Frequent monitoring of disk space (using DataFabric® Manager) is strongly recommended.**

*Example 2*

Our application needs two 50GB disks, so we will need to create two 50GB LUNs.

We are using *file guarantees* in which the aggregate will only guarantee there will be enough space for space-reserved LUNs.

We have determined that the change rate of their data is never over 30%. However, the file-based guarantee forces fractional\_reserve to 100%. This means the space required for the first Snapshot copy will be equal to the size of the LUN. They also want to keep the last three Snapshot copies. Now our volume size calculations become:

Total size of LUNs =	size of LUN0 + size of LUN1	
Space reservation =	total size of LUNs * fractional reserve percent	
Change rate =	% of the total size of LUNs on that volume that changes between creation of Snapshot copies (the new snap delta command assists in this determination)	
Snapshot space =	(number of Snapshot copies) * (change rate * total size of LUNs)	
Traditional volume size =	(total size of LUNs + space reservation + Snapshot space) * 1.10	
Flexible volume size =	(total size of LUNs + space reservation + Snapshot space)	

Adding the numbers to the above formula provides the following:

Total size of LUNs =	50GB + 50GB =	100GB
Space reservation =	100GB * 100% =	100GB
Fractional_reserve = 100%		
Change rate = 30% of the total of the luns change between creation of Snapshot copies		
Snapshot space =	3 * (30% * 100GB) =	90GB
Traditional volume size =	(100GB + 100GB + 90GB) * 1.10 =	319GB
Flexible volume size =	(100GB + 100GB + 90GB) =	290GB

If you plan to use LUN clones, each of these clones and their Snapshot copies must also be factored into this sizing exercise.

## Best Practices

In order to ensure a timely cluster failover time, keep the total number of volumes on a filer to 50 or less.

Use FlexVol volumes to take advantage of the flexibility and scalability they provide.

Don't create LUNs on /vol/vol0. This is the root volume, and thus it is not used for storing application data.

Make sure snap reserve is set to zero and no automatic Snapshot copies are scheduled on volumes containing LUNs.

Do not put any CIFS or NFS files in the same volume as one or more LUNs. They typically have different Snapshot copy requirements that can be automated on the NetApp storage system.

Verify the desired space reservation guarantee (volume, file, none)

Prior to Data ONTAP 7G, a volume was tied to one or more RAID groups, sometimes making multiple LUNs on a volume necessary. With aggregates and flexible volumes, more volumes may be created, allowing separation of LUNs.

Use FlexVol volumes to organize LUNs according to host, application, or function.

Leave the fractional reserve set at 100% until you truly understand the change rate of the data; then lower it accordingly with frequent monitoring.

Place LUNs that have varying change rates of data on their own FlexVol volumes. Since NetApp storage systems create Snapshot copies on the volume level (traditional or FlexVol), it is important to group LUNs with similar change rates of data from the same host or application within the same volume. In this fashion, the space that is required can be understood in detail and planned for, versus multiple LUNs with different change rates of data being stored on the same volume, creating confusion.

## 5) LUN Management

### Snapshot Copies

In Data ONTAP 7G, LUN Snapshot copies are handled in the same way as before. Space has to be allocated in the capacity planning stage in order to accommodate the number of Snapshot copies to be kept and the value of the fractional reservation option. NetApp storage systems also use Snapshot copies as the basis for mirroring or replication as well as data protection.

### LUN Reallocation Scans

Reallocation scans ensure that blocks in a LUN or a volume are laid out sequentially. If they are not, read and write commands can take longer to complete due to additional disk seeks. Sequential block layout improves the read/write performance of applications that access data on the filer.



A reallocation scan evaluates how the blocks are laid out in a LUN or a volume. Data ONTAP performs the scan as a background task, so applications can write blocks in the LUN or volume without interruption while the scan is running. Layout rechecks are performed during a scan to ensure that the sequential block layout is maintained. A reallocation scan does not necessarily rewrite every block in the LUN. Rather, it rewrites whatever is required to optimize the layout of the LUN. Keep in mind that this also could affect the size of Snapshot copies.

Data ONTAP performs a reallocation scan in the following steps:

1. Scans the current block layout of the LUN.
2. Determines the level of optimization of the current layout on a scale of three (moderately optimal) to 10 (not optimal).
3. Depending on the optimization level set, the scan will either stop if the layout is optimal or reallocate blocks if the layout is not optimal.
4. Scans the new block layout.
5. Steps 2 and 3 will be repeated until the layout is optimal.

You can perform reallocation scans on LUNs when they are online. You do not have to take them offline.

Reallocation scans can be set up by enabling the reallocation scans, then either defining a reallocation scan to run at specified times (such as every 24 hours) or defining a reallocation scan to run on a specified schedule that you create (such as every Thursday at 3:00 p.m.).

A LUN may have only one reallocation scan defined.

You can also start scans at any time, force Data ONTAP to reallocate blocks sequentially regardless of the layout optimization level, and monitor and control the progress of scans.

The reallocation scan process requires additional temporary storage space when reallocating blocks. The longer the period between scans, the less optimum the LUN will be, assuming a very active LUN. This has two implications: more temporary storage will be needed, and the scan process will take longer. Consider this when setting an interval or schedule.

When performing both a Snapshot copy and reallocation scan at around the same time, it is much better to do the reallocation scan before the Snapshot copy scan. By performing the steps in this order, the Snapshot copy scan will be pointing to an optimum block layout. For example, if a Snapshot copy of a LUN is created at 2:00 a.m. each day, then start a reallocation scan at 12:00 a.m., depending on the daily dynamics of the LUN.

If a LUN is deleted, the reallocation scan defined for it remains. If you take the LUN offline, delete it, and then reconstruct it, the reallocation scan remains in place. However, if you delete a LUN with a scan and you do not restore the LUN, the filer console displays an error message the next time the scan is scheduled to run.

Reallocation scans are *disabled* by default. You must *enable* reallocation scans first if you want to run a scan or schedule regular scans.

To enable reallocation scans, complete the following step:

```
Filer>reallocate on
```

To define a reallocation scan for a LUN, complete the following step:

On the filer's command line, enter the following command:

```
reallocate start [-t threshold] [-n] [-i interval] LUN_path
```

*-t threshold* is a number between three (layout is moderately optimal) and 10 (layout is not optimal). The default is four.

The LUN's block layout is checked before any blocks are reallocated. If the layout is below the threshold, then no blocks are reallocated. If the current layout is equal to or above the threshold, the scan reallocates blocks in the LUN.

*-n* reallocates blocks in the LUN without checking its layout.

*-i interval* is the interval, in terms of hours, minutes, or days, at which the scan is performed. The default interval is 24 hours. Specify the interval as follows:

```
n[m|h|d]
```

For example, 30m is a 30-minute interval.

The countdown to the next scan begins only after the first scan is complete. For example, if the interval is 24 hours, and a scan starts at midnight and lasts for an hour, the next scan begins at 1:00 a.m. the next day—24 hours after the first scan completed.

### Examples:

The following example creates a new LUN and a simple reallocation scan that runs every 24 hours.

```
LUN create -s 100g /vol/vol2/LUN0
```

```
reallocate start /vol/vol2/LUN0
```

You can run reallocation scans according to a schedule. The schedule you create replaces any interval you specified when you entered the `reallocate start` command.

To create a reallocation scan schedule, complete the following step:

Enter the following command:

```
reallocate schedule [-s schedule] LUN-path
```

*-s schedule* is a string with the following fields:

```
"minute hour day_of_month day_of_week"
```

minute is a value from 0 to 59

hour is a value from 0 (midnight) to 23 (11:00 p.m.)

day\_of\_month is a value from 1 to 31

day\_of\_week is a value from 0 (Sunday) to 6 (Saturday)

A wildcard character (\*) indicates every value for that field. For example, a \* in the day\_of\_month field means every day of the month.

You can define a comma-separated list of values for a field. For example, entering "0,1" in the day\_of\_week field means Sundays and Mondays. You can also define a range of values. For example, "0-3" in the day\_of\_week field means Sunday through Wednesday.

-d deletes an existing schedule. If you delete a schedule, then the reallocation scan uses the interval you defined using the reallocate start command.

**Example:**

The following example schedules a reallocation scan for every Saturday at 11:00 p.m.

```
reallocate schedule -s "0 23 * 6" /vol/myvol/LUN1
```

You can delete an existing reallocation scan schedule that is defined for a LUN. If you delete a schedule, the scan runs according to the interval that you specified when you initially defined the scan using the reallocate start command.

To delete a reallocation scan schedule, complete the following step:

Enter the following command:

```
reallocate schedule -d LUN-path
```

**Example:**

```
reallocate schedule -d /vol/myvol/LUN1
```

**You perform the following tasks to manage reallocation scans:**

**Start a one-time reallocation scan**

Enter the following command:

```
reallocate start -o -n
```

-o performs the scan only once.

-n performs the scan without checking the LUN's layout.

**Start a scan that reallocates every block in a LUN or volume, regardless of layout.**

Enter the following command:

```
reallocate start -f LUN-path | volume-path
```

Display the status of a reallocation scan

Enter the following command:

```
reallocate status [-v] LUN-path
```

-v provides verbose output

*LUN-path* is the path to the LUN for which you want to see reallocation scan status. If you do not specify a LUN-path, then the status for all scans is displayed.

**Result:** The `reallocate status` command displays the following information:

State—whether the scan is in progress or idle.

Schedule—schedule information about the scan. If there is no schedule, then the `reallocate status` command displays n/a.

Interval—interval at which the scan runs, if no schedule is defined.

Optimization—information about the LUN layout.

### ***Stop a reallocation scan***

You use the `reallocate stop` command to permanently delete a scan you defined for a LUN. The `reallocate stop` command also stops any scan that is in progress on the LUN.

To delete a scan, complete the following step:

```
reallocate stop LUN-path
```

**Result:** The `reallocate stop` command stops any scan in progress on the LUN, including a scheduled scan that is not running or a scan that is quiesced, and deletes that scan.

### ***Quiesce a reallocation scan***

You can quiesce a reallocation scan that is in progress and restart it later. When you quiesce a scan and restart it, the scan restarts from the beginning of the reallocation process. For example, you quiesce a scan if you want to back up a LUN, but the scan is already in progress.

To quiesce a reallocation scan, complete the following step:

### ***Restart a reallocation scan***

You might restart a scan for the following reasons:

You quiesced the scan by using the `reallocate quiesce` command, and you want to restart it.

You have a scheduled scan that is idle (it is not yet time for it to run again), and you want to run it immediately.

Enter the following command:

```
reallocate restart LUN-path
```

**Result:** The command restarts a quiesced scan. If there is a scheduled scan that is idle, the `reallocate restart` command runs the scan.

### ***Disable reallocation***

When you disable reallocation scans, you cannot start or restart any new scans. Any scans that are in progress are stopped. To disable reallocation scans, complete the following step:

On the filer's command line, enter the following command:

```
reallocate off
```

### **Best Practices**

NetApp recommends the following best practices for using reallocation scans:

Define a reallocation scan when you first create the LUN. This ensures that the LUN layout remains optimized because of regular reallocation scans.

Define regular reallocation scans by using either intervals or schedules. This ensures that the LUN layout remains optimized. Do not wait until most of the blocks in the LUN layout are not sequential. A scan takes longer for LUNs in which the layout ratio is not optimized.

Define intervals according to the type of read/write activity associated with the LUN:

Long intervals—Define long reallocation scan intervals for LUNs in which the data changes slowly: for example, LUNs in which data changes as a result of infrequent large writes.

Short intervals—Define short reallocation scan intervals for LUNs that are characterized by workloads with many small random writes and many sequential reads. These types of LUNs might become heavily fragmented over a shorter period.

If a LUN has an access pattern of random writes followed by periodic large sequential reads (for example, it is accessed by a database or a mail backup).

When performing both a Snapshot copy and reallocation scan at around the same time, it is much better to do the reallocation scan before the Snapshot copy scan.

Schedule reallocation scans during nonpeak hours in order to avoid any impact on production applications.

Use of the commands described above, available through the Manage ONTAP™ API, allows scripts to be written to automate these tasks when many LUNs are involved.

## Snap Delta

To effectively plan volume sizing, space guarantees, and Snapshot copy creation, space must be taken into consideration. Snapshot copy space is dependent on the number of Snapshot copies to be kept, the frequency of Snapshot copy creation, and the amount of data that changes between creation of the Snapshot copies. Understanding the actual change rate of the data between creation of Snapshot copies can be a challenge. Data ONTAP 7G introduces the `snap delta` command to assist. The use of this command results in:

- More accurate information for estimating volume sizing

- More effective utilization of volume space

- Improved use of current storage resources

The `snap delta` command will display the amount of data that has changed between creation of Snapshot copies. Knowing this information will allow more efficient space planning for Snapshot copies and allow proper estimation of the `fractional_reserve` value for LUN space guarantees. The use of the command is described below.

```
snap delta [-A] vol_name snapshot
```

`-A` displays the rate of change of data between creation of Snapshot copies for all aggregates in the system.

`vol_name` is the name of the volume.

`snapshot` is the name of the Snapshot copy.

If you do not specify an argument, the `snap delta` command displays the rate of change of data between creation of Snapshot copies for all volumes in the system.

Example: The following example displays the rate of change of data between creation of all Snapshot copies in `vol0`.

```
filer_1> snap delta vol0
```

```
Volume vol0
working...
From Snapshot To KB changed Time Rate (KB/hour)
hourly.0 Active File System 1460 0d 02:16 639.961
nightly.0 hourly.0 1492 0d 07:59 186.506
hourly.1 nightly.0 368 0d 04:00 91.993
hourly.2 hourly.1 1420 0d 04:00 355.000
hourly.3 hourly.2 1960 0d 03:59 490.034
hourly.4 hourly.3 516 0d 04:00 129.000
nightly.1 hourly.4 1456 0d 08:00 182.000
hourly.5 nightly.1 364 0d 04:00 91.000
Summary...
From Snapshot To KB changed Time Rate (KB/hour)
hourly.5 Active File System 9036 1d 14:16 236.043
```

**Interpreting snap\_delta output.** The `snap delta` command displays the above information in two tables:

The first row of the first table displays the rate of change between the most recent Snapshot copy and the active file system. The following rows provide the rate of change between successive Snapshot copies. Each row displays the names of the two.

Use of the `snap delta` command facilitates a more accurate measurement of the change rate of data between creation of Snapshot copies. This assists in more efficient space planning.

### Snap Reclaimable

To estimate the amount of space freed by deleting a set of Snapshot copies, complete the following step.

Enter the following command:

```
snap reclaimable vol_name snapshot snapshot
```

*vol\_name* is the name of the volume.

*snapshot* is the name of the Snapshot copy.

Example: The following example shows the approximate amount of space that would be freed by deleting two Snapshot copies.

```
filer_1> snap reclaimable vol0 hourly.1 hourly.5

Processing (Press Ctrl-C to exit) ...

snap reclaimable: Approximately 1860 Kbytes would be freed
```

## 6) Cloning and Snapshot Copies

### LUN Cloning

LUN cloning has been modified in Data ONTAP 7G to more closely emulate the way volume cloning works. In releases prior to ONTAP 7G, a cloned LUN was a completely independent LUN built from the original "backing store LUN." This "backing store LUN," also called by some a "writable Snapshot copy," was based upon the original LUN and a specified Snapshot copy. Once the cloning was complete, the cloned LUN did not need the original or the Snapshot copy it was based upon.

In Data ONTAP 7G, this is no longer the case. When a LUN is cloned, it essentially becomes what we used to refer to as a "backing store LUN." Initially, the clone and its parent share the same storage; more storage space is consumed only as the cloned LUN changes (see Figure 3).

```
lun clone create clone_LUNpath [-o noreserve] -b parent_LUNpath parent_snap
```

The `parent_snap` is locked in the parent volume, preventing its deletion until the clone is either destroyed or split from the parent using the `lun clone split start` command. The `lun clone create` command fails if the chosen `parent_LUNpath` is currently involved in a LUN clone split operation.

**lun clone show [LUN\_path]**

If no LUN\_path is given, information about all LUNs being cloned is shown. The above command displays the cloning status of the LUN in terms of percentage complete.

Data ONTAP 7G	Before Data ONTAP 7G
<b>lun clone split start LUN_path</b>	<b>lun clone start LUN_path</b>

This command begins separating clone *LUN\_path* from its underlying parent. New storage is allocated for the clone LUN that is distinct from the parent. This process may take some time and proceeds in the background. Use the **lun clone split status** command to view the command's progress. Both clone and parent LUNs remain available during this process of splitting them apart. Upon completion, the Snapshot copy on which the clone was based will be unlocked in the parent volume.

Data ONTAP 7G	Before Data ONTAP 7G
<b>lun clone split status LUN_path</b>	<b>lun clone status LUN_path</b>

This command displays the progress in separating clone LUNs from their underlying parent LUNs. This gives details about the number of blocks processed until now.

Data ONTAP 7G	Before Data ONTAP 7G
<b>lun clone split stop LUN_path</b>	<b>lun clone stop LUN_path</b>

This command stops the process of separating a clone from its parent LUN. All of the blocks that were formerly shared between *LUN\_path* and its backing LUN that have already been split apart by the **LUN clone split start** will remain split apart.

### Volume Cloning

With Data ONTAP 7G, cloning of a flexible volume, a FlexVol volume, is now possible. A detailed explanation of the FlexClone process is provided in the paper *A Thorough Introduction to FlexClone Volumes*. FlexVol volumes and FlexClone volumes have an impact on the management of LUNs.

Figure 3 summarizes the relationship of LUNs, cloned LUNs, space reservation, and cloned volumes. In Figure 3-A, there is a 100GB LUN with no Snapshot copies; hence no space has actually been reserved. If the volume upon which the LUN is created is cloned, then a LUN of the same size is created but placed offline until mapping is resolved. This LUN will rely on the parent LUN (in the original volume) for read operations but will use space in the cloned volume for write operations.

In Figure 3-B, there is a 100GB LUN (LUN 0) with a Snapshot copy; hence additional space equal to the size of the LUN has been reserved (fractional reservation left at default of 100%). Subsequently, this LUN is cloned (LUN 1), relying on the parent LUN and Snapshot copy for read operations and new space for write operations.

If the volume upon which the LUN and its clone are created is itself cloned, then LUNs of the same size are created on the cloned volume but placed offline until mapping is resolved.



These LUNs will rely on the parent LUNs (in the original volume) for read operations but will use space in the cloned volume for write operations.

Figure 3-C demonstrates the case where the cloned LUN from Figure 3-B is split, making it a fully independent LUN in terms of read and write operations. In this case, when the volume is cloned, it will operate the same as in Figure 3-A, where a new LUN of the same size is created on the cloned volume and placed offline until mapping is resolved. It will then rely on the parent LUN for read operations. Writes will go to the cloned volume.

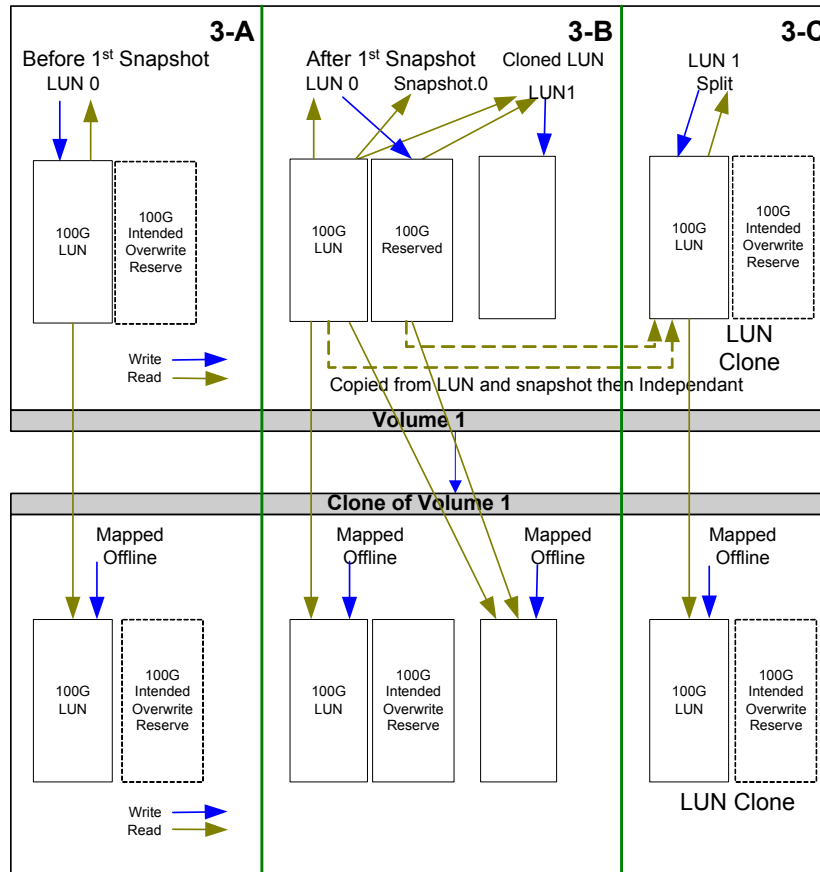


Figure 3)

**Clone a Volume or Clone a LUN?**

So, if duplicates of a LUN are required, does one clone the LUN or clone the volume? Is there a case where one does both?

LUN clones, though they share blocks, have an extra level of indirection on reads (assuming they have not been split). This is not the case with LUNs on FlexClone volumes. There is no performance penalty due to redirection on reads.

Snapshot copies created in the volume after a LUN clone has been created will also "lock" the base Snapshot copy from which the LUN clone was created. Even after the clone is split, a user will be unable to delete the base Snapshot copy unless all Snapshot copies created since the clone was created but before it was split have been deleted. FlexClone volumes, in

contrast, do not have this problem, since Snapshot copies are created in the clone volume and are not locked with the parent volume after clone creation.

Unless the admin provisions the storage such that all applications store their data in individual FlexVol volumes, perhaps FlexClone volumes won't help, because the clone might contain unrelated data, possibly making granularity a factor. However, if the goal is to clone all the data (perhaps in separate LUNs) in one step, and all LUNs are stored in one flexible volume, then volume cloning makes sense.

FlexClone volumes allow creating writable volumes even from read-only volume SnapMirror destinations. Therefore, it is possible to use LUNs in flexible volumes on secondary or disaster recovery sites for purposes such as backup or verification. This isn't possible with LUN clones unless the clone is created within the same volume. It is possible with QSM and LUN clones, but not with volume SnapMirror.

There are limits on the number of FlexVol volumes a system and a cluster can have in Data ONTAP 7G, and there are further limits on how many FlexVol volumes are recommended for takeover/giveback times (which are more important for Fibre Channel). This may be a factor when considering how to provision LUNs in volumes.

How does this answer our question? In most cases, it would be better to create a FlexClone volume in order to duplicate a LUN. The case where it may not make sense is when a volume contains data unrelated to the LUN or there are not enough volumes left to create a volume for each LUN.

## 7) Summary

The new FlexVol and FlexClone capabilities provided in Data ONTAP 7G bring new flexibility and scalability to the creation and management of block-based LUNs. Multiple LUNs, each with separate Snapshot copy schedules, can share in the performance provided by a pool of disks.

As always, proper volume sizing is vital. The choice of traditional or flexible volumes and the new space guarantee capabilities provide increased flexibility but necessitate proper planning in order to maximize efficiency.

With the ability to create flexible volumes, create FlexClone volumes, and split clones, the structure of LUN commands has been modified to match those that now exist at a volume level. A "writable Snapshot copy" or "backing store LUN" from an earlier version of Data ONTAP is now referred to as a LUN clone in Data ONTAP 7G. A LUN clone from an earlier version of Data ONTAP is now referred to as a split LUN clone.

Data ONTAP 7G also provides new utilities to aid in the optimization of LUNs and allows better planning for space guarantees. Reallocation scans ensure that blocks in a LUN are laid out sequentially for performance. The snap delta command provides LUN change rate information in order to ensure optimum space guarantees.

## 8) Additional Resources

*Data ONTAP 7G—Block Access Management Guide for FCP*

*Data ONTAP 7G Storage Management Guide*

*A Thorough Introduction to FlexClone Volumes* (by Miroslav Klivansky, NetApp)



**Network Appliance, Inc.**  
495 East Java Drive  
Sunnyvale, CA 94089  
[www.netapp.com](http://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.