How do Virtual Machine Snapshots work in VMware

Piyush Chordia | | *Visits (48436)*

When you read the word snapshot what is the first thing that comes to your mind is , "A Photograph which preserves the best moments of your life". Technically snapshot is very much the same with the difference that it preserves the state of some digital resource. In VMware a disk "snapshot" is a copy of the VM's disk file (.vmdk) captured at a certain point in time. This snapshot preserves the disk file system and the files stored on it which can be of any type (including all the operating system files). So if something goes wrong with your virtual machine, you can restore it to a snapshot which was working previously.

One can also create snapshots for different versions/service-packs on an OS.Hence snapshots can also be looked upon as version controlling mechanism at OS level. So if your computer was shut down abruptly or gets infected by virus, just revert to a snapshot.

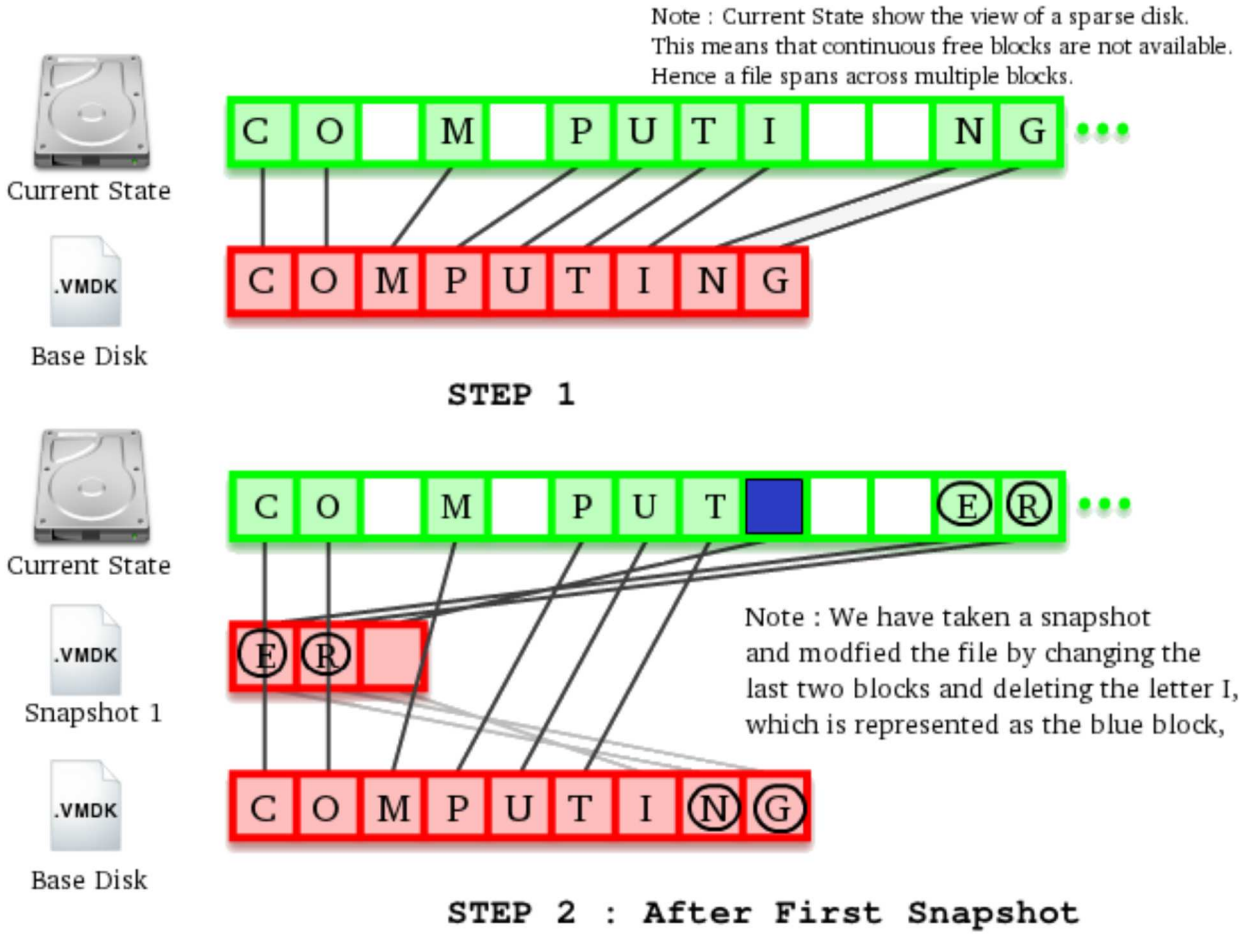
So how do snapshots really work ? There's just one **super rule** to VMware's snapshot technology :

Thumb Rule: "*Snapshots only store the differences between the current state and the original state*".

It follows the copy-on-write scheme with every subsequent disk access. Lets try to understand what that means ...

Consider that you have a text file with the word "COMPUTING" stored in it.This file is a sparse in nature : which means it spans across multiple blocks on the disk.Step 1 below demonstrates this scenario. The black lines indicate

the links on to the stored data. For demonstration purpose lets consider that each block on disk has only one character.



Note : The blocks shown above contain only one character and is purely for example purpose. In real the block size could be of say 1MB or a sector on disk.

Now when you take a snapshot another file named Snapshot1.vmdk will be created. When you create a snapshot, any changes made on the original virtual disk image are not made on the original disk, but they are written to a new (snapshot) disk file. This action is very fast as there is no need to copy whole virtual disk image.

Thumb Rule : *"While saving changed data blocks in a snapshot, all modified block will be saved first , followed by blocks which were deleted as compared to base disk blocks."*

As seen in Step 2 , blue block is linked at the end of the snapshot1.vmdk

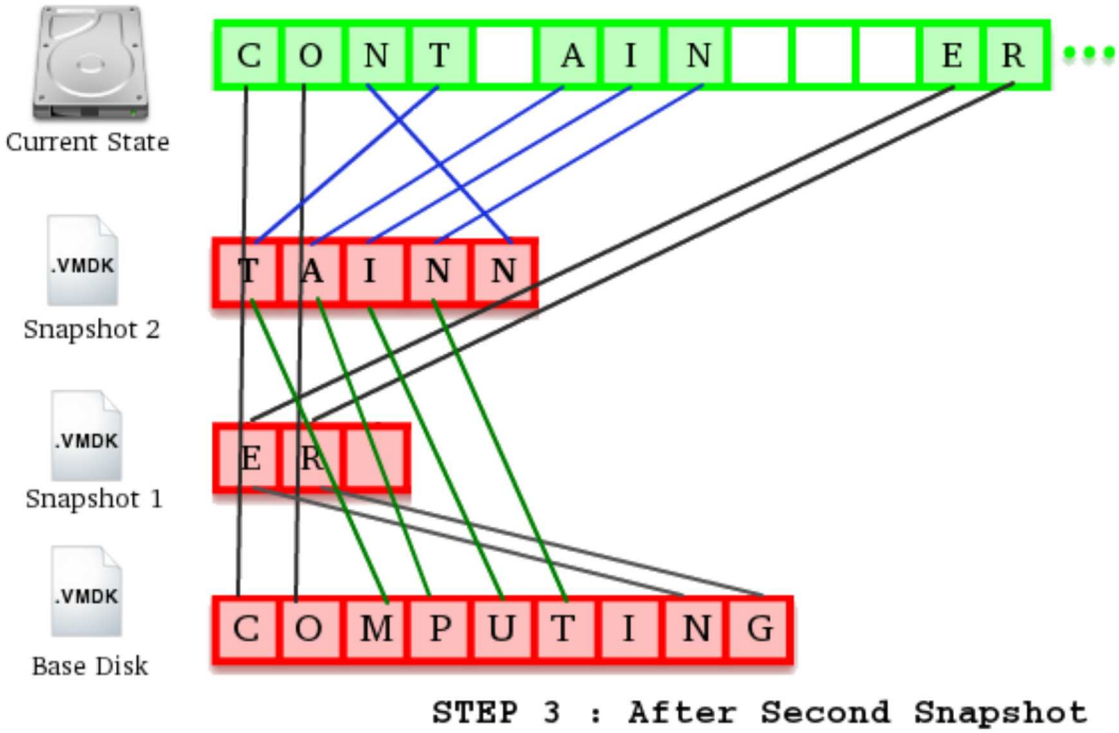
Lets suppose that you take a snapshot after you have saved the word "COMPUTING" in the file. After the snapshot you modify the file by changing its last two blocks (letters N and G circled above) and clear the letter I. The new changed word is "COMPUTER" as show in the Step 2. The blue block above is nothing but an empty block created by deleting letter I. The blocks in RED represents the new snapshot1.vmdk disk which contains only the changed characters.

Thumb Rule : *"While reading any file in current state read only the data accessible by first level links, irrespective of number of snapshots and original data of the file."*

Reading the first links of the "Current State" disk from Step 2 (Green blocks) , word "COMPUTER" is retrieved and the size of snapshot1.vmdk is 3 blocks (2 filled and one empty). However since its a differential snapshot file its size is much less than original base disk (9 blocks). Snapshot image size grows as you continue to change more and more

data from your original virtual disk image (which remains untouched from the moment you took the snapshot).

Thumb Rule : *"Size of a snapshot will always be less than the base disk, but in worst case it will be exactly the same size if all blocks were to be changed."*



As seen in Step 3 , we now take another snapshot after saving the word "COMPUTER" in the file. On making more changes after snapshot2 , similar process is followed to create snapshot2.vmdk file. The new changed word in the file is "CONTAINER". As a result now neither snapshot 1 nor the base file are written two, but are still referenced. Snapshot 2 will store the new changes as compared to snapshot 1. If you read the first level links of the green blocks from top to bottom , the word "CONTAINER" is read with the fact that only 5 letters are stored in snapshot2.

Below are the list of changes made to the file

Step1 - Base Disk = COMPUTING

Step2 - Snapshot1 = COMPUTER

Step3 - Snapshot2 = CONTAINER

From the above scenarios its clear that taking snapshots in VMware involves only writing the differences in files changed from the time of the snapshot, not the complete virtual machine disk. This mechanism is similar to taking diff and patch in Unix, but in a more sophisticated way that diffs on a binary level with the knowledge of how a VMFS (Virtual Machine File System) is structured.

Now we have a clear idea about the Copy-On-Write Protocol - Every time a block from the base disk or previous snapshot is changed , copy it to the current delta or snapshot file, which implies that when you perform a snapshot restoration, it only has to rewrite the sectors that were modified since you took the snapshot. As a result snapshot revert is also super fast.

But the Question is, What happens when you revert to an older snapshot? The VM software throws away the contents of snapshot2.vmdk and starts over with reading contents from snapshot1.vmdk. During this all the Blue links in Step3 are replaced with Green Links. (this is a similar strategy followed in deleting a node from a linklist ... Pure programming stuff). Note : Links to Snapshot1 and Base are not changed.

There are two more important aspects of Snapshot management : Discarding a snapshot and Merging a snapshot (reverting to a non-immediate parent snapshot). I'll discuss it in my future articles.

Original Article

Tags: *virtualization vmware snapshots*

Add a Comment More Actions

Comments (0) *Add a Comment* *Edit* *More Actions* *Quarantine this Entry*

Previous Entry Main Next Entry