2012 International Conference on Image, Vision and Computing (ICIVC 2012)
IPCSIT vol. 50 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V50.58

The Method of Multi-Point Data Backup in Disaster Recovery Systems

Zhong Hui-Hui[+] and Wang Zheng-Xia

College of Computer Science, Sichuan University, Chengdu, China

Abstract. A multi-point data backup method for disaster recovery systems is proposed in this paper. It can effectively increase the data redundancy and improve the system reliability by backing up the data to multiple copies stored in different places. In this method, first, the local production server backs up data to the primary backup server which is selected by the server's load, and then the other backup servers will back up the data from the primary server regularly .In this way, it not only improved the stability of the system, but also reduced the impact on the local server performance greatly.

Key words: disaster recovery system; multi-point backup; data security; load balance;

1. Introduction

With the rapid development of the economic and technological, information technology utilization continues to improve, more and more enterprises have realized the importance of data security. There were many kinds of disasters which cannot predict and avoid can cause damage or loss of corporate data. It will be enormous, even catastrophic [1] [2]. As a result, establish a sound disaster recovery system has become an urgent in today's information society, so that the integrity and security of data and the continuity and stability of business will be ensured [3].

Disaster recovery system establishes and maintains one or more redundant systems which are identical or similar to the original system. It uses of geographically dispersed and data system redundancy to withstand disaster [4]. Therefore, the key of disaster recovery system is to increase data redundancy. When disaster occurred in the system, the probability that all copies of the data were destroyed at the same time will reduce to an acceptable level. There were two ways as following to reduce the probability of this:

- Increase the number of copies. The more copies, the more difficult to destroy these copies at the same time and the lower the probability will be. For example, assume that each copy is stored in different hard drives, and an accident make one of the hard disk damaged. The probability of this cause one copy destruction is 0.1%.Then the probability of two copies damage at the same time was also reduced to 0.0001%.In this way, the probability of four copies damage was only 0.0000000001%.Thus increasing the number of copies have a significant effect to improve the ability of the system to against the data disaster caused by the hardware or media failure.
- Geographically dispersed. If all the copies of the data in the same room or the same building, it could not improve the disaster recovery capabilities for the disasters such as fires, earthquakes by increasing the number of data copies. In this case, the geographically dispersed of the copies will be more effective.

Based on these, a multi-point data backup method for disaster recovery systems (DMBM) is proposed. Compared to traditional disaster recovery systems, this method can effectively increase the data redundancy

[+] Corresponding author. Tel.: +13551063807.
E-mail address: huihui_zhong@163.com.

by increasing the number of backup copies and distance between copies. It also can improve the reliability and stability of the system, and reduce the impact on the local server performance greatly.

2. Overview of DMBM

2.1. Architecture

In DMBM, the local production server connects to every remote backup server through the Internet. Each local production server with multiple remote backup servers forms a multi-point structure. The relationship of the local production server and the remote backup server is M to N. First, backup the data from the local production server to a remote backup server. Next, backup the data from this remote backup server to the others. Then the data of the local production server has been backed up to several remote backup servers. The system architecture of DMBM is shown in figure1.

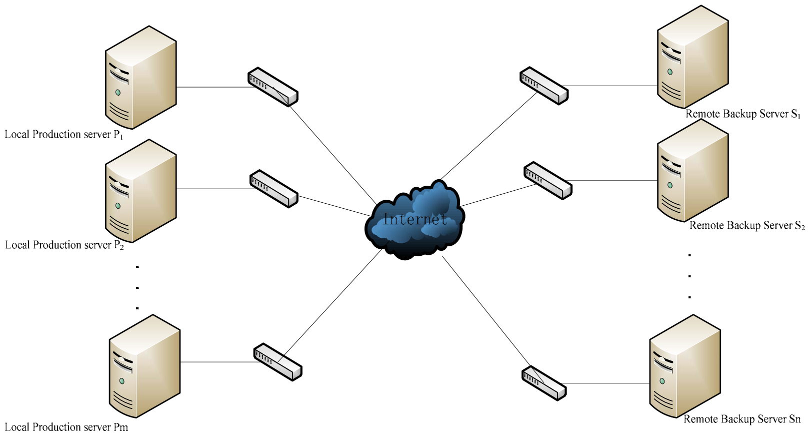
Local Production server P_1
Local Production server P_2
Local Production server Pm
Internet
Remote Backup Server S_1
Remote Backup Server S_2
Remote Backup Server Sn

Fig. 1 DMBM System Architecture

2.2. Definition

- Source Data Node: In disaster recovery system, data of the local production server is backed up to each remote backup server to save copies. The local production server is the source of the data, so it is called source data node.
- Primary Backup Node: The remote backup server which is selected to backup the data of the local production server first, takes charge of backing up data to the other remote backup server. This special remote backup server is called the primary backup node.
- Copy Nodes: The other backup servers except Primary Backup Node.
- Massive Cache: The method to store the backup data of the local production server temporarily using the external memory.
- Digest Value: The only value used to mark the data for checkout. It is calculated with using algorithm such as MD5、CRC、SHA-1 and so on. In DMBM, we use MD5 algorithm for this.
- Error Control: The way to check whether an error occurred during the network transmission of the data [5]. It works as compare the digest value of the data that it had received with the digest value it received.

2.3. Basic Thought

The basic thought of DMBM is shown in figure2. The source data node selects the primary backup node by the loads of the remote backup servers. After the initialization is achieved, data changes of the logical volume will be monitored in real-time, and sent to multiple backup servers, then replayed on the primary backup node. And back up data to the primary backup server is achieved. The different data is backed up from the primary backup node to the copy nodes regularly, when the primary backup is seldom used such as night. In this way, we complete the data backup of the source data node.

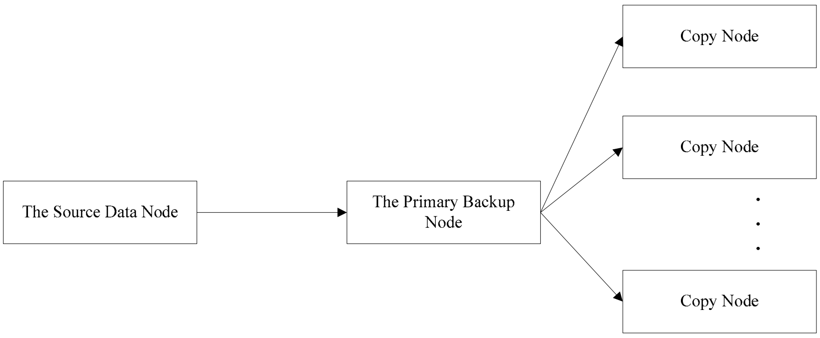

Fig. 2 The basic thought of DMBM

3. Method process

3.1. Initialization

The main work of initialization is to obtain the load of each remote backup server, and select the primary backup node, then synchronize the data of source data node and the primary backup node.

How to determine the load of the remote backup server is the key of MDBM. In order to quantify the load of the remote backup servers, assuming each load of the N remote backup servers is $Load_i(t)$,i=1,2,···, N. $Load_i(t)$ expresses the load of backup server i at time t. Select the CPU utilization $C_i(t)$,memory usage $M_i(t)$,disk IO access efficiency $IO_i(t)$, total number of processes $Pr_i(t)$, and request response time $Rt_i(t)$ as load detection information. Then the load of each remote backup server can be calculated as:

$$Load_i(t)= \mu_1C_i(t)+ \mu_2 M_i(t)+ \mu_3 IO_i(t)+ \mu_4 Pr_i(t)+ \mu_5Rt_i(t), \quad i\in 1,2,\ldots,N \tag{1}$$

$$\sum_{i=1}^{5}\mu_i = 1 \tag{2}$$

Where, μ_1、μ_2、μ_3、μ_4、μ_5 represent the load impact factor of the CPU utilization, memory usage, disk IO access efficiency, total number of processes, and request response time. In different environments, the parameter has different influence about load. Therefore, the value of each factor based on specific environmental should be analysis and comparison to determine by experiments.

At the beginning, select the remote backup server whose load is the lowest as the primary backup node. To ensure the data of the source data node and the primary backup node are consistent, it should synchronize the data of source data node and the primary backup node. Read the data in the volume of the source data node. Then the backup data、disk offset and other information are encapsulated in packets and sent to the primary backup node. After the primary backup node gets and analysis the data packets, writes data in its corresponding offset on the volume. The initial data of the primary backup node and the source data node is consistent after the initialization complete.

3.2. Data Backup To the Primary Backup Node

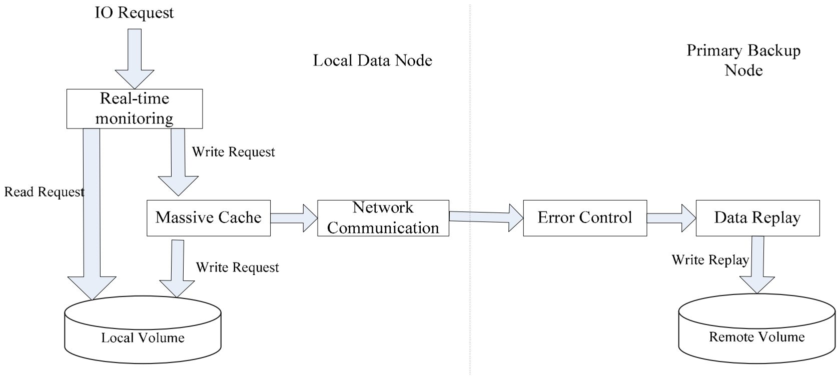

Fig. 3. Local Data Node backup to Primary Backup Node process

After the primary backup node is selected, data changes of the source data node's logical volume will be monitored in real-time [6], and sent to the primary backup server, then replayed on the primary backup node. Because the external memory can store much more data than the memory, we use the massive cache to store the monitored data temporarily. Then send the data from the massive cache to the primary backup server. In this way, it can not only improve the speed of the backup task, but also reduce the impact on the local server performance greatly.

The backup progress is shown in figure 3, specific steps are as follows:

1) The source data node is monitoring the data changes in real-time using kernel-driven [7]. Capture system or the user's IO request packet to analyze whether it's the read or write requests. For a read request, send it directly to the logical volume to read. And send the packet into the massive cache for a write request.

2) Read data packets from the massive cache. Calculate its digest value using the MD5 algorithm. The packets are sent to the primary backup node via TCP after re-packaged with their digest value. Then the packets are formatting as:

$$\{<tid,offset,len,data,md5v>\}$$

Where, *tid* is the unique identifier of the backup task. The *offset* intercepts the location of the data to be written in the logical volume. The *len* shows the length of the data. The *md5v* expresses the digest value of the data. It is calculated as:

$$\text{md5v=MD5 (tid + offset + len +data)} \qquad (3)$$

3) Primary backup node receives the packets which are sent by the source data node. unpackage the packate and check whether data error by comparing the digest. Calculate its digest value using algorithms as the second step. If it is equal to the digest value recorded in the packet, it shows that the backup data is correct. Otherwise, discard the packet and ask the source data node to resend the packet.

4) The kernel creates a write request according to the valid information that received by the primary backup node. Then write the backup data in the corresponding offset of the volume. After the data replayed on the primary backup node, the data is backuped to the primary backup node.

3.3. Data Backup to the Copy Nodes

At the time that users operate seldom, the data in the primary backup node will be backed up to other remote backup servers. When the first backup, there are no backup data in every copy node. The valid data of the primary backup node should be copied into every copy node completely. Only backup the different data between the primary backup node and each copy node when do another backup.

3.3.1. The First Backup

In the first backup, there is a great difference between the primary backup node and each copy node. Copy all the data to the remote, in order to ensure that the data of each copy node and the primary backup node is exactly consistent. The realization is shown in figure 4.

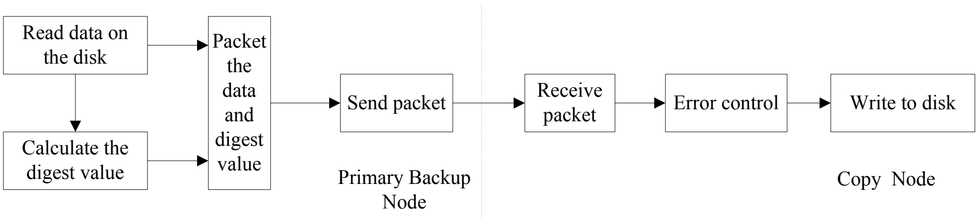

Fig. 4. The process of the first backup

The primary backup node reads the data on the logical volume with a read thread. Then calculate the digest value of each data block for error checking. The packets are sent to the copy nodes using the multiple sending threads which correspond to each copy node. Each backup node calculates the digest value of data that they received, and compares with the digest value in the packet. If they are equal, write the backup data into each backup node's disk. The backup task is achieved until all the backup data is copied to the copy nodes.

3.3.2. Another Backup

When backup again, considering that there is only little part of the difference between the primary node and each copy node. We only backup the difference between the primary node and each copy node. The realization is shown in figure 5.

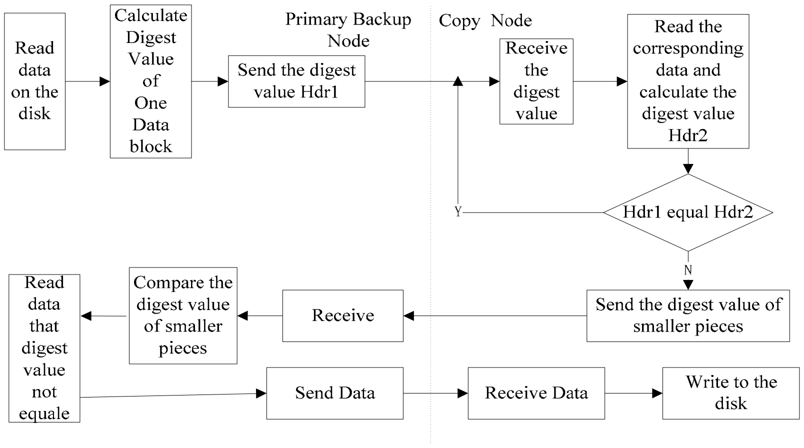
Read data on the disk
Calculate Digest Value of One Data block
Primary Backup Node
Send the digest value Hdr1
Copy Node
Receive the digest value
Read the corresponding data and calculate the digest value Hdr2
Hdr1 equal Hdr2
Y
N
Send the digest value of smaller pieces
Receive
Compare the digest value of smaller pieces
Read data that digest value not equale
Send Data
Receive Data
Write to the disk

Fig. 5 The process of another backup

The primary backup node reads the data of volume into data block. And calculate the digest value of each data block. Send the data blocks and their digest value to each copy node. Each copy node reads the corresponding data block and calculates its digest value. If it's equal with the digest value of primary backup's block, this data block does not need to be backed up. Otherwise, the data block will be divided into smaller pieces. Calculate the digest value of each data piece and send to the primary backup node. The primary backup node receives the data packet and reads each corresponding piece data. Then calculate the digest value of each piece. Read the piece data of the volume and send those pieces whose digest value is not the same as the copy node sent. The copy node receives the data pieces and writes the backup data into the volume. The backup task is achieved until all the backup data is copied to the copy nodes.

4. Conclusion

A multi-point data backup method for disaster recovery systems is proposed. The source data node selects the primary backup node by the loads of the remote backup servers. First the data is backed up to the primary node. In the time that users operate seldom, backup the data of the primary backup node to each copy node. Then a multi-point data backup is achieved. It will not only increase the number of copies, but also increase the distance between each copy in this method. At the same time, the system's load balancing can be ensured in this method as taking into account the server's load. What's more, the impact on the local production server is reduced and insures the system stability.

5. References

[1] Velpuri, Rama. Oracle backup & recovery handbook [M], 1997.

[2] Paul Hrabal, Disaster Recovery White Paper[DB/OL], www.usdatatrust.com, 2004

[3] P. Fallara, Disaster recovery planning [J], IEEE Potentials, 2003, 22(5).

[4] R. Cegiela .Selecting technology for disaster recovery[C] //Proc of the International Conference on Dependability of Computer Systems 2006:160-167.

[5] E. Serrelis, N. Alexandris Fault tolerant production infrastures in practice[C]//The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communicatons 2007:1-5.

[6] JF. Lennert, W. Retzner,et al. The automated backup solution-safeguarding the communications network infrastructure[J].BellLabs Technical,2004,9:59-84

[7] R. Mark, S. David Microsoft Windows Internals [M]. 5th Edition. Washington: Microsoft Press, 2009.

[8] Silver Peak System Inc. An Overview of SliverPeak's WAN Accleration Technology [DB/OL].http://www.silver-

peak.com 2008

[9] K. Keeton, A. Merchant. A framework for evaluating storage system dependability[C]//Proc of International Conference on Dependable

[10] S J. Lloyd, P. Joan,L. Jian, et al. RORIB: An Economic and Efficient Solution for Real-time Online Remote Information Backup[J]. Journal of Database Management, 2003, 14(3): 56-73.