Journal of Theoretical and Applied Information Technology
15th April 2012. Vol. 38 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FINGERPRINT LOOKUP-AN EFFECTIVE AND EFFICIENT BACKUP USING DEDUPLICATION TECHNIQUE

[1]VELVIZHI J, [2]BALAJI C G

[1]M.Tech Student, School of Computing, Sastra University, Tamilnadu, India

[2]Asstt. Prof., School of Computing, Sastra University, Tamilnadu, India

E-mail: [1]jvelvizhi@gmail.com, [2]cgbalaji@cse.sastra.edu

ABSTRACT

Data deduplication avoids redundant data in the storage of backup operation. Deduplication reduces storage space and overall amount of time. In this system, files that contain data are split into chunks by using context aware chunking and fingerprints lookup to each chunk. Backup storage process is for avoiding duplicate data using fingerprints lookup. In this paper, we compare three methodology of backup systems such as full backup, cumulative incremental backup and differential incremental backup. Full backups contain all data file blocks. Cumulative incremental backups contain blocks from level *n*-1 or lower. Restoration speed is faster than differential incremental backup but storage space occupy much more. Differential incremental backups contain only modified blocks from level *n* or lower. The processes of differential incremental backup in which data objects changes made since the last full backups are copied.

Keywords: *Deduplication, Chunking, Fingerprint lookup, LRU Table Management.*

1. INTRODUCTION

The recent introduction of digital TV, digital camcorders, and other communication technologies has quickly accelerated the quantity of data being maintained in digital form. In 2006, for the first time ever, the total volume of digital contents exceeded the global storage capacity, and it is estimated that by 2011 only half of the digital information will be stored. Further, the volume of automatically generate information exceeds the amount of human generate digital information. Combining the problem of storage space, digitized information has a more fundamental problem: it is more vulnerable to error compared to the information in legacy media, e.g., paper, book, and film. When data is stored in a computer storage system, a single storage error or power failure can put a large amount of information in danger.

To protect against such problems, a number of technologies to reinforce the availability and reliability of digital data have been used, including mirroring, replication, and adding parity information. In recent times there are need more requirements to backup such as database, email, file server, web servers, and transaction servers. Now a day's backup accepted as standard method by many industries to protect the important business and enterprise data. To reduce backup storage capacity, the deduplication mechanism is widely used in the traditional backup system.

Data deduplication, in computer storage, refers to the elimination of redundancy in data backup storage. In the deduplication technique, singular chunks of data, or byte patterns, are chunked and stored during a process of analysis. As the analysis, chunks are compared with other already stored chunks and whenever a similarity occurs, the redundant chunk is substitute with a small reference that points to the stored chunk. The match frequency of chunk is a factor of the chunk size occurs dozen, hundreds and thousands of times, in this process the amount of data that must be stored or transferred can be reduced much more. Divide object into logical segments called chunks. Identify duplicate chunks using hash function for each chunk to produce unique identifier. Compare each chunk identifier with index to determine whether chunk is already stored.

LRU (Least Recently Used) technique is used for table management and bloom filter is used to check whether fingerprint available or not in the table. In this system, there are three methods used for backup operation - full backup, differential incremental backup, cumulative incremental backup.

Journal of Theoretical and Applied Information Technology
15th April 2012. Vol. 38 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. RELATED WORKS

The three existing technique are delta encoding, duplication elimination and compression that are used independently or combined to reduce the space efficiently and network bandwidth utilization. Duplicate data elimination is a method that identify and coalescing data block. In this design it combination of content based hashing, copy-on-write and lazy update. For an ease implementation it is also build on Storage Tank's Flash Copy function [1]. Prun system eliminates redundancy information from intra-file and inter-file [2]. From prun system adopts filter based main memory index lookup structure to minimize restructuring of on-disk overhead and improve buffer cache miss rate [5].

An ADMAD scheme makes use of different file chunking method based on certain metadata to reduce inter-file level duplication. In this ADMAD process speedup I/O performance and as well as ease the data management also [3]. Compare-by-hash is techniques frequently read or write data that is identical to already existing data. Disk-based deduplication is a storage the data centers that perform weekly backup from primary storage system to secondary storage [6]. Pastiche techniques achieve excess disk capacity to perform peer-to-peer backup and with no administrative costs [13]. However, we also found that chunking significantly increases the fingerprint lookup overhead. By increasing the target pattern size from 11 bits to 13 bits, the deduplication detection rate decreased by two percent and the chunking performance decreased from approximately 150 MB/sec to 100 MB/sec with files being in memory. However, the overall deduplication speed increased from 51 MB/sec to 77 MB/sec. Normal backup occupy much more storage space than the deduplication backup [7].

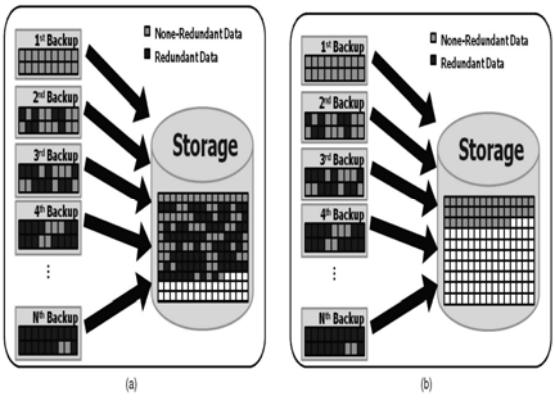

Figure.1: *Data Deduplication. (a) Normal Backup. (b)Deduplication Backup*

3. SYSTEM ORGANIZATION

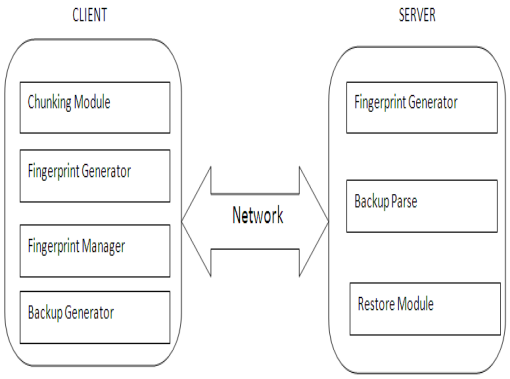

Figure.2: *System Organization*

In this design the term client and server technology of backup data stream. There are four modules from client side are chunking module, fingerprint generator, fingerprint manager and backup generator. Chunking module partitions the file into number of chunks. Each chunk has unique identification known as fingerprint. Fingerprint manager is responsible for insertion, deletion of fingerprints into the table and for searching. Backup generator transfers the backup data.

The server side contains fingerprint generator, backup parse and restore module. Backup parse receive backup history. Restore module is used for restoration process. Fig.2 illustrates overall system organization of PRUN.

4. CHUNKING MODULE

Chunking module helps to partition the file into number of chunks. Each piece of file is known as a chunk. The four types of chunking are whole file chunking, fixed size chunking, variable size chunking and format aware chunking. The processes are defined.

(a) Whole file chunking

Each file is treated as a single chunk. No detection of duplicate data at sub files level is done.

(b) Fixed-size chunking

Chunk boundaries occur at fixed intervals, irrespective of data content. This method is unable to detect duplicate data if there is an offset difference, because redundant data is shifted due to insertion/deletion and redundant data is embedded within another file or contained in a composite structure.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(c) Variable-size chunking
Rolling hash algorithm is used to determine chunk boundaries to achieve an expected average chunk size and it can detect redundant data, irrespective of offset differences. Often referred to as fingerprinting (e.g., Rabin fingerprinting).

(d) Format-aware chunking
In setting chunk boundaries, this algorithm considers data format/structure for example: awareness of backup stream formatting; awareness of PowerPoint slide boundaries; awareness of file boundaries within a composite.

5. FINGERPRINT GENERATOR

Fingerprint generator is used in data to determine a unique signature for each chunk. Fingerprint algorithm maps an arbitrarily large data item (such as a computer file) to a much shorter bit string, its fingerprint that uniquely identifies all data blocks (chunks). Signature values are compared to identify all duplicates. Fingerprint generates for each chunks and it send to fingerprint manager.

6. FINGERPRINT MANAGER

Fingerprint Manager is responsible to client and server side to deduct the redundant data. Manger is responsible for insert chunk to the repository. In the existing system it is used to maintain the table using array of pointer whose entry pointer to individual table. In our system, we use fingerprint manager at the client and server side that contain fingerprint value to detect the duplicate data.

7. TABLE MANAGEMENT

In List of tables current fingerprint is inserted.

7.1 LRU-Based Table Management

Least Recently Used (LRU): select the item which is least recently used. In this process it requires tracking of what was used and when it was used, in this process that make sure that it always selects the least recently used item. It is used to maintain the list of fingerprint in the tables, to reduce the number of tables to examine. It maintains the temporal locality on fingerprint search. The recently hit table moved to the head of the list.

7.2 Bloom Filter:

Bloom filter is verified whether fingerprint available or not in the table. The need of storage space, bloom filter is compared to stored set of fingerprint table. It checks whether the given fingerprint independently contained in the fingerprint table. If it contain in the fingerprint table it is known as positive, otherwise it is known as false positive and then it is insert into the fingerprint table.

8. BACKUP STORAGE:

There are three types of backup storage, Full backup storage, Cumulative increment (CI), Differential incremental backup (DI). In this paper we evaluate these backup storage techniques.

8.1 Full Backup Storage

Full backup storage is a starting point of other types of backup. It backup all files and folders that are selected. In this method, we cannot use deduplication techniques as they cannot be successful and it occupy more disk space as shown in the below figure.

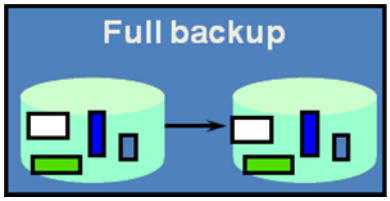

Figure.3: *Storage for Full Backup*

8.2 Cumulative Incremental Backup

A backup data is altered since the last full backups are copied. The most recent full backup as well as every incremental backup made since the last full backup is needed. The cumulative incremental backup process that backup's up all chunks that are made after a starting from level 0 backup. The main disadvantage of cumulative backup over differential is more disk usage as shown in the figure.5.

There are three cumulative incremental backup:

Direct Cumulative Incremental: Primary Storage on a target system copy directly to the attached Secondary Storage on the target system in cumulative incremental backup.

Network Cumulative Incremental: Primary Storage on a target system copies to Secondary Storage through network.

Synthetic Cumulative Incremental: In Cumulative Incremental ,synthetic cumulative incremental is one of the special types that will merge some or

Journal of Theoretical and Applied Information Technology
15th April 2012. Vol. 38 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

else all of the files that have been created, in this backup system from most recent level n-1 and at the lower level as shown figure.4. In this process of backup system contain without any interaction of primary storage. It contains same data that have been taken from the target system at the period of time.

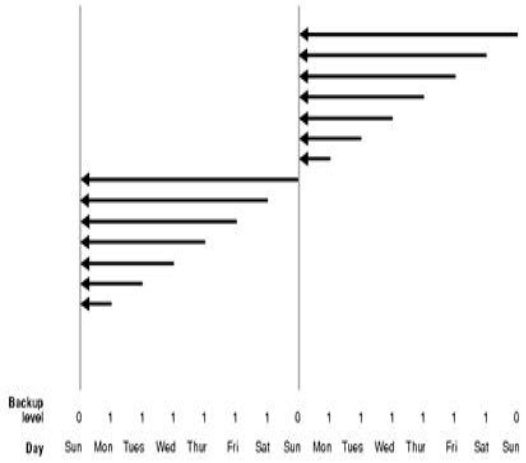

Figure.4: *Cumulative Incremental Backup*

In this cumulative incremental backup storage system, full backup is taken along with the recently updates - it cannot avoid duplication. So it is not a successful deduplication technique.

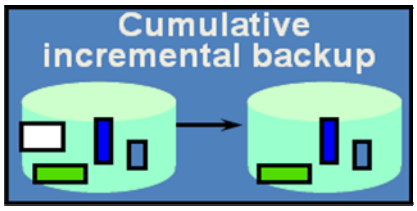

Figure.5: *Storage for Cumulative Incremental Backup*

8.3 Differential Incremental Backup:

Differential backup contain all modified files that were made till last full backup. The default technique in differential backup takes copy from last level 1 or level 0 backup. In this system, storage speed is faster than other methods because less data block is stored, as shown in figure.5. For a complete restore, the latest full backup and the latest differential backup are needed. Backup speed, restoration speed, storage speed backup are medium in this process. Using deduplication technique in this method, we can avoid duplicate data. In differential incremental backup it is sufficient to use deduplication techniques, for example we are taking backup weekly. On Friday, the full backup ABCDEFGH is taken then on Monday ABJF is taken as the data. For avoiding the redundancy of these data, ABJF is compared with the previous data ABCDEFGH and the result J is stored as backup. This process continues for rest of the days in that week as shown in the table.1.

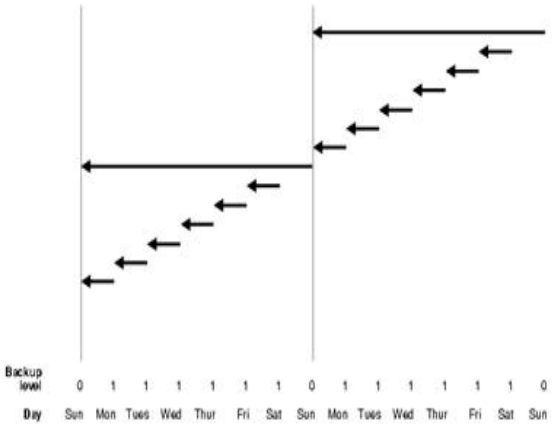

Figure.6: *Differential Incremental Backup*

Table.1: *Example differential incremental backup using de-duplication technique*

DATA	BACKUP ORDER	DAYS	STORAGE DATA
ABCDFAGHI	Full Backup	Friday	ABCDFGHI
ABJF	Incremental	Monday	J
JHKL	Incremental	Tuesday	KL
ABMN	Incremental	Wednesday	MN
ABOJ	Incremental	Thursday	O
ABCDFGRHIJKLMO	Second Full Backup	Friday	R

Using de-duplication technique backup operation in differential incremental algorithm is successful and performance is better than other backup system.

Journal of Theoretical and Applied Information Technology
15th April 2012. Vol. 38 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

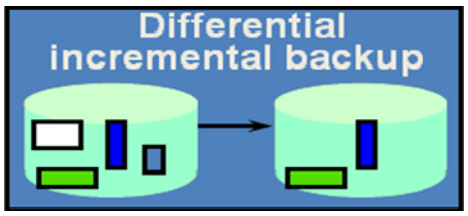

Figure.7: *Storage for Differential Incremental Backup*

9. EXPERIMENTAL RESULT

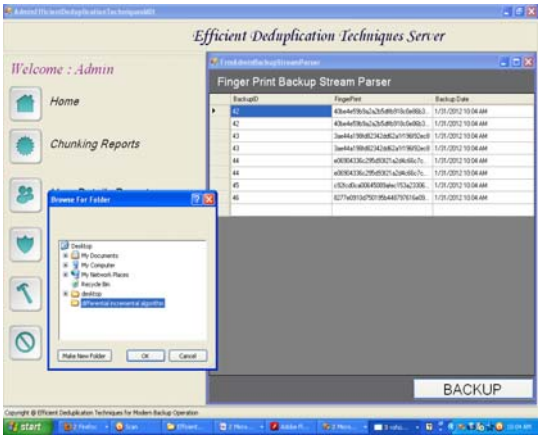

Figure.8: *Backup Stream Parse*

The above figure shows backup stream parse containing all fingerprints, backup ID and date and time that have been used.

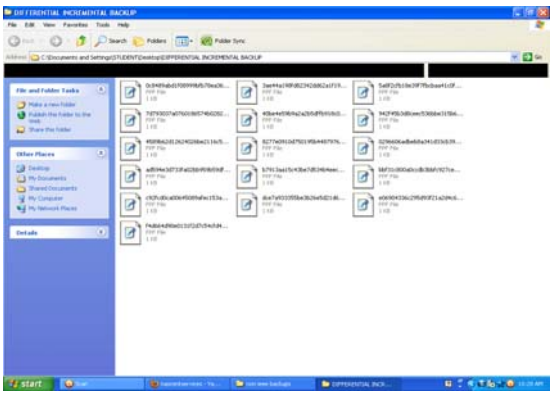

Figure.9: *Differential Incremental Backup*

In this Differential incremental backup, folders containing files avoid duplication, it take only files without redundancy as shown in the Figure.9.

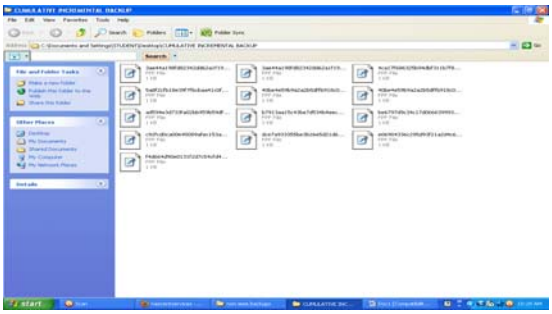

Figure.10: *Cumulative Incremental Backup*

The cumulative incremental folder contains files with duplicate data. Folders containing similar fingerprints occur, as shown in Figure.10.

10. PERFORMANCE MEASURE

The experiment focus on comparison of the performance of three backup methodologies viz., full backup, cumulative incremental backup and differential incremental backup as shown in the Figure.11. In the full backup process, backup speed will be slow in this system, restoration speed is fast and it occupies much more high storage space. Redundant data is the main drawback of this system. In the cumulative incremental backup process, backup speed will be fast, restoration speed is slow in the system and it occupies medium storage space when compared to full backup system. This backup system also consists of redundant data. Compare to both backup system, differential backup provide better result. The main advantage of this system is redundant data is eliminated.

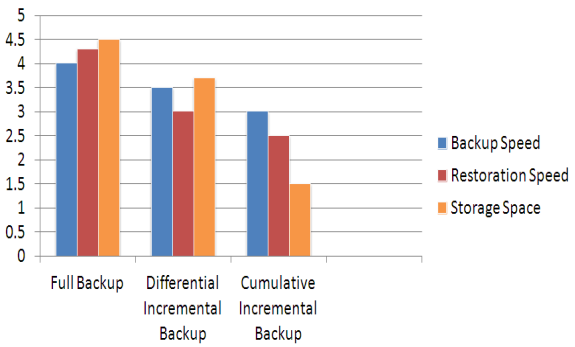

Figure.11: *Performance Measure*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

11. CONCLUSION

In this work, we compare three backup methods using deduplication technique but we put a great effort to understand relationship between chunking module and fingerprinting lookup. Performance of differential incremental is better and this method avoids duplication data and it occupies less storage space. It is great deal to understand full backup, cumulative incremental backup and differential backup. The experiment results show that the storage space is used efficiently and that the performance is distinctly improved.

REFERENCES:

[1] B. Hong and D.D.E. Long, "Duplicate Data Elimination in a San File System," Proc. 21st IEEE / 12th NASA Goddard Conf. Mass Storage Systems and Technologies (MSST), pp. 301-314, Apr. 2004.

[2] Y. Won, R. Kim, J. Ban, J. Hur, S. Oh, and J. Lee, "Prun: Eliminating Information Redundancy for Large Scale Data Backup System," Proc. IEEE Int'l Conf. Computational Sciences and Its Applications(ICCSA '08), 2008.

[3] C. Liu, Y. Lu, C. Shi, G. Lu, D. Du, and D. Wang, "ADMAD: Application-Driven Metadata Aware De-Duplication Archival Storage System," Proc. Fifth IEEE Int'l Workshop Storage Network Architecture and Parallel I/Os (SNAPI '08), pp. 29-35, 2008.

[4] Int'l, Symp. "Modeling, Analysis and Simulation of Computers and Telecomm. Systems". (MASCOTS '08), pp. 1-3, Sept. 2008.

[5] V. Henson, "An Analysis of Compare-by-Hash," Proc. Conf. Hot Topics in Operating Systems (HOTOS '03), 2003.

[6] C. Policroniades and I. Pratt, "Alternatives for Detecting Redundancy in Storage Systems Data," Proc. Conf. USEXNIX '04, June 2004.

[7] Jaehong Min, Daeyoung Yoon, and Youjip Won, "Efficient Deduplication Techniques for Modern Backup Operation". IEEE transaction on computers, Vol. 60, No. 6, June 2011.

[8] L.Aronovich, R. Asher, E. Bachmat, H. Bitner, M. Hirsch, and S. Klein, "The "Design of a Similarity Based Deduplication System," Proc. SYSTOR '09: The Israeli Experimental Systems Conf., pp. 1-14, May 2009.

[9] D.R. Bobbarjung, S. Jagannathan, and C. Dubnicki, "Improving Duplicate Elimination in Storage Systems," ACM Trans. Storage, vol. 2, no. 4, pp. 424-448, 2006.

[10] J. Burrows and D.O.C.W. DC, "Secure Hash Standard," Federal Information Processing Standards Publication, Apr. 1995.

[11] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "R-ADMAD: High Reliability Provision for Large-Scale De-Duplication Archival Storage Systems," Proc. 23rd Int'l Conf. Supercomputing, (ICS '09), pp. 370-379, 2009.

[12] B. Zhu, K. Li, and H. Patterson, "Avoiding the Disk Bottleneck in the Data Domain Deduplication File System," Proc. FAST '08: Sixth USENIX Conf. File and Storage Technologies, pp. 1-14, 2008.

[13] L. P. Cox, C. D. Murray, and B. D. Noble. "Pastiche: making backup cheap and easy". SIGOPS Oper. Syst. Rev., 36(SI):285{298, 2002.

[14] P. Efstathopoulos and F. Guo, "Rethinking Deduplication Scalability," HotStorage '10, Second Workshop Hot Topics in Storage and File Systems, June 2010.

[15] P. Kulkarni, F. Douglis, J. LaVoie, and J. Tracey, "Redundancy Elimination within Large Collections of Files," Proc. USENIX Ann. Technical Conf., General Track, pp. 59-72, 2004.

[16] P. Kulkarni, F. Douglis, J. Lavoie, and J. M. Tracey. "Redundancy elimination within large collections of files". In Proceedings of the annual conference on USENIX Annual Technical Conference, 2004.