01 Jan 2001

Volume Management in SAN Environment

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

Gyoung-Bae Kim

Bum-Joo Shin

Volume Management in SAN Environment

Chang-Soo Kim, Gyoung-Bae Kim, Bum-Joo Shin
ETRI(Electronics and Telecommunications Research Institute)
{cskim7, gbkim, bjshin}@etri.re.kr

Abstract

Logical volume managers have long been a key components of storage system. Their key features are creation of logical or virtual view of physical storage devices and support for various software RAID levels. These make it possible to overcome the limits to capacity, availability and performance of a physical storage device.

Most logical volume managers are operated in a single system environment. They are not adequate for SAN Environments where several hosts share and access a logical volume at the same time.

Some recent logical volume managers are run in a multi-host environment. However, they can't support the enterprise computing environments in which the system must support 24x7x365 uptime operations such as online resizing and online backup.

In this paper, we propose a logical volume manager called 'SANtopia Volume Manager' that supports multi-host environments and provides various volume management features to support enterprise computing. Also it is a cluster enabled logical volume manager that maximizes the parallelism for high performance, and provides high scalability and high availability.

1. Introduction

The activation of internet, e-business, and the services of multimedia data cause the amount of data processed in IT(Information Technology) fields to grow exponentially. As the needs for more high performance and capacity have grown in various situations, the storage system used in traditional IT systems has limits to satisfy those. This initiated the improvements in hardware and software areas.

Recent advances in switching technology, fiber optics and the convergence of network and channel interfaces are allowing order-of-magnitude improvements in network latency and bandwidth through new technologies like Fibre Channel[2]. The Fibre Channel standard integrates both storage and networking capabilities into a single serial interface. Hundreds of Fibre Channel disks and host computers may be combined in shared-bandwidth loops or across switches capable of maintaining several simultaneous gigabit transfers. This network-like connection among disk drives and hosts has prompted a shift in the way storage systems are viewed[2]. In contrast, today's parallel SCSI technology supports only about 8 devices per bus with each bus extending at most 25m making the technology effectively unscalable[1].

A Storage Area Network(SAN) is the combination of network attached FC storage devices and computers with FC network adapters on a loop or fabric. Each computer has effectively local access to all the drives. Fibre Channel combines the high channel bandwidths of disk interfaces with the high connectivity of network interfaces.

The ability to provide users of enterprise servers with maximum uptime, optimal performance, and high storage capacities and file sizes is limited by traditional disk technology. The capacity of several disk drives is often needed for a large file system. Because a file system is created on a single device, individual disks are combined into one volume called a logical device. A file system is then created on the logical device and a logical volume manager in the OS translates between the logical device and underlying physical disks.

A 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.

In the simplest case, logical volume manager will store data on the underlying devices from beginning to end, one disk to the next. This concatenation of devices results in linear mapping. Additionally, it may support various software RAID levels including RAID-0(striping), RAID-1(mirroring), RAID-5(striping with parity) and so on.

In this paper, we propose a logical volume manager called 'SANtopia Volume Manager' that is operated in SAN environments to take advantages of the Fibre Channel interface. It allows multiple hosts to access and share devices on Fibre Channel network, and provides several online volume management features. Additionally, it will be a high performance cluster volume manger helping to bring the availability and scalability.

0-7695-1153-8/01 $10.00 © 2001 IEEE

The remainder of this paper is organized as follows. In section 2, we represents some related works. Section 3 gives an overview of the SANtopia. In section 4, we describe the SANtopia volume management schemes. In section 5, we summarize our works and give some future works.

2. Related works

There have been many works for the software RAID systems. Most of them concentrated on the single system environment. A representative system is the Linux LVM[3]. Constructing flexible virtual views of storage with Linux LVM is possible because of the abstraction layers formalized in the standard. User space tools used to configure each virtual level follow a similar set of operations. With these tools, online allocation or deallocation of storage to virtual groups is possible. Higher level virtual devices can then be expanded or shrunk online.

The lowest level in the Linux LVM storage hierarchy is the Physical Volume(PV). A PV is a single device or partition. On each PV, a Volume Group Descriptor Area (VGDA) is allocated to contain the specific configuration information. Multiple Physical Volumes are merged into a Volume Group (VG). The newly created VG has the combined capacity of the participating PV's. More PV's can be added at any time to existing VG. A Volume Group can be viewed as a large pool of storage from which Logical Volumes (LV) can be allocated. LV's are actual block devices on which file systems can be created.

Linux LVM is a powerful logical volume manager in single system environments, but it can't be used in SAN environments where multiple computer systems share the storage devices.

Another work for the logical volume manager is GFS's Pool Driver[4, 5]. Pool Driver is for SAN environments and can be used to share SAN storage devices with GFS file systems. The Pool Driver is for the Linux and joins a collection of individual disk partitions into a logically contiguous block device. It is a mid-level block driver built on SCSI and FC driver.

Like the Linux LVM, Pool Driver's user tools are used to build a large logical block device. 'Ptool' command creates a pool, similar to LVM's Logical Volume, on disk by writing a label to each device's header. 'Ptool' uses a parameter file defining the pool's configuration. The command 'passemble' then scans all accessible devices for pool labels and create logical device node in /dev/pool/ directory with the given pool name. In order to support multiple computer systems, Pool Driver services SCSI command to handle the lock objects existed in special hardware disk devices.

Pool Driver has some drawbacks in volume managements. It supports only RAID level 0 and 1. And it can't support online resizing/reconfiguration and snapshot which is very important features for enterprise users.

3. Overview of the SANtopia

New networking technologies like Fibre Channel(FC) allow multiple systems to share the same storage devices. The file systems that allow these systems to simultaneously mount and access files on those shared devices are called shared file systems[12,13].

The SANtopia proposed in this paper is a cluster shared file system operating in SAN environments where the systems directly access the storage devices connected with SAN. The SANtopia collects many network attached storage devices into a large storage pool and provides a logical or virtual view of it. The storage pool provided by the SANtopia volume manager is not owned or controlled by any systems and used as a shared storage space in the systems on the network.

Figure 1 represents the architecture of the SANtopia. The SANtopia consists of 4 modules : file manager, global buffer manager, global lock manager, volume manager.

The volume manager collects various storage devices connected with SAN, and creates a big logical volume. Additionally, it implements a software RAID technique for each volume. And it considers the cluster environment in which many host systems share and access a logical volume, and administrative requests that are very important features in an enterprise computing environment including online resizing, snapshot and so on. Most of this paper will concentrate on this volume manager which will be described in the section 4 in detail.

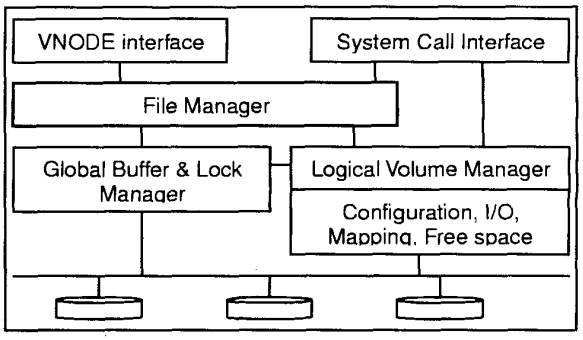

Figure 1. Architecture of the SANtopia

3.1 File manager

The SANtopia file manager uses 64-bit address space to support very large file size and file system. In the SANtopia file system layout, the inode area has no a fixed portion unlike to other normal file systems. In other words, inodes are scattered across the entire logical address space.

This results in eliminating the limit to the number of files allowed in a file system.

In order also to support the online resizing feature, an allocation bitmap to keep track of logical address usage is placed into the tail of the file system layout. When the file system size is increased in the future, the allocation bitmap portion for the increased storage space is added to the original allocation bitmap, and the new combined allocation bitmap is placed into the tail of the entire logical address space including the newly added storage spaces.

The SANtopia allocates and deallocates the storage space in extent unit. The extent is a contiguous storage space that consists of multiple blocks. An extent may be used for both normal data and metadata. The SANtopia gives two bits to the allocation bitmap for an extent in order to distinguish these usages of an extent. The value 00 is given to an extent for the free space, 01 is for an inode, 10 is for a directory entry and 11 is for a data extent.

An inode has dynamic multi-level structure unlike to general unix inode structure, which results in that the file system doesn't limit the size of an file. In addition, the SANtopia manages the directory by using extendible hashing technique to minimize the search time.

3.2 Global buffer and lock manager

In common, a file system maintains buffers in order to increase bandwidth by reducing the number of I/O to disks. The buffer management scheme is very sensitive to file system performance.

The SANtopia that is a cluster shared file system tries to enhance the effects of the buffer by utilizing the buffers in other servers as well as its own buffers. To achieve this, the SANtopia maintains a local buffer manager and a global buffer manager in each node of the cluster.

Because a block is accessed by several nodes in the cluster, it is needed to ensure the consistency of the corresponding buffer that each node has. The SANtopia uses a locking mechanism to solve this problem. The locking unit is a file and the lock manager locks a file using its inode number.

In the SANtopia, the buffer management and lock management are tightly coupled and operating together. A local buffer manager handles information about the buffers managed in its own node. A global buffer manager manages globally the buffering of blocks that it must handle according to the value of hash function to each block. Similar to the buffer management, the lock management is done by two managers, local lock manager and global lock manager. In the case of a lock request, the local lock manager in the node where the request is occurred sends a buffer list managed by itself to the corresponding global manager. At the grant or reject of a lock, a global lock manager sends a global buffer list managed by itself to the local buffer manager of the target node.

With this method, we can reduce the number of messages moved between cluster nodes. In the cluster environments, the number of messages between nodes is very sensitive to the overall performance.

4. SANtopia volume manager

4.1 Configuration management

In order to support a large file system, a hardware RAID collects several disks and provides one large device node. It also supports various redundant schemes including RAID-0, RAID 1, RAID-5 and so on. But, it is an expensive solution if one simply needs to make many small disks look like a single big disk.

The SANtopia volume manager collects various physical disks connected with the SAN and gives a virtual view of storage pool. It then allocates or deallocates partitions to build a logical volume with a correct RAID level on which a file system is build, in accordance with user requests. The lowest level in the SANtopia volume manager storage hierarchy is a disk partition.

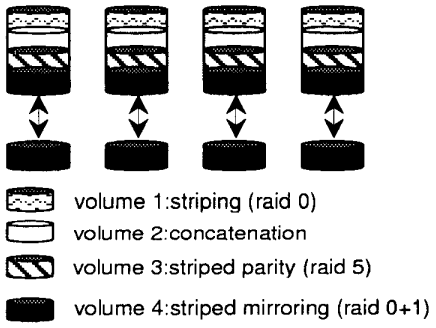

Figure 2. An available volume configuration with 8 disks.

Figure 2 shows an available software volume configuration in the environment with eight physical disks.

The SANtopia volume manager supports striping(RAID-0), mirroring(RAID-1), striped mirroring(RAID 0+1), striping with parity(RAID 5) and concatenation(linear) as RAID levels. The implementation of each RAID level is done by the mapping manager described in later. The mapping manager determines the location of each extent according to the underlying RAID level.

The volume configuration information is stored in the headers of each partition participating in the volume and read at the system boot time.

In addition, the configuration manager of the SANtopia performs various management features including volume creation/deletion, online resizing and movement of data between partitions in accordance with user requests. These management features are performed with help of the other internal modules. Additionally, It provides user with the configuration information of each volume that needed at management work of a volume.

4.2 Mapping management

The Mapping Manager makes it possible to provide the file manager with the virtualized view of the physical storage devices. A logical volume is represented as a block device node and can be used just like a real device by the file manager. All I/O operations by the file manager are performed on logical address space of a logical volume and this logical address is mapped to a physical address of an underlying physical storage device.

The mapping manager maps the logical address used in the upper modules to the corresponding physical address by an extent unit. The Extent is a unit of allocation or deallocation of physical storage and its size is a multiplication of basic block size and configurable at the volume creation time. The default size is 64 KB.

This mapping is done according to the RAID level of the corresponding volume that contained in configuration information which maintained in each participating partition's header. The real implementation of software RAID is therefore done by the mapping manager.

While many volume managers maps a logical address to a physical address by using one of the equation-like methods. With these methods, the mapping is fixed and they cannot provide the flexibility in many situations.

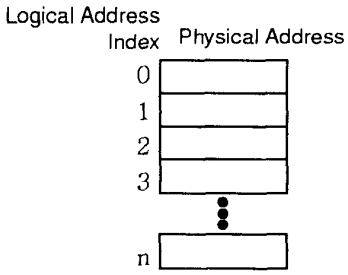

Figure 3. The structure of mapping table

The SANtopia volume manager uses a variable mapping technique. It maintains a table that associates a logical address with a physical address. We call this table as mapping table. Figure 3 shows the structure of this mapping table. The index key value of the mapping table is determined by the following equation.

index key value = logical address / extent size

The content of the table is a physical address corresponding to a logical address.

The mapping table is stored in a distributed manner, into the headers of disk partitions that are participating in a volume. Because the contents of a mapping table must be protected in the case of physical disk error, it needs to be duplicated. The structure of the mapping table makes the mapping information related with a specific partition to be scattered over several other partitions including itself. From this fact, we conclude that the duplication of contents within same partition is useless for recovery of a physical error. To solve this problem, we adopt the chained declustering technique introduced in [8]. The figure 4 is an example of the mapping for a volume containing 4 physical partitions.

1	16'	6	1'	11	6'	16	11'
2	17'	7	2'	12	7'	17	12'
3	18'	8	3'	13	8'	18	13'
4	19'	9	4'	14	9'	19	14'
5	20'	10	5'	15	10'	20	15'

Figure 4. The Mapping Table for a volume constructed on 4 partitions.

The mapping table maintained in the disks will be accessed by several SANtopia server systems. At this point, if the servers access the table in a free style, the contents will be corrupted. There are many solutions to solve this problem. The most popular method to maintain the consistency is to use a locking mechanism.

Nevertheless, because the access frequency of the mapping information is very high, we have to find a method with a little overhead. In the SAN environments, we assume that the servers trust with each other. Therefore, we distribute the mapping information with servers evenly and each server can access only its own area. If a server needs to access a part managed by another server, it sends a request to target server.

The mapping management using a mapping table provides much flexibility in managing the logical volume. The mapping table makes it possible to change the relationship between a logical address and a physical address at any time. With this property in the mapping management, the mapping manager can provides various useful features.

During the operation of system, the user will have a need to increase or decrease the size of a volume in online. We call this feature as online resizing of a volume. To implement this feature, the SANtopia uses two techniques according to users' favorites.

The first is to unify the configuration scheme of the entire volume including old and new. This technique is

depicted in figure 5.

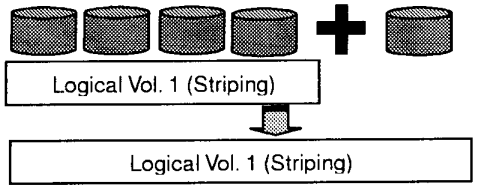

Figure 5. Applying an unified scheme to the entire volume.

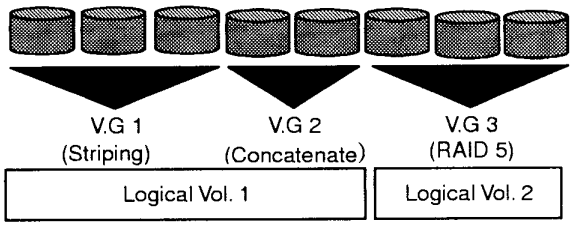

Figure 6. Applying different schemes to each storage group.

Second is that the newly added region has a different configuration scheme from old region's and is connected with old region logically. This technique is represented in figure 6.

In the case of figure 5(the volume configuration is striping), it is needed for the data distributed across the old volume components to be moved to newly added components, because the number of components constructing the volume is increased by one. In the previous technique using a fixed equation-like method, normal I/O operation is limited at this redistribution of data. The I/O operation occurred during the redistribution of data is performed cannot find the correct location of data interesting in the operation because the data can be existed in the original or new location.

While the SANtopia can handle the redistribution of data and normal I/O operation at the same time because it keeps track of the movement of data using the mapping table.

In the case of the figure 6, we apply the same concept as the Volume Group of Linux LVM. Each Volume Group has its own redundancy scheme. This simplifies the implementation but sacrifices the performance gained from striping entirely.

In the enterprise computing environments, the backup is very important to recover a physical error. But, we have to be able to backup of entire volume within the normal operation in order to support the 24x7x365 operating environment. To solve this problem, the SANtopia provides a feature called as snapshot. Snapshots are 'frozen' images of an entire volume that is consistent and can be used to back up the resident file system. The mapping management technique makes a snapshot operation to be possible smoothly.

A copy-on-write technique is used to implement snapshot in the SANtopia. It allows continued updates to the volume while maintaining a previous frozen state. A snapshot command replicates corresponding mapping table in an area. This is the complete action of snapshot command and consumes a few of seconds. Any new writes will result in a copy of the original extent into a newly allocated extent. Then the update is performed in old extent. The old and new locations of each modified extents are maintained by mapping manager in two mapping tables.

Each disk drives have different characteristics like transfer rate. The system administrator monitors the usage of each disk continuously and performs the tuning tasks to distribute the load of hot spot disks. At this point, the location of original data in the hot spot disk is changed to another disk. The mapping manager changes the location of the data and reflects this update to the mapping table.

4.3 Free space management

The SANtopia manages physical and logical addresses independently in order to provide more flexibility. It causes that we must separate physical allocation bitmap from the bitmap for managing usage of logical address. The logical allocation bitmap is a part of the file manager.

A scenario occurred when new storage space is needed by the file manager is following. The file manager reserves a logical address for new storage space, and then requests the volume manager to allocate new storage space. The volume manager passes it to the mapping manager. The mapping manager determines the disk partition in which new storage space will be allocated in accordance with the volume configuration. With this information, the mapping manager calls the free space manager to allocate new storage space in that partition. The free space manager allocates a free space in that partition, sets the corresponding allocation bitmap and then returns the physical address of it to the mapping manager. Then the mapping manager updates the mapping table with the physical address.

The volume manager maintains the physical allocation bitmap in the headers of each partition. Because the physical allocation bitmap is subject to its own partition, the bitmap is duplicated within its partition to recover an error of physical disk. Figure 7 represents the physical allocation bitmap in each partition.

The physical allocation bitmap is also accessed by several server systems and have to be protected from corruption caused by random accesses. The free space manager splits the bitmap into several units in fixed size and associates a lock object with each unit. If a server

needs to allocate new storage, it obtains a lock corresponding to one of several units in the interesting partition and then uses the bitmap unit exclusively.

There are several locks in each partition. A server can obtain at least one lock in each partition and use storages in entire partitions. From this technique, we can ensure high scalability and maximize the parallelism of servers. Also it minimizes the metadata search time.

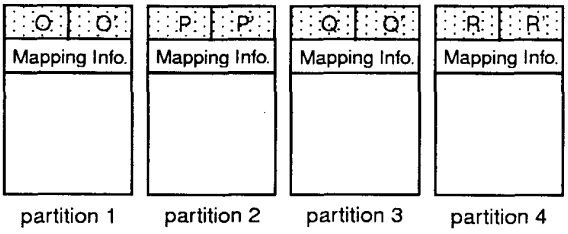

Figure 7. The physical allocation bitmap of each partition

5. Conclusions and future works

In this paper, we proposed a SANtopia volume manager for SANs. The SANtopia volume manager supports various software RAID levels. The software RAID technique provides user with flexible configurations of many small physical disk drives. In addition, the users of the SANtopia volume manager can resize the logical volumes in online.

When the volumes are resized, newly added parts can be unified with old configuration or have its own configuration. The unified method causes the blocks within original configuration to move into new partitions.

One of the key features of a logical volume manager is to provide virtualization of physical storages. This virtualization can be possible by mapping between logical spaces and physical spaces. The snapshot of a volume can be taken in a few seconds, which is a very important feature in enterprise computing environments.

The mapping information and physical allocation bitmaps are maintained in each partition's header. These are duplicated with appropriate schemes to ensure physical failure-safe works. Additionally, this information is handled in a manner to guarantee the consistency constraints and provides high scalability with maximized performance.

There are more works which can be done on the SANtopia volume manager. Taking advantage of special disk characteristics and load balancing are required to optimize the performance more.

Another area of works is recovery in the face of the server failure. In current SANtopia, all metadata including mapping information and physical allocation bitmaps are synchronized to the disk in order to remove needs of recovery in the case of server failure. The synchronization technique is a very safe technique, but has I/O overhead to the disk.

6. 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] Chandramohan A. Thekkath, Timothy Mann, Edward K. Lee. Frangipani: A Scalable Distributed File System. ACM Operating Systems Review, Vol. 31, no.5, Dec. 1997.

[7] 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.

[8] 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.

[9] 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

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

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

[12] 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.

[13] 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.