



NetApp® Unified Storage Capacity Management Using Open Interfaces

Network Appliance, Inc.

March 2010

Executive Summary

NetApp unified storage systems support multiprotocol data access and can be configured as Fibre Channel and iSCSI SAN and NAS devices simultaneously. NetApp storage systems support different type of storage objects like aggregates, volumes, LUNs, qtrees etc. and provides open interfaces like Data ONTAP APIs, SNMP, SMI-S agent for monitoring and managing various components of the NetApp storage system. This document provides the details of how to use NetApp open interfaces for unified storage capacity management and how to simplify capacity management of NetApp storage systems when multiple protocols are supported and multiple objects are being managed.

Table of Contents

1	INTRODUCTION.....	3
1.1	BACKGROUND.....	3
1.2	UNIFIED CAPACITY MANAGEMENT.....	3
1.3	PURPOSE AND SCOPE.....	3
2	NETAPP OPEN INTERFACES.....	4
2.1	Data ONTAP APIs.....	4
2.2	SNMP.....	4
2.3	DATA ONTAP SMI-S AGENT.....	5
3	NETAPP UNIFIED STORAGE CONCEPTS.....	6
3.1	STORAGE CONTAINERS.....	6
3.1.1	<i>Physical Storage Containers</i>	7
3.1.2	<i>Logical Storage Containers</i>	9
3.1.3	<i>Data Storage Containers</i>	10
3.1.4	<i>Exporting the data</i>	10
3.2	LIMITS ON STORAGE CONTAINERS.....	12
3.3	STORAGE CAPACITY CONCEPTS.....	12
3.3.1	<i>Snapshot</i>	12
3.3.2	<i>Space reservation</i>	12
3.3.3	<i>Fractional (Overwrite) reserves</i>	13
3.3.4	<i>Space guarantees</i>	13
3.3.5	<i>Snapshot reserves</i>	13
3.3.6	<i>Quotas</i>	13
3.3.7	<i>WAFL Reserve</i>	14
3.3.8	<i>Raid Space</i>	14
3.4	STORAGE LAYOUT.....	15
3.4.1	<i>Disk Layout</i>	15
3.4.2	<i>Aggregate Layout</i>	17
3.4.3	<i>Volume Layout</i>	19
4	CAPACITY CALCULATION FOR STORAGE SYSTEM.....	21
4.1	TOTAL RAW CAPACITY INSTALLED.....	21
4.2	TOTAL FORMATTED (RIGHT-SIZED) CAPACITY.....	22
4.3	TOTAL SPARE CAPACITY.....	23
4.4	TOTAL CAPACITY IN RAID SPACE.....	24
4.5	TOTAL CAPACITY IN WAFL RESERVE.....	26
4.6	TOTAL CAPACITY IN RESERVED SPACE.....	27
4.7	TOTAL CAPACITY USABLE FOR PROVISIONING.....	30
4.8	TOTAL CAPACITY ALLOCATED.....	31
4.9	TOTAL CAPACITY OF USER USABLE DATA.....	33
4.10	TOTAL CAPACITY AVAILABLE.....	36

1 Introduction

1.1 Background

NetApp storage systems (FAS appliances, and NearStore systems) function as both NAS and SAN storage devices. They support a multiprotocol environment for data access and so they are termed as Unified Storage Devices (USD). NetApp storage systems export data *as files* via two primary protocols, NFS and CIFS, corresponding to the UNIX and Windows way of doing things. They can also export data as *blocks*, via FCP or iSCSI, and operate as SAN-attached disk arrays. They also support other file service protocols like HTTP, FTP and WebDAV. A single NetApp storage system can be configured to serve data over all these protocols.

NetApp's unified storage devices are based on the novel and patented design of its operating system, Data ONTAP[®] and its integrated file system, WAFL[®]. This integrated file system has different storage objects (both physical and logical) that form the storage hierarchy of the system.

1.2 Unified Capacity Management

The Unified Storage capabilities of NetApp storage systems require a different approach for managing the capacity of the storage system. The same storage object can be accessed through different protocols and can be represented as different host side objects like *files* or *block devices*. Also there are many different storage objects, both physical and logical, which form the storage hierarchy of the system and simplistic methods of capacity calculation are inadequate if the best usable storage capacity is to be achieved. Thus a unified view of capacity of the system is required to manage the capacity of the NetApp storage systems.

1.3 Purpose and Scope

NetApp provides following Open Interfaces to remotely manage the NetApp devices:

- NetApp Manageability SDK
- SNMP
- Data ONTAP SMI-S Agent

This document provides details of different capacity utilization scenarios in the NetApp storage systems and how to use NetApp open interfaces to manage these capacity utilization scenarios. This document covers only the use of NetApp Manageability SDK and SNMP interfaces for calculating capacity details, though SMI-S interfaces can also be used to do this.

2 NetApp Open Interfaces

2.1 *Data ONTAP APIs*

The ONTAPI interface is a set of foundational APIs for managing NetApp Storage Systems. This interface is developed by NetApp for advance management of NetApp Storage Systems.

ONTAPI interfaces use XML as a format and may be configured to use either HTTP, HTTPS or DCE/RPC as the transport mechanism. HTTP is important when managing devices which may be outside of the corporate firewall, and locked down so that only port 80 is available, and is the default transport configuration. HTTPS can be used for secure communication through ONTAPI.

NetApp Manageability SDK also provides a small set of core interfaces that marshal and un-marshal ONTAP API arguments using XML as the description language. At present, core interfaces are provided in C/C++, Perl and Java.

2.2 *SNMP*

SNMP (Simple Network Management Protocol) is a well-known standard for network management. NetApp storage systems support the SNMP version 1 compatible agent. This agent supports both MIB-II and the NetApp™ custom MIB. For reasons of security, NetApp supports only monitoring using SNMP which means that SNMP SET operations are not permitted.

If SNMP is enabled in Data ONTAP, SNMP managers can query your storage system's SNMP agent for information (specified in your storage system's MIBs or the MIB-II specification). In response, the SNMP agent gathers information and forwards it to the SNMP managers using the SNMP protocol. The SNMP agent also generates trap notifications whenever specific events occur and sends these traps to the SNMP managers. The SNMP managers can then carry out actions based on information received in the trap notifications.

The latest versions of the Data ONTAP MIB files are available online on the [NetApp on the Web \(NOW™\) site](#).

2.3 Data ONTAP SMI-S Agent

Data ONTAP SMI-S agent provides standards based storage management interface to discover, monitor, and manage NetApp storage systems. The specifications for SMI-S are developed by SNIA (Storage Networking Industry Association) and DMTF (Distributed Management Task Force) standards organizations.

The SMI-S agent is implemented as a proxy-based management solution. The agent needs to be installed on an external server (i.e. it is not implemented within ONTAP). Currently the supported platforms for SMI-S agent installation are Windows and Linux (Red Hat and SUSE) based hosts.

3 NetApp Unified Storage Concepts

3.1 Storage Containers

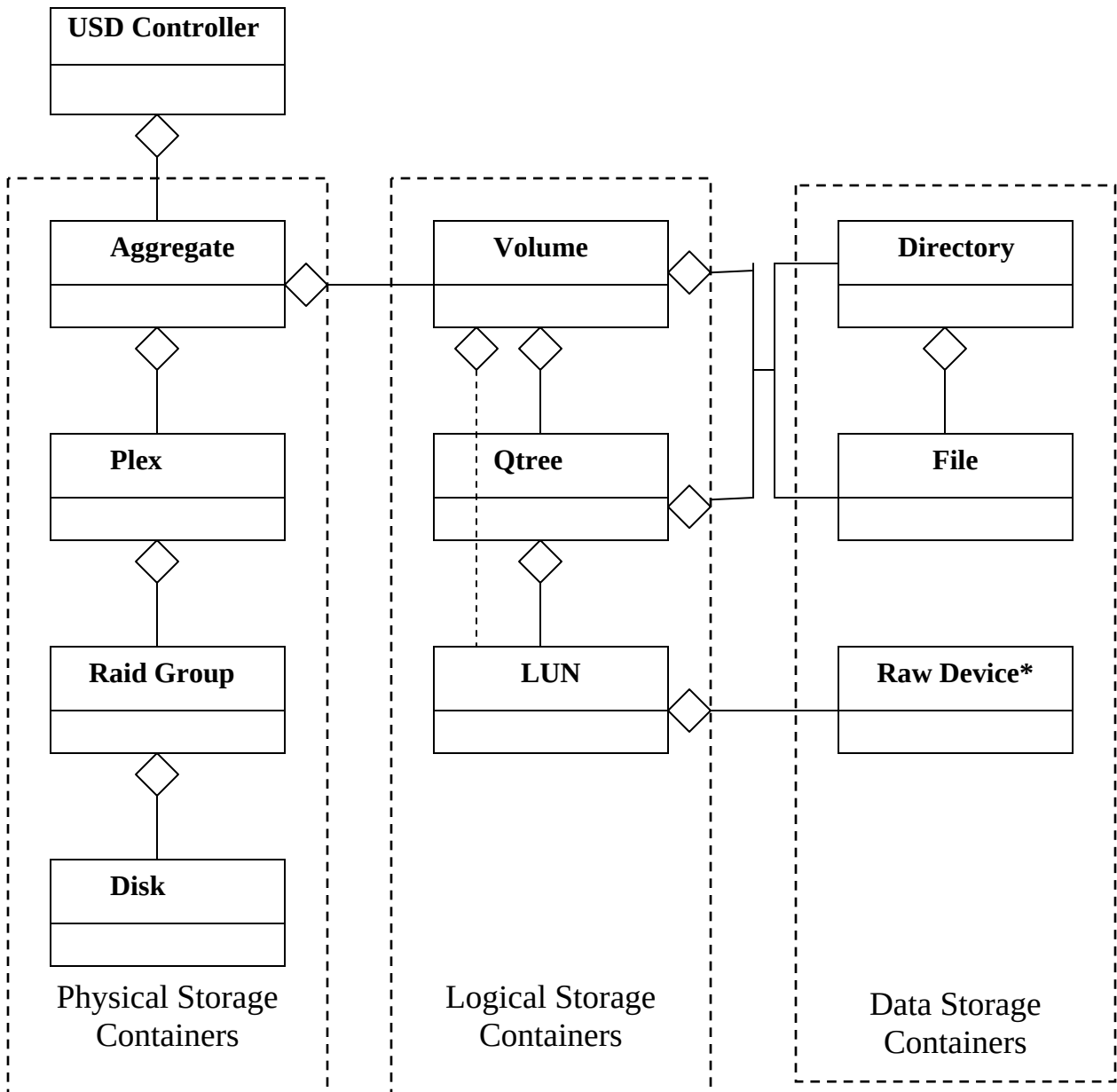


Fig: Storage Containers Hierarchy [Note:  indicates a “Contains” relationship]

* LUNs are exported as raw block devices to the hosts, hosts can create filesystem on these raw devices

3.1.1 Physical Storage Containers

The following storage entities form the physically identifiable storage objects on NetApp storage systems:

Disk:

Disks form the basic storage device in the NetApp storage systems. ATA disks, Fibre Channel disks, SCSI disks, SAS disks or SATA disks are used, depending on the storage system model.

Before the storage system is configured, all the disks are in an unassigned state. A disk must be assigned to a storage system before it can be used as a global spare or in a RAID group. If disk ownership is hardware based, disk assignment is performed by Data ONTAP. Otherwise, disk ownership is software based, and you must assign disk ownership.

Data ONTAP assigns and makes use of four different disk categories to support data storage, parity protection, and disk replacement. The disk category can be one of the following types

- Data disk - Holds data stored on behalf of clients within RAID groups (and any system management data)

- Global hot spare disk - Does not hold usable data, but is available to be added to a RAID group in an aggregate. Any functioning disk that is not assigned to an aggregate functions acts as a hot spare disk.

- Parity disk - Stores information required for data reconstruction within RAID groups.

- Double-parity disk - Stores double-parity information within RAID groups, if RAIDDP is used.

When you add a new disk, Data ONTAP reduces the amount of space on that disk available for user data by rounding down. Most storage vendors leveraging disk drives from multiple sources perform this process. This maintains compatibility across disks from various manufacturers. The available disk space listed by informational commands such as sysconfig is, therefore, less for each disk than its nominal rated capacity. This is called “rightsizing” of the disk.

Raid Group:

Data ONTAP organizes disks into RAID groups, which are collections of data and parity disks to provide parity protection. For Data ONTAP 6.5 onwards the following RAID types are supported for NetApp storage systems:

RAID4 technology: In this RAID, within each RAID group, a single disk is assigned for holding parity data, which ensures against data loss due to a single disk failure within a group.

RAID-DP™ technology (DP for double-parity): RAID-DP provides a higher level of RAID protection for Data ONTAP aggregates. Within its RAID groups, it allots one disk for holding parity data and one disk for holding double-parity data. Double-parity protection ensures against data loss due to a double disk failure within a group.

NetApp recommends the use of RAID-DP due to its much higher level of protection against errors caused by disk failure compared with single parity RAID, and its equal performance and capacity requirements.

Plex:

A plex is a collection of one or more RAID groups that together provide the storage for one or more WAFL® file system volumes. Data ONTAP uses plexes as the unit of RAID-level mirroring when the SyncMirror® feature is enabled. All RAID groups in one plex are of the same level, but may have a different number of disks.

Aggregate:

An aggregate is a collection of one or two plexes, depending on whether you take advantage of RAID-level mirroring. If the aggregate is unmirrored, it contains a single plex. If the SyncMirror feature is licensed and enabled, Data ONTAP adds a second plex to the aggregate, which serves as a RAID-level mirror for the first plex in the aggregate.

You use aggregates to manage plexes and RAID groups because these entities only exist as part of an aggregate. You can increase the usable space in an aggregate by adding disks to existing RAID groups or by adding new RAID groups. Once you've added disks to an aggregate, you cannot remove them to reduce storage space without first deleting the aggregate.

When you create an aggregate, Data ONTAP assigns data disks and parity disks to RAID groups, depending on the options you choose, such as the size of the RAID group (based on the number of disks to be assigned to it) or the level of RAID protection.

If the SyncMirror feature is enabled, you can convert an unmirrored aggregate to a mirrored aggregate and vice versa without any downtime.

3.1.2 Logical Storage Containers

Volume:

A volume is a logical file system whose structure is made visible to users when you export the volume to a UNIX host through an NFS mount or to a Windows host through a CIFS share. A volume is the most inclusive of the logical containers. It can store files and directories, qtrees, and LUNs.

Each volume depends on its containing aggregate for all its physical storage. The way a volume is associated with its containing aggregate depends on whether the volume is a traditional volume or a FlexVol volume.

Traditional volume: A traditional volume is contained by a single, dedicated, aggregate. A traditional volume is tightly coupled with its containing aggregate. The only way to increase the size of a traditional volume is to add entire disks to its containing aggregate. It is impossible to decrease the size of a traditional volume. The smallest possible traditional volume must occupy all of two disks (for RAID4) or three disks (for RAID-DP). Thus, the minimum size of a traditional volume depends on the size and number of disks used to create the traditional volume. No other volume can use the storage associated with a traditional volume's containing aggregate.

FlexVol volume: A FlexVol volume is loosely coupled with its containing aggregate. Because the volume is managed separately from the aggregate, FlexVol volumes give you a lot more options for managing their size. FlexVol volumes provide the following advantages:

- You can create FlexVol volumes in an aggregate. They can be as small as 20 MB and as large as the volume capacity that is supported for your storage system. These volumes stripe their data across all the disks and RAID groups in their containing aggregate.

- You can increase and decrease the size of a FlexVol volume in small increments (as small as 4 KB).

- You can increase the size of a FlexVol volume to be larger than its containing aggregate, which is referred to as aggregate over commitment

- You can clone a FlexVol volume, which is then referred to as a FlexClone™ volume.

A FlexVol volume can share its containing aggregate with other FlexVol volumes. Thus, a single aggregate is the shared source of all the storage used by the FlexVol volumes it contains.

Qtree:

A qtree is a logically-defined file system that exists as a special top-level subdirectory of the root directory within a volume. It can be created both in traditional and FlexVol volumes. You can use qtrees to organize files and directories, as well as LUNs.

You might create a qtree for either or both of the following reasons:

You can easily create qtrees for managing and partitioning your data within the volume.

You can create a qtree to assign user- or workgroup-based soft or hard usage quotas to limit the amount of storage space that a specified user or group of users can consume on the qtree to which they have access.

In general, qtrees are similar to volumes. However, they have the following key differences:

Snapshots can be enabled or disabled for individual volumes, but not for individual qtrees.

Qtrees do not support space reservations or space guarantees.

LUN:

In SAN environments, NetApp storage systems are targets that have storage target devices, which are referred to as LUNs. With Data ONTAP, you configure the storage systems by creating traditional volumes to store LUNs or by creating aggregates to contain FlexVol volumes to store LUNs. You can use LUNs to serve as virtual disks in SAN environments to store files and directories accessible through a UNIX or Windows host via FCP or iSCSI.

3.1.3 Data Storage Containers

Files & Directories:

A file is the smallest unit of data management. Data ONTAP and application software create system-generated files, and users create data files. Users can also create directories in which to store files. Volumes, Qtrees and LUNs (with host support) can be used to store files and directories. File properties can be managed by managing the volume or qtree in which the file or its directory is stored.

3.1.4 Exporting the data

File Data:

NetApp storage exports data *as files* via two primary protocols, NFS and CIFS, corresponding to the UNIX and Windows way of doing things. UNIX and Windows actually have slightly different terminology, and very different file sharing protocols.

NFS:

UNIX uses NFS file sharing protocol. What gets exported on a UNIX server is called a filesystem, and is usually specified in the system file `/etc/fstab`. An `exportfs` command is used on the NetApp storage system to export the filesystem (volumes, qtrees, directories, files) to the hosts. A local host that wants to mount a exported filesystem uses a command like

```
# mount remotestorage:/usr/users /local_users
```

to mount the filesystem `/usr/users` on NetApp storage 'remotestorage' over the existing local directory `/local_users`. After this is done, the NFS file sharing protocol is used to get files from the storage system as they are accessed by the user who's logged into the local host.

CIFS:

Windows, on the other hand, uses the CIFS file sharing protocol (more on CIFS and NFS later). What gets exported on a Windows server is known as a share, and is usually specified via the GUI using the Sharing... menu item that pops up when you right click on a folder in the Explorer. A local host that wants to mount a share called 'users' on the NetApp storage 'remotestorage' and call it drive G: uses a command like

```
> net use G: \\remotestorage\users
```

This mounts 'remotestorage' share named 'users' on drive G: and all the files and subdirectories in the remote share 'users' appear as though they were local files in the G: drive. This drive is called a network drive. The CIFS protocol is used to get files and directory listings as they're accessed by the user logged into the local host.

Block Data:

NetApp storage system exports data as *blocks*, via FCP or iSCSI. You can configure the storage for Block access by creating LUNs.

3.2 Limits on Storage Containers

Storage Container	Max Limit	Comments
Disk	Depends on the NetApp Storage Model	
RAID Group	400 per storage system or cluster	
Plex	2 per aggregate	
Aggregate	100 per storage system	Traditional volumes are counted as aggregates
Volume	200 per storage system (FAS200) 500 per storage system (All other storage models)	Only 100 Traditional volumes can be present per storage system
Qtree	4,995 per volume	
LUN	1024 to 2048 per volume (depending on the storage model)	
Sub-Directories	Maximum of 99,998 subdirectories per directory	
Files	33,554,432 per volume	This limit can be increased using the 'maxfiles' command. Also this may change with the latest releases of ONTAP and NetApp storage models

3.3 Storage capacity concepts

3.3.1 Snapshot

A snapshot is a performance and space efficient, point-in-time image of the data in a volume or an aggregate. Snapshots are used for such purposes as backup and recovery.

3.3.2 Space reservation

Space reservation is a LUN/file attribute. Space reservation determines when space for the LUN/file is reserved or allocated from the volume. With reservations enabled (default for LUNs) the space is subtracted from the volume total when the LUN/file 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.

3.3.3 Fractional (Overwrite) reserves

Fractional reserve is a volume attribute. Fractional reserve determines how much space Data ONTAP will reserve for overwriting the data backed-up by the snapshots of LUNs and space-reserved files. The default value is 100%¹ but in practice this can be significantly lower when following other NetApp best practices. Data ONTAP removes or reserves the space for space-reserved LUNs and files from the volume as soon as the first snapshot copy is created.

3.3.4 Space guarantees

Space guarantee is a volume attribute applicable to FlexVol volumes. Guarantees on a FlexVol volume ensure that write operations to a specified FlexVol volume or write operations to LUNs or files with space reservation set do not fail because of lack of available space in the containing aggregate. Guarantees determine how the aggregate pre-allocates space to the FlexVol volume. There are three types of guarantees:

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. Fractional reserve is adjustable from the default of 100 percent only when a FlexVol volume has this type guarantee.

file - The aggregate guarantees that space is always available for overwrites to space-reserved LUNs or files. In this case, Fractional reserve for the volume is set to 100 percent and is not adjustable.

none - A FlexVol volume with a guarantee of none reserves no space, regardless of the space reservation settings for LUNs /files in that volume. Write operations to space-reserved LUNs/files in that volume might fail if its containing aggregate does not have enough available space. There are tools that can report these events to ensure that the writes don't fail due to lack of space.

3.3.5 Snapshot reserves

Snapshot reserve is set at the volume level and is set as a percentage of the volume. Data ONTAP reserves the defined percentage (20% by default, but it can be changed as per the user requirement) of volume from being available for configuring LUNs or for CIFS or NFS files to use. As Snapshot copies need space, they consume space from the snapshot reserve. By default after the snapshot reserve is filled, the Snapshot copies start to take space from the general volume.

3.3.6 Quotas

Quotas are used to restrict and track the disk space and number of files used by a user, group, or qtree.

¹ For Best Practises in setting the Fractional Reserve settings please refer the TR:
<http://www.netapp.com/library/tr/3483.pdf>

You specify a quota for the following reasons:

- To limit the amount of disk space or the number of files that can be used by a quota target

- To track the amount of disk space or the number of files used by a quota target, without imposing a limit

- To warn users when their disk space or file usage is high

A quota target can be

- A user, as represented by a UNIX ID or a Windows ID.

- A group, as represented by a UNIX group name or GID.

Note : Data ONTAP does not apply group quotas based on Windows IDs.

- A qtree, as represented by the path name to the qtree.

The quota target determines the quota type, as shown in the following table:

Quota target	Quota type
user	user quota
group	group quota
qtree	tree quota

3.3.7 WAFL Reserve

WAFL never let the file system get more than 90% full. This is done by pretending that the file system is 10% smaller than it actually is, so if you have a volume capable of storing 100GB, WAFL tells the outside world that there's only 90GB there: "df", filerview and Operations Manager all say 90; the free space is calculated by subtracting the used space from 90, not 100; any client operation that would result in there being more than 90GB in the volume is failed with an ENOSPACE error. This 10% space is called WAFL Reserve.

WAFL Reserve is required because the file system works better when it's no more than 90% full. The major advantage is that with one block in every ten always free, WAFL is guaranteed that it can find a free block when it needs one fairly quickly, and have more flexibility in how it places data to optimize the disk writes.

3.3.8 Raid Space

To support RAID some extra space is consumed. The amount of this space depends on the type of the RAID configured. For RAID4, there is one parity disk for each RAID group. So RAID space is the capacity of this parity disk. For RAID-DP, there are parity disks for each RAID group. So RAID space is the sum of the capacity of the two parity disks. If the aggregate is configured for Sync-Mirror, then the RAID space is the capacity of all the disks in the mirrored Plex of the aggregate.

3.4 Storage Layout

3.4.1 Disk Layout

The usable space on the disk is not equal to the raw capacity of the disk. The disk space is carved as described below:

20MB per disk for **kernel, boot block** etc., at the beginning of each disk.

Each drive is rightsized (as mentioned in section 3.1.1). For example, if the storage system has a 280GB drive from one vendor and a 290 GB drive from another one, then it rightsizes both of them down to 250GB and use the rest of the space at the end of the disk for things like core dump & raid labels.

10% of the rest of the space reserved as **WAFL reserve**.

Of the remaining space, 5% is used for the **Aggregate snapshot** (this is the default reserve and can be changed.)

Of the rest of the space, 80% is reserved for the active **file system**, 20% for **snapshots** (this is the default split and can be changed).

There is also the RAID **parity** space. The parity space refers to amount of data on the parity drive vs. the data drive in RAID-4 or RAID-DP RAID groups. It's dependent on the size of the RAID group (default is 16 : 14 data disks, 2 parity disks for a RAID-DP RAID group on a Fibre Channel based storage system).

For ATA drives, the storage system uses an "**8/9ths**" encoding scheme to emulate block checksums. Every 9th 512-byte disk sector is used to store the checksum for the previous 8 512-byte sectors (one 4K WAFL block). Thus the usable space on an ATA drive is 8/9ths of the stated capacity. But the significant cost difference of ATA compared to FC disks more than compensates for this extra overhead. FC disks have slightly larger sectors (520 bytes instead of 512). 8 sectors are enough for 4K block + checksum (64 bytes).

One more thing to consider in space calculations is the **definition of GB**. For disk vendors, 1GB = 1000 * 1000 * 1000 bytes. For Data ONTAP (when measuring disks), **1GB = 1000 * 1024 * 1024 bytes**. So for the FC disks, Data ONTAP reported sizes are ~95% of nominal disk sizes: $(1000 * 1000) / (1024 * 1024)$. For ATA disks, Data ONTAP reported sizes are ~85% of nominal disk sizes: $(8 * 1000 * 1000) / (9 * 1024 * 1024)$.

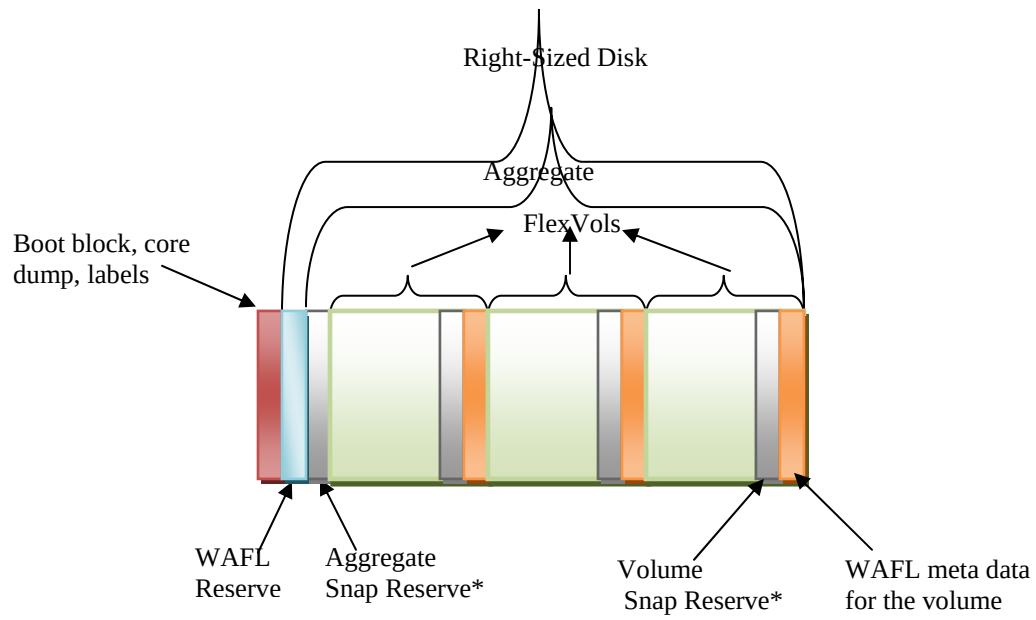
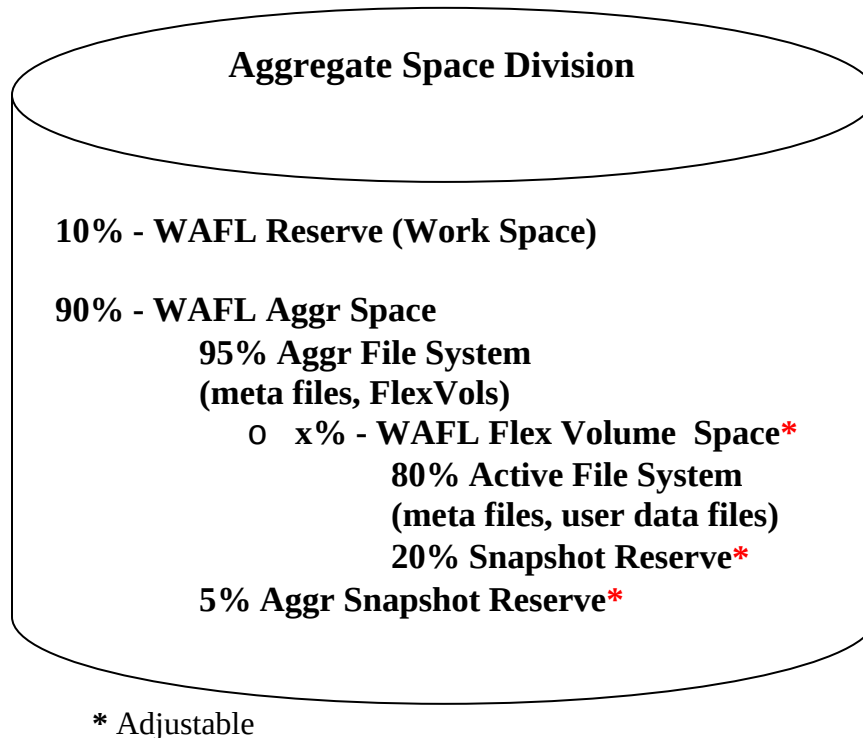


Fig: Disk Layout [* - Snap reserve space can be adjusted depending on the user requirements]

3.4.2 Aggregate Layout

Aggregate space is divided between the WAFL reserve, Aggregate snapshots, and the FlexVol allocations. The following figure depicts how the storage in the aggregate is laid out:



The aggregate snapshot reserve is by default 5% but it can be changed as per the user requirement. It can be set to '0%' also but it is recommended to be set as per the best practices of Data ONTAP storage management. For example, to change the aggregate reserve space from storage system CLI, use the command below:

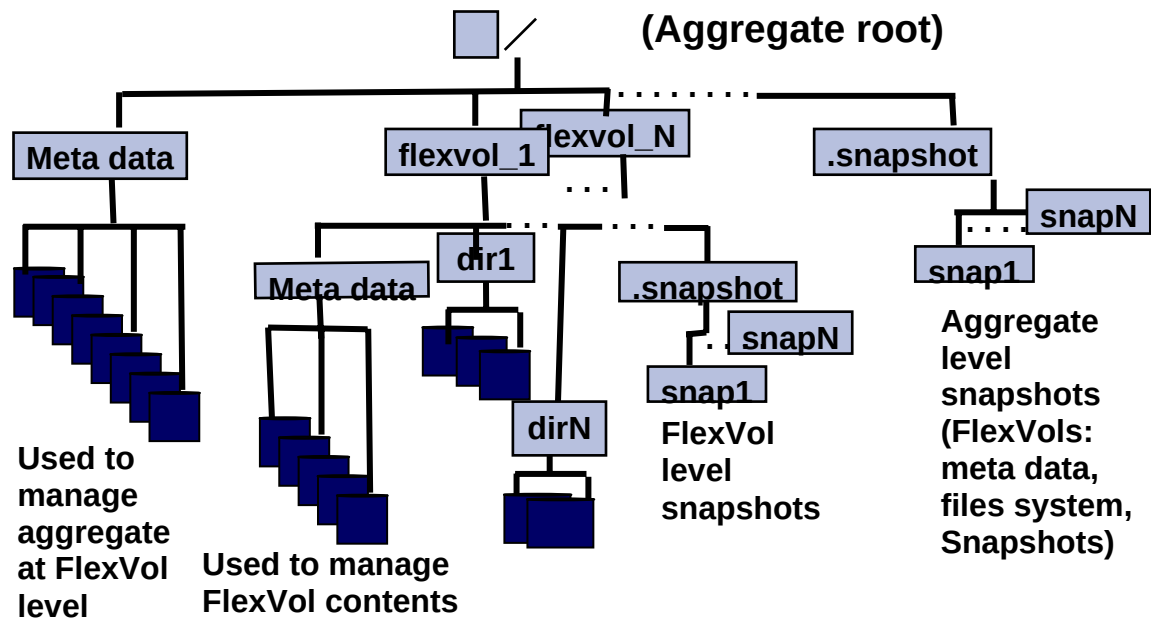
```
$snap reserve -A 'aggregate name' [percent]
```

Similarly the volume snapshot reserve can also be changed as per the user requirement. To change the volume snapshot reserve, use the command below:

```
$ snap reserve -V 'volume name' [percent]
```

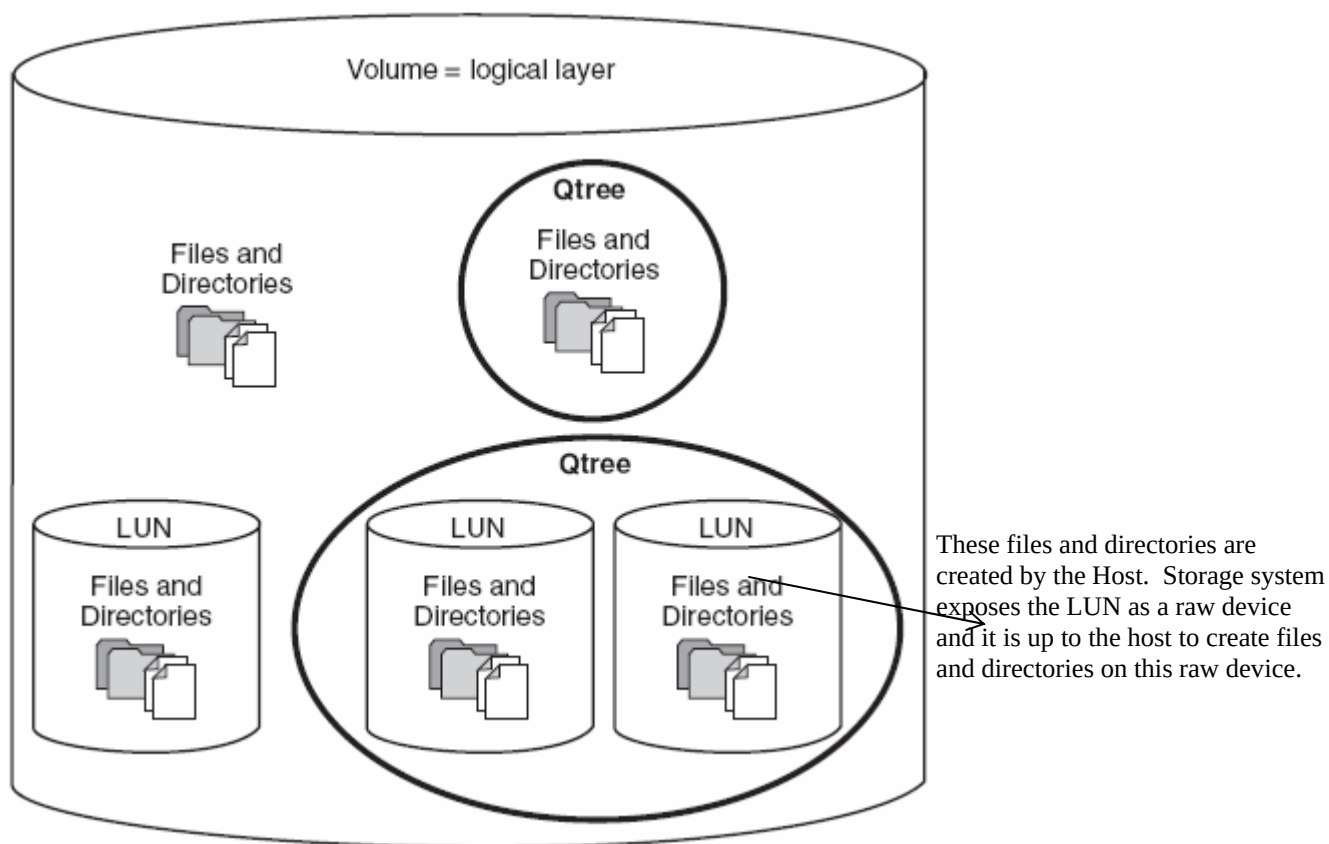
The following figure shows the different components of an aggregate.

Aggregate Infrastructure



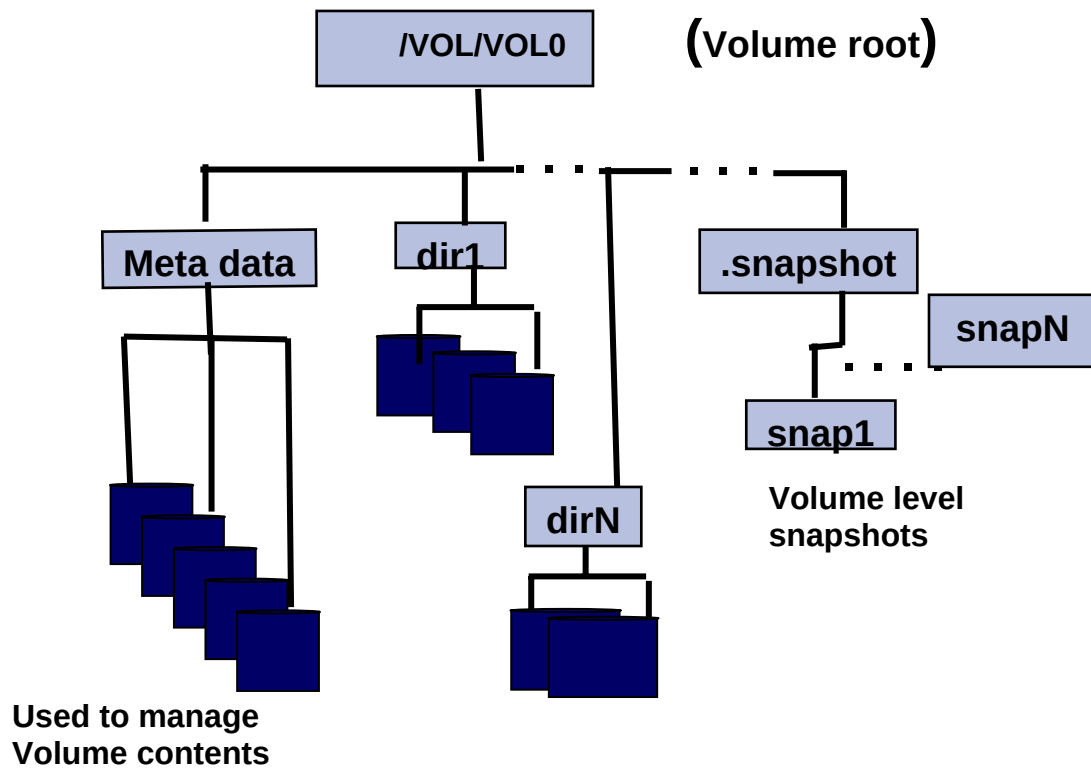
3.4.3 Volume Layout

The following figure depicts how the different logical storage entities are embedded inside a Volume.



The following figure shows the different components of a volume.

Volume Infrastructure



The metadata shown in the above figure corresponds to the file-system data like inodes.

4 Capacity calculation for storage system

4.1 Total Raw Capacity installed

The total installed capacity of the system is obtained by adding up the disk capacity of all the disks assigned to the storage system.

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the details of disks in the system	API	<i>disk-list-info</i>	API returns <i>disk-detail-info[]</i> array. The fields ' <i>raid-state</i> ' and ' <i>physical-space</i> ' will be used for calculating the RAW capacity.
	SNMP	<i>raidPNumber</i> (1.3.6.1.4.1.789.1.6.8.0)	This gives the number of elements in the raidPTable
		<i>raidPTotalMb</i> (1.3.6.1.4.1.789.1.6.10.1.22)	This gives the number of physically available Mbytes for the given disk drive
		<i>raidPStatus</i> (1.3.6.1.4.1.789.1.6.10.1.2)	This gives the status of the given disk drive

Steps to calculate the Total RAW capacity of the system:

API:

```
For each disk in the disk-detail-info[] array
    If(raid-state != "broken"){
        Total RAW Capacity += disk-detail-info-> physical-space
    }
```

SNMP:

```
For each disk in the raidPTable (raidPNumber gives total disks),
    If(raidPStatus != "failed (6)" and raidPStatus != "prefailed (9)") {
        Total RAW Capacity += raidPTotalMb
    }
```

4.2 Total Formatted (Right-Sized) Capacity

The total formatted capacity of the system is obtained by adding up the right-sized disk capacity of all the disks assigned to the storage system.

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the details of disks in the system	API	<i>disk-list-info</i>	API returns <i>disk-detail-info[]</i> array. The fields 'raid-state' and 'used-sapce' will be used for calculating the Right-sized capacity.
	SNMP	<i>raidPNumber</i> (1.3.6.1.4.1.789.1.6.8.0)	This gives the number of elements in the raidPTable
		<i>raidPUsedMb</i> (1.3.6.1.4.1.789.1.6.10.1.20)	This gives the number of right-sized Mbytes for the given disk drive
		<i>raidPStatus</i> (1.3.6.1.4.1.789.1.6.10.1.2)	This gives the status of the given disk drive

Steps to calculate the **Total Formatted (Right-sized) Capacity** of the system:

API:

```
For each disk in the disk-detail-info[] array
    If(raid-state != "broken"){
        Formatted Capacity += disk-detail-info->used-space
    }
```

SNMP:

```
For each disk in the raidPTable (raidPNumber gives total disks),
    If(raidPStatus != "failed (6)" and raidPStatus != "prefailed (9)") {
        Formatted Capacity += raidPUsedMb
    }
```

4.3 Total Spare Capacity

The total Spare capacity of the system is obtained by adding up the formatted capacity of all the spare disks in the storage system. The disks with the raid state as “spare”, “pending” or “reconstructing” are considered as spare disks.

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the list of disks in the system	API	<i>disk-list-info</i>	API returns <i>disk-detail-info[]</i> array. The fields ‘ <i>raid-state</i> ’, ‘ <i>physical-space</i> ’ and ‘ <i>used-space</i> ’ will be used for calculating the Spare capacity.
Get the details of the spare disks in the system	SNMP	<i>spareNumber</i> (1.3.6.1.4.1.789.1.6.6.0)	This gives the number of elements in the spareTable
		<i>spareTotalMb</i> (1.3.6.1.4.1.789.1.6.3.1.7)	This gives the number of physically available Mbytes for the given spare disk drive (NOTE: The correct size to be used for this is the right-sized capacity of the spare disk, but that details is not available in the spare disk Table.)
		<i>spareStatus</i> (1.3.6.1.4.1.789.1.6.3.1.3)	This gives the status of the given spare disk drive

Steps to calculate the **Total Spare Capacity** of the system:

API:

```
For each disk in the disk-detail-info[] array
    If(raid-state == “spare” or
       raid-state == “pending” or
       raid-state == “reconstructing”){
        Spare Capacity += disk-detail-info->used-space
    }
```

SNMP:

```
For each disk in the spareTable (spareNumber gives total spare disks),
    If(spareStatus != “unknown (4)”) {
        Spare Capacity += spareTotalMb
    }
```

4.4 Total Capacity in RAID Space

The total capacity of the system in RAID Space is obtained by adding up the formatted capacity of all the parity disks and the capacity of the all the disks in the Sync plexes.

Steps for getting the system RAID Space

1. Get the details of all the aggregates in the system
2. Get the capacity of all the parity disks and the capacity of the plex if the aggregate is sync mirrored
3. Add the capacity of the parity disks and the sync mirror plex

Raid Space of the system = Capacity of (Parity disks + Sync mirror plex disks) for all the aggregates in the system

Calculating the **Total RAID Space** of the system using API:

Operation	Open Interface	Command Details (API for ONTAPI)	Comments
Get the list of aggregates	API	<i>aggr-list-info</i>	This API returns the array <i>aggr-info[]</i> which contains the details of all the aggregates in the system
Get the details of the given disk	API	<i>disk-list-info disk</i>	Set the input parameter ' <i>disk</i> ' to the name of the disk for which the details needs to be got.

API:

For each aggregate in the *aggr-info[]* array

1. Get all the disks in the aggregate
aggr-info->plex[]->raid-group[]->disk[]
2. For each of the disk, get the disk details
disk-list-info disk-name
3. Check the raid type of the disk
If (raid-type == "parity" or raid-type == "dparity") {
RaidSpace += disk-detail-info-> used-space
}
4. Check if the aggregate is sync-mirrored
If (aggr-info->plex-count > 1) -> implies the aggregate is sync-mirrored
{ / Get all the disks in one of the plex of the aggregate */*
aggr-info->plex-info[]->raid-group-info[]->disk-info[]

/ For each of the disk, get the disk details */*
disk-list-info disk-name

```

/* Add up the capacity of all the disks in this plex */

    RaidSpace += disk-detail-info->used-space
}

```

Calculating the **Total RAID Space** of the system using SNMP:

Operation	Open Interface	Command Details (OID for SNMP)	Comments
Get the details of the disks	SNMP	<i>raidPNumber</i> (1.3.6.1.4.1.789.1.6.8.0)	This gives the number of elements in the raidPTable
		<i>raidPUsedMb</i> (1.3.6.1.4.1.789.1.6.10.1.20)	This gives the number of right-sized Mbytes for the given disk drive
		<i>raidPDiskName</i> (1.3.6.1.4.1.789.1.6.10.1.10)	This gives the name of the given disk drive
Get the details of the number of plexes	SNMP	<i>raidPPlexNumber</i> (1.3.6.1.4.1.789.1.6.10.1.6)	This gives the number of plexes in the given aggregate

SNMP:

For each disk in the raidPTable (*raidPNumber* gives total disks),

```

    If(strstr(raidPDiskName,"parity") != NULL) {
        Total RAID Space += raidPUsedMb
    }

```

For each aggregate in the raidPTable

```

    If(raidPPlexNumber == 2) {
        For each disk in one of the plex of the corresponding aggregate {
            Total RAID Space += raidPUsedMb
        }
    }

```

4.5 Total Capacity in WAFL Reserve

Each aggregate has 10% of the space reserved for the WAFL reserve. This space is used to as a reserve for better file system performance.

Steps for getting the system WAFL Reserve

1. Get all the aggregates in the system
 2. For each aggregate calculate its capacity
- The system WAFL Reserve = capacity of all the aggregates * 0.1

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the list of aggregates	API	<i>aggr-list-info</i>	This API returns the array <i>aggr-info[]</i> which contains the details of all the aggregates in the system
Get the list of disks	SNMP	<i>raidPNumber</i> (1.3.6.1.4.1.789.1.6.8.0)	This gives the number of elements in the raidPTable
		<i>raidPUsedMb</i> (1.3.6.1.4.1.789.1.6.10.1.20)	This gives the number of right-sized Mbytes for the given disk drive
		<i>raidPStatus</i> (1.3.6.1.4.1.789.1.6.10.1.2)	This gives the status of the given disk drive

API:

For each aggregate in the *aggr-info[]* array

- a. Get all the disks in the aggregate
aggr-info->plex[]->raid-group[]->disk[]
- b. For each of the disk, get the disk details
disk-list-info disk-name
- c. Add the capacity of all the disks in the aggregate

```

If(raid-state != "broken"){
    Aggregate size += disk-detail-info->used-space
}

```

WAFL Reserve = Sum of the "Aggregate size" for all the aggregates in the system * 0.1

SNMP:

For each disk in the raidPTable (raidPNumber gives total disks),

```

    If(raidPStatus != "failed (6)" and raidPStatus != "prefailed (9)") {
        Formatted Capacity += raidPUsedMb
    }
    WAFL Reserve = Formatted capacity * 0.1

```

4.6 Total Capacity in Reserved space

The total reserve is calculated by adding up all the snapshot reserves and the Overwrite reserves for all the aggregates and the volumes.

a. Calculating the snapshot space for the aggregates

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the space usage details of all the aggregates	API	<i>aggr-space-list-info</i>	This API returns the array <i>aggr-space-info[]</i> which contains the space information details of all the aggregates in the system. The field " <i>size-snap-used</i> " gives the space used by the aggregates snapshots Note: <i>aggr-space-list-info</i> API is available only from ONTAP 7.1 onwards
Get the snapshot space details of all the aggregates	SNMP	<i>dfFileSys</i> (1.3.6.1.4.1.789.1.5.4.1.2)	This gives the name of the container (check for <i>"/.snapshot"</i> in the container name for getting snapshot related data)
		<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
		<i>dfKBytesTotal</i> (1.3.6.1.4.1.789.1.5.4.1.3)	This gives the total bytes reserved for the container
		<i>dfKBytesUsed</i> (1.3.6.1.4.1.789.1.5.4.1.4)	This gives the total bytes used by the container

API:

Total snapshot space reserved in the aggregate = Greater of (snapshot reserve of the aggregate, size of the snapshot used space)

The API does not provide the snapshot reserve details for the aggregate, so we can take the snapshot usage as an approximation for the snapshot reserve.

Total snapshot space reserved for all the aggregate = Sum of *aggr-space-info[]* -> *size-snap-used* for all the aggregates

SNMP:

For each container in the dfTable

```
If the container type is an aggregate (dfType == 3) {
  Get the snapshot of the container ( strstr (dfFileSys, “/.snapshot”) != NULL) {
    Snapshot space reserved in the aggregate = Greater of (dfKBytesTotal,
dfKBytesUsed)
  }
}
```

Total snapshot space reserved for all the aggregates = Sum of the Snapshot space reserved in each aggregate

b. Calculating the snapshot space for the volumes

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the list of all the volumes in the system	API	<i>volume-list-info</i>	This API returns the array <i>volume-info[]</i> which contains the details of all the volumes on the system
	SNMP	<i>dfFileSys</i> (1.3.6.1.4.1.789.1.5.4.1.2)	This gives the name of the container (check for “/.snapshot” in the container name for getting snapshot related data)
		<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
Get the snapshot reserve detail for each volume	API	<i>Snapshot-get-reserve ‘volume’</i>	This API returns the ‘ <i>blocks-reserved</i> ’ which provides the number of 1024 byte blocks that has been set aside as reserve for snapshot usage in the volume. From ONTAP 7.3 onwards this value can also be got from <i>volume-info</i> structure returned by the ‘ <i>volume-list-info</i> ’ from the field ‘ <i>snapshot-blocks-reserved</i> ’
	SNMP	<i>dfKBytesTotal</i> (1.3.6.1.4.1.789.1.5.4.1.3)	This gives the total bytes reserved for the container
Get the	API	<i>snapshot-list-info ‘volume’</i>	This API returns the array

snapshot usage details for each volume			<i>snapshot-info[]</i> which contains the details of all the snapshots of the given volume. The field “ <i>cumulative-total</i> ” of the last snapshot in this array gives the total space of volume used by all the snapshots in the volume in 1024 byte blocks.
	SNMP	<i>dfKBytesUsed</i> (1.3.6.1.4.1.789.1.5.4.1.4)	This gives the total bytes used by the container

API:

1. Get the list of the all the volumes in the storage system
 - i. *volume-list-info*
2. For each volume, get the snapshot details
 - i. *snapshot-list-info 'volume'*
Walk through *snapshot-info[]* array and for the last element, extract the field “*cumulative-total*”
Snapshot space used in the volume = *cumulative-total*
Snapshot bytes in the volume = Larger of the (Snapshot space used in the volume, Snapshot blocks reserved)*1024 bytes
3. Total snapshot space used in all the volumes = Sum of the ‘*Snapshot bytes in the volume*’ for all the volumes

SNMP:

For each container in the dfTable

```
If the container type is a FlexVol or Traditional Volume(dfType = 2 || dfType = 1) {
  Get the snapshot of the container ( strstr (dfFileSys, “/.snapshot”) != NULL) {
    Snapshot space reserved in the volume = Greater of (dfKBytesTotal,
    dfKBytesUsed)
  }
}
```

Total snapshot space reserved for all the volumes = Sum of the Snapshot space reserved in each volume

- c. Calculating the Overwrite reserve space for the volumes

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the details of the over-write	API	<i>volume-list-info</i>	This API returns the array <i>volume-info[]</i> which contains the details of all the volumes on the

reserve of all the volumes in the system			system. The fields ‘ <i>reserve</i> ’ and ‘ <i>reserve-used</i> ’ of the volume-info structure are used for calculating the overwrite reserve of the volume
	SNMP	Not Available	

API:

1. Get the list of the all the volumes in the storage system
 - i. *volume-list-info*
2. For each volume, get the overwrite reserve as below:
 - i. Overwrite reserve bytes of the volume = larger of (*reserve*, *reserve-used*)
3. Total overwrite reserve space in all the volumes = Sum of the “Overwrite reserve bytes of the volume” for all the volumes

Total capacity in the reserve space = Total snapshot space used in all the aggregates + Total snapshot space used in all the volumes + Total overwrite reserve space in all the volumes

4.7 Total Capacity Usable for Provisioning

The total space useable for provisioning is the amount of the space in the aggregate after deducting the WAFL Reserve and aggregate Snapshot reserve space.

Calculating the useable space for provisioning

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the list of aggregates	API	<i>aggr-list-info</i>	This API returns the array <i>aggr-info[]</i> which contains the details of all the aggregates in the system. The field ‘ <i>size-total</i> ’ of the <i>aggr-info</i> structure is used to calculate the usable space for provisioning.
	SNMP	<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
		<i>dfKBytesTotal</i> (1.3.6.1.4.1.789.1.5.4.1.3)	This gives the total bytes reserved for the container, for aggregates this is capacity of the aggregate after excluding the WAFL reserve and the snapshot reserve

Aggregate's usable space for provisioning
 = Aggregate's total space – WAFL Reserve – Snapshot reserve space

API:

For each aggregate in the *aggr-info[]* array
 Total capacity of the aggregate = *aggr-info[]*->'size-total'

SNMP:

For each container in the dfTable
 If the container type is an aggregate (dfType == 3) {
 Get the total capacity of the container
 Total capacity of the aggregate = dfKBytesTotal
 }

Total capacity usable for provisioning = Sum of the Total capacity of all the aggregates in the system

4.8 Total Capacity Allocated

This is the sum of the space reserved for the volume and the space used by non reserved data. For volume guaranteed volumes, this is the size of the volume, since no data is unreserved. For volumes with space guarantee of none, this is the used space of the volume since no unused space is reserved. The Allocated space value is the amount of space that was requested by the user but not the total space actually being used by the volume.

Calculating the total space allocated

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the space information list of all the aggregates	API	<i>aggr-space-list-info</i>	This API returns the array <i>aggr-space-info[]</i> which contains the space usage details of all the aggregates in the system. The field 'size-volume-allocated' of the <i>aggr-space-info</i> structure is used to calculate the total space allocated. Note: <i>aggr-space-list-info</i> API is available only from ONTAP 7.1 onwards

	SNMP	<i>Not Available</i>	

Aggregate's space allocated

= space reserved for the volume + the space used by non reserved data.

= *aggr-space-info->size-volume-allocated*

For each aggregate in the *aggr-space-info[]* array

i. Get the field '*size-volume-allocated*'

Total capacity allocated = Sum of the '*size-volume-allocated*' field for the all the aggregates

4.9 Total Capacity of User usable data

This is the amount of space that is taking up disk blocks used to store the data that is usable by the user. This value includes the space used in the volumes created by the user and the space consumed by the aggregate spanshots. It excludes all the data that is used by the system viz., WAFL reserve and the metadata required to maintain the flexible volume.

Calculating the total user used space

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the space usage details of all the aggregates	API	<i>aggr-space-list-info</i>	This API returns the array <i>aggr-space-info[]</i> which contains the space information details of all the aggregates in the system. The field “ <i>size-snap-used</i> ” gives the space used by the aggregate snapshots and the field ‘ <i>size-volume-used</i> ’ gives the space used by the volumes. Note: <i>aggr-space-list-info</i> API is available only from ONTAP 7.1 onwards
	SNMP	<i>dfFileSys</i> (1.3.6.1.4.1.789.1.5.4.1.2)	This gives the name of the container (check for “ <i>/.snapshot</i> ” in the container name for getting snapshot related data)
		<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
Get the snapshot reserve detail for each volume	API	<i>Snapshot-get-reserve ‘volume’</i>	<i>This API returns the ‘blocks-reserved’ which provides the number of 1024 byte blocks that has been set aside as reserve for snapshot usage in the volume. From ONTAP 7.3 onwards this value can also be got from volume-info structure returned by the ‘volume-list-info’ from the field ‘snapshot-blocks-reserved’</i>
	SNMP	<i>dfKBytesTotal</i> (1.3.6.1.4.1.789.1.5.4.1.3)	This gives the total bytes reserved for the container

Get the snapshot details for each volume	API	<i>snapshot-list-info 'volume'</i>	This API returns the array <i>snapshot-info[]</i> which contains the details of all the snapshots of the given volume. The field “ <i>cumulative-total</i> ” of the last snapshot in this array gives the total space of volume used by all the snapshots in the volume in 1024 byte blocks.
	SNMP	<i>dfKBytesUsed</i> (1.3.6.1.4.1.789.1.5.4.1.4)	This gives the total bytes used by the container

Aggregate’s space used for user usable data

= the space used by aggregate snapshot data + space used in each volume of the aggregate + space used by the snapshots in each volume – space used by the snapshots beyond the snapshot reserve of the volume (this is required because the size-used for the volume also includes any snapshot space used by the snapshots beyond the snapshot reserve)

API:

- i. Issue the command *aggr-space-list-info* to get the space details of all the aggregates in the system
- ii. For each aggregate in the *aggr-space-info[]* array
 - a. Get the field ‘*size-volume-used*’
 - b. Get the field ‘*size-snap-used*’
- iii. For each volume in the *aggr-space-info[]* array, get the snapshot details as below
 - a. Issue Snapshot-get-reserve command for the volume and extract the field ‘*blocks-reserved*’. This indicates the amount of space reserved for the snapshot usage in the volume
 - b. Issue snapshot-list-info command for the volume and extract the field ‘*cumulative-total*’ for the last snapshot listed in the command. This indicates the actual snapshot space used by all the snapshots in the volume
 - c. Snapshot space of the volume

= ‘*blocks-reserved*’ (If *cumulative-total* > *blocks-reserved*, then the amount of extra space consumed by the snapshots beyond the snapshot reserve is already accounted in the ‘*size-volume-used*’ calculation.

= *cumulative-total* (if *cumulative-total* <= *blocks-reserved*)

Total capacity of User usable data

= Sum of [*aggr-space-info->size-snap-used* +
aggr-space-info->size-volume-used +

Sum of (Snapshot space of the volume) for all the volumes in the aggregate)] for all the aggregates in the systems

SNMP:

For each container in the dfTable

```
If the container type is an aggregate (dfType == 3) {  
  Get the snapshot of the container ( strstr (dfFileSys, “/.snapshot”) != NULL) {  
    Snapshot space used in the aggregate = dfKBytesUsed  
  }  
}
```

Total snapshot space used for all the aggregates = Sum of the Snapshot space used in each aggregate

For each container in the dfTable

```
If the container type is a FlexVol or Traditional Volume(dfType = 2 || dfType = 1) {  
  Space used in the volume = dfKBytesUsed  
  Get the snapshot space used in the volume (strstr (dfFileSys, “/.snapshot”) != NULL)  
  {  
    Snapshot space used in the volume = dfKBytesUsed  
  }  
  Get the snapshot reserve of the volume ( strstr (dfFileSys, “/.snapshot”) != NULL) {  
    Snapshot space reserved in the volume = dfKBytesTotal  
  }  
  Snapshot space of the volume  
    = dfKBytesTotal (if snapshot space used is greater than snapshot  
    reserve, because the extra space used by the snapshots is already  
    accounted while calculating the space-used value of the volume.)  
    = dfKBytesUsed (if snapshot space used is less than snapshot  
    reserve)
```

Total space used for all the volumes = Sum of the snapshot space of each volume +
Space used by each of the volume

Total capacity of User usable data = Total snapshot space used for all the aggregate +
Space used by all the volumes + Space used by the snapshots of all the volumes

4.10 Total Capacity Available

This is the space that is available in the system for the user. The space available could be in one of the following forms:

- a. Disk space available as spare disks which are not assigned to any aggregate
- b. Amount of disk space available in the aggregates which is not provisioned for any volume yet
- c. Amount of space available in the volumes which is not used for any user data yet

Available Capacity calculations:

- a. Calculating total spare disk capacity
– Refer Section 4.3
- b. Calculating total capacity available for provisioning:

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the list of aggregates in the storage system	API	<i>aggr-list-info</i>	This API returns the array <i>aggr-info[]</i> which contains the details of all the aggregates in the system. The field ‘ <i>size-available</i> ’ of the <i>aggr-info</i> structure is used to calculate the available space for provisioning.
	SNMP	<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
		<i>dfKBytesAvail</i> (1.3.6.1.4.1.789.1.5.4.1.5)	This gives the total bytes available in the container

API:

1. Issue the command *aggr-list-info* to get the details of all the aggregates in the system
2. For each aggregate in the *aggr-info[]* array
Total capacity available in the aggregate = *aggr-info[]* -> ‘*size-available*’

SNMP:

For each container in the *dfTable*

```
If the container type is an aggregate (dfType == 3) {  
  Get the total capacity available in the container  
  Total capacity available in the aggregate = dfKBytesAvail }  
}
```

Total capacity available for provisioning = Sum of the space available in all the aggregates of the system

c. Calculating total capacity available for user data:

Operation	Open Interface	Command Details (API for ONTAPI, OID for SNMP)	Comments
Get the details of the space available in the all the volumes in the system	API	<i>volume-list-info</i>	This API returns the array <i>volume-info[]</i> which contains the details of all the volumes on the system. The fields ‘ <i>size-available</i> ’ volume-info structure is used for calculating the available space in the volume for user data
	SNMP	<i>dfType</i> (1.3.6.1.4.1.789.1.5.4.1.23)	This gives the type of the container
		<i>dfKBytesAvail</i> (1.3.6.1.4.1.789.1.5.4.1.5)	This gives the total bytes available in the container

API:

1. Issue the command *volume-list-info* to get the details of all the volumes in the system
2. For each volume in the *volume-info[]* array
Total capacity available = *volume-info[]*->‘*size-available*’

SNMP:

For each container in the dfTable

If the container type is a FlexVol or Traditional Volume (dfType = 2 || dfType = 1) {
Get the total capacity available in the volume
Total capacity available in the volume = dfKBytesAvail }

Total capacity available for user data = Sum of the available space in all the volumes of the system