



## **Thin Provisioning in a NetApp SAN or IP SAN Enterprise Environment**

**Richard Jooss, Network Appliance, Inc.**

**June 2006, TR-3483**

### **Abstract**

Traditional disk arrays force the IT storage specialist to allocate and dedicate storage space to a particular volume or LUN at creation time. The problem is that at creation time the true requirements are often not known, which forces overprovisioning and higher costs. Network Appliance™ storage systems offer powerful, flexible provisioning capabilities that give IT storage specialists the flexibility to lower cost while meeting capacity and performance requirements. This document describes the configurations and recommendations for efficiently provisioning NetApp SANiSAN storage.

## Table of Contents

1. Introduction .....	3
2. Intended Audience and Assumptions .....	3
3. Thin Provisioning.....	3
3.1 Definition.....	3
3.2 WAFL—The Enabling Technology.....	3
4. LUN Space Usage .....	4
4.1 Host versus Storage View.....	4
4.2 LUN Capacity Used over Time.....	6
5. Data ONTAP Configuration Parameters.....	7
5.1 LUN Reservation .....	7
5.2 Guarantees.....	7
5.3 Fractional_reserve.....	9
5.4 Snap Reserve.....	10
5.5 Volume Filling Example.....	10
5.6 Autosize.....	14
5.7 Autodelete .....	15
6. Thin Provisioning of LUNs .....	16
6.1 The Two Different Types of Thin Provisioning.....	16
6.2 Monitoring of Available Space.....	16
6.3 Thin Provisioning of LUNs.....	16
6.4 Thin Provisioning through LUN Growing.....	17
7. Thin Provisioning of Snapshot Space.....	18
7.1 Thin Provisioning of Snapshot Space .....	18
7.2 Thin Provisioning of Snapshot Space Configurations.....	18
8. Summary.....	19

## 1. Introduction

This document describes how to efficiently provision storage with NetApp storage systems in a Fibre Channel or iSCSI deployment. Storage efficiency has historically been extremely low because of the difficulty provisioning or changing provisioning to adapt to changing needs with classic storage array architectures. NetApp storage systems, particularly with the release of FlexVol™ technology with Data ONTAP® 7.0, provide an extremely flexible paradigm for provisioning, allowing the user to minimize unused space. This paper does not cover thin provisioning through the use of LUN and FlexClone™ as these methods are adequately covered in other papers.

## 2. Intended Audience and Assumptions

This paper is intended for system and storage architects designing NetApp storage appliances into IT environments. For the concepts, methods, and procedures in this document to be useful to the reader, several assumptions are made:

- The reader has at a minimum general knowledge of Network Appliance hardware and software solutions, particularly in the areas of Fibre Channel and/or iSCSI
- This paper is based on the functionality of Data ONTAP release 7.1

## 3. Thin Provisioning

### 3.1 Definition

The definition of thin provisioning is presenting more storage space to the hosts or servers connecting to the storage system than is actually available on the storage system. NetApp storage systems have always had this sophisticated capability for FC and iSCSI provisioning and FlexVol technology released with Data ONTAP 7.0 further increased this flexibility. An example of thin provisioning is when a storage system contains 5,000GB of usable storage capacity but the storage administrator has mapped LUNs of 500GB each to 15 hosts. In this example the storage administrator makes 7,500GB of storage space visible to the hosts even though the storage system has only 5,000GB of usable space. Obviously, if all 15 hosts immediately use all 500GB provisioned to them there will be a problem. The storage administrator will have to monitor the system and add storage as needed.

In other areas or industries this practice is prevalent. For example, a water utility company's infrastructure does not have the capacity to maintain acceptable water pressure if all of its customers turn on their taps simultaneously. The company plans on having only a certain percentage of its customers simultaneously using water. In the banking industry, clearly banks do not have enough cash on hand to allow all of their customers to withdraw all the money in their accounts at one time. They plan that only a small percentage of their customers will be withdrawing money at one particular time. Both of these examples illustrate that thin provisioning is more applicable in larger environments, including the storage world. The larger the storage system and the more users or applications utilizing the storage typically the better the chance to take advantage of thin provisioning.

### 3.2 WAFL—The Enabling Technology

WAFL, which can be thought of as the virtualization layer of Data ONTAP, enables thin provisioning. When a LUN is created it does not dedicate specific blocks out of the NetApp volume for the LUN or for Snapshot™ copies of the LUN. Instead, it allocates the blocks from the NetApp volume when the data is actually written. This allows the administrator to provision more storage space, as seen from the connected servers, than is actually physically present in the storage system.

## 4. LUN Space Usage

### 4.1 Host versus Storage View

There is often confusion over the space usage inside a LUN. This confusion stems from the fact that the vast majority of LUNs have a file system installed on them and how full the file system is is not the same as how full the LUN is. Assuming the file system is configured to have access to the complete LUN, the LUN will always be at least as full as the file system. From a storage system perspective, the percentage of a LUN used will always increase and never decrease.

The graphic below helps show why an individual LUN's usage never decreases. The graphic shows a theoretical case of how a file system and a LUN are used as files are written and deleted from the file system. Steps 1 and 2 show, as expected, that the file system and the LUN both report the same percentage used as files are written to the file system. Step 3 is where a difference occurs between what the host or file system reports and what the storage reports. In this step, files 1 and 2 are deleted, meaning the blocks used by these files are available to the file system. The only file still visible in the file system is file 3, so the file system reports it is 25% full. However, the storage is still reporting it is 75% full. The reason that the storage shows 75% full is because the storage has no way of knowing that the data written for files 1 and 2 is no longer needed. The space in the graphic is shaded because from the storage perspective those blocks still contain valid data. When a file is deleted the file system does not send any type of erase or delete command to the storage array. It simply changes a pointer in its tables that pointed to the file and makes those blocks available for future use. The fact that the blocks containing the files are not actually erased is what allows "undelete" utilities to recover files after the user has deleted them. In Step 4, file 4 is shown being written to the last 25% of the LUN. This step depicts how a file system doesn't necessarily attempt to keep data at the beginning of a LUN. This is also shown in Figure 2, which shows the layout of an NTFS file system that is 75% full. After Step 4, every block in the LUN has been written and is therefore full from a storage perspective. As mentioned, the fill percentage of a LUN will never go down and Steps 5 through 7 show that, regardless of the host activity on the LUN, it will remain 100% full.

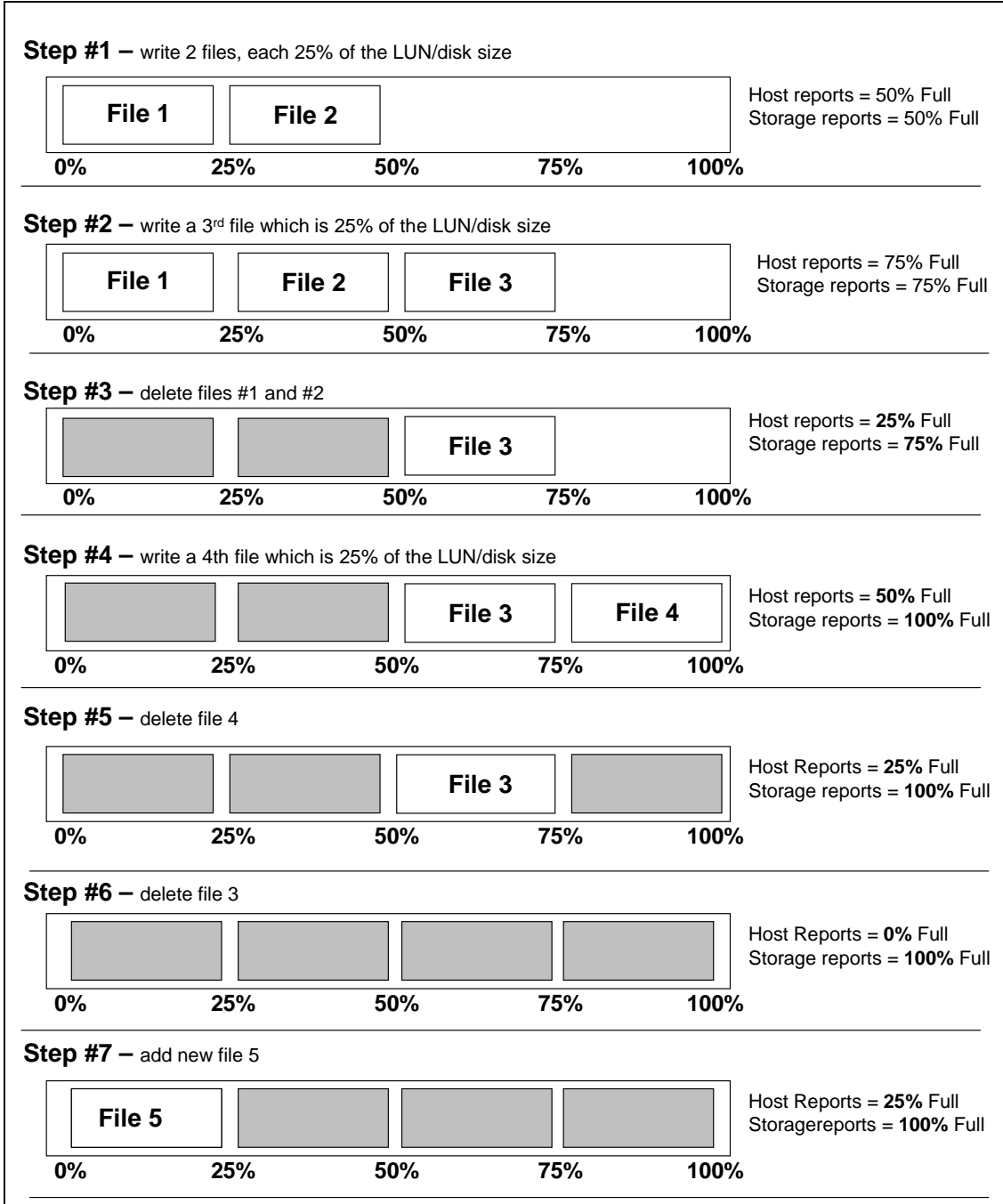


Figure 1) LUN capacity—host versus storage view.

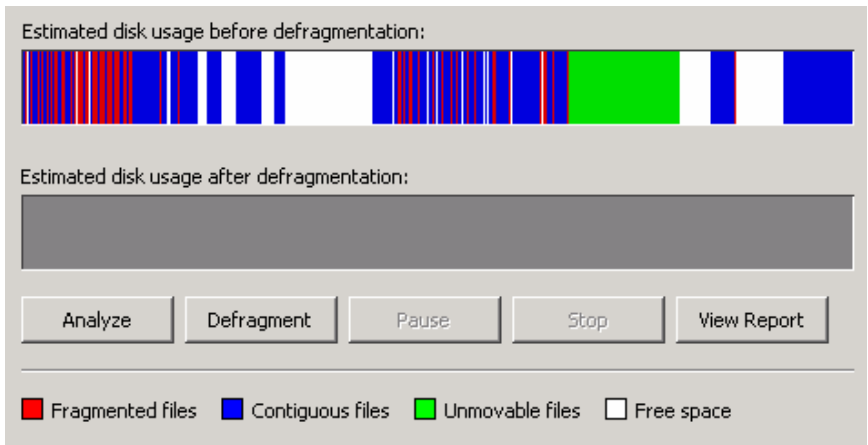


Figure 2) LUN usage.

#### 4.2 LUN Capacity Used over Time

The figure below shows how much of a LUN's capacity is used over time from a storage perspective. The steepness of the curve and how fast the LUN will get to 100% is affected by many parameters such as the type of file system or application, the size of the files being written to the file system, and the percentage of the file system itself that is being used. The time it takes for a LUN to reach 100% will typically be measured in days, weeks, or months, not seconds or years.

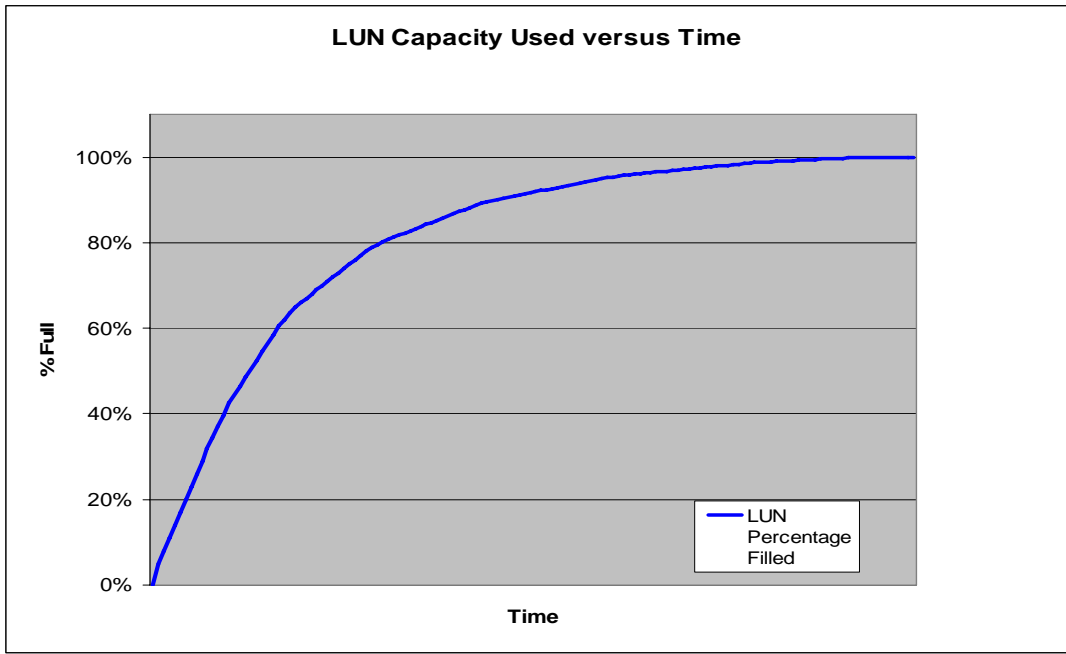


Figure 3) LUN capacity used versus time.

## 5. Data ONTAP Configuration Parameters

There are a number of variables or options in Data ONTAP that are important to understand before configuring for thin provisioning.

### 5.1 LUN Reservation

LUN reservation (not to be confused with SCSI2 or 3 logical unit locking reservations) determines when space for the LUN is reserved or allocated from the volume. With reservations enabled (default) the space is subtracted from the volume total when the LUN is created. For example, if a 20GB LUN is created in a volume having 80GB of free space, the free space will go to 60GB at the time the LUN is created even though no writes have been performed to the LUN. If reservations are disabled, space is first taken out of the volume as writes to the LUN are performed. If the 20GB LUN was created without LUN space reservation enabled, the free space in the volume would remain at 80GB and would only go down as the LUN was written to.

### 5.2 Guarantees

With flexible volumes, introduced with Data ONTAP 7.0, there is the new concept of space guarantees, which allow the user to determine when space is reserved or allocated from the containing aggregate.

- Volume—A guarantee of “volume” ensures that the amount of space required by the FlexVol volume is always available from its aggregate. This is the default setting for FlexVol volumes. With the space guarantee set to “volume” the space is subtracted, or reserved, from the volume’s available space regardless of whether it is actually used for data storage or not.

The example shown here shows the creation of a 20GB volume. The `df` commands showing the space usage aggregate before and after the volume create command show how the 20GB is removed from the aggregate as soon as the volume is created, even though no data has actually been written to the volume.

```
netappl> df -A -g aggr0
```

Aggregate	total	used	avail	capacity
aggr0	85GB	0GB	85GB	0%
aggr0/.snapshot	4GB	0GB	4GB	0%

```
netappl> vol create flex0 aggr0 20g
```

Creation of volume 'flex0' with size 20g on hosting aggregate 'aggr0' has completed.

```
netappl> df -g /vol/flex0
```

Filesystem	total	used	avail	capacity	Mounted on
/vol/flex0/	16GB	0GB	15GB	0%	/vol/flex0/
/vol/flex0/.snapshot	4GB	0GB	4GB	0%	

```
netappl> df -A -g aggr0
```

Aggregate	total	used	avail	capacity
aggr0	85GB	<b>20GB</b>	65GB	23%
aggr0/.snapshot	4GB	0GB	4GB	0%

Since the space has already been reserved from the aggregate, write operations to the volume will not cause more space from the aggregate to be used.

- **None**—A FlexVol volume with a guarantee of “none” reserves no space, regardless of the space reservation settings for LUNs in that volume. Write operations to space-reserved LUNs in that volume might fail if the containing aggregate does not have enough available space. Space is first taken from the aggregate when data is actually written to the volume. The example here shows how, in contrast to the example with the volume guarantee, the volume creation does not affect the used space in the aggregate. Even LUN creation, which by default has space reservation enabled, does not reserve space out of the volume.

```
netappl> df -A -g aggr0
Aggregate          total      used      avail capacity
aggr0              85GB      0GB      85GB      0%
aggr0/.snapshot   4GB       0GB      4GB       0%
netappl>
netappl> vol create noneflex -s none aggr0 20g
Creation of volume 'noneflex' with size 20g on hosting aggregate
'aggr0' has completed.
netappl> df -g /vol/noneflex
Filesystem          total  used  avail capacity  Mounted on
/vol/noneflex/      16GB  0GB   16GB    38% /vol/noneflex/
/vol/noneflex/.snapshot 4GB   0GB   4GB     0%
/vol/noneflex/.snapshot
netappl> df -A -g aggr0
Aggregate          total      used      avail capacity
aggr0              85GB      0GB      85GB      0%
aggr0/.snapshot   4GB       0GB      4GB       0%
netappl> lun create -s 10g -t windows /vol/noneflex/foo
Mon Nov 24 15:17:28 EST [array1:
lun.vdisk.spaceReservationNotHonored:notice]: Space reservations in
noneflex are not being honored, either because the volume space guarantee
is set to 'none' or the guarantee is currently disabled due to lack of
space in the aggregate.
lun create: created a LUN of size:  10.0g (10742215680)
netappl> df -A -g aggr0
Aggregate          total      used      avail capacity
aggr0              85GB      0GB      85GB      0%
aggr0/.snapshot   4GB       0GB      4GB       0%
```

- **File**—The aggregate guarantees that space is always available for overwrites to space-reserved LUNs. Fractional reserve, a volume level option discussed later in this paper, is set to 100% and is not adjustable with this type of space reservation. The “file” guarantee is basically the same as the “none” guarantee with the exception that space reservations for LUNs and space-reserved files are honored. The example below looks the same as the previous example under the “none” reservation except that the LUN creation takes space from the aggregate because it is a space-reserved object by default. Since the space reservation is honored, the “lun create” command also doesn’t issue the warning shown in the previous example.

```
netappl> df -A -g aggr0
Aggregate          total      used      avail capacity
aggr0              85GB      0GB      85GB      0%
aggr0/.snapshot   4GB       0GB      4GB       0%
netappl>
netappl> vol create noneflex -s none aggr0 20g
```

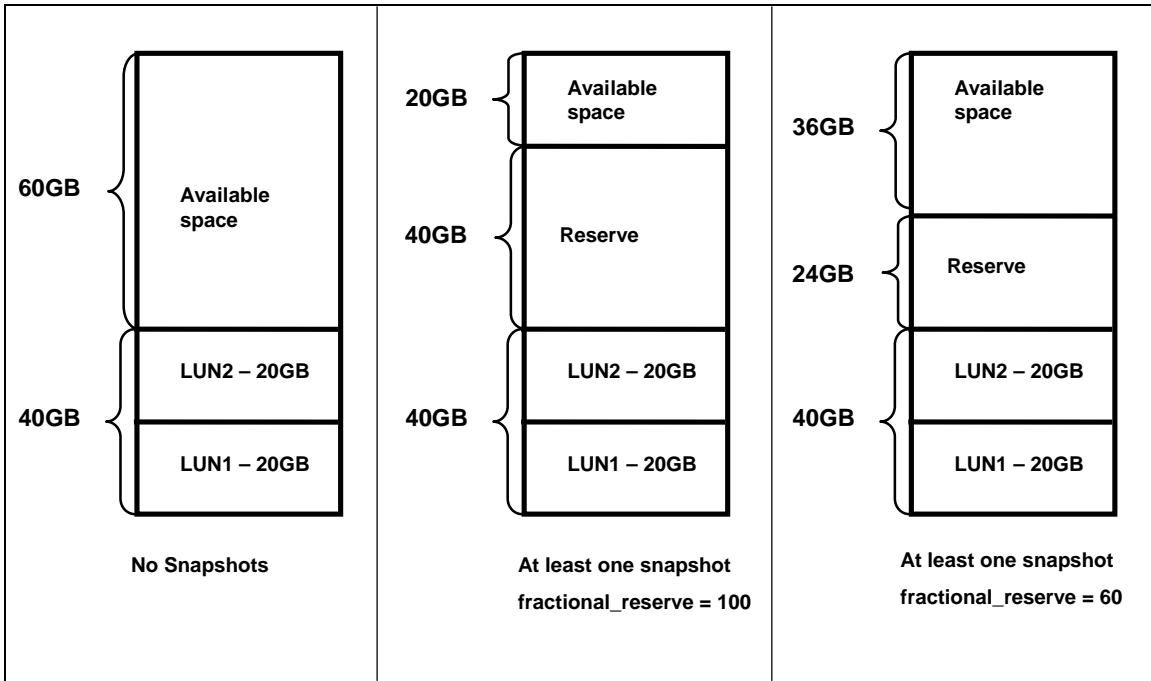
```

Creation of volume 'noneflex' with size 20g on hosting aggregate
'aggr0' has completed.
cnrll>
netappl> df -g /vol/noneflex
Filesystem                total      used      avail capacity  Mounted on
/vol/noneflex/            16GB       0GB       16GB      38%
/vol/noneflex/
/vol/noneflex/.snapshot   4GB        0GB        4GB        0%
/vol/noneflex/.snapshot
netappl>
netappl> df -A -g aggr0
Aggregate                  total      used      avail capacity
aggr0                      85GB       0GB       85GB        0%
aggr0/.snapshot            4GB        0GB        4GB        0%
netappl>
netappl> lun create -s 10g -t windows /vol/noneflex/foo
lun create: created a LUN of size:  10.0g (10742215680)
netappl> df -A -g aggr0
Aggregate                  total      used      avail capacity
aggr0                      85GB      10GB       75GB        0%
aggr0/.snapshot            4GB        0GB        4GB        0%

```

### 5.3 Fractional\_reserve

Fractional reserve is a volume option available in Data ONTAP 6.5 and later that determines how much space Data ONTAP will reserve for Snapshot overwrite data for LUNs and space-reserved files. The default value is 100%. Data ONTAP removes or reserves this space from the volume as soon as the first Snapshot copy is created. For example, as shown in Figure 4 on the left-hand side, a 100GB volume is shown with two 20GB LUNs. Assuming the LUNs are full and a Snapshot copy is created, Data ONTAP by default will reserve 40GB (2 x 20GB) of space in the volume to assure that there is always enough space for both the LUNs and all the Snapshot data, even if the LUNs are completely overwritten. This is depicted in the middle of Figure 4. As depicted on the right-hand side, if fractional\_reserve is set to 60% when the Snapshot is created, instead of reserving 40GB in the volume, Data ONTAP will reserve only 24GB (60% \* [2 x 20GB]).



**Figure 4) Fractional Reserve.**

The amount of space reserved can be seen using the “-r” option of the df command. This reserve area is used only when Data ONTAP reports that the volume is full. Until the volume is full, space for Snapshot overwrites are taken from the volume and only when the volume is 100% full will this reserved space be used.

#### 5.4 Snap Reserve

This variable is set at the volume level and is set as a percentage of the volume. Data ONTAP removes the defined percentage (20% by default) of volume from being available for configuring LUNs or for CIFS or NFS files to use. As Snapshot copies need space, they consume space in the snap reserve area. By default after the snap reserve area is filled, the Snapshot copies start to take space from the general volume. Of course, because of WAFL write anywhere technology, snap reserve doesn’t actually reserve specific physical blocks for Snapshot usage and can be thought of as a logical space accounting mechanism. As a best practice prior to having available the functionality provided in Data ONTAP 7.1, the snap reserve for volumes using Fibre Channel or iSCSI LUNs was set to zero. If the automatic Snapshot management in 7.1 is not used, this is still the recommended setting.

#### 5.5 Volume Filling Example

The following figures walk you through an example of how space is used from volume creation through filling up the volume with Snapshot data. In this example “lun reserve,” “fractional\_reserve,” and “snap reserve” all have been left at their default values.

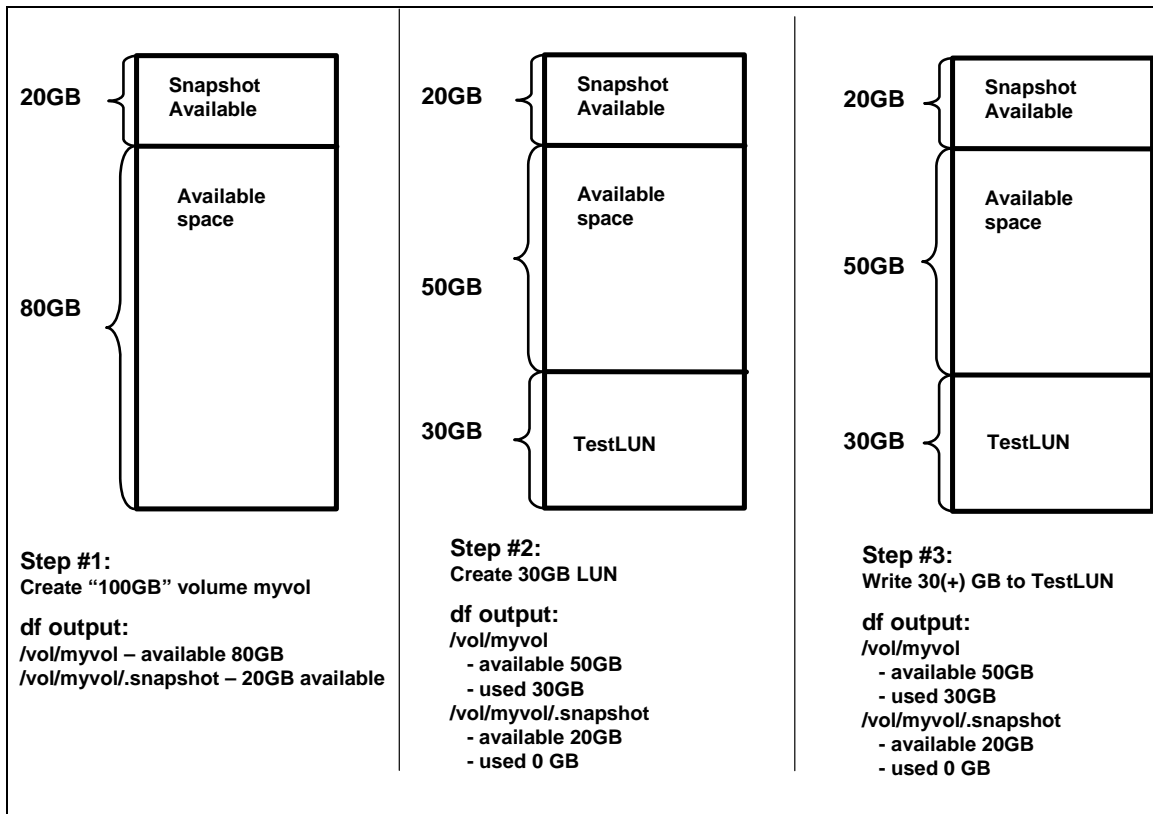


Figure 5) Volume filling—Actions #1–#3.

- Step #1—A 100GB volume is created. The available space is shown as 80GB since by default there is a 20% snap reserve area.
- Step #2—A 30GB LUN called "TestLUN" is created. Because LUNs are by default space-reserved objects, the 30GB is immediately taken from the available space from the volume.
- Step #3—The LUN is filled with data and since the space was already reserved there is no change in reported space usage.

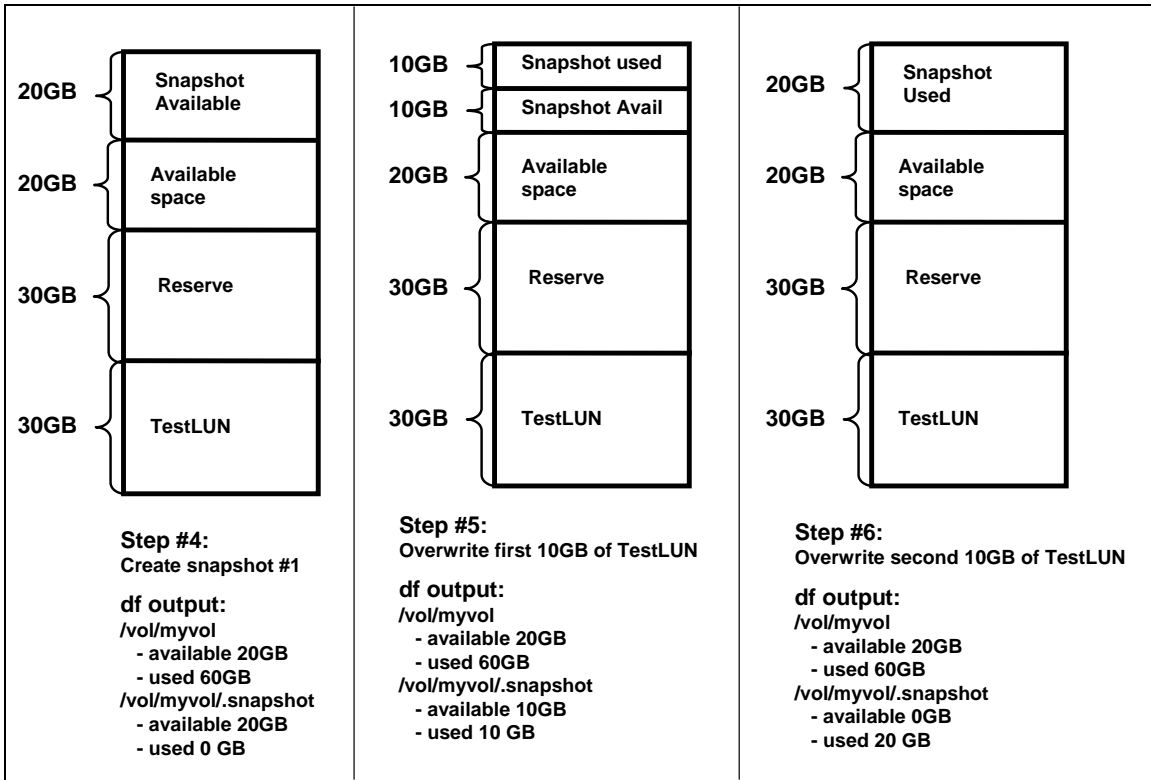


Figure 6) Volume filling—Actions #4–#6.

- Step #4—A Snapshot is created causing Data ONTAP to reserve 30GB of space from the volume, ensuring that there is space for Snapshot data even if the LUN is completely overwritten. As shown in Figure 4, if fractional\_reserve was set to less than 100, the amount of space reserved would be less than 30GB.
- Step #5—10GB of data is overwritten in the TestLUN. Since a Snapshot exists, overwriting data in a LUN means that space will start to be consumed for Snapshot data. The first space used for this is from the Snapshot reserve area. This is seen in the df output in the /vol/myvol/.Snapshot line where 10GB is now shown as used.
- Step #6—Another 10GB of data is overwritten in the TestLUN. This uses up the rest of the Snapshot reserve area, which now shows up as full.

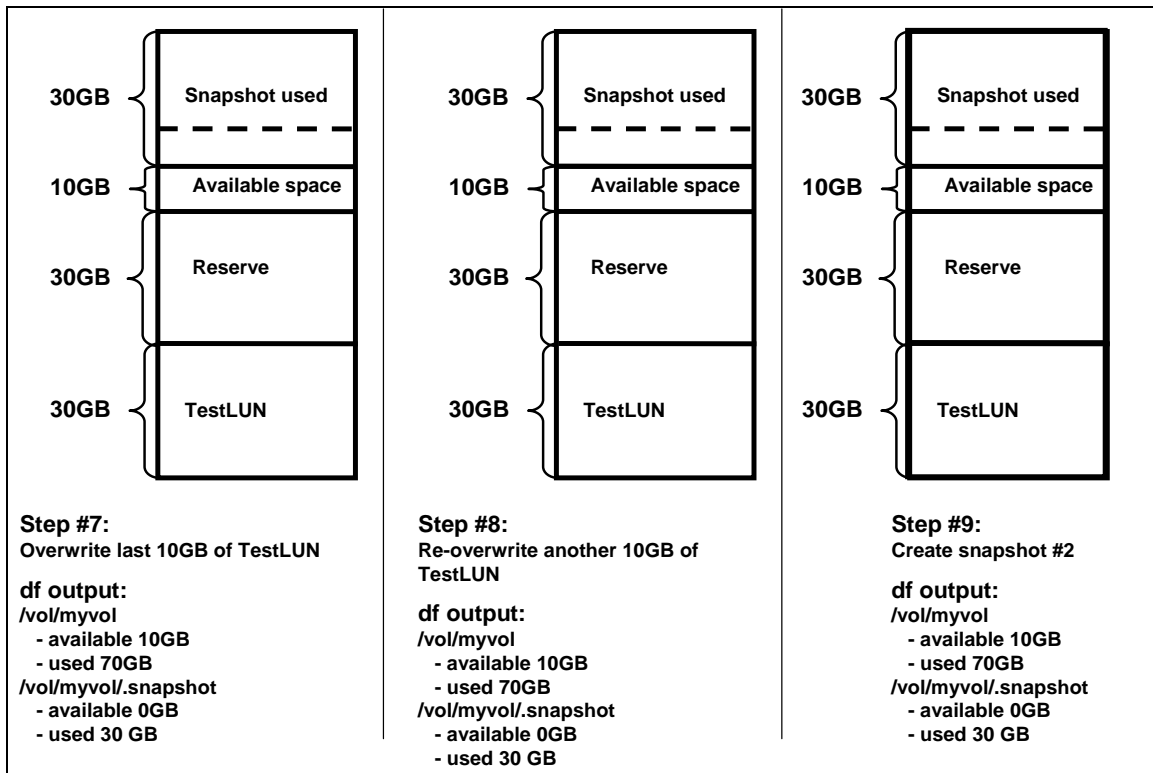


Figure 7) Volume filling—Actions #7–#9.

- Step #7—Another 10GB of data is overwritten in the TestLUN. Even though the Snapshot reserve area is already full, the writes are allowed to happen and no Snapshot copies are lost or corrupted because there is still available space in the volume. These writes simply take space from the available space in the volume, which goes down from 20GB as shown in the last step to 10GB after Step #7. The Snapshot space usage that goes above the amount of snap reserve can be considered counted twice in the df command output since it shows up in the used column of both the general volume and the snap reserve space. It’s interesting to note that the used column df output for the snap reserve area can go above 100%.
- Step #8—Another 10GB of data is written to the TestLUN. In this case, because in the previous steps the entire LUN was already overwritten, no additional Snapshot data requires space. There is no change to the space usage.
- Step #9—A second Snapshot is created. Unlike the creation of the first Snapshot, the creation of more Snapshot copies doesn’t cause any immediate changes to the space usage. The creation of additional Snapshot copies will not directly cause space to be consumed. However, as the count of Snapshot copies increases, the possibility that the Snapshot reserve space will be needed goes up, since multiple overwrites of a block are more likely to be caught in separate Snapshot copies.

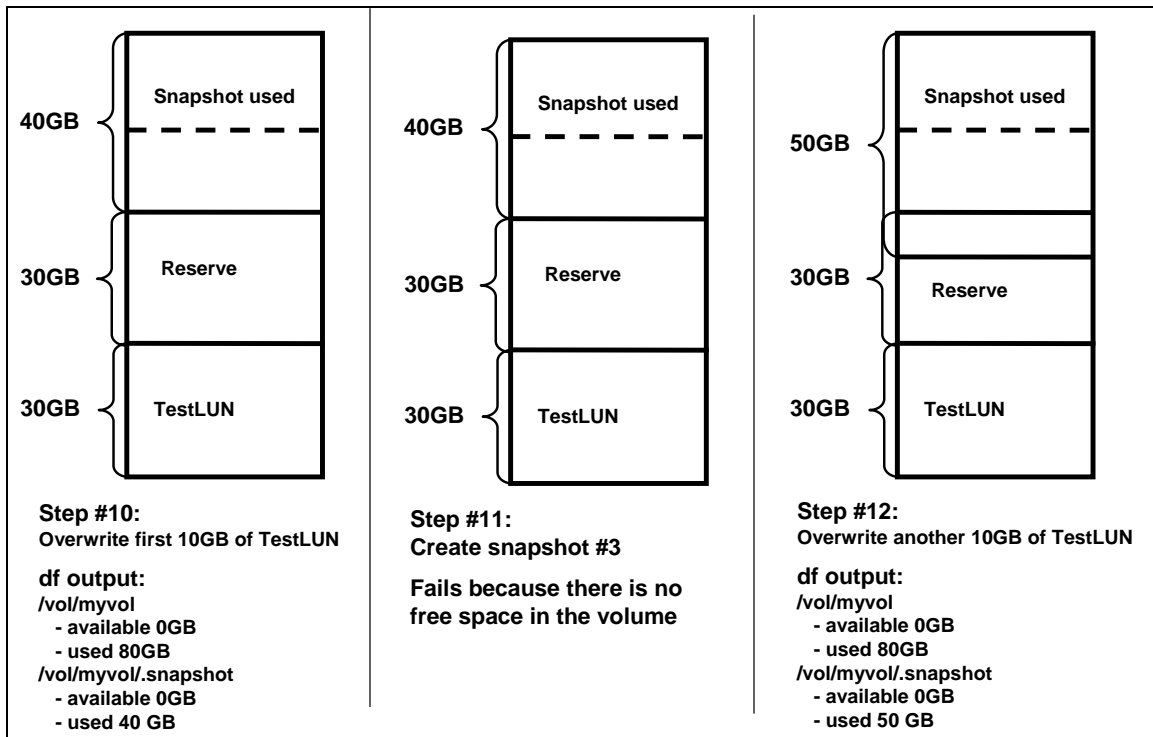


Figure 8) Volume filling—Actions #10–#12.

- Step #10—10GB of the LUN is overwritten. Since a Snapshot copy was just created in the previous step, it doesn't matter which 10GB of the LUN is overwritten, since space is still required to hold the Snapshot data. In this case, the 10GB of available space is taken out of the volume. At this point the volume reports as full and it is not possible to create any more LUNs, create any files via NFS/CIFS, create more Snapshot copies, or write to non-space-reserved files.
- Step #11—An attempt is made to create a third Snapshot. This operation fails because the volume is full.
- Step #12—Another 10GB of data is overwritten in the TestLUN. Even though the volume is full, the writes to the TestLUN are successful and the Snapshot copies are not lost because there is still space available in the reserve area. The only change to the df command output will be in the used column of the snap reserve area. Because the fractional\_reserve is set to 100%, writes to the LUN will never fail for lack of space. If the fractional\_reserve were set to less than 100% and overwrites continued, at some point writes to the LUN might not be possible because of lack of space.

This example shows how the reserve area is used only when there is no other space left in the volume.

## 5.6 Autosize

This is a volume setting available in Data ONTAP 7.1 and later that defines whether a volume should automatically grow to avoid filling up to capacity. This option is available only for flexible volumes. It is possible to define how fast the volume should grow with the “-i” option. The default growth increment is 5% of the volume size at creation. It is also possible to define how large the volume is allowed to grow to with the “-m” option. If volume autosize is enabled, the default maximum size to grow to is 120% of the original volume size.

## 5.7 Autodelete

This is a volume setting available in Data ONTAP 7.1 and later that allows Data ONTAP to delete Snapshot copies if a definable threshold is met. This threshold is called a “trigger” and can be set so that Snapshot copies will be automatically deleted under one of the following conditions:

- **Volume**—The volume is near full. The volume will report that it is full even though there might still be space in the `snap_reserve` and `space_reserve` areas. This is reported in the first line reported for each volume in the “df” command.
- **Snap\_reserve**—The snap reserve space is near full. The snap reserve space will report being full and is considered independent of whether there is space in the `snap_reserve` and `space_reserve` areas.
- **Space\_reserve**—The overwrite reserved space is full. This is the space determined by the LUNs with space reservations enabled and the `fractional_reserve` option. The reserve space will never be full until after both the volume and the `snap_reserve` areas are full.

There are three options that can be set to define the order in which Snapshot copies should be deleted:

- **Delete\_order**—This determines whether the oldest or newest Snapshot copies should be deleted first.
- **Defer\_deleted**—This allows the user to define a group of Snapshot copies that should be deleted first when no other Snapshot copies are available. It is possible to defer the deletion of user created or scheduled Snapshot copies or Snapshot copies beginning with a configurable prefix.
- **Commitment**—This option determines how Snapshot copies used for SnapMirror® and dump operations should be handled. If set to “try,” it will only delete these Snapshot copies if they are not locked. If set to “disrupt,” these Snapshot copies will be deleted even if they are locked. These will always be the last Snapshot copies deleted. In many configurations, deleting the last SnapMirror Snapshot copy is not desired because a new full baseline copy will be required to resume mirroring operations. If, for example, the source and destination are at different sites, this can be a time-consuming and costly process.

The algorithm to determine which Snapshot copy to delete will first look for a Snapshot that does not lie in the “defer\_delete” criteria and use the `delete_order` to determine whether to delete the oldest or more recent Snapshot. If no such Snapshot copy is found, the “defer\_delete” criteria will be ignored in the selection process. If a Snapshot copy is still not available for deletion, then the SnapMirror and dump Snapshot copies will be targeted depending on the “commit” option.

Snapshot copies will stop being deleted when the free space in the trigger criteria reaches the value of the `target_free_space` variable, which defaults to 80%.

If autosize is enabled and the autodelete trigger is set to “volume,” the “try\_first” volume option will determine whether a volume grow or Snapshot copy delete will be attempted first.

## 6. Thin Provisioning of LUNs

### 6.1 The Two Different Types of Thin Provisioning

Given the widespread use of the industry-leading NetApp Snapshot implementation, it is generally believed that there are two different types of thin provisioning:

- Thin provisioning of LUN space. The classic case of provisioning more space than is available.
- Thin provisioning of Snapshot space. If Snapshot copies are being used, having available less actual disk space than the total size of the LUNs will rarely, if ever, be practical. In this case the reserved space for Snapshot data is thinly provisioned.

### 6.2 Monitoring of Available Space

If thin provisioning will be employed, it is imperative that the free space on the storage system be well monitored. This is because if the available space for a critical application is getting too low, a plan can be executed quickly enough to make more space available for that particular application. A well-managed system (i.e., software is up to date, etc.) is not the same thing as a well-monitored system.

### 6.3 Thin Provisioning of LUNs

Given the previous explanation of how LUNs will typically fill up close to 100%, strict thin provisioning of LUNs often will not provide significant savings for an extended period of time. However, with a large number of LUNs or very large LUNs, there is certainly the potential for savings.

There are many ways to configure the NetApp storage appliance for LUN thin provisioning; each has advantages and disadvantages. The following are best practice configurations:

- Volume Guarantee=None Configuration
  - volume guarantee = none
  - LUN reservation = not applicable
  - volume auto\_grow = off
  - volume fractional\_reserve = not applicable since Snapshot copies are not being used

This configuration has the advantages that the free space in the aggregate is used as a shared pool of free space and that it works with all versions of SnapDrive® without issues. The disadvantage of this configuration is that the level of thin provisioning cannot be tuned on an individual volume basis.

- Autogrow Configuration
  - volume guarantee = volume
  - LUN reservation = disabled
  - volume auto\_grow = on
  - volume fractional\_reserve = not applicable since Snapshot copies are not being used

This configuration has the advantage that it is possible, if desired, to finely tune the level of thin provisioning for each application. With this configuration the volume size defines or guarantees an amount of space that is only available to LUNs within that volume. The aggregate provides a shared storage pool of available space for all the volumes contained within it. If the LUNs require more space than available in the volume, the volumes will automatically grow, taking more space from the containing aggregate. The degree of thin provisioning is done on a per-volume level, allowing an administrator to, for example, set the volume size to 95% of the cumulative LUN size for a more critical application and to 80% for a less critical application. It is possible to tune how

much of the shared available space in the aggregate a particular application can consume by setting the maximum size to which the volume is allowed to grow.

The disadvantage is that this configuration does not work with SnapDrive for Windows (as of version 5.0 and earlier). Also, the current Data ONTAP limitation of 16TB aggregates should be considered in this scenario. The sum of all volumes on the aggregate should be sized in an autogrow configuration so that this limit is not approached.

- LUN Reserve Disabled Configuration
  - volume guarantee = volume
  - LUN reservation = disabled
  - volume auto\_grow = off

volume fractional\_reserve = not applicable since Snapshot copies are not being used

This configuration is useful for traditional volumes and cases with multiple LUNs or LUNs for multiple applications in a single volume. For data management purposes in normal circumstances, having each application in separate volumes is recommended. However, since Snapshot copies are not used in this configuration, other considerations might lead to multiple applications in a single volume.

This configuration ensures that the LUNs in a particular volume have access to the space reserved by the volume but limits those LUNs from consuming more than that amount of space. If more space is needed, the volume can still be grown manually using either the “vol add” command for traditional volumes or the “vol grow” command for flexible volumes.

#### **6.4 Thin Provisioning through LUN Growing**

Because of the way that LUNs are utilized, one of the best ways to thin provision LUNs in a Network Appliance environment is to size the LUNs to what is needed at the time and then grow them as needed. Unlike other storage arrays, Data ONTAP enables this type of strategy with online, flexible, and simple provisioning from the storage side. In addition, Network Appliance offers SnapDrive for Windows and SnapDrive for UNIX®, which completely automate the growing of the file system, taking care of all necessary actions on the host as well as on the storage. These technologies allow the storage administrator to grow the storage as needed while eliminating the overprovisioning necessitated by solutions that don't enable quick and seamless LUN growth.

## 7. Thin Provisioning of Snapshot Space

### 7.1 Thin Provisioning of Snapshot Space

The vast majority of Network Appliance customers take advantage of the industry-leading Data ONTAP Snapshot technology. Any time Snapshot copies are used, space needs to be reserved to hold the Snapshot data. The amount of space needed for Snapshot data varies greatly and is determined by a number of factors. A few of the most critical factors include application behavior, number of Snapshot copies being held, and the length of time for which they are held. For example, if Snapshot copies are being created solely by SnapMirror, the life of the copies is relatively short and a small number of copies are retained so that it is not necessary to have the default 100% reserve for the Snapshot space.

### 7.2 Thin Provisioning of Snapshot Space Configurations

Given the large number of tunable options, there are a large number of possible configurations for thin provisioning of Snapshot space. Each of them has various advantages and disadvantages. The following are best practice configurations:

- **Autodelete Configuration**

guarantee	= volume
LUN reservation	= enabled
fractional_reserve	= 0%
snap_reserve	= Y%
autodelete	= snap_reserve / oldest_first
autosize	= off
try_first	= snap_delete

In this configuration, Snapshot copies are limited to using no more space than what's defined by `snap_reserve`. This is by default set to 20%. Once the `snap_reserve` area fills up, the oldest Snapshot copy will be deleted to maintain free space. This configuration makes it a priority to keep the LUNs accessible over maintaining Snapshot copies.

One of the advantages of this configuration is that it is easy to monitor and understand the space usage by just monitoring the volume and reserve areas, seen under the "df" command as ".Snapshot". The volumes are also independent of each other, meaning each application can be tuned independently. This configuration also has the advantage that only the oldest Snapshot copies are deleted while the most recent ones are maintained. The disadvantage of this configuration is that it doesn't use the available space in the aggregate as a shared pool of available space.

The "auto\_grow" option could also be enabled for extra safety if, for example, SnapMirror is being used and those Snapshot copies should not be deleted. If those Snapshot copies cause the space to run out, the autogrow operation would allow the volume to take extra space from the aggregate to avoid running out of space.

- **Autosize Configuration**

guarantee	= volume
LUN reservation	= enabled
fractional_reserve	= 0%
snap_reserve	= 0%
autodelete	= off
autosize	= on
try_first	= volume_grow

The big advantage of this configuration is that it takes advantage of using the free space in the aggregate as a shared pool of available space. Since a guarantee of volume is being used, a different level of thin provisioning per application can easily be achieved through individually sizing the volumes and tuning how much each volume is allowed to grow. Space usage is also very easy to monitor and understand by simply looking at the volume and aggregate usage.

As with all configurations using a shared free space, a disadvantage of this configuration is that the volumes are not 100% independent of one another since they are competing for that space when they need to grow.

- **Fractional\_reserve Configuration**

guarantee	= volume
LUN reservation	= enabled
fractional_reserve	= Y%
snap_reserve	= 0%
autodelete	= space reserve / oldest_first
autosize	= on
try_first	= snap_delete

This configuration is similar to the default setting with a 100% setting for “fractional\_reserve,” but it reduces the reserve space to below 100% and adds Snapshot autodelete as a final safety factor to prevent running out of space. The big advantage of this configuration is that SnapManager® for Exchange will monitor the space available in the reserve area and delete SnapManager for Exchange–created Snapshot copies to free space as needed. SnapDrive for Windows (version 4.1 or later) can also monitor the space available in the space or fractional reserve area and issue alerts that allow the administrator to react.

## 8. Summary

As discussed, when using NetApp storage and leveraging FlexVol technology, it is possible to thin provision both LUN and Snapshot space to allow greater efficiency and lower cost. The best practice configurations detailed in this paper are excellent starting points and can be modified as needed to precisely fit a particular environment. There are multiple mechanisms in both Data ONTAP and higher level applications such as

SnapDrive and SnapManager for Exchange that allow the administrator to customize the level of thin provisioning to match the environment, thereby enabling maximum cost savings.



[www.netapp.com](http://www.netapp.com)

© 2006 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp, the Network Appliance logo, DataFabric, FAServer, FilerView, NetCache, NearStore, SecureShare, SnapManager, SnapMirror, SnapRestore, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, and WAFL are registered trademarks and Network Appliance, ApplianceWatch, BareMetal, Camera-to-Viewer, ContentDirector, ContentFabric, Data ONTAP, EdgeFiler, HyperSAN, InfoFabric, MultiStore, NetApp Availability Assurance, NetApp ProTech Expert, NOW, NOW NetApp on the Web, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapCopy, SnapDirector, SnapDrive, SnapFilter, SnapMigrator, Snapshot, SnapSuite, SnapVault, SohoCache, SohoFiler, SpinMirror, SpinShot, SpinStor, The evolution of storage, Vfiler, VFM, Virtual File Manager, and Web Filer are trademarks of Network Appliance, Inc. in the U.S. and other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.