NetApp Snapshot Technology, when does a snapshot grow?

Source : http://blog.hernanjlarrea.com.ar/index.php/netapp-snapshot-technology-when-does-a-snapshot-grow/

by Hernán J. Larrea

Let's take a quick but deeper view on how does snapshots work and behave, the concept is simple save volume data as it was in certain point in time, but actually it is not so simple or it has some secrets. NetApp snapshot technology is famous because the ability it has to preserve data by using little space and it is the key to many of the features which make NetApp such a great storage technology such as, snapmirror, flex clones, snap vault, deduplication and many others.

Understanding the basics

First of all take a look at the environment I've set up to run the tests and some concepts we need to understand:

Environment

A simulated NetApp filer is exporting via NFS a volume to a server running Linux operating system. This server will create, modify and delete files in the NFS export while it is being monitored in the filer side, on file system utilization, snapshots creation and growth rate.

Tools

Let's see which tools can we use to monitor snapshots behavior:

- (CLI) snap delta: this command displays the rate of change of data between snapshots. (**TIP:** I have found particularly useful this command to determine how much data has changed between last snapmirror snapshot taken and the AFS (or new snapmirror snapshot if transfer is is progress), this way you can determine how much data would be transferred if you update the snapmirror relationship).
- (CLI) snap reclaimable: this command allows you to determine how much space would be released from a certain volume after deleting the given names of the snapshots.
- (Filer View) Volumes –> Snapshots –> Manage: in this part of Filer View (which is particularly useful) you will be able to determine how much does a snapshot occupy (haven't found a way yet to do this through the CLI) and you can calculate the reclaimable space as you can do using *snap reclaimable* CLI command.

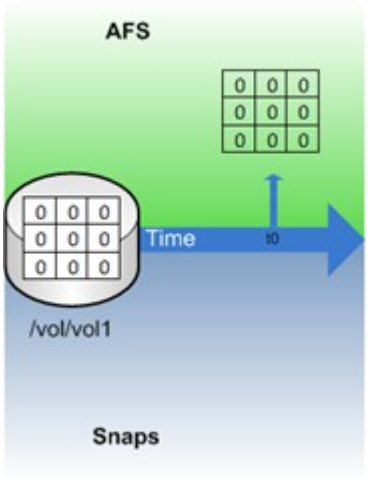
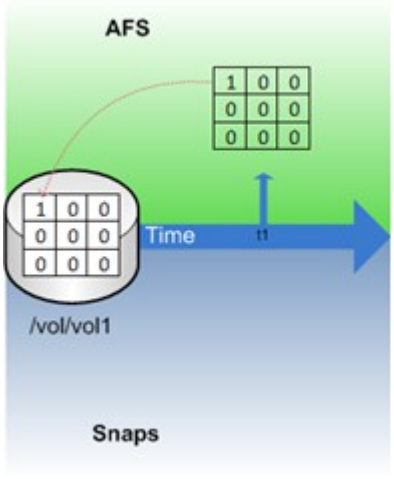
Concepts

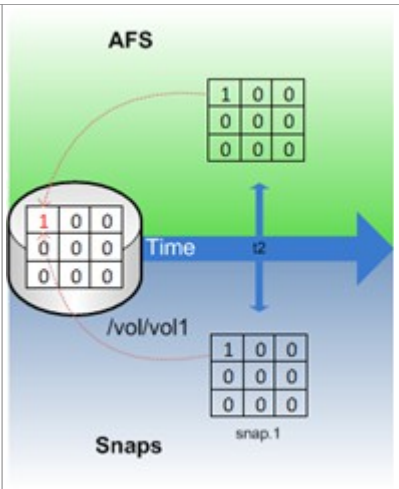
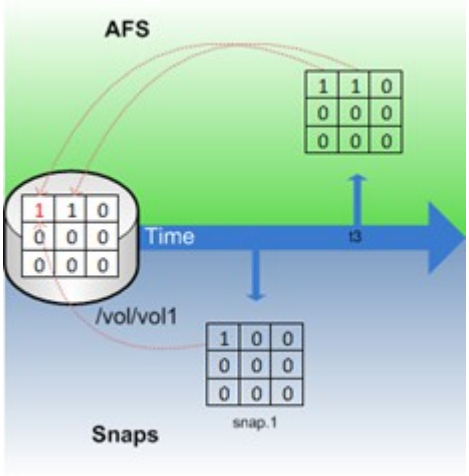
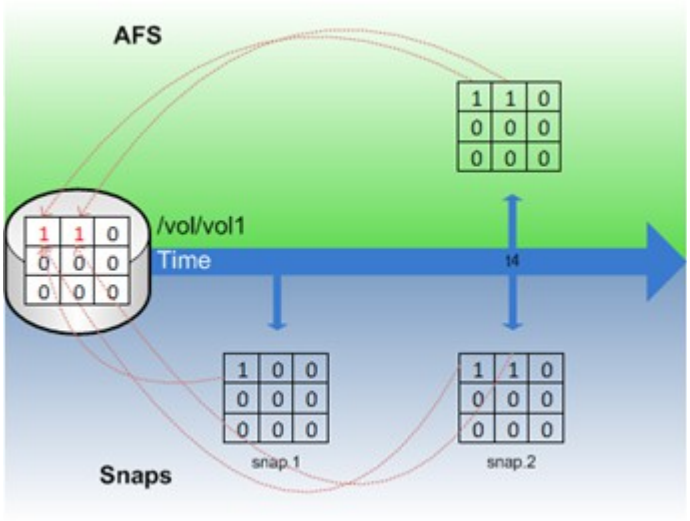
Active File System or AFS, it's a concept very similar to Snapshot, it is a table, but this one points to the production data blocks within a volume being presented to the server. In other words a server accessing a volume (CIFS, NFS export or LUN, there is no difference) is only aware of the blocks pointed by the AFS table as the data available on the volume.

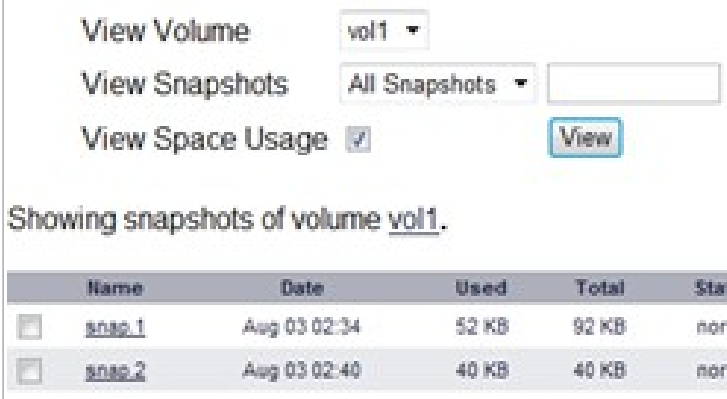
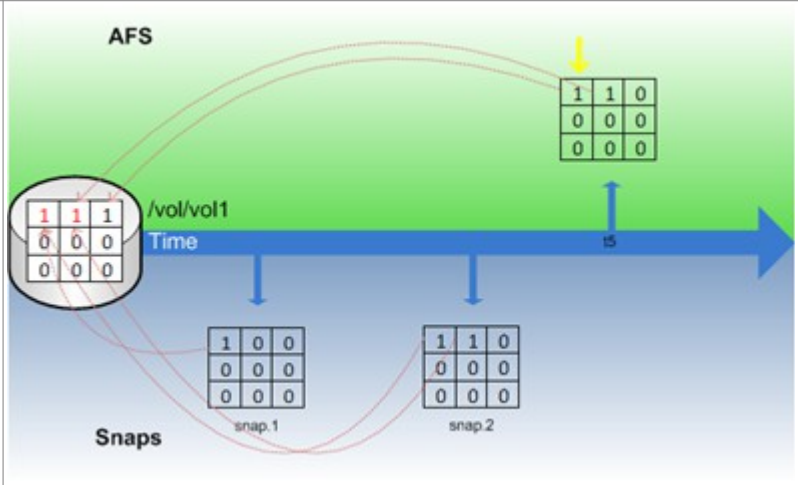
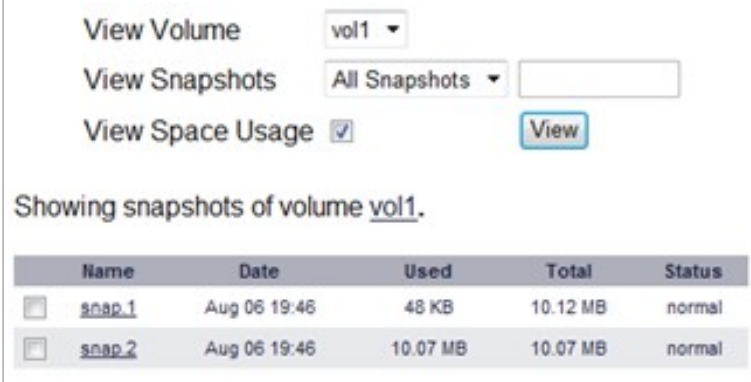
Watching snapshots in action

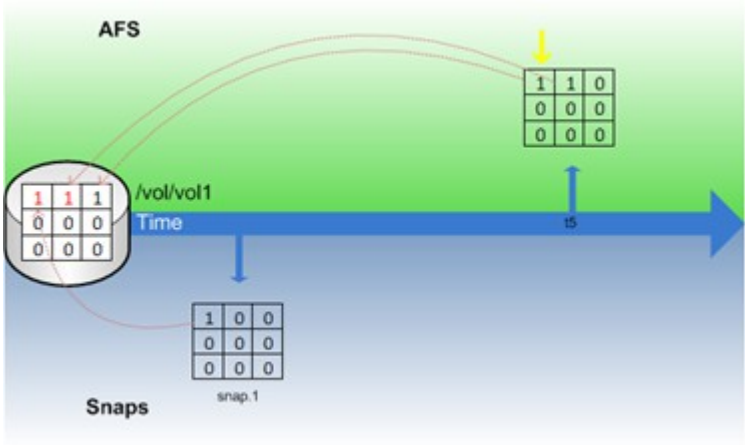
And now let's take a look at the following time line to understand how snaps behave:

Overwriting a file

Step	Description
t0	 The initial state, volume empty (500MB), exported and mounted in the server (AFS stands for Active File System, the table which point to the data blocks being presented to the server).
t1	 A file named *file1* (10MB) is created.

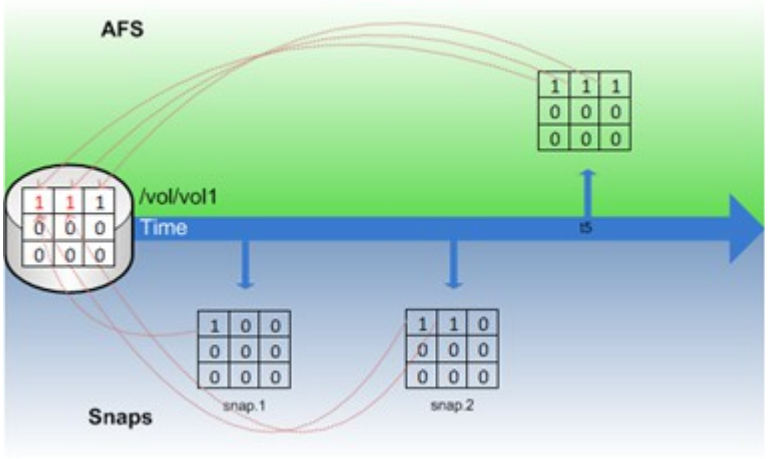
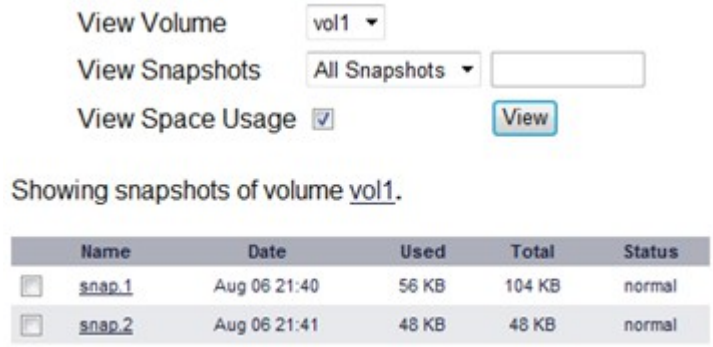
t2	 A snapshot named *snap.1* is created. As you can see in the drawing the snap holds pointers to the data blocks the AFS was pointing at the time the snap was taken.
t3	 A new file named *file2* (10MB) is created.
t4	 A new snapshot named *snap.2* is created. Now the blocks used by the

	new file also get locked by the new snap. Check out Filer View, as we didn't modify any files snaps have their original size: 
t5	 *file1* is fully overwritten by 10MB bunch of data. Lets take a new look to Filer View to see snaps sizes.  As you can see, the most recent snapshot has grew, this is quite a logical event as the file which was 10MB has been overwritten by another 10MB of data, and you need to be able to go back to the time before the overwrite, those original 10MB of data must be stored somewhere, and

	that somewhere is the snapshot area of the volume  .
t6	 Now let's delete the most recent snapshot (*snap.2*), and let's take a look at the snapshots size:  As you can see, the snapshot *snap.1* which previously was only couple of KBs now it is 10MB size, why is that? Well, it is because snapshots tend to merge, *snap.2* was 10MB, in other words it had 10MB of data blocks locked by it, but between *snap.1* and *snap.2* there was no modifications to *file1*, so when we overwritten the file in t5, *snap.2* saved the data for *file1*, but *snap.1* should be also able to restore to the same state (remember between *snap.1* and *snap.2, file1* didn't suffer any changes) so now *snap.2* is not any more here, the 10MB of original data from *file1* must be saved by some one in order to ensure the ability to restore to that point in time. This is called merge, and it is a cool way to save space. In other words, when we deleted the snapshot *snap.2*, the following one (*snap.1*) toke over those data blocks which were from interest to him. I know this concept might sound estrange and little hard to understand, but play a little with your volumes and snapshots, and if you can do some diagrams.

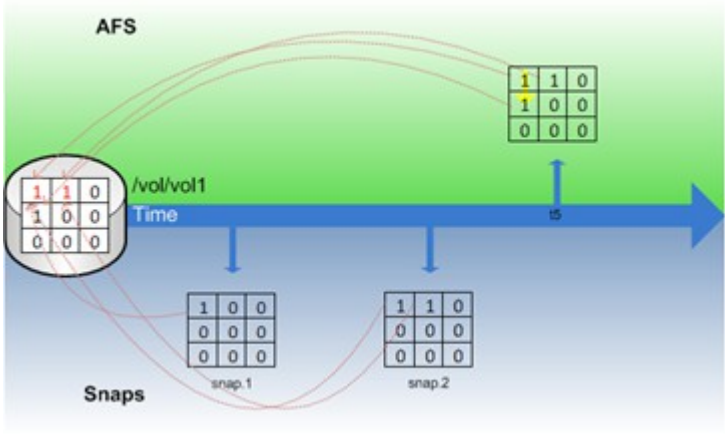
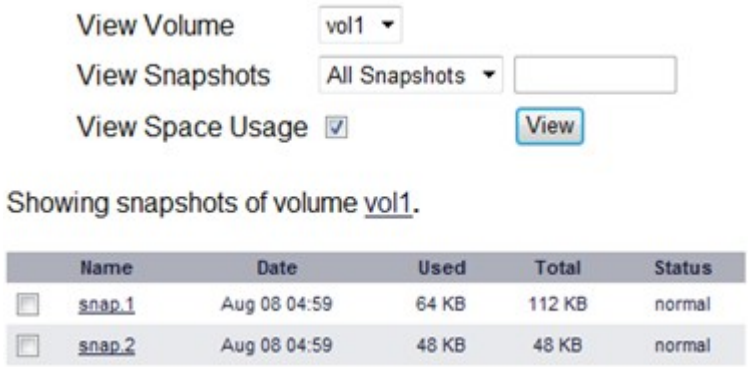
Adding a new file

So until this moment we have seen, a snapshot will grow when it is the last snapshot taken (most recent) and when an already existing file at the moment of the snapshot take is modified, and when it is the second newest snap and the newest one get's deleted. Let's now go back to the time line, back to t5 (now t5'), but now will create a new file instead of modifying an existing one:

Step	Description
t5'	 A new file named *file3* (10MB) is created. Let's take a look at the snaps sizes:  As you can see, none of them grew, this is because the files, actually the blocks used by the file we just created are not locked by any snapshot.

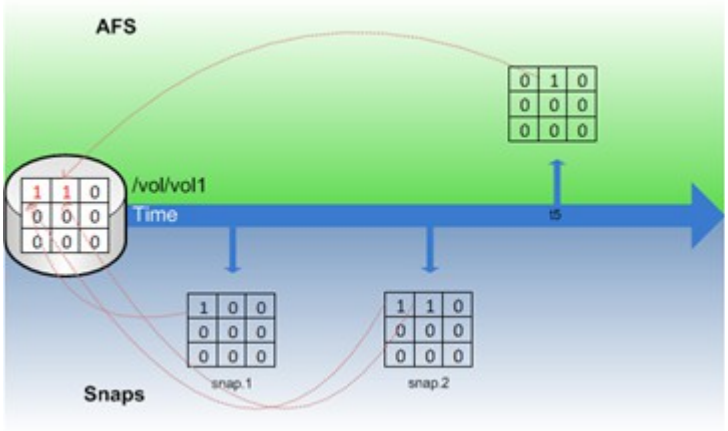
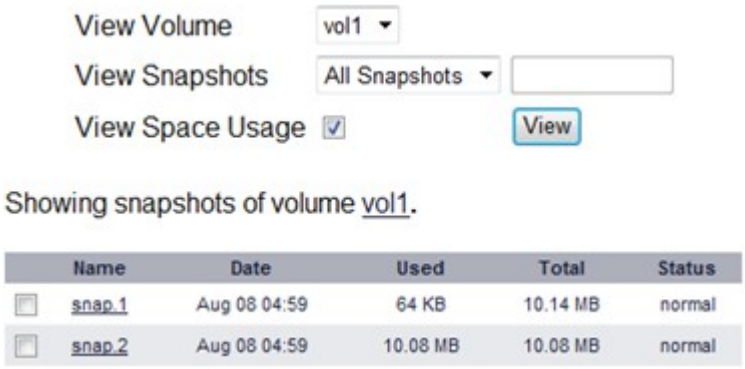
Appending data to an already existing file

And now let's look what happens when you append data to files in a volume that is being snapped:

Step	Description
t5"	 10MB bunch of data is appended to *file1*. Let's see what happens to the snapshot sizes:  As you can see, none of the has grew since new blocks are being added but not locked by any snap.

Deleting files

Now let's go to t5 (now t5‴) and let's see what happens when you try to delete a file. I've used the case of t5‴

Step	Description
t5‴	 After appending 10MB bunch of data to *file1* in t5‴ (now *file1* is 20MB size). We deleted *file1,* Let's see what happens to the snapshot sizes:  As you can see, the most recent snapshot has grew, but only in 10MB. This is because only 10MB of *file1* were protected by snapshots, the additional 10MB were added before any other snapshot was taken, so now, if you try to restore from *snap.2* or *snap.1* you could only take *file1* to the status previous to the appending of 10MB of data.

Conclusion

At this point we have seen what happens when you, create a new file, overwrite a file, append data to a file, and delete files in a volume being snapped. So in conclusion, we can say:

A snapshot will grow when…	**A snapshot won't grow when…**
• It is the most recent snap and the files (actually data blocks) existing at the moment of the snapshot take are modified. • When a snap is deleted the next one will take over the data blocks the deleted one has released and are from interest to him, this will make the next snapshot to grow. • When files are deleted, the most recent snapshot will grow since this one will take over the data blocks it is locking.	• A new file is added • When appending data to a file, no matter if the file (actually it's blocks) were already locked by a snap.

Always remember a volume has space to store data and to store snapshots, an Active File System and a snapshot area. And you must try to find equilibrium between these two, which is actually not an easy task to achieve, specially in big volumes, when snapshots are involved, many things must be taken in considerations. Imagine you need to release space from a CIFS volume which is being snapped and you don't want to extend the volume, it sounds really simple, require users to delete files, but remember when you try to delete files your snaps will grow, so actually you will see no space is actually being released even though your users are deleting. Here you will find some tools or features of volumes, such as snap reserve, which will help you to solve this, and eventually if you need to delete old snaps to release some space, remember it is a good practice to start deleting snapshots from the oldest ones and always to use snap reclaimable command in order to determine how much space are you actually releasing.

Again, I know this isn't easy to understand in the first reading, I know it might require couple of re reads to understand, it wasn't easy to write it either, but it has really helped me to gain much deeper understanding on how snapshots behave, hope you can enjoy it too  .

Hernán J. Larrea