ELVM: A LVM-based Remote Replication System

Xiaoguang Liu, Zhenhai Zhao, Gang Wang, Julei Sui, Yongzhi Song
Nankai-Baidu Joint Lab, College of I.T., Nankai University, Tianjin, China, 300071
{liuxg74, wgzwp}@yahoo.com.cn

ABSTRACT: Remote replication, also called remote mirror, is the key technology for data disaster tolerance. Replication system can be divided into synchronous and asynchronous mode according to whether the reply from remote server is required. In this paper, a remote replication system based on LVM (Logical Volume Manager) and NBD (Network Block Device) is presented in terms of the user requirements and Linux characteristic. Experimental results show that the system performance is acceptable both on LAN and WAN.

KEYWORDS: LVM; NBD; Replication; Linux

I. INTRODUCTION

Storage as a Service is popular as a business model, in which third-party providers rent space on their storage to end users that lack the capital budget and/or technical personnel to implement and maintain their own storage infrastructure. There is nothing new about Storage as a Service, but the complexity of today's backup, replication, and disaster recovery needs has renewed its popularity, particularly among small and medium-sized businesses. Building an enterprise data center (EDC) is the core issue to Storage as a Service.

Data corruption is one of the key problems to EDC. Data disaster tolerance technologies help EDC deal with data corruption by maintaining multiple copies on data centers which located in different place [1].

Remote replication, also called remote mirror, is the most important technology for data disaster tolerance. Here, remote replication means one or more than one copies are kept in different data centers. Meanwhile, remote replication also implies the maintenance of data consistency among all data centers.

According to the time that data be copied, remote replication can be divided into two categories, timing replication and real-time replication. Timing replication means the data will be copied in accordance with the plan. To real-time replication, the copy will be done immediately after the data are changed. Obviously, real-time replication can support stricter data consistency than timing replication. Real-time replication is more important to data disaster tolerance.

Continuous replication systems also can be divided two categories in terms of implementation. They are synchronous mode and asynchronous mode. Synchronous mode replication, also named mirror, requires data should be stored on source and target devices in the same time. Synchronous mode replication ensures data consistency completely. To asynchronous mode replication, a little difference may occur between the source and target devices during a period time.

Many works about data replication or recovery have been done. Some products provided by storage enterprises also appeared in market. Data replication can be implemented at block level [2, 3], file level [4], logical volume level [5] and database level [6].

In this paper, an asynchronous remote replication system based on LVM and NBD, named ELVM (Enhanced LVM), is designed and implemented in terms of the user requirements and Linux OS characteristic. The remote replication mechanism is embedded into the LVM module of Linux. ELVM uses the superior disk management capability of LVM to shield the design complexity and the difficulties caused by the discrepancy in physical storage device.

II. THE SYSTEM ARCHITECTURE

In ELVM, the remote replication is a module which is embedded into the LVM in Linux. Using the storage management of LVM, the discrepancy of storage devices and replication process are transparent to file systems and applications.

The basic operation principle is, the production center intercepts the update requests in LVM, send them to the I/O operation queue through the logical mapping in LVM, so the update for domestic is done. Meanwhile, the replications and requests are sent to the remote replication request queue in previous order. The requests should be sent to the disaster tolerance center by NBD (Network Block Device) protocol and the updates will be completed from production center to disaster tolerance center. As shown in figure 1,

Figure 1. The system architecture

Due to network delay, the write operation in production center is much quicker than that in disaster tolerance center. In ELVM, after a write request is completed, production center can continue to deal with other I/O requests. Production center need not care about whether the update in disaster tolerance center is completed or not.
III. DATA CONSISTENCY

In the design of remote replication system, the key problem is how to maintain the data consistency between production center and disaster tolerance center.

ELVM can works under both synchronous mode and asynchronous mode. When system is working under synchronous mode, the write requests order in disaster tolerance centers will be same to that in production center. As a result, the data will be copied to disaster tolerance centers from production center according to same request queue. Obviously, data consistency can be ensured under synchronous mode. But the system performance is poor under synchronous mode.

When system is working under the asynchronous mode, the write requests in disaster tolerance centers may be delayed because the data is transferred via network. The delay may cause the write queue is different between disaster tolerance center and production center. It will lead to data inconsistencies.

A case of asynchronous mode remote replication system is shown in figure 2. The write queue is WB1, WB2, WB3 and WB2. They will write data to logical block B1, B2, B3 and B2 respectively. It means the data in B2 is updated in period t. In production center, the write queue is completed by local storage system. The write requests are done according to the right queue. But in disaster tolerance centers, the two write requests to B2 are merged. So the write queue changes to WB1, WB3 and WB2. After B1 and B3 are transmitted from production center to disaster tolerance centers, B2 is not transmitted for some reasons, such as production center failed. In this case, the data between production center and disaster tolerance center will be inconsistent.

![Figure 2. A case of asynchronous mode](image)

Actually, if the dependent write requests are not completed by disaster tolerance centers totally, then the data in disaster tolerance centers will be marked as dirty. The dirty data will be recopied in the future. In asynchronous remote replication system, it’s difficult to determine the tradeoff between data consistency and system performance.

In ELVM, strict written consistency strategy is used for data consistency. This strategy means the entire write requests during a period time in production centers will be copied to disaster tolerance center according to same order strictly.

It also means the snapshots for production centers are kept by disaster tolerance center. After production centers failed, the data of previous period can be recovered by the snapshots.

The strict written consistency strategy is easy to be implemented. It’s only required to keep the written order. Because there is no write coverage optimization, the performance of strict written consistency strategy is weak in some cases.

IV. DATA REPAIR SYSTEM

After the failure of the network, the replication system will run under degrade mode. The data need to be copied will be marked. After the resumption of the network, these data will continue to be copied to disaster tolerance centers. In order to guarantee data consistency under this condition, a bitmap-based data repair system is used in ELVM. There are three parts in this bitmap-based data repair system. They are bitmap marking, detection of network connection and the maintenance of data consistency.

A. Bitmap marking

Bitmap uses a bit to express a certain space on disks. For example, one bit corresponds to 2 MB space on disks. During the process of data synchronization, if the process is interrupted by the network disconnection, then all the remaining data blocks will be marked as 1 in the bitmap. After the resumption of the network, all the data blocks marked as 1 will be copied by the repair process.

During the initialization of data repair system, the bitmap is created. After the repair job is completed and the replication system returns to normal mode, the memory space of bitmap is released.

B. Detection of network connection

How to detect the disconnection of network is an important problem to data repair system. There are many reasons that may result in disconnection of network. For examples, errors of physical link, server side anomaly, the client program anomaly and failure of TCP connection. Moreover, different reasons show different error information.

In some cases, the errors are returned by SEND function. In other cases, the errors are returned by RECEIVE function. Specially, these two functions may be deadlock in some special cases. It requires the remote replication system can treat the errors caused by different reasons.

In the replication system, a special method is used to detect the disconnection of network. There are only two processes to treat sockets in ELVM. One is send process of NBD, another is receive process of NBD. After any errors, which produced by the two processes, are detected, the system will try to connect the network again.

Meanwhile, the KEEPLIVE function, provided by TCP, is used to detect the status of network. The SEND or RECEIVE function will reports network errors if KEEPLIVE find some TCP error during a period of time.
C. The maintenance of data consistency

If the repair process or write requests want to do some I/O operations to a same data block in the same time, it may result in data inconsistency. For example, after a read request has been sent by repair process, another process with higher priority sent a write request to the same data block. In this case, the data that the repair process read isn’t the data which the repair process wants to read. In order to solve this problem, a data consistency protection mechanism is designed in ELVM. The main idea of this mechanism is that a hash table is used to record write requests. Before repair process doing data synchronization, it will check hash table firstly. If there are conflicts, repair process will wait until the conflicting threads complete their jobs. As shown in figure 3,

V. EXPERIMENTAL RESULTS

In order to test the performance of ELVM, we use some benchmark software, called Iometer[7] and WANem[8], to test system performance on LAN and WAN respectively.

A. Experiments on LAN

In the experiments on LAN, the tests are completed based on some servers which connected by Gigabit Switch. The server has an AMD Opteron processor with 1.8 GHZ, and 512 MB memory. The local data and remote data are stored on some SCSI disks which are integrated together by RAID 0.

Iometer is a special benchmark software, which distributed by Intel, to test the IO performance of system. By simulating the IO requests produced by practical applications, Iometer can evaluate the IO performance of system.

To evaluate the effect of request’s size, a performance experiment is done. The results are shown in figure 4. According to the results, with the increasing of request’s size, there is no obvious difference between local storage and remote storage.

Random write requests are used by many practical applications. As shown in figure 5, with the increasing of random write requests, there is no obvious difference between local storage and remote storage also.

B. Experiments on WAN

Just as its name implies, remote replication system is used on WAN in most cases. The experiments on WAN will show the performance of ELVM in reality.

A WAN simulator, called WANem, is used to simulate the action of WAN. WANem can simulate more than 10 physical networks. The performance of replication system in different physical networks is shown in figure 6.

Figure 3. the maintenance of data consistency

Figure 4. the effect of request’s size

Figure 5. the performance of random write

Figure 6. the performance on WAN
In figure 6, the left bar is the speed of remote replication system, the right bar is the maximal speed which the network can provide. As shown in figure 6, the use rate of network bandwidth near to 90% in most cases.

As shown in figure 7, an experiment to evaluate the effect of network delay on remote replication system is done. When the network delay is more than 5ms, the performance decreases obviously. It shows long network delay will hurt the performance of remote replication system.

![Figure 7. the effect of network delay](image1)

Packet loss rate also affects the system performance obviously. As shown in figure 8, with the increasing of packet loss rate, the speed of remote replication system decreases linearly almost. Specially, when packet loss rate increases to 1%, the system performance drops to 3 MB/s. It’s very poor.

![Figure 8. the effect of packet loss rate](image2)

**VI. CONCLUSIONS**

Remote data replication is the key technology to data disaster tolerance. In this paper, a LVM-based remote replication system, name ELVM, is presented. ELVM is embedded into the LVM module of Linux. It uses NBD as data communication protocol. Experimental results show the performance of ELVM is acceptable both on LAN and WAN.

In the future work, optimizations on communication and data consistency will be done to improve the performance on WAN.

**ACKNOWLEDGMENT**

This paper is supported partly by the National High Technology Research and Development Program of China (2008AA01Z401), NSFC of China (60903028), SRFDP of China (20070055054), and Science and Technology Development Plan of Tianjin (08JCYBJC13000).

**REFERENCES**


