



**NetApp™**  
Go further, faster

NETAPP TECHNICAL REPORT

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

Rick Jooss  
Jan 2008 | 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

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
<b>2</b>	<b>INTENDED AUDIENCE AND ASSUMPTIONS .....</b>	<b>3</b>
<b>3</b>	<b>THIN PROVISIONING .....</b>	<b>3</b>
3.1	<b>DEFINITION .....</b>	<b>3</b>
3.2	<b>WAFL—THE ENABLING TECHNOLOGY .....</b>	<b>4</b>
<b>4</b>	<b>LUN SPACE USAGE .....</b>	<b>4</b>
4.1	<b>HOST VERSUS STORAGE VIEW .....</b>	<b>4</b>
4.2	<b>LUN CAPACITY USED OVER TIME .....</b>	<b>6</b>
4.3	<b>SPACE RECLAMATION .....</b>	<b>7</b>
4.4	<b>HOST VERSUS STORAGE VIEW WITH SPACE RECLAMATION .....</b>	<b>7</b>
<b>5</b>	<b>DATA ONTAP CONFIGURATION PARAMETERS .....</b>	<b>8</b>
5.1	<b>LUN RESERVATION .....</b>	<b>8</b>
5.2	<b>GUARANTEES .....</b>	<b>8</b>
5.3	<b>FRACTIONAL_RESERVE .....</b>	<b>10</b>
5.4	<b>SNAP RESERVE .....</b>	<b>11</b>
5.5	<b>VOLUME FILLING EXAMPLE .....</b>	<b>12</b>
5.6	<b>AUTOSIZE .....</b>	<b>15</b>
5.7	<b>AUTODELETE .....</b>	<b>16</b>
<b>6</b>	<b>EFFICIENT PROVISIONING WITH SNAPSHOT COPIES .....</b>	<b>18</b>
6.1	<b>PROVISIONING WITH SNAPSHOT COPIES .....</b>	<b>18</b>
6.2	<b>BEST PRACTICE CONFIGURATIONS: PROVISIONING WITH SNAPSHOT COPIES .....</b>	<b>18</b>
<b>7</b>	<b>THIN PROVISIONING OF LUNS .....</b>	<b>19</b>
7.1	<b>THIN PROVISIONING OF LUN SPACE .....</b>	<b>19</b>
7.2	<b>MONITORING OF AVAILABLE SPACE .....</b>	<b>20</b>
7.3	<b>SNAPDRIVE FOR WINDOWS AND THIN PROVISIONED LUNS .....</b>	<b>20</b>
7.4	<b>BEST PRACTICE CONFIGURATIONS .....</b>	<b>20</b>
7.5	<b>THIN PROVISIONING THROUGH LUN GROWING .....</b>	<b>21</b>
<b>8</b>	<b>SUMMARY .....</b>	<b>21</b>

# Table of Contents

## 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 7.2.4**

## 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 diverse users and applications utilizing the storage are typically the better the chance to take advantage of thin provisioning.

### 3.2 WAFL—The Enabling Technology

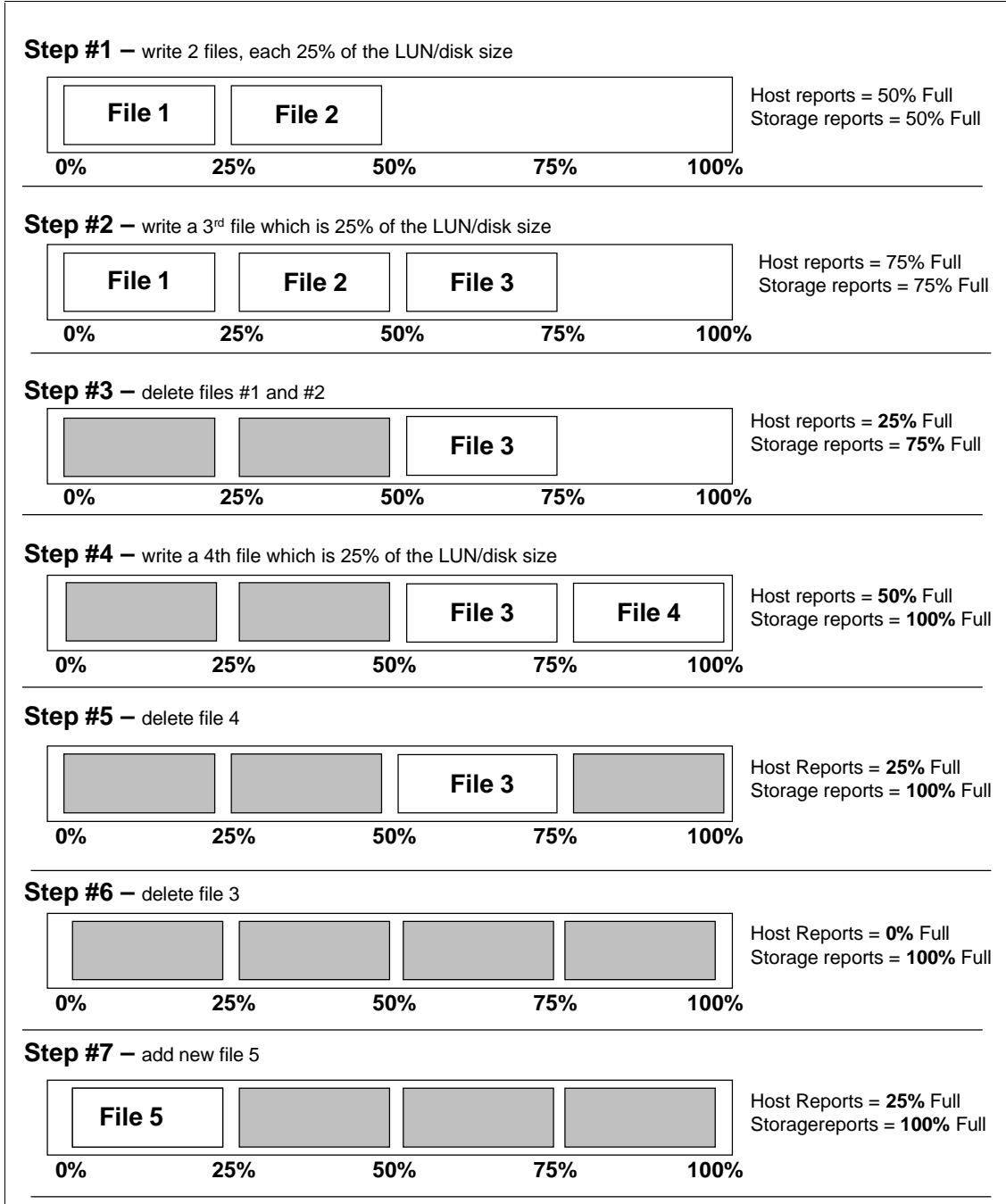
WAFL, which can be thought of as the virtualization layer of Data ONTAP, enables thin provisioning. When a LUN is created it does not dedicate specific blocks out of the NetApp volume for the LUN or for Snapshot™ copies of the LUN. Instead, it allocates the blocks from the NetApp volume when the data 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

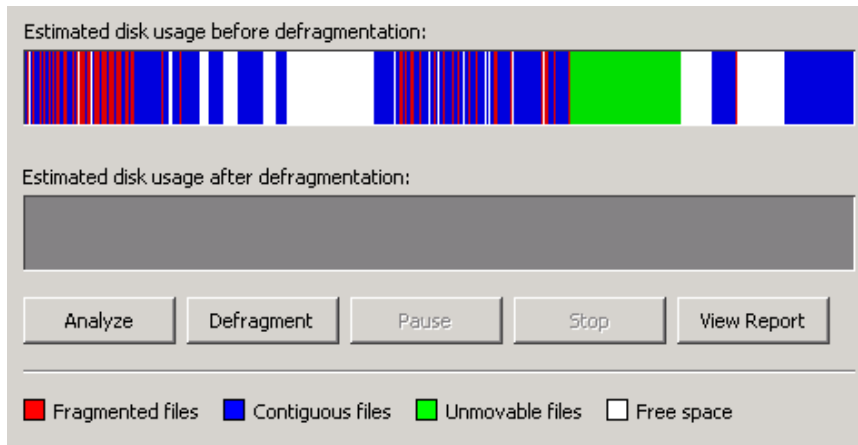
### 4.1 Host versus Storage View

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 much space from the LUN is used. 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 unless space reclamation is used which is discussed later.

The graphic below helps show why an individual LUN's usage never decreases without space reclamation. 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 that 75% of the LUN is used. 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 the storage system's perspective. As mentioned, the fill percentage of a LUN will never go down without space reclamation and Steps 5 through 7 show that, regardless of the host activity on the LUN, it will remain 100% full.



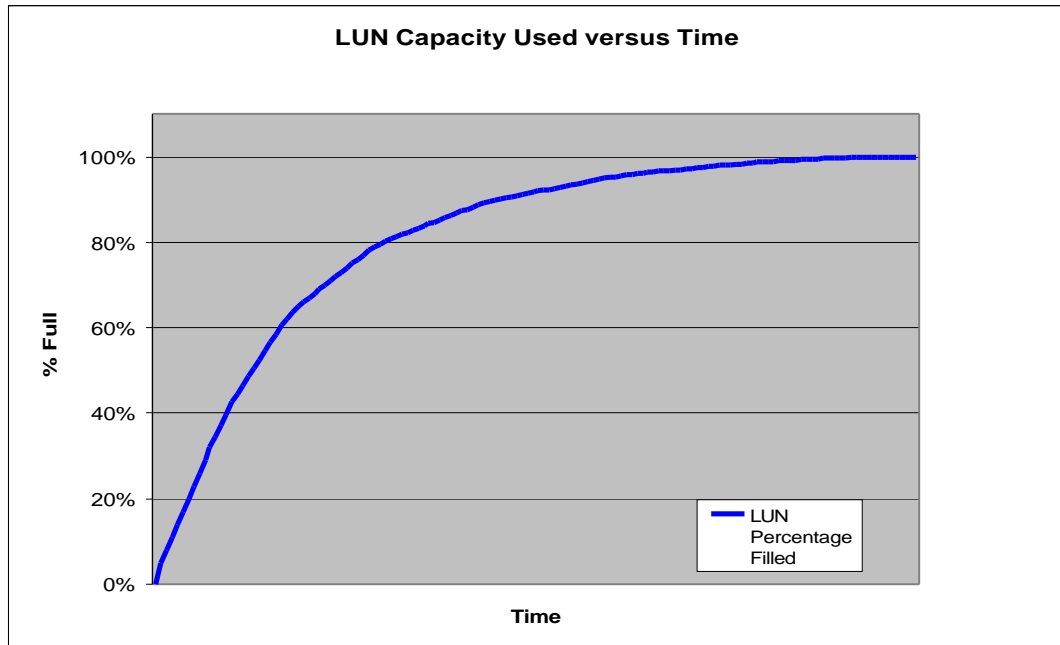
**Figure 1) LUN capacity—host versus storage view.**



**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 assuming no space reclamation. The steepness of the curve and whether it goes 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 to fill to its maximum value can, therefore, vary from hours to years. The important point of this graph is that the used capacity never goes down from the storage system perspective unless space reclamation is used.



**Figure 3) LUN capacity used versus time.**

### 4.3 Space Reclamation

Space Reclamation is the process of allowing the storage system to free the blocks no longer used by the host operating system. These blocks can then be used by the storage system for other purposes such as holding data for SnapShot copies. SnapDrive for Windows version 5.0 introduced space reclamation for Windows NTFS filesystems.

### 4.4 Host versus Storage View with Space Reclamation

The example shown above in section 4.1 showed how space usage between the host filesystem and the LUN diverged over time. Space Reclamation can in theory eliminate the difference between the capacity used from the LUN and the amount used within the host filesystem. In practice space reclamation can not completely eliminate the difference but can greatly reduce it.

The first three steps in the example below show the same results as the example show without space reclamation. Step #4 shows the results after space reclamation has run. Space reclamation is done by SnapDrive for Windows going through the host filesystem, locking a group of free blocks on the host filesystem and sending a special command to the storage system for those blocks telling the storage system that those blocks are no longer holding used data. The blocks are temporary locked to avoid an application trying use those blocks during the reclamation process. The storage system then frees those blocks.

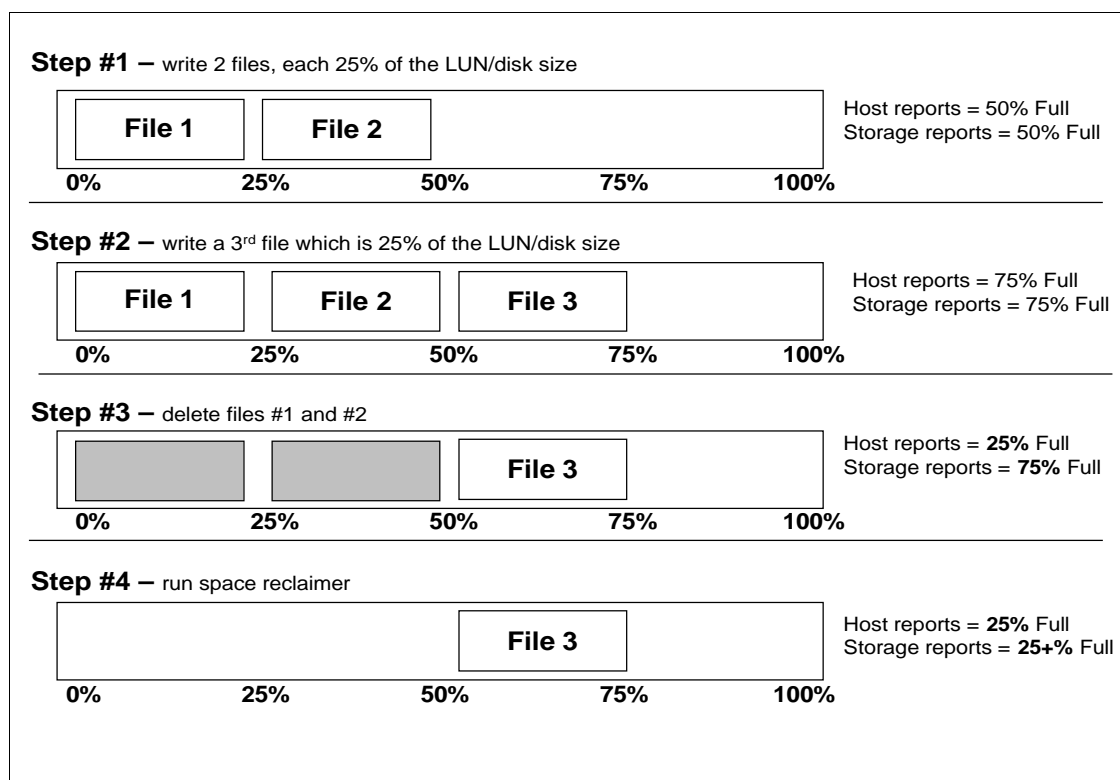


Figure 3) LUN capacity—host versus storage view with space reclamation

## 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 free space 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 data was written to the LUN.

### 5.2 Guarantees

With flexible volumes, introduced with Data ONTAP 7.0, there is the 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 aggregate’s available space at volume creation time. The space is reserved from the aggregate 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 of the aggregate before and after the volume create command display 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.

```
netapp1> df -A -g aggr0
Aggregate                total      used      avail capacity
aggr0                    85GB      0GB      85GB      0%
aggr0/.snapshot          4GB      0GB      4GB      0%
```

```
netapp1> vol create flex0 aggr0 20g
Creation of volume 'flex0' with size 20g on hosting aggregate 'aggr0' has
completed.
```

```
netapp1> df -g /vol/flex0
Filesystem                total      used      avail  capacity  Mounted on
/vol/flex0/                16GB      0GB      16GB      0% /vol/flex0/
/vol/flex0/.snapshot       4GB      0GB      4GB      0% /vol/flex0/.snapshot
```

```
netapp1> df -A -g aggr0
Aggregate                total      used      avail capacity
aggr0                    85GB      20GB      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 from the aggregate during volume creation. 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 above with the volume guarantee, the volume creation does not reduce used space in the aggregate. Even LUN creation, which by default has space reservation enabled, does not reserve space out of the aggregate. Write operations to space-reserved LUNs in a volume with guarantee=none will fail if the containing aggregate does not have enough available space. LUN reservation assure that the LUN has space in the volume but guarantee=none doesn't assure that the volume has space in the aggregate.

```
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>
netappl> df -g /vol/noneflex
Filesystem          total  used  avail capacity  Mounted on
/vol/noneflex/      16GB  0GB   16GB      0%  /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> 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>
netappl> df -g /vol/noneflex
Filesystem          total  used  avail capacity  Mounted on
/vol/noneflex/      16GB  10GB   6GB      0%  /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%
```

- **File**—With `guarantee=file` the aggregate assures 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 guarantee. 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 with `guarantee=none` except in this example the LUN creation takes space from the aggregate because it is a space-reserved object. 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 file 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         0%  /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>
netappl> df -g /vol/noneflex
Filesystem                total    used    avail capacity  Mounted on
/vol/noneflex/            16GB   10GB    6GB         0%  /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         10GB         75GB         12%
aggr0/.snapshot          4GB         0GB         4GB         0%

```

### 5.3 Fractional\_reserve

Fractional reserve is a volume option that determines how much space Data ONTAP will reserve for Snapshot overwrite data for LUNs and space-reserved files to be used after all other space in the volume is used. The default value is 100%, however, with the introduction of the autodelete functionality, NetApp generally recommends setting `fractional_reserve = 0` and using the autodelete functionality. Data ONTAP removes or reserves this space from the volume as soon as the first Snapshot copy is created and will

only use this space when the rest of the volume has been filled. For example, as shown in 0 on the left-hand side, a 100GB volume is shown with two 20GB LUNs. Assuming the LUNs are full Data ONTAP will with the default `fractional_reserve=100` 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 would be completely overwritten. This is depicted in the middle of 0. As depicted on the right-hand side, if `fractional_reserve` is set to 60% when the Snapshot copy is created, instead of reserving 40GB in the volume, Data ONTAP will reserve 24GB (60% \* [2 x 20GB]).

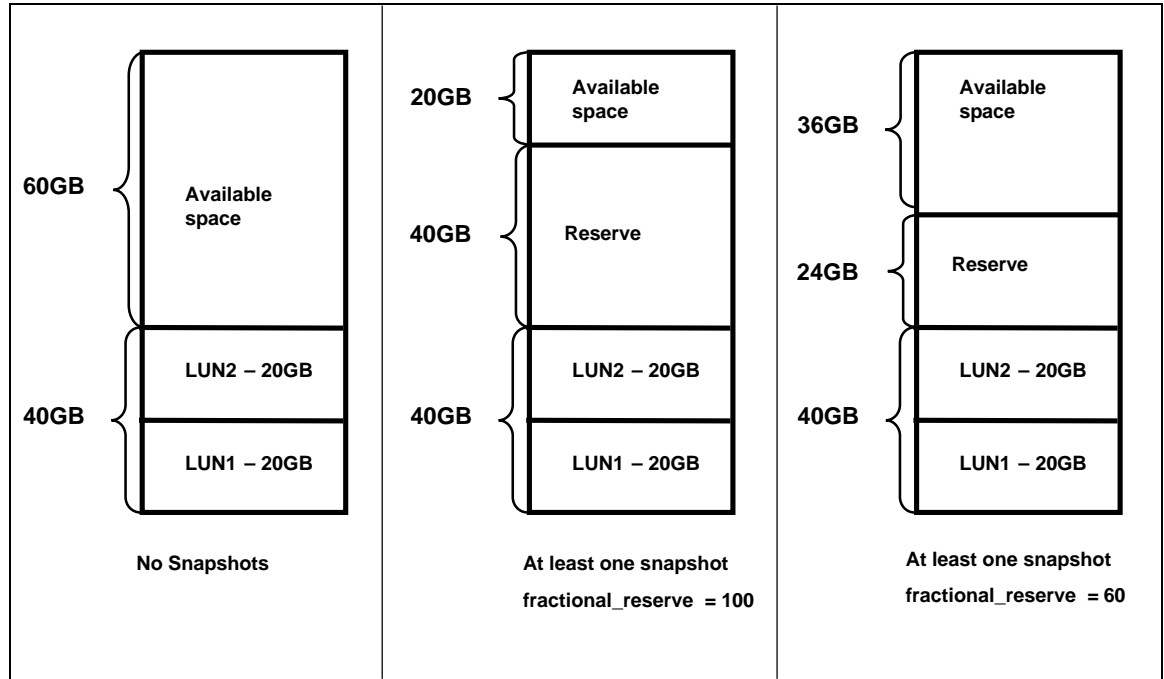


Figure 4) Fractional\_reserve.

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 file usage with CIFS or NFS. 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.

## 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.

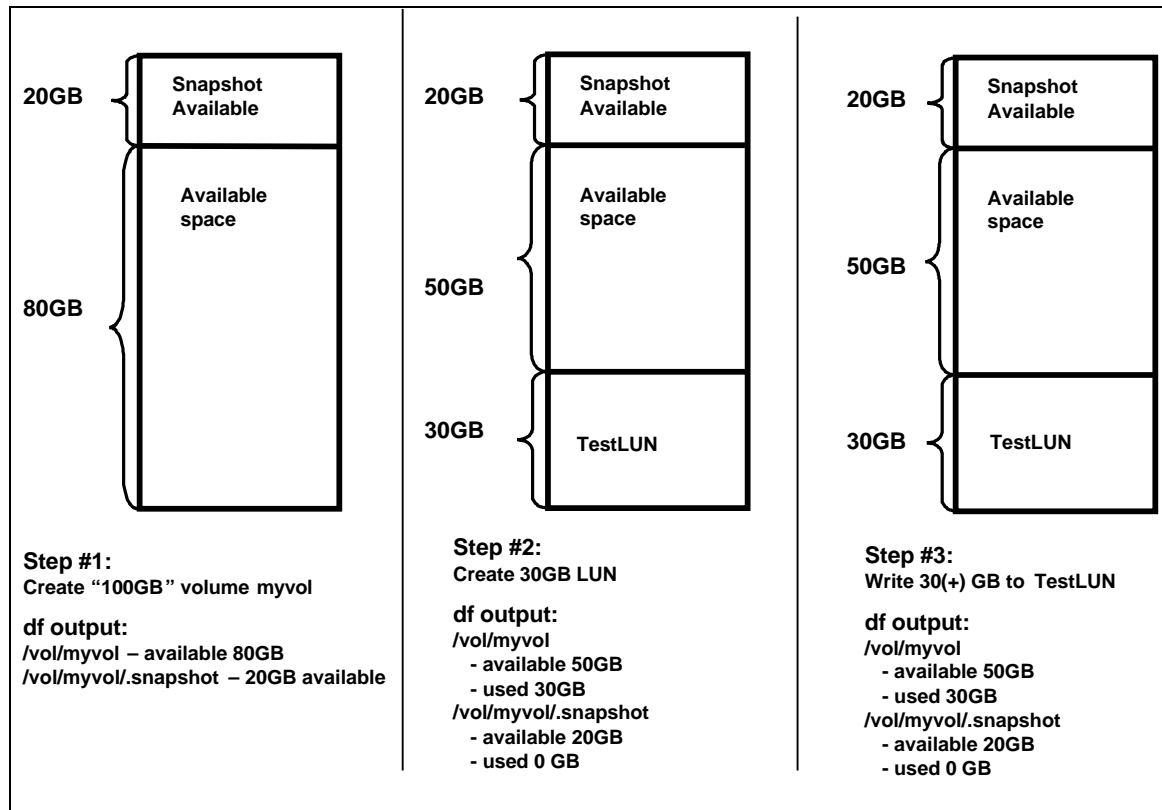
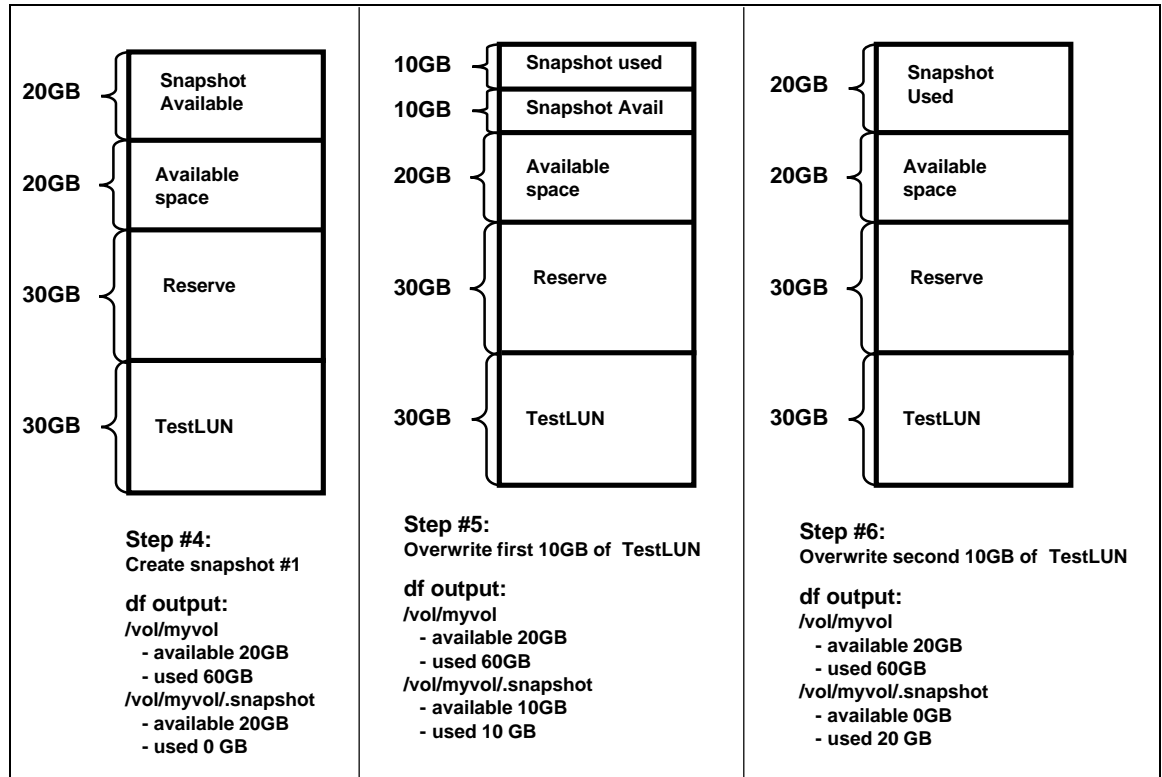


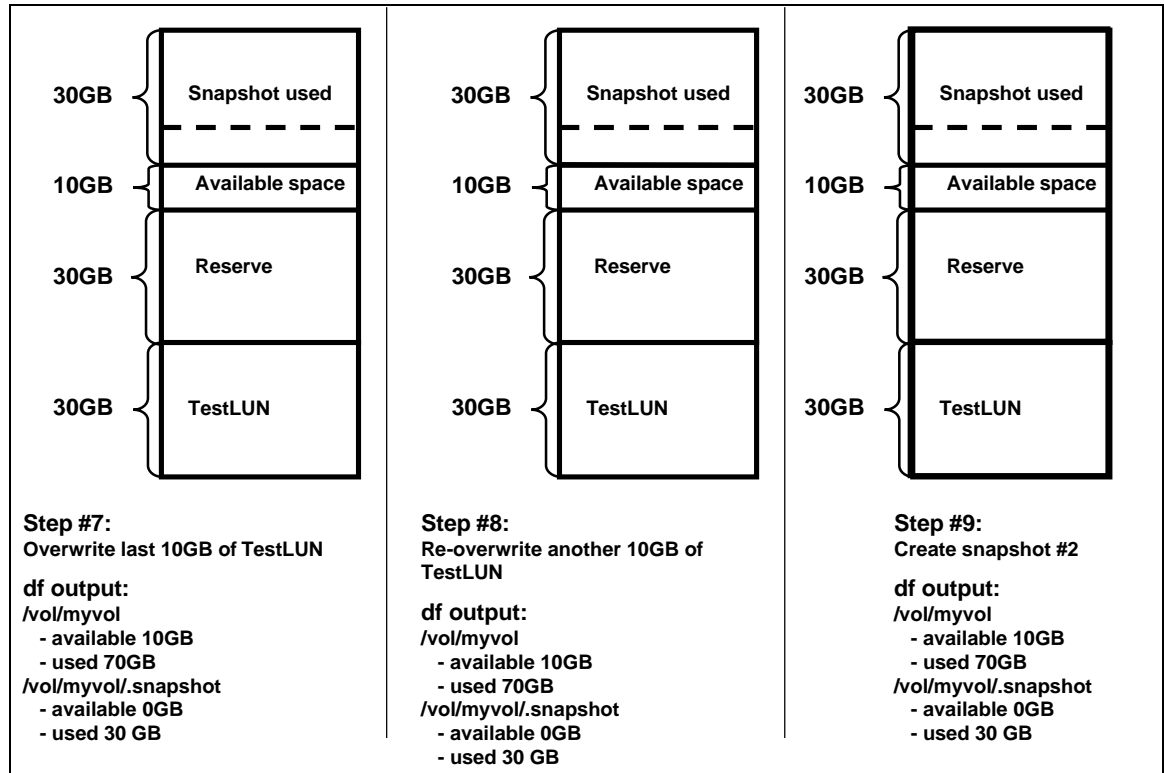
Figure 5) Volume filling—Actions #1–#3.

- 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.



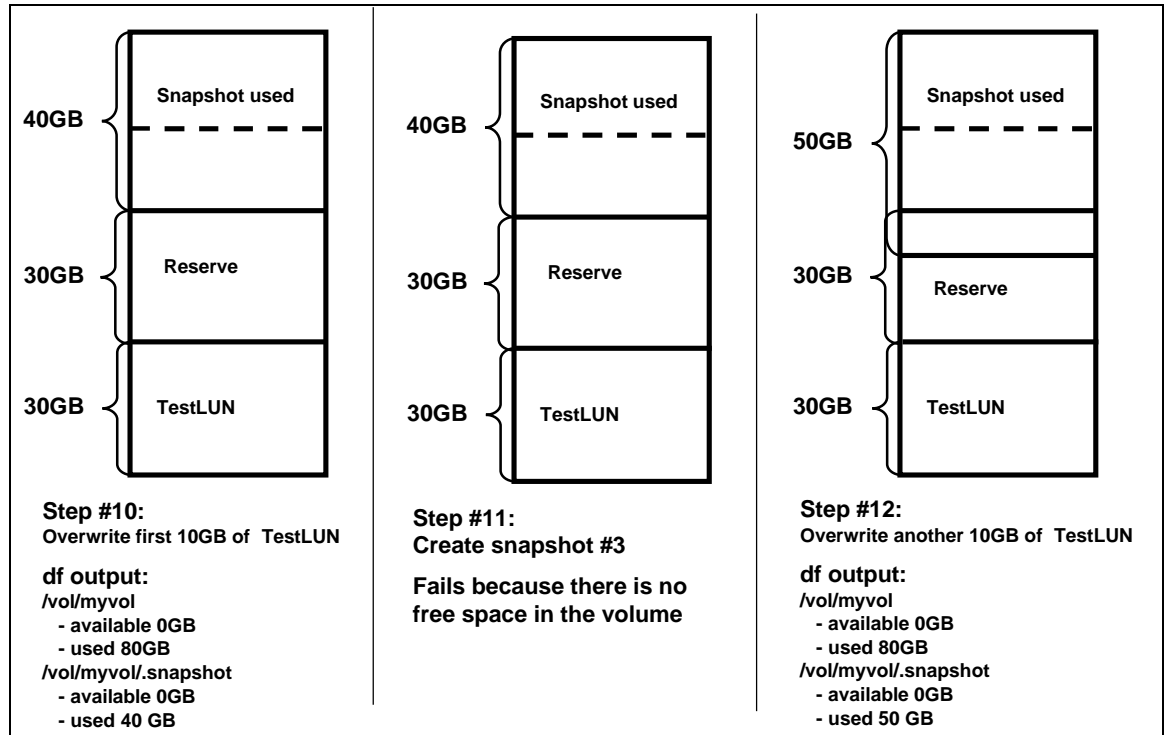
**Figure 6) Volume filling—Actions #4–#6.**

- 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 0, 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 volume setting (available in Data ONTAP 7.1 later) 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 volume setting (available in Data ONTAP 7.1 and later) allows Data ONTAP to delete Snapshot copies if a 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. This is reported in the first line reported for each volume in the “df” command. It should be noted that the volume can be full even though there might still be space in the snap\_reserve areas.
- **Snap\_reserve**—The snap reserve space is near full.
- **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 filled until both the volume and the snap\_reserve areas are full.

It is strongly recommended to set the trigger to volume.

The order in which Snapshot copies are deleted is determined by the following three options:

- **Delete\_order**—This option determines whether the oldest or newest Snapshot copies should be deleted first.
- **Defer\_deleted**—This option allows the user to define a group of Snapshot copies that should first be deleted when no other Snapshot copies are available. It is possible to defer the deletion of user created Snapshot copies, 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, re-creating this baseline 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 dum 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 both autosize and autodelete are 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 EFFICIENT PROVISIONING WITH SNAPSHOT COPIES

### 6.1 Provisioning with Snapshot copies

The vast majority of NetApp 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 copies varies greatly and is determined by a number of factors. A few of the most critical factors include application behavior, the length of time for which Snapshot copies are held, and the number of Snapshot copies being 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 enormous amounts of space reserved.

### 6.2 Best Practice Configurations: Provisioning with Snapshot copies

Given the large number of tunable options, there are a variety of possible configurations for provisioning for configurations taking advantage of Snapshot copies. Each of them has various advantages and disadvantages. The following are best practice configurations recommended by NetApp:

- **Autogrow Configuration**

guarantee	= volume
LUN reservation	= enabled
fractional_reserve	= 0%
snap_reserve	= 0%
autodelete	= on / oldest_first
autogrow	= 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, the volumes are not 100% independent of one another since they are competing for that space when they need to grow. Meaning that if other volumes have already consumed the space in the aggregate it might not be possible for a volume to grow.

Care needs to be taken if clones (FlexClone or LUN clones) are being created as Snapshot copies which are backing this clones cannot currently be deleted by the autodelete functionality. This risk can be avoided if the volume is sized at 2X or greater of the LUN size and there is only one locked snapshot.

When using this configuration the volume would typically be sized to hold the LUNs and the Snapshot copy data. For example, if the volume needs to hold the following:

- 5 x 100GB LUNs
- 3% change rate per day
- Snapshots copies to be retained for 14 days

it would be need to be sized at a minimum of  $500\text{GB} + ((500\text{GB} * 3\%) * 14) = 710\text{GB}$ . Ideally, it would be sized slightly larger in case of any unexpected growth in Snapshot copies. If the Snapshot copies require more space than planned the volume would attempt to autogrow and if not possible would autodelete to free space.

- **Autodelete Configuration**

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

This configuration is the same as the autogrow configuration with the removal of the autogrow functionality. This configuration makes it a priority to keep the LUNs accessible over maintaining Snapshot copies. If the volume starts to fill up Snapshot copies will be deleted to allow the LUNs to remain online.

One of the advantages of this configuration is that it is easy to monitor and understand the space usage by just monitoring the volume space usage. 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.

As with the previous configuration, care needs to be taken if clones (FlexClone or LUN clones) are being created as Snapshot copies which are backing these clones cannot currently be deleted by the autodelete functionality.

When using this configuration the volume would typically be sized to hold the LUNs and the Snapshot copy data. For example, if the volume needs to hold the following:

- 5 x 100GB LUNs
- 3% change rate per day
- Snapshots copies to be retained for 14 days

it would be need to be sized at a minimum of  $500\text{GB} + ((500\text{GB} * 3\%) * 14) = 710\text{GB}$ . Ideally, it would be sized slightly larger in case of any unexpected growth in Snapshot copies.

## 7 THIN PROVISIONING OF LUNS

### 7.1 Thin provisioning of LUN space

As defined above 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. For systems whose storage needs will grow over time thin provisioning allows the storage manager to provide the need storage more quickly and cost effectively.

For long term thin provisioning where the storage system will never grow to the full size of the storage being presented out to the servers thin provisioning enable long term savings.

## 7.2 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.

## 7.3 SnapDrive for Windows and thin provisioned LUNs

Starting with SnapDrive for Windows 5.0 it is possible to create LUNs with space reservations disabled. Since deciding to thin provision LUNs is the responsibility of the storage system administrator and not the server administrator, it is necessary to create a configuration file on the storage system itself indicating to SnapDrive for Windows to create non-space reserved LUNs in a particular volume. For each volume where SnapDrive for Windows should create LUNs with space reservations a file with the name `sdparams.conf` should be created with the line:

```
space_reservation_new_lun = off (no spaces)
```

The file can be created through the storage system console using the `wrfile` command or through a NFS or CIFS share to the volume. When this is set all LUNs created by SnapDrive for Windows will be created with space reservations disabled. Starting with SnapDrive for Windows 5.0 all checks for enabled space reservations during SnapDrive start-up and LUN connect operations have been removed.

## 7.4 Best Practice Configurations

There are many ways to configure the NetApp storage appliance for LUN thin provisioning; each has advantages and disadvantages. It should be noted that it is possible to have thinly provisioned volumes and non-thinly provisioned volumes on the same storage system or even the same aggregate. LUNs for critical production application might be configured without thin provisioning while LUNs for other types of applications may be thinly provisioned. The following are considered to be best practice configurations:

- **Volume Guarantee=None Configuration**

<code>guarantee</code>	= none
<code>LUN reservation</code>	= enabled
<code>fractional_reserve</code>	= 0%
<code>snap_reserve</code>	= 0%
<code>autodelete</code>	= volume / oldest_first
<code>autosize</code>	= off
<code>try_first</code>	= snap_delete

This configuration has the advantages that the free space in the aggregate is used as a shared pool of free space. The disadvantages of this configuration is that there is a high level of dependency between volumes and the level of thin provisioning cannot easily be tuned on an individual volume basis. When using this configuration the total size of the volumes would be greater than the actually storage available in the host aggregate. With this configuration the storage administrator will generally size volume so that they only need to manage and monitor the used space in the aggregate.

- **Autogrow/Autodelete Configuration**

guarantee	= volume
LUN reservation	= disabled
fractional_reserve	= 0%
snap_reserve	= 0%
autodelete	= volume / oldest_first
autosize	= on
try_first	= autogrow

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 or Snapshot copies 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 as explained in the description of the autogrow feature.

In cases where snapshots are also being used the volume might also be configured larger than the size of the LUNs contained within the volume. The advantage of having the LUN space reservation disabled in that case is that snapshots can then use the space that is not needed by the LUNs. The LUNs themselves are also not in danger of running out of space because the autodelete feature will remove the snapshots consuming space. It should be noted that, currently, snapshots used to create clones will not be deleted by autodeleted.

## 7.5 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 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.

## 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.

© 2006 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp, the Network Appliance logo, DataONTAP, SnapDrive, SnapMirror, and SnapManager are registered trademarks and Network Appliance, FlexClone, FlexVol, and Snapshot are trademarks of Network Appliance, Inc. in the U.S. and other countries. UNIX is a registered trademark of The Open Group. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.