International Journal of Advances in Computer Science & Its Applications – IJCSIA
Volume 4: Issue 3 [ISSN 2250-3765]
Publication Date : 30 September,2014

An Investigation on Data De-duplication Methods And it's Recent Advancements

Neha Kaurav

Abstract—Entire world is adapting digital technologies, converting from legacy approach to Digital approach. Data is the primary thing which is available in digital form everywhere. To store this massive data, the storage methodology should be efficient as well as intelligent enough to find the redundant data to save. Data de-duplication techniques are widely used by storage servers to eliminate the possibilities of storing multiple copies of the data. De-duplication identifies duplicate data portions going to be stored in storage systems also removes duplication in existing stored data in storage systems. Hence yield a significant cost saving. In this paper, we, investigate about data de-duplication its techniques and changes introduced in de-duplication due to virtualized data centre and evolution of current cloud computing era.

Keywords—- Data de-duplication; data storage; hash index; Inline and post process de-duplication.

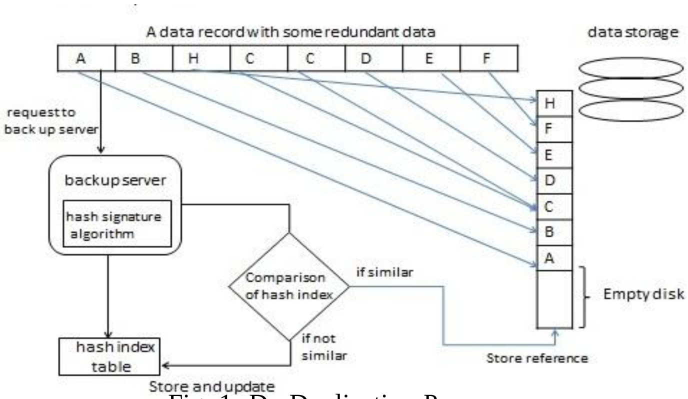

Fig. 1: De Duplication Process

I. Introduction

The world is producing the large number of digital data that is growing rapidly. According to a study, the information producing per year to the digital universe will increase more than six fold from 161 Exabyte to 988 Exabyte between 2006 and 2010, growing by 57% annually. This whopping growth of information is imparting a considerable load on storage systems. The terror attacks of the 9/11 events and the data lost of enterprises in those attacks proved that data loss is devastating to a modern enterprise. So it is critical to back up the data regularly to a disaster recovery site for data availability and integrity [1]. Enterprise Data consists of pictures, audio , video , email conversations , scanned documents and many more. Every organization archives this data for business and legal issues. Rapidly developing data arises many challenges to the existing storage systems. A large number of data requires more storage medium to be used. As the data increases, more data is for backup [2]. The cost of the storage media has decreased, but the main problem is to manage number of disks in the back-up systems.

In fact, In storage archives a large quantity of data is redundant and slight changed to another chunk of data. There are many techniques exists for eliminating redundancy from the stored data. At present data de-duplication has gained popularity in the research community . Data de-duplication is a specialized data compression technique for eliminating redundant data, typically to improve storage utilization . In the de-duplication process , redundant data is left and not stored , leaving distinctive copy of the data chunk to be stored, and a pointer to the unique copy of data [3]. De-duplication is a method to reduce the required Storage capacity since only the unique data is stored. (Refer Figure 1)

II. Data De-duplication Mechanism

In storage servers, De-duplication algorithm detects redundant data by creating cryptographic hash of the data to be stored. Hash is a fixed length representation of arbitrary length message. Hashing reduces the complexity of comparing two data chunks or data record because the size of the hash is much smaller as compared to the size of the data. For each incoming record, server first calculates its hash signature and searches this hash signature in already maintained hash index in the system. If the server finds that an entry is available for this signature in the hash index (i.e. data is already stored), then, in place of storing again, server only creates a reference for this, which points to the location of block on the disk. Otherwise server stores this record on the disk and adds an entry for its hash signature in the hash index , Refer Fig 1 .

III. De-duplication Benefits

Data de-duplication can achieve data reduction levels ranging from ten to one to fifty to one . With less storage needed, storage costs are reduced, because this means smaller disks and less frequent disk purchases. Less data also means smaller backups , which translates into smaller backup windows and faster recovery time. So, by using data de-duplication technology in data centers, it can serve more number of user with the same available space.

Neha Kaurav
Department of Computer Science & Engineering
Acropolis Institute of Technology & Research
Indore,India

IV. De-Duplication Types

A. *De-Duplication Based on Time*

Data de-duplication can be broadly classified into two types i.e. based on time of operation is ”In-line”(at the time of flowing of data) and other is ”Post-process” after the data has been written. These two classifications are described in following sections.

1) Post-process Data De-duplication: With post-process de-duplication , de-duplication analysis and calculations are made after the data is stored in storage device. Once the data is stored then only the process will be applicable. A benefit of using post process is no one need to wait for hash based calculations. The lookup is completed before storing the data also ensuring about performance degradation not achieved. On the negative side of this process,one may unnecessarily save redundant data for a small time which could be an important issue if the system is near to full capacity [4].

2) In-line De-duplication: With inline data de-duplication, process applied at the target device. When the data enters the device in the real instance of time , de-duplication and hash calculations are performed. At this time if the device found a data or block already available in the system, in this case it does not store a new one rather it will reference to the existing block. An advantage of using inline data de-duplication is that it needs lesser amount of storage as data is not redundant. On the other side because of the lookup and hash calculations takes a long time, it leads to slower data ingestion due to this decrement in throughput of backup of the device [4].

B. *De-Duplication Based on Location*

On the basis of location the de-duplication may occur in two locations. First process is applied where data is produced and known as “source de-duplication” and second process is applied where the record is stored and referred as "target de-duplication” [4].

1) Source versus Target De-duplication: When we describe de-duplication process for backup systems and architectures there are two kinds of de-duplication which can be apply. They are referred as source based and target based data de-duplication.

De-duplication process applied at the data source is known as source de-duplication. This is one of the type of location based de-duplication. Source de-duplication process commonly implemented within the file system directly. In such process a periodic scan is performed by file system, in which de-duplication process will scan new files creating hash and then after it compares these hashes with existing hash indexes of records. If any file or record found with similar hash index, it will remove the copy of a file and this new file will point to the old file. Duplicated copies are separately stored and in case of duplicate file , it is modified after sometime , than by performing the copy on write, another copy of modified file is produced. While applying de-duplication for backup the file system causes the redundancy that results with bigger backup rather source data. In target de-duplication removing of redundant data is done at the secondary storage . Backup store as a virtual tape library or data storing repository are the general types of backups those are provided in target based de-duplication. In this target based de-duplication, at the time of data comes to storing,it apply post process or in-line de-duplication. Choice of in-line or post process method depends on requirements at the target side storage [3].

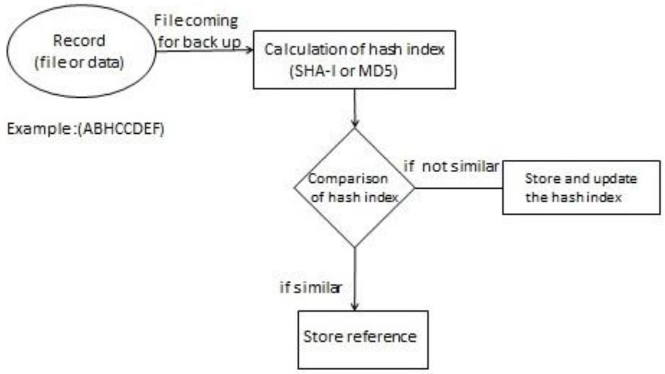

Fig. 2: File Level De-Duplication

C. De-duplication Levels

Data de-duplication technology is a process of removing duplicate data, identification of redundant files and to decrease the requirement to store data in order to utilize the overall storage capacity. Duplication of data or record can occur within a data block; within a file or in a specific data byte. While investigating the recent methods and advancements we found three levels where duplication usually occurs and requires to de-duplicate that data. At present mainly three levels of data de-duplication is available

1) File level de-duplication
2) Block-level de-duplication
3) Byte-level de-duplication

These strategies can be optimized for utilizing storage capacity [4].

1) File Level De-Duplication: File level de-duplication is a method in which whole file is considered as a record. The time file comes to back up, the hash signature of incoming record are compared with hash signatures of already stored files. It stores a reference to the file if it exists otherwise it will store entire record and a new entry for hash signature of this record in the hash table. (Refer fig 2)

2) Byte Level De-Duplication: Byte stream data is another level. In this, the incoming data stream is divided into the number of bytes and then the hash signature of each incoming bytes are compared with the stored bytes on the disk and take appropriate action i.e. if any byte stream is already available then it will reference to it otherwise data stream will be stored on the disk with new hash signature entry will be created in the hash table (Refer fig 3). Byte level de-duplication gives the highest accuracy as compared to file level de-duplication and block level de-duplication. But byte level de-duplication lead to many problems, which are as follows

1) Size of the hash table will become very large.
2) It may lead to large file fragmentation.

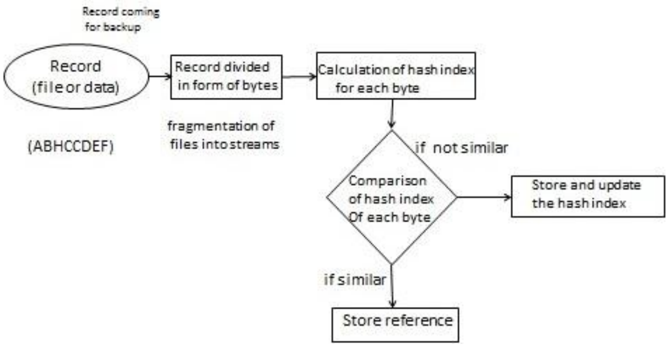
Record coming for backup
Record (file or data)
(ABHCCDEF)
Record divided in form of bytes
fragmentation of files into streams
Calculation of hash index for each byte
if not similar
Comparison of hash index Of each byte
Store and update the hash index
if similar
Store reference

Fig. 3: Byte Level De-Duplication

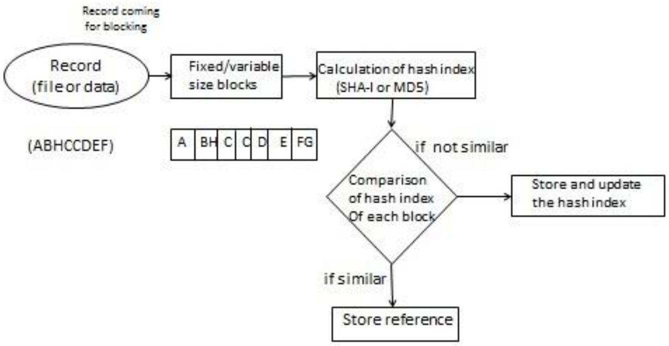
Record coming for blocking
Record (file or data)
(ABHCCDEF)
Fixed/variable size blocks
A
BH
C
C
D
E
FG
Calculation of hash index (SHA-I or MD5)
if not similar
Comparison of hash index Of each block
Store and update the hash index
if similar
Store reference

Fig. 4: Block Level De-Duplication

3) Finally, byte level de-duplication will lead to performance degradation.

D. Block Level De-duplication

Block-level data de-duplication method is the next level wherein data block de-duplication is applied. In this process the incoming data stream is divided into blocks, and then it is compared with the hash of data block. After comparison, it determines whether it is same as with the previously saved data block (using the hash algorithm for each data block to form a digital signature or unique identifier). If the hash of the data block is distinct , then store this block to disk, and store its identifier in the hash index; otherwise, store the reference to the same data block's real location. It stores a reference of a comparatively small size in place of the data block, rather than storing redundant data blocks again, hence a significant saving of disk storage space. Hash algorithm used to examine and judge duplicate data. It may lead to conflict between the hash signatures, so generally it uses SHA-1 algorithm for generating hash signatures because it generates 160 bit of hash signature and can create different hash signature for 2^{160} blocks of the data.(Refer fig 4).

V. Types of Block Level De-Duplication

There are two types of Block Level de-duplication:

A. Fixed Size Block De-Duplication

Fixed Block de-duplication involves determining a block size and segmenting files/data into those block sizes. Then, those blocks are what are stored in the storage subsystem Suppose we take a fixed size 1 byte to divide an incoming file.

Whole File	-	Low	Lowest
Fixed Size	Block Size	Time Complexity > (1)	Middle
Variable size	Average Block Size	Time Complexity > (2)	greater than fixed blocking
Byte Level	-	Most Complex	Highest

TABLE I: Comparison of De-Duplication Methods

B. Variable Size Block De-Duplication

Variable Block de-duplication involves using algorithms to determine a variable block size. The data is split based on the algorithm's determination. Then, those blocks are stored in the subsystem.

VI. Advantages of Block Level vs File Level

File internal changes, will cause the entire file need to store. PPT and other files may need to change some simple content, such as changing the page to display the new report or the dates, which can lead to re-store the entire document. Block-level data de-duplication technology stores only one version of the paper and the next part of the changes between versions [5]. File level technology, generally less than 5:1 compression ratio, while the block-level storage technology can compress the data capacity of 20: 1 or even 50: 1. Table I can comprehend these de-duplication methods.

VII. Various Methodologies of De-Duplication

At present, the research of de-duplication focuses on two aspects. One is the effectiveness of data reduction, that is, remove the duplicate data as much as possible,and then reduce the storage capacity requirement. The other is the efficiency of data de-duplication, i.e. the resources required to achieve the effectiveness.

Most of the available traditional backup systems use file-level de-duplication [6]. However the data de-duplication technology can exploit inter-file and intra-file information redundancy to eliminate duplicate or similarity data at the granularity of file, block or byte. Some of the available architecture follows the source de-duplication approach and provide the de-duplication technology in the available user's file system [7]. However because of this file system de-duplication, user has to face delay in sending data to the backup store, and the rest of the available architectures which support target de-duplication strategy provide single system de-duplication that means at the target side only single system (Server) handles all the user requests to store data and maintains the hash index for the number of disks attached to it.

Some previously proposed architectures are VENTI, LBFS, SIS (single instance store), and PASTICHEL. VENTI and SIS adopt fixed-size file dividing method to partition the file into blocks [5] [8]. LBFS and PASTICHEL divides a file into variable sized blocks [7] [9]. Fixed-size file dividing method is simple and easy,but the salient disadvantage is that all the blocks after the change point will be affected, and then misjudged as non-duplicate blocks.

Another problem with the available architectures is that the hash algorithm may lead to the hash collision, that is, different blocks produce the same hash codes, which will result in discarding unique block mistakenly. However, LBFS [7], fingerdiff [10],REBL [11], 3DNBS [12] and SDD [13] used hash algorithm (SHA-1 or MD5), and most of them considered that the probability of hash collision is extremely lower than the probability of hardware errors.

Zhu ET uses the Summary Vector, an in-memory, conservative summary of the segment index, to reduce the number of times that the system goes to disk to look for a duplicate segment only to find that none exists. Then they use Stream-Informed Segment Layout (SISL) to create spatial locality and to enable Locality Preserved Caching (LPC) to prefetch hash codes of adjacent segments into cache. LPC method avoids disk operation and accelerates the process of identifying duplicate segments [14].

Extreme Binning [2] exploits file similarity instead of locality and splits up the chunk index into two tiers. The top tier called primary index resides in RAM. It is used to identify a file. The second tier called bin is kept on disk. It stores all de-duplicate chunks of a file. Thus Extreme Binning makes a single disk access for chunk lookup per file instead of per chunk to alleviate the disk bottleneck problem. But one disadvantage of Extreme Binning is that it allows some duplicate chunks.

In the present scenario, many organizations are involved in working with data de-duplication concept. Few of the organizations are IBM, SYMANTEC, and NetApp. NetApp de-duplication is a fundamental component of Data ONTAP operating system. NetApp de-duplication is the first that can be used broadly across many applications, including primary data, backup data, and archival data. Symantec also provides backup appliances that provide three step reduction processes. First it provides data de-duplication at source and targets both and reduces the data de-duplication complexity. IBM's TS7610 ProtecTIER De-duplication Appliance Express provides fast , reliable easy backup with de-duplication technology.

Sengar and Mishra et al. [15] proposed a very scalable and efficient in-line data de-duplication , this algorithm support bloom filter to reduce the disk access time for segments which are not present in the Disk. It supports load balancing in storage nodes.

VIII. Conclusion

We thoroughly studied about various Data De-Duplication methods widely used in storage servers worldwide. This study is also useful for a new researcher who wants to work in field of Data de-duplication and this study can be a start guide for it. We are focusing on the new load balanced algorithms which are scalable as well as described in the work of sengar mishra et al. [15].

References

[1] D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, "The expanding digital universe," 2007.

[2] D. Bhagwat, K. Eshghi, D. D. E. Long, and M. Lillibridge, "Extreme binning: Scalable, parallel deduplication for chunk-based file backup." in MASCOTS. IEEE, 2009, pp. 1–9. [Online]. Available: http: //dblp.uni-trier.de/db/conf/mascots/mascots2009.html#BhagwatELL09

[3] WikiPedia, "Data deduplication." [Online]. Available: http://www.wikipedia.com/deduplication

[4] Q. He, Z. Li, and X. Zhang, "Data deduplication techniques," in Future Information Technology and Management Engineering (FITME), 2010 International Conference on, vol. 1, 2010, pp. 430–433.

[5] S. Quinlan and S. Dorward, "Awarded best paper! - venti: A new approach to archival data storage," in Proceedings of the 1st USENIX Conference on File and Storage Technologies, ser. FAST '02. Berkeley, CA, USA: USENIX Association, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1083323.1083333

[6] G. Wang, Y. Zhao, X. Xie, and L. Liu, "Research on a clustering data de-duplication mechanism based on bloom filter," in Multimedia Technology (ICMT), 2010 International Conference on, 2010, pp. 1–5.

[7] A. Muthitacharoen, B. Chen, and D. Mazières, "A low-bandwidth network file system," SIGOPS Oper. Syst. Rev., vol. 35, no. 5, pp. 174–187, Oct. 2001. [Online]. Available: http://doi.acm.org/10.1145/502059.502052

[8] W. J. Bolosky, S. Corbin, D. Goebel, and J. R. Douceur, "Single instance storage in windows&#174; 2000," in Proceedings of the 4th Conference on USENIX Windows Systems Symposium - Volume 4, ser. WSS'00. Berkeley, CA, USA: USENIX Association, 2000, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267102.1267104

[9] L. P. Cox, C. D. Murray, and B. D. Noble, "Pastiche: Making backup cheap and easy," in Proceedings of the 5th Symposium on Operating Systems Design and implementation Copyright Restrictions Prevent ACM from Being Able to Make the PDFs for This Conference Available for Downloading, ser. OSDI '02. New York, NY, USA: ACM, 2002, pp. 285–298. [Online]. Available: http://doi.acm.org/10.1145/1060289.1060316

[10] D. R. Bobbarjung, S. Jagannathan, and C. Dubnicki, "Improving duplicate elimination in storage systems," Trans. Storage, vol. 2, no. 4, pp. 424–448, Nov. 2006. [Online]. Available: http://doi.acm.org/10.1145/1210596.1210599

[11] 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, ser. ATEC '04. Berkeley, CA, USA: USENIX Association, 2004, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=1247415.1247420

[12] T. Yang, D. Feng, J. Liu, Y. Wan, Z. Niu, and Y. Ke, "3dnbs: A data de-duplication disk-based network backup system," in Networking, Architecture, and Storage, 2009. NAS 2009. IEEE International Conference on, 2009, pp. 287–294.

[13] C. Liu, D. Ju, Y. Gu, Y. Zhang, D. Wang, and D.-C. Du, "Semantic data de-duplication for archival storage systems," in Computer Systems Architecture Conference, 2008. ACSAC 2008. 13th Asia-Pacific, 2008, pp. 1–9.

[14] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system."

[15] S. S. Sengar and M. Mishra, "A parallel architecture for in-line data de-duplication," Advanced Computing & Communication Technologies, International Conference on, vol. 0, pp. 399–403, 2012.