2012 International Conference on Image, Vision and Computing (ICIVC 2012)
IPCSIT vol. 50 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V50.17

An Improved Method of File-set Archive

CHEN Shi-jue[+]

Dept. of Computer Science, Sichuan University, Chengdu, P.R.China

Abstract. To improve the efficiency of data recovery of differential archive method, an improved method of file-set archive is proposed. This method use a total file-set to store all files in every archive point without duplicate, and the method also save the mark file of the file-set in every archive point. During recover the file-set of an archive point, the method analyzes the mark file of target archive point, then reconstruct the file-set of target archive point from the total file-set. This method can recover every archive point by doing only once reconstruction. Experiment shows that, compared with the differential archive method, the method is more stable, and it can improve the recover efficiency obviously.

Keywords: Differential Archive; File-Set ; Mark File ; Archive Point

1. Introduction

As increasing attention of information security by enterprise and individuals, Disaster Recovery System becoming more popular. It is normal that the server of Disaster Recovery System has a large number of historical documents. In order to manage those historical documents efficiently, an improved method of file-set archive is proposed.

In the specific user environment，such as banks, government, financial system，the data are often composed in the form of a large number of small files. In order to achieve higher efficiency of data backup and archive management, it is wise to save this data in the form of file-set.

Traditional archiving methods can be divided into two categories. The first method keeps complete data of every archive points，called Complete Archive Method；the other method is called Differential Archive Method，which save the complete file-set of the first archive point or the last archive point, while other archive point only save the differential data.

The former method called Complete Archive Method keeps complete data of every archive points，during recover, there is no need to reconstruct the target archive point, so it is fast and efficiency. But it consumes lots of storage space.

The latter method called Differential Archive Method, which can divide into two categories. One is Positive Differential Archive Method, which keeps complete data of the first archive point and differential data of other archive points, when recovers the first archive point, there is no need for the method to do any reconstruct, so it is fast. but when recovers other archive point, reconstruction needs to be done from the first archive point to the target archive point, the latter the target archive point is, more time will cost. The other Differential Archive Method is called Reverse Differential Archive Method, in contrast with the positive one, keeps complete data of the latest archive point and differential data of other archive points, so it is fast to recover the latest archive point, but the former archive point is, more time the recover will cost.

In the specific user environment as bank, government, financial system, in those systems, random archive point is often required. If the Differential Archive Method is used in this kind of system, the cost time of recovering different archive point will be different, so the efficiency will be low. In order to solve the

[+] Corresponding author.
E-mail address: 598324626@qq.com.

problem, this paper proposed an improved archive method. This method can cover every archive point by doing only once reconstruction, the cost time of recovering different archive point will be stable. This method is more suitable for the above issue.

2. Differential Archive Method

2.1. Terms

1) f_j represent a file which user has archived, $F_i = \{ f_j \mid j = 1,2,3...n \}$ represent the file-set of archive point i, which is made up of $f_1, f_2, f_3 ..... f_n$. $X_i = \{ f_j \mid j = 1,2,3...n \}$ represent the XML file of file-set F_i, which means $f_1, f_2, f_3 ..... f_n$ are in the file-set F_i。
2) $\Delta F_k = F_k - F_j$ （$k > j$）represent the positive differential of F_k to F_j; $\Delta F_k' = F_k - F_j$ （$k < j$）represent the reverse differential of F_k to F_j.
3) Suppose $\Delta F_k = F_k - F_j$ （$k > j$）, $\Delta F_k = \{+ f_i\}$ means f_j is in file-set F_k but not in file-set F_j, f_j is added into F_k. $\Delta F_k = \{- f_i\}$ means f_j is in file-set F_j but not in file-set F_k, f_j is deleted from F_k.
4) A_i represent archive point i
5) $sizeof(f_i)$ indicates the storage space f_i consumes, $sizeof(F_i)$ indicates the storage space F_i consumes.

2.2. Reverse Differential Archive Method

Reverse Differential Archive Method is more efficient, it is more widely used.

Reverse Differential Archive Method keeps the complete file-set of latest archive point called F_c, but only keeps the reverse differential of previous archive point. For example, archive point A_i only keeps the reverse differential of A_{i+1} to A_i, which is represented as $\Delta F_i' = F_i - F_{i+1}$.

When a archive point is wanted, such as A_i. The method reconstructs A_{n-1} by using of F_c and $\Delta F_{n-1}'$, then reconstructs A_{n-2} by using of A_{n-1} and $\Delta F_{n-2}'$, and A_{n-3}, A_{n-4} …… until A_i.

Assume the archive server has n archive points, the storage structure is like Figure 1.

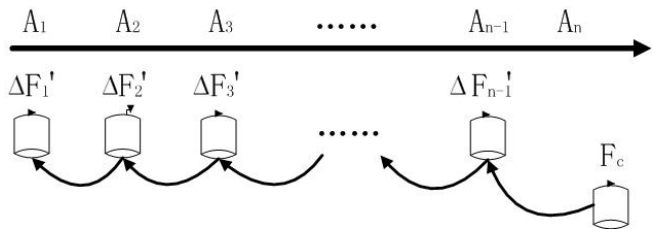

Figure 1 storage structure of Reverse Differential Archive Method

Reverse Differential Archive Method is efficient when we recover the latest archive point. But in the specific user environment where random archive point is often required, Reverse Differential Archive Method is not suitable. In order to solve the issue above, the improved archive method is designed.

3. An Improved Method of File-set Archive

3.1. System structure of the method

System structure of the method is like Figure 2.

Figure 2 system structure of the method

The client transmits archive data to the archive server when it has important data to be backup. The server accepts the data and archive them. When the client loses the important data, it can recover them from archive server. That is how the system works.

3.2. The main idea

The method keeps all files in every archive point in a file-set called F_t, there is no duplicate file in F_t. For example, at the first backup, the client transmits $F_1 = \{f_1, f_2, f_3\}$ to the server, the server set $F_t = \{f_1, f_2, f_3\}$; at the second backup, the client transmits $F_2 = \{f_3, f_4, f_5\}$ to the server, then the server set $F_t = \{f_1, f_2, f_3, f_4, f_5\}$.

Every time after client transmits backup data, the server create a archive point to store the XML file for the file-set has just accepted, then add new files into F_t. Such as, the i th backup, after client transmits data, create archive point A_i to restore X_i, then add new files into F_t.

When the client sends a recover request. For example the client send a request to recover A_i,the server accept the request and begin to analyze X_i, gets all files whose are marked in X_i from F_t, then create the target file-set F_i by using of the files before. After the steps above, transmits F_i to the client and complete the recover process.

3.3. Basic process

1) Backup process

The first backup process is complete backup. For example the client scans the target file-set $F_1 = \{f_1, f_2 ..... f_m\}$ which will be transmits to the server, calculates the mark file $X_1 = \{f_1, f_2 ..... f_m\}$, then transmit F_1 and X_1 to the server at the same time. The server accepts those data, create archive point A_1 to store X_1, set $F_t = F_1 = \{f_1, f_2 ..... f_m\}$, complete the first backup.

In the second backup process, the client scans the target file-set $F_2 = \{f_3, f_4 ..... f_m, f_{m+1}\}$, calculates the mark file $X_2 = \{f_3, f_4 ..... f_m, f_{m+1}\}$, then calculates the differential between F_2 and F_t, which is called $\Delta F_2 = F_2 - F_t = \{-f_1, -f_2, +f_{m+1}\}$. After all those done, the client transmit ΔF_2 and X_2 to the server. The server create archive point A_2 to restore X_2, then process ΔF_2, deletes f_1 and f_2 from F_t, and adds f_{m+1} to F_t.

The following backup process is similar with the second one. Assume n times of backup was done, there are n archive points which are A_1、A_2 ….. A_n, every archive point store mark file X_i in them. Figure 3 shows the storage structure in this case.

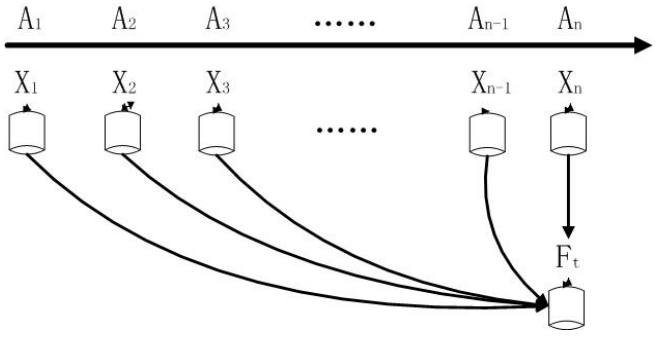

Figure 3 storage structure

2) Recover process

Assume n times of backup was done, there are n archive points which are A_1、A_2 ….. A_n, and A_i is required. In the recover process, analyzes X_i first, gets all files whose are marked in X_i from F_t, then create the target file-set F_i by using of the files before. Then transmits F_i to the client and complete the recover process. The process shows in Figure 4.

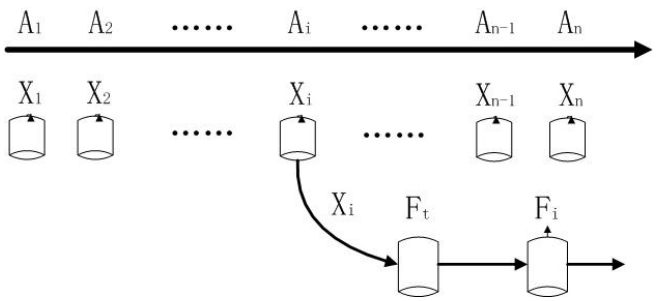

Figure 4 the process of recover A_i

3.4. The mark method of file-set

In order to represent the file-set clearly, and simplify the mathematical operation, we use XML to mark the file-set. The method is below.

The mark of file-set

<tree >…… </ tree> tag is used to mark file-set

The mark of folder

<directory fpath='path'>…… </directory> is used to mark folder， path represents the relative path in F_t.

The mark of file

<file> path</file> is used to mark file, path represents the relative path in F_t.

Such as， in archive folder /archive, there is a archive point A_i, which stores $X_i = \{f_1, f_2 ..... f_m\}$ for $F_i = \{f_1, f_2 ..... f_m\}$. The absolute path of F_t is /archive/total/. X_i will be like below.

<tree >
<file>f1</file>
<file>f2</file>
……
<file>fm</file>
</tree>

3.5. Recover method for archive point

Every archive point has its own mark file X_i, the files marked in X_i are all existed in F_t. In the recover process, we should analyze X_i, then get all files marked in X_i from F_t, after this, construct target file-set F_i by using the files below. The server transmits F_i to client to complete the recover process.

For example， to recover A_i， whose mark file is X_i, see X_i below.

<tree >
<file>f3</file>
<file>f4</file>
<file>fm-1</file>
<file>fm</file>
</tree>

Assume $F_t = \{f_1, f_2 ..... f_m\}$。During recover, get f_3, f_4, f_{m-1}, f_m that are marked in X_i from F_t, then create target file-set $F_i = \{f_3, f_4, f_{m-1}, f_m\}$.After this, transmit F_i to the client to complete the process.

Compared with Differential Archive Method, this method has the similar operation in the data backup. But during recover, the method is more efficient by doing only once reconstruction.

4. Experiment

4.1. Experimental environment

Experimental environment is CPU：Inter(R) Celeron(R) CPU 3.33GHz、Main Memory 504M、Hard Disk HDS728080PLA380、Operation System MS Windows XP SP3, File System is NTFS.

4.2. Experimental process and results analysis

The experiment set 10 archive points ($A_1 \sim A_{10}$), compare the recover time of every archive points with the Reverse Differential Archive Method.

The experiment use a test file-set F , which consume 100M storage space. Each archive point has 10% random change with adjacent archive point.

During the recover of the 10 archive point （ $A_1 \sim A_{10}$ ） , we have recorded the time consumed by each archive point. Table 1 shows the time of the two methods. Figure 5 shows the comparing time of the two method.

Table 1 the time of recover archive point

Archive Point	Reverse Differential Archive Method	This method
	Time T(s)	Time T(s)
A_1	32.27	26.35
A_2	31.22	25.92
A_3	30.61	26.07
A_4	29.56	26.13
A_5	29.13	25.96
A_6	28.57	26.04
A_7	27.71	26.88
A_8	26.64	26.45
A_9	26.32	26.33
A_{10}	25.56	25.98

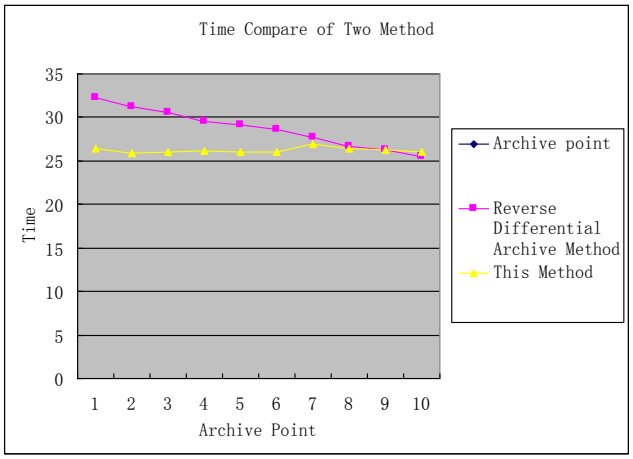

Figure 5 Time Compare of Two Method

As it is shown above, the two method has similar efficiency when recovering A_{10}, which is the latest archive point. Because the Reverse Differential Archive method keeps complete data of the latest archive point. But as we can see in Figure 5, the older the archive point is the lower efficiency the Reverse Differential Archive method has.

Compared with the Reverse Differential archive method, this method is more stable. The method has similar efficiency of recovering each archive point as recovering latest archive point A_{10}. Because only once reconstruction is needed when recovering any archive point. In the specific user environment, which random archive point is often required, this method is more suitable than Differential archive method.

5. Conclusion

Increasingly high degree of information technology in modern society, the protection of the data is more important. In different environments, different archive method is used. This method is has high efficiency in specific situation. I believe that in the future practical application, this method will be increasingly used.

6. References

[1] Li Tao. Introduction to Network Security [M].Beijing: Electronic Industry Press,2004. 474-490

[2] Cerra, D. Datcu, M. A Model Conditioned Data Compression Based Similarity Measure. Data Compression Conference,2008.DCC 2008 25-27 March 2008 Page(s): 509-509

[3] Behnke, J.; Watts, T.H..EOSDIS petabyte archives: tenth anniversary. Mass Storage Systems and Technologies, 2005. Proceedings. 22nd IEEE / 13th NASA Goddard Coference on 11-14 April 2005 Page(s):81 - 93.

[4] Cunhua Q.Syouji N, Toshio N. Optimal Backup Policies for a Database System with Incremental Backup[J]. Fundamental Electronic Science, 2002, 85 (4): 1-9.

[5] Rizwana Mehboob,Shoab A.Khan, Zaheer Ahmed.High Speed Lossless Data Compression ArchitectureMultitopic Conference, 2006. INMIC '06. IEEE 23-24 Dec. 2006 Page(s):84 (R) 88.

[6] Andrew Tridgell Efficient algorithms for sorting and synchronization[D] Canberra The Australian National University 1999.