

## VICS: a Storage Virtualization Management System for SAN

Li Bigang, SHU Ji-wu, ZHENG Wei-min  
 Department of Computer Science and Technology  
 Tsinghua University, Beijing 100084, China  
[lbg01@mails.tsinghua.edu.cn](mailto:lbg01@mails.tsinghua.edu.cn)

### Abstract

Storage Area Networks (SANs) have the virtues of high scalability, high availability and high performance. On the other hand, their storage virtualization systems are not compatible with multi-operating systems, and it is hard for the virtualization storage management system to manage multi-type storage. This paper proposes a new virtualization storage management model for SANs: Virtual Intelligent Control System (VICS). It includes three layers, the logical storage management layer, the virtualization layer and the storage resource management layer. With the logical management layer and the storage resource management layer the VICS can manage multi operating systems and is compatible with various storage systems. The VICS controls the storage resources through the FC network and applies the LUNs for various operating systems, giving users a uniform management interface. The VICS system employs functions such as storage virtualization, and LUN zoning, and it supports the management of disks and tapes. Furthermore, a cache mechanism is also designed in the VICS, which improves the SAN's performance. We implemented a prototype of the VICS, subsequent testing proved that the VICS makes the SAN systems more compatible and easier to manage.

### 1. Introduction

Storage Area Networks (SANs)<sup>[1,2]</sup> use a net-oriented storage structure, which enables the separation of data processing and data storage. SANs have the virtue of high availability and scalability, high I/O performance, and data sharing. SANs employ backup, remote mirroring, and virtualization functions, which has made them more popular. The storage virtualization management system can manage various storage systems which still provide one uniform interface for users. But at this time the management of the SAN systems is not compatible with multi operating systems. Various storage systems, such as XIOTech<sup>[3]</sup>, IBM<sup>[4]</sup>, EMC<sup>[5]</sup>, all have their own virtualization management systems, which add extra complexity and difficulty. Furthermore, the incompatibility between them makes the management of SANs more complex, and unified storage management is difficult to achieve.

The LVM<sup>[6]</sup> and EVMS<sup>[7]</sup> storage virtualization systems run on host servers, but they can only be used for one certain operating system. The XIOTech and other storage systems can implement storage virtualization management at the device level and are suitable for multi operating systems, but they can only manage their own storage system. StorAge<sup>[8]</sup> has developed an out-band virtualization system which

can manage various storage system. It has limited compatibility with few determinate operating systems through the agent program running on its servers. It introduces extra work for the hosts, and it is still complicated for users to manage. What's more, the extra agent running on the servers bring additional risk and complication for users.

This paper introduces the Virtual Intelligent Control System (VICS), a storage virtualization management system based on storage area networks. The VICS splits the traditional SAN into a host SAN and a device SAN. The host SAN is made up of the multi servers and the networks; the various storage resources and the network for them make up the device SAN. The VICS controls both the device SAN and the host SAN and offers one uniform management interface for the user. The VICS has broad compatibility with multi operating systems and with various storage resources. Figure 1 illustrates the architecture of a SAN system with a VICS.

In order to validate the VICS, we implemented a prototype of it. This prototype could apply functions such as volume online resizing/creating, snapshot, LUN-mapping and other virtualization functions. The VICS is also compatible with multi operating systems. Furthermore, it is also suitable for IP networks and FC networks. The VICS has excellent compatibility and scalability. We tested the performance of the prototype, and the result showed that the VICS only introduced a little latency for SANs, and the cache mechanism in the VICS could greatly improve the SAN's performance.

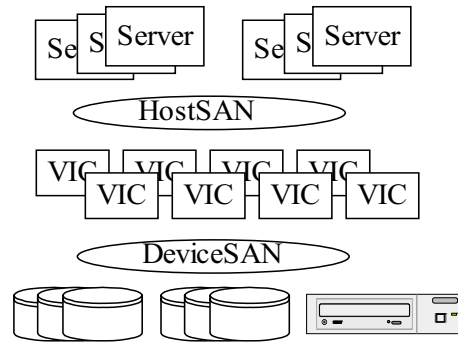


Figure 1 Architecture of a SAN with VICS

## 2. Related Work

Now the most popular networks for SANs are FC and IP. With the publication of the iSCSI standard, the development of the IP SAN has progressed greatly. Now the Intel<sup>[9]</sup> and the University of New Hampshire have their own iSCSI implementation<sup>[10]</sup>.

There have been many studies on the volume management software. We researched the Sistina Company, which used the global file system (GFS)<sup>[11]</sup> as a parallel file system in a SAN environment, and issued the logical volume manager (LVM) as a part of a Linux kernel. They provided a plan for the virtualization of storage in single systems. IBM also used EVMS to solve this problem<sup>[7]</sup>. The SANtopia volume management is a storage virtualization system based on hosts<sup>[12]</sup>. Another example of work on the logical volume is the GFS's Pool Driver<sup>[13]</sup>. It is a logical volume manager for SANs in Linux. It also builds virtual volumes for file systems and is a cluster aware driver, but it can only be used with the Linux OS.

We also researched the SCSI initiator driver and target driver for the ISP HBA in TH-MSNS<sup>[14]</sup>. Based on it we developed the VICS.

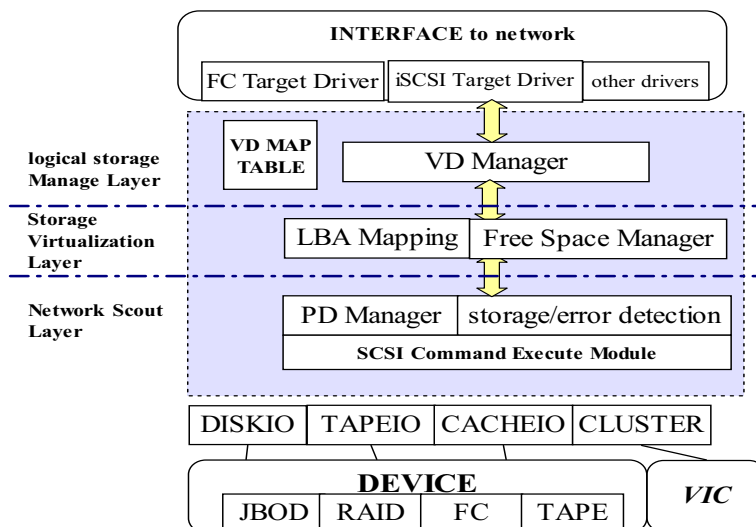


Figure 2 the architecture of the VICS

### 3 Architecture of the VICS

Figure 2 shows the VICS’s architecture. The system works on the network layer. It manages all the storage resources and implements storage virtualization, and LUN zoning. The VICS captures all the SCSI I/O commands and transfers them to the proper storage to implement.

#### 3.1 Overview of the VICS

The VICS splits the SAN into a device SAN and a host SAN. The host SAN enable the VICS to be compatible with multi operating systems, and the VICS controls the different storage systems through the device SAN. Figure 2 shows the software architecture of the VICS, which includes three layers: the logical storage management layer, the virtualization layer and the storage resource management layer. The logical storage management layer implements the LUN zoning and enables the multiple operating systems to become uniform. The virtualization layer provides the virtualization functions. The storage resource management layer

controls the device SAN, detects various storage systems and implements the cache mechanism.

#### 3.2 The logical storage management layer

The logical storage management driver receives the SCSI commands and messages from the target driver (iSCSI or FC HBA target driver), and then sends them to the proper logical disks. The interface between the target driver and the logical storage management is designed to be suitable for the IP-SAN and the FC-SAN. The main functions of the logical storage management layer are explained below.

(1) The logical storage management layer allows the VICS to be compatible with different target drivers. With this function the VICS can be implemented in both the FC-SAN and the IP-SAN. The interface is designed as Ref [14].

Figure 3 shows the SCSI command flow. When a new command is received, the target driver calls the *rx\_cmnd()* to notify the VICS. After handling the SCSI command, the VICS calls the *xmit\_reponse()* function to notify the

target driver of the completion of the command. If the command is a write command, the VICS first informs the target driver that the buffer for the data is ready first. This is implemented through the *rdy\_to\_xfer()* function.

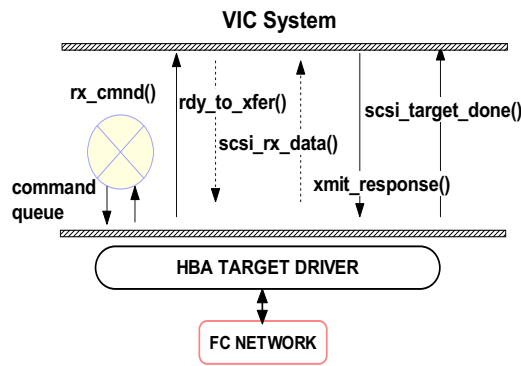


Figure 3 the SCSI command flow in VICS

With this interface the VICS is able to work well with the iSCSI target driver and the FC HBA target driver.

(2) The logical storage management layer should implement the LUN zoning function to manage the logical storage resources. The logical storage resource access mode includes several grades:

RW: the corresponding host can read from and write to the logical volume freely.

RO: the corresponding host can only read data from the logical volume.

DENY: the corresponding host cannot access the logical volume.

To implement this function, the logical storage management layer identifies one unique number for every host (for example, the MAC address, IP or WWN of the FC Network). The logical storage management layer keeps one resource table dynamically. One possible example is shown in table 1. The storage manager can read/modify the grades through the management software applied by the VICS.

Table 1 LUN Zoning Table

	VD1	VD2	VDn
ServerA	RO	DENY	RW
ServerB	RW	RO	RO
ServerX	DENY	RW	RO

### 3.3 The virtualization layer

The virtualization layer implements the storage virtualization management function for the VICS. It connects several physical disks (PD) to form storage container (SC). All the storage space is split into the same size of segments, which default is 32MB. The unassigned space of segment named free segment (FS), others, which are used for virtual disks (VD), are named physical regions. Then, all the storage resource is organized as figure 4. All the metadata for the virtualization layer is stored at the first part of the physical disks, including the UUID number for the physical disk. Generally speaking, the PD information and the VD information of SC is less than 1MB for a 73GB disks, so the space for the metadata of virtualization layer is very few.

As shown in figure 4, all the VD are made up of the PS. Normally, these PS come from different PD to improve the performance with this structure, the virtualization layer can offer the functions such as online resizing/creating and so on. For example, if the VD\_A need to extend from 50GB to 100GB, the virtualization layer can just arrange FS link to the end of VD\_A for 100GB. The VD can be extended if there were free space (or FS) in the SC.

The address-mapping is also very important for this layer. If the SCSI command's logical address is LA and the logical volume number is LV, the address-mapping manager would map them to the proper PS and offset.

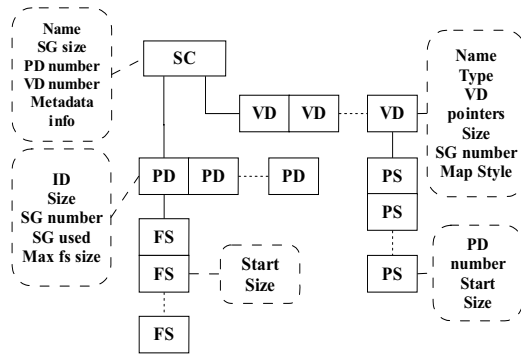


Figure 4 the architecture of space organization

The address-mapping maybe is linear mode or stripe mode, which are very simple. But VICS’s mapping mode is complex. With Stripe Mode, data with a fixed size are sent to different PDs one by one, and the performance of the VD would be better. The virtualization layer implemented the address mapping work in three steps:

- (1) Find the proper VD according to the LUN number with the SCSI commands;
- (2) Find the corresponding PS by comparing the LBA in SCSI command and PS size, then get the proper PD information;
- (3) Convert logical address to actual address and read/write the data.

With these 3 steps this layer can convert the (CD, LBA) to the (PD, offset) and read/write data. Figure 5 shows it.

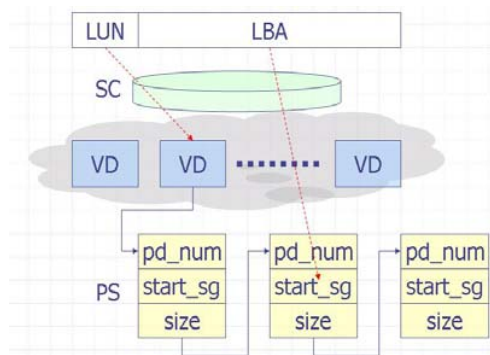


Figure 5 address-mapping

Another important function of the virtualization layer is snapshot. The snapshot function can make a static data copy of a whole VD very fast and without stopping the data service; so many online backup systems use this characteristic to backup data. The snapshot is implemented with a technology named COW (copy on write). Kim and others improved the traditional snapshot technology and we have adopted this improved method to implement this function<sup>[15]</sup>. The detail information can be found in the ref. 15.

### 3.4 Storage resource management layer

The storage resource management layer deals with various disks, and employs the determinate interface for the virtualization layer. It receives the SCSI commands from the virtualization layer, and sends them to the various disks or tapes. The main functions of this layer include:

- (1) Providing the virtualization layer with a uniform interface. This layer screens the difference between the disks for the virtualization layer. The RAID algorithm is also implemented in this layer.

(2) Implementing the DISKIO and TAPEIO. With the DISKIO, the VICS can control the disks. The VICS also can control the tapes with the TAPEIO. The DISKIO deals with the SCSI Block Commands, and the TAPEIO deals with the SCSI Stream Commands. The virtualization layer uses the disks to construct virtual disks, while the backup system uses the tape to back up data. The storage resource management layer distinguishes the SCSI commands and sends them to the proper IO driver.

- (3) Implementing the

communication mechanism. Multi VIC nodes communicate with each other. For example, two VIC nodes can form a failover system through the communication mechanism. The metadata and the configuration data can be kept synchronous with each other.

(4) Implementing the cache mechanism for the VICS. The VICS can use the RAM as a data buffer to improve the performance of the storage system. In this mechanism, many algorithms for cache systems, such as read-ahead optimizations, should be implemented, as they dramatically improve block device's I/O performance.

The VICS is constructed of the three layers mentioned above. These three layers have clear functions and a clear interface, so the VICS can provide a uniform management interface for users and it is compatible with multi operating systems and various storage systems. The storage virtualization management system gives users a more intelligent and accessible storage management technique.

#### 4. Performance Evaluation

In order to prove the benefits and compatibility of the VICS and test its performance, we implemented a prototype of the VICS based on the TH-MSNS<sup>[14]</sup>. This prototype supported the disk arrays and tape devices. The cache mechanism was also implemented.

In testing configuration, the server's operating systems included the RedHat9, Windows 2000, FreeBSD, Windows XP, NetWare and Solaris SAPRC, and the storage systems included FC-DISK JBOD, XIOTech FC-DISK, SCSI disk array, IDE disk array and a tape device.

These devices were connected with each other through on FC network with a bandwidth of 2Gbps. The different operating systems and various storage systems formed a complex SAN. Figure 6 shows the hardware architecture of the test environment.

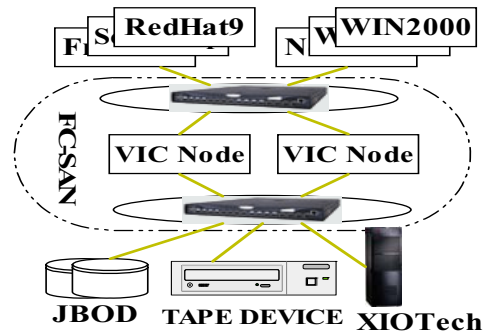


Figure 6 hardware architecture of the test environment

#### 4.1 Testing configuration

The initiator server machines include a Xeon 2.4G CPU, 1GB RAM and a Qlogic2300 HBA for the Fiber Channel. The operating system is different with each to test the compatibility of VICS. The VIC servers included two Xeon 2.4G CPUs, 1GB RAM and two Qlogic 2300 HBAs for the Fiber Channel. One Qlogic HBA was working in the initiator mode and the other was working in the target mode. The VICS modules were running on these servers. The FC switch type was the Brocade Silkworm 3200. This switch provides 2Gbps bandwidth for the FC Channel.

The storage device for the SAN was a little complicated. One FC\_JBOD was connected to the device SAN. This JBOD included five Seagate FC disks. A IDE and a SCSI array were connected to the device SAN through the I/O control machine, same with a tape device.

#### 4.2 Testing Results

We evaluated the performance of

the VICS with the iometer, which is a standard benchmark used for measuring I/O performance. This benchmark was originally developed by Intel, and it can measure the read/write performance in a sequential/random manner and test the I/O latency.

Figure 7 shows the SAN's read throughput, and Figure 8 shows the SAN's write throughput. The cache throughput represents the VICS performance with the cache mechanism. The results show that the VICS and the SAN had the same performance. This proves that the VICS had little influence on performance, as the VICS only modifies and transmits the SCSI commands in the RAM, and these operations are much faster than those on the disks. The VICS enhanced the SAN's functions, so the slight effect on latency (< 1%) is acceptable. Figure 9 shows the average response time in the three conditions. The result shows that VICS has little influence on the latency of IO operations, since the latency of the VICS and the FC network is much less than the disks. What's more, the cache mechanism can greatly improve the VICS's performance. The cache system used 512MB RAM as the data buffer in the test environment. With the cache implementation, the throughput of the VICS increased from 110MB/s to 130MB/s. The average response time of IO was reduced greatly, proving the cache mechanism of VICS can improve the SAN's performance.

In testing environment, all the servers with different OS could access LUN correctly, and the VICS could control the various storage systems well. This proves that the VICS is compatible with multi OS and with various storage

systems.

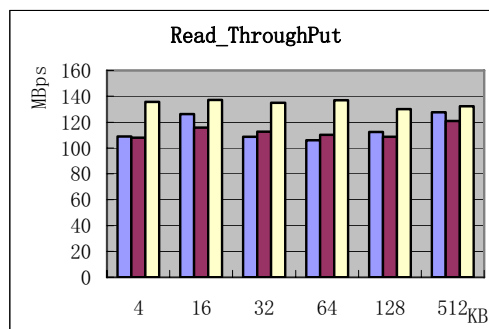


Figure 7 the Read Throughput

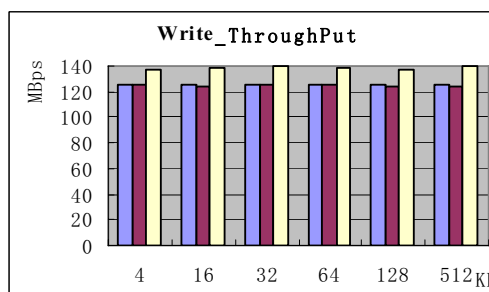


Figure 8 the Write Throughput

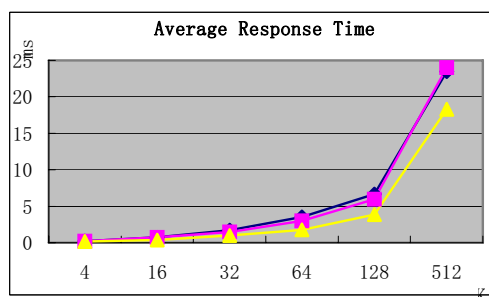


Figure 9 the average response time

## 5. Conclusion

This paper proposed a storage virtualization management system based on the storage area network. The new model includes three layers: the logical storage management layer, the virtualization layer and the storage resource management layer. The three layers, which have their own interface and functions, make up the VICS, which is a storage management system with high scalability and compatibility. Compared with other virtual storage systems for SANs, the VICS has some

obvious advantages:

(1) By splitting the SAN into a device SANs and a host SANs and introducing the network management layer, the VICS can control and manage the SANs more directly and easily.

(2) The VICS is compatible with multi operating systems. The VICS does not need an additional client/agent/driver to run on the hosts. This conserves the power of the hosts and predigests the management complexity.

(3) The VICS centralizes the storage resource and provides one uniform interface for the manager. It can manage the various storage resources and screen the differences between them for users.

(4) The VICS slightly influences the performance of the SAN, and the cache mechanism in the VICS can greatly improve the performance.

## Acknowledgement

The work described in this paper was supported by the National Key Basic Research and Development (973) Program of China (Grant No. 2004CB318205).

## Reference

- [1] B.Phillips, "Have storage area networks come of age?" [J] IEEE Computer, vol.31, no.7, 10-12, July 1998
- [2] R. Khattar, et al., *Introduction to Storage Area Network: Redbooks Publications (IBM)*,1999
- [3] XIOTech Corp., <http://www.xiotech.com/>, May 2004.
- [4] IBM Corp. <http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg245470.pdf>, March 2003,
- [5] EMC Corp., [http://www.emc.com/products/storage\\_manag](http://www.emc.com/products/storage_manag)

[ement/controlcenter/pdf/H1140\\_cntrlctr\\_srm\\_plan\\_ds\\_ldv.pdf](http://www.emc.com/controlcenter/pdf/H1140_cntrlctr_srm_plan_ds_ldv.pdf), May 2004.

- [6] David C. Teigland, Heinz Mauelshagen, *Volume Managers in Linux*, Sistina Software Inc. <http://www.sistina.com>,2001
- [7] Steven Pratt, *EVMS:A Common Framework for Volume Management*, Linux Technology Center, IBM Corp., <http://evms.sf.net>
- [8] StoreAge Networking Technologies Ltd., *High-Performance Storage Virtualization Architecture*, <http://www.store-age.com>, 2001
- [9] Intel Corp, "Intel iSCSI project", <http://sourceforge.net/projects/intel-iscsi>,2001
- [10] Ashish Palekar. *Design and Implementation of A Linux SCSI Target for Storage Area Networks*. Proceedings of the 5th Annual Linux Showcase & Conference. 2001
- [11] Sistina Software, Inc. *Global File Systems*, <http://www.sistina.com>
- [12] Chang-Soo Kim, Gyoung-Bae Kim, Bum-Joo Shin, *Volume Management in SAN Environment*. ICPADS 2001: 500-508. 1997
- [13] Teigland D. *The Pool Driver: A Volume Driver for SANs* [Master Degree Dissertation]. Minnesota: Department of Electrical and Computer Engineering University of Minnesota, 1999
- [14] Shu Ji-wu, Li Bigang, Zheng Wei-min: *Design and Implementation of a SAN System Based on the Fiber Channel Protocol*, IEEE Transactions on Computers, 54(4), 2005: p439-448
- [15] Kim, Chang-Soo; Bak, Yu-Hyeon etc: *A method for enhancing the snapshot performance in SAN volume manager*, 6th International Conference on Advanced Communication Technology, Broadband Convergence Network Infrastructure, 2004, p 945-948