01 Jan 2004

# A Method for Enhancing the Snapshot Performance in SAN Volume Manager

Chang-Soo Kim
*Missouri University of Science and Technology*, ckim@mst.edu

Dong-Jae Kang

Young-Ho Kim

Hag-Young Kim

*et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/790*

## Recommended Citation

C. Kim et al., "A Method for Enhancing the Snapshot Performance in SAN Volume Manager," *Proceedings of the 6th International Conference on Advanced Communication Technology, 2004*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2004.
The definitive version is available at https://doi.org/10.1109/ICACT.2004.1293007

# A Method for enhancing the snapshot performance in SAN Volume Manager

Chang-Soo Kim, Yu-Hyeon Bak, Dong-Jae Kang, Young-Ho Kim, Hag-Young Kim, Myoung-Jun Kim
*ETRI(Electronics and Telecommunications Research Institute)*
{cskim7, bakyh, djkang, kyh05, kimh0, joonkim }@etri.re.kr

***Abstract* — Linux-based storage clustering software, SANtopia Volume Manager (SVM), has provided various online volume management features to support enterprise computing in SAN environment in which a number of storages and host computers are connected with each other through fiber channel. The online snapshot is one of the representative features provided by SVM and enables online backup of data without system downtime.**

**In this paper, we propose an advanced online snapshot technique to enhance the overall system performance. The SVM has taken relatively long time according to the size of the logical volume in performing a snapshot creation or deletion operation. Additionally it consumed more time to perform a normal I/O operation on the original data with snapshot than without snapshot. In order to solve the problems described above, we update the metadata structure of SVM and enhance the performance by reducing the execution time of snapshot creation, snapshot deletion, and normal I/O operation with snapshot. Also, we can provide confidence of system by ensuring that all I/O operations have same performance regardless of existence of snapshot.**

***Keywords* — SAN, Volume Manager, Snapshot**

## 1. Introduction

Paradigm shift to information society has made information to be enlarged, and it has required high performance data processing system. Storage Area Network (SAN) is a system environment proposed to overcome problems caused by those environments. SAN with fiber channel interface connects a number of storage devices with many computer systems independently. From this, very fast data access service, scalability and availability of both storage and computing power are available.

However, SAN itself can't utilize its embedding power fully. In order to utilize full benefits of SAN, many other features are needed. Since many computer systems and a number of storage devices are inter-connected with each other, a centralized management system is needed to manage its physical components in SAN environment efficiently. Additionally, since a number of computer systems share same storage devices, it is also needed to control those computer systems correctly in SAN.

File systems and logical volume managers used before SAN's appearance can be applied to these environments in which only one computer system manages a number of storage devices. After SAN's appearance, many file systems and logical volume managers has been researched. Global File System (GFS), Veritas Cluster File System and SANtopia File System are representative file systems supporting the SAN [4, 5, 13]. Also, Pool Driver, Veritas Cluster Volume Manager and SANtopia Volume Manager are representative logical volume managers and can be used in the SAN [4, 5, 13].

Logical volume manager groups a number of disks into a large virtual logical volume. A logical volume can have one of various software RAID levels according to its necessity in order to provide availability of data or performance of I/O operations. In addition, logical volume manager has to support various online management features for the non-stop system environment [13].

SANtopia Volume Manager (SVM) is a storage clustering software supporting SAN environment and implemented in Linux operating system. SVM provides various online management features including online-resizing of a logical volume and online snapshot [13].

In this paper, we propose a method for enhancing the snapshot performance of the SVM. The SVM has taken relatively long time according to the size of the logical volume in performing a snapshot creation or deletion operation. Additionally it consumed more time to perform a normal I/O operation on the original data with snapshot than without snapshot. These natures of snapshot operations are similar in most logical volume managers.

The remainder of this paper is organized as follows. In section 2, we describe the method for providing the snapshot feature and problems in previous SVM. In section 3, we explain the new efficient method for snapshot feature proposed in this paper. Finally in section 4, we conclude with brief summary and our contribution.

## 2. SANtopia Volume Manager

### 2.1. System Architecture

A logical volume is simply the grouping of several storage devices into what appears to be a single large storage device. The logical volume is represented as a block device node and can be used just like a real device. Every I/O operation specifies a device and block number pair. A request for a logical device and block must be mapped to a physical device and block for the low level disk driver.

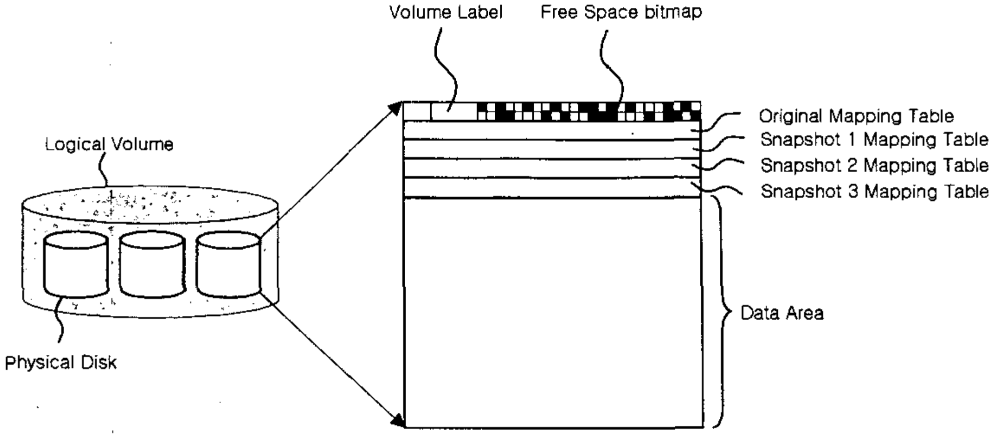

**Figure 1. The structure of metadata**

This mapping is done by a logical volume manager in the kernel. A logical volume can be configured to have various RAID levels.

The SVM consists of configuration manager, mapping manager, free space manager, Input/Output manager and distributed lock manager. The configuration manager collects various storage devices connected with SAN according to the user request, and creates a logical volume that is a virtual large block device. The mapping manager maps the logical address to the corresponding physical address. The logical address is an address that other parts of the system use to access the logical volume. The free space manager processes the allocation or deallocation of a physical block (or extent). The Input/Output manager performs real I/O operations appropriate to the RAID level of underlying logical volume. Finally, the distributed lock manager controls the concurrent access of data from a number of computer systems.

Figure 1 shows the structure of metadata that the SVM manages in a physical disk for a logical volume that consists of 3 physical disks. The volume label is managed by the configuration manager, and contains the volume name, information for disks participating in the logical volume, software RAID level and so on. The free space bitmap is managed by the free space manager, and allocated a bit per one block (or extent) in data area. Four mapping table is managed by the mapping manager. Original mapping table is for address mapping in normal I/O operations. Other snapshot mapping tables are for address mapping in snapshot I/O operations. As shown in the figure, the SVM supports at most three snapshots.

## 2.2. Problems

In current computing environment, it is very important to support non-stop services. The SVM provides online snapshot feature for supporting non-stop services. The snapshot technique eliminates the downtime of system that can be caused by backup of data, by storing the status of data at specific time. After creation of snapshot, if any data is updated, the SVM allocates a block or extent from free data area and copy the original data block to the newly allocated block. This technique is so called as Copy-On-Write (COW) technique.

Figure 2 shows steps for performing update I/O operation on a logical volume with snapshot. When user requests to create a snapshot, the SVM copies the original mapping table to an appropriate snapshot mapping table according to the snapshot creation order. After completion of snapshot creation, user can update a data block.

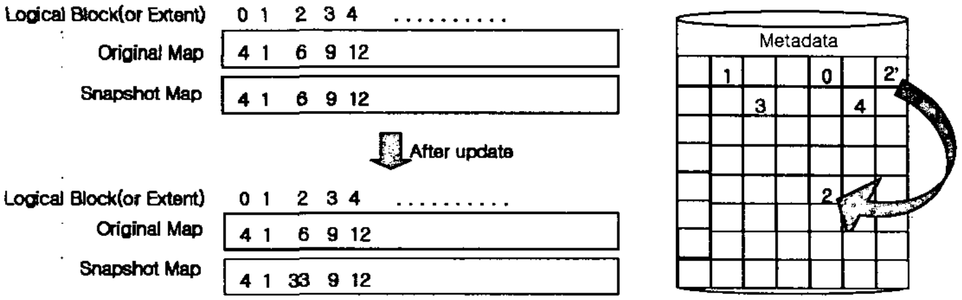

**Figure 2. update operation after creation of snapshot**

When an update operation is issued on a logical volume with snapshot, the SVM has to check if the update operation on the data block is first update operation after creation of snapshot. If so, the SVM must perform COW operation. For this decision, the SVM compares the original mapping entry to the snapshot entry. At this time, if all three snapshots were created, the number of comparison is three. If any mapping entry for snapshots is same with the original mapping entry, this means that the data block is first updated after creation of corresponding snapshot(s). In other words, COW operation is needed for the corresponding snapshot(s). The COW operation allocates a new data block, copy the contents of the original data block to the newly allocated data block, and updates the corresponding snapshot mapping entry to point to the newly allocated data block. When the COW operation is completed, the update I/O operation is performed on the original data block. In figure 2, an update I/O operation is issued on the logical block 2.

The size of the original mapping table is determined by the capacity of underlying physical disk and consists of several blocks with 512 byte unit size. Therefore, I/O operations on four disk blocks are needed in order to compare four mapping information. This means that the time consumed in mapping a logical address to the physical address is four times when all allowed snapshots are existed. This different time consuming in mapping makes it difficult to support confident services in enterprise computing environments.

In case a snapshot is no more useful, user can request to remove the snapshot. The removal of a snapshot requires deallocation of data blocks allocated for COW operations. For this deallocation, the SVM compares each mapping entry for the snapshot to mapping entries for other snapshots and original mapping entry. If a mapping entry for the snapshot is different from all other entries, the corresponding data block can be deallocated because the data block is only used for the removed snapshot. Similar to comparison for checking the needs of COW operation, four times of comparisons per each mapping entry are required to determine that a data block allocated for the snapshot can be deallocated.

## 3. Advanced snapshot method

In this section, we propose a new snapshot method to solve the problems described in section 2.2. The new method always consumes same time to map a logical address to the corresponding physical address regardless of existence of snapshots and the number of snapshots. The reason is that the new method ensures that only one disk I/O operation is sufficient to determine necessity of COW operation at snapshot creation time and deallocation of data blocks allocated for snapshots at snapshot deletion time. This can enhance overall system performance and confidence of services.

We update the structure of mapping table in new snapshot method. The updated structure of mapping table is shown in figure 3.

As shown in figure 1, in the structure of mapping table for previous SVM, data blocks for original mapping information are listed in continuous manner and then data blocks for mapping information of first, second and third snapshot follows in turn. From this structure of mapping table, four disk I/O operations are needed for comparison of mapping information at worst case.

However, as shown in figure 3, in the updated structure of mapping table, a pair of mapping entries for original, snapshot 1, snapshot 2 and snapshot 3 is integrated. In this structure, one unit block with the size of 512 bytes can contain 16 integrated mapping entries. The calculation is following.

*The number of integrated mapping entries per one unit block = 512 bytes / ((8 bytes per one mapping entry) * 4 entries)* (1)

When user requests to create a snapshot, the SVM determines snapshot number. After determining the snapshot number, the SVM reads the mapping table blocks into memory in turn. After reading of each mapping table block, for each integrated mapping entry, the SVM copy the first 8 bytes that is a original mapping information to the position at 8*snapshot number. After completion of coping for all mapping table blocks, the snapshot creation operation is completed.

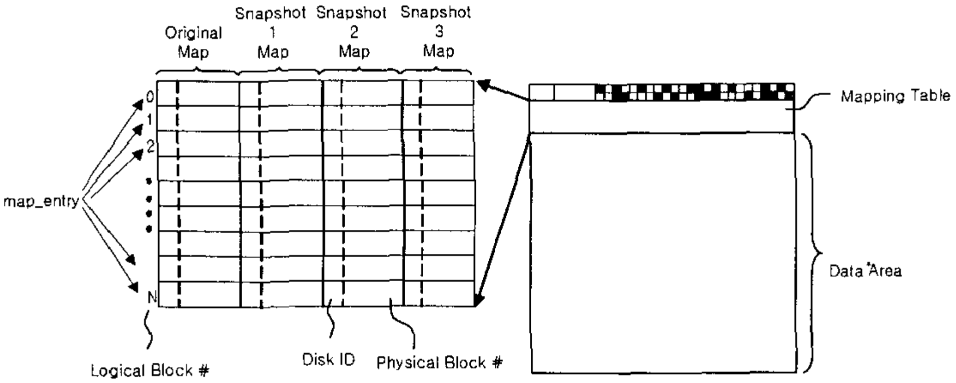

**Figure 3. New structure of mapping table**

When user issues an update I/O operation on original logical volume with snapshot(s), the SVM reads corresponding mapping table block into the memory to perform address mapping from logical address to physical address. The mapping table block contains all information needed to determine whether COW operation is required. The information contains all snapshot mapping information as well as original mapping information. We can see that only one disk I/O is sufficient for this determination.

Additionally, if user requests delete operation on a snapshot, the SVM must compare the snapshot mapping information corresponding to the snapshot with all other mapping information. This comparison operation requires also only one disk I/O operation per each mapping block.

## 4. Conclusions

In this paper, we proposed an advanced snapshot technique to enhance the performance in performing normal I/O operations on original logical volume with snapshot(s). Through the modification of the structure for mapping information table, we can eliminate the necessity of additional disk I/Os to determine that a COW operation is needed when user requests an update I/O on original logical volume with snapshot(s) and that the SVM can deallocate the data blocks allocated for a snapshot by COW operations when user requests deletion of the snapshot.

Those eliminations of additional disk I/Os give the system with increased performance. Additionally, those ensure that the system can service with same performance regardless of existence of snapshot(s).

## REFERENCES

[1] Friedhelm Schmidt. The SCSI Bus & IDE Interface. Addison-Wesley, second edition, 1998.

[2] Alan F. Benner. Fibre Channel: Gigabit Communications and I/O for Computer Network. McGraw-Hill, 1996.

[3] Heinz Maulschagen. Logical Volume Manager for Linux. http://linux.msede.com/lvm/.

[4] David C. Teigland, Heinz Mauelshagen. Volume Managers in Linux. http://www.sistina.com.

[5] David Teigland. [Slides] The Pool Driver: A Volume Driver for SANs. http://www.sistina.com.

[6] Hsiao H-I, DeWitt DJ. Chained declustering : a new availability strategy for multiprocessor database machines. 6th International Conference on Data Engineering, IEEE Comput. Soc. 1990.

[7] Chao C, English R, Jacobson Dstepanov A, Wilkes J. Mime: a high performance storage device with strong recovery guarantees. [Report] HPL-CSP-92-9 rev 1, March 1992

[8] Edward K. Lee, Chandramohan A. Thekkath, Chris Whitaker, Jim Hogg. A Comparison of Two Distributed Disk Systems. http://www.research.digital.com/SRC/

[9] Amiri K, Gibson GA, Golding R. Highly concurrent shared storage. Proceedings 20th IEEE International Conference on Distributed Computing Systems. 2000.

[10] Edward K. Lee, Chandramohan A. Thekkath. Petal: Distributed Virtual Disks, The Proc. 7th International Conference on Architectural Support for Programming Languages and Operating Systems, 1996.

[11] Matthew T. O'Keefe, Standard file systems and fibre channel, In The Sixth Goddard Conference on Mass Storage System and Technologies in cooperation with the Fifteen IEEE Symposium on Mass Storage Systems. pp.1-16, College Park, Maryland, March 1998.

[12] Steve Soltis et al. The design and performance of a shared disk file system for IRIX, in the Sixth Goddard Conference on Mass Storage System and Technologies in cooperation with the Fifteen IEEE Symposium on Mass Storage Systems, pp.41-66, College Park, Maryland, March 1998.

[13] Chang-Soo Kim, Gyoung-Bae Kim, Bum-Joo Shin. Volume Management in SAN Environments, Proceedings of the Eighth International Conference on Parallel and Distributed Systems, pp.500-505, Gyoung-Ju, Korea, june 2001.