ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN INVESTIGATION ON DATA DE-DUPLICATION METHODS AND ITS RECENT ADVANCEMENTS

Arjita Singh Baghel*, Arnaaz Khan

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.

INTRODUCTION

Data de-duplication is a method to reduced redundancy in disk storages while backing up data. Data de-duplication methods uses the concept of hashing (hash is a fingerprint or summary of a message), by which a hash signature is created for each data and record [4]. Hash makes comparison easier because the size of hash is smaller than size of data. Whenever any request is occurred for storing any data, firstly a hash signature is calculated in server itself. After this it checks for this signature in hash table. If the hash signature is similar then it only points that data from storage rather saving it again in disk. Data de-duplication helps achieving data reduction with better compression ratio. Generally a de-duplication method consists of two approaches for data de-duplication storage systems: hash based (fingerprinting-based) and delta-based data de-duplication. Now-a-days, hash based de-duplication is prevalent in practice and research, and this thesis deals exclusively with this type. A Broad view of de-duplication classification is presented in Figure 2-1.

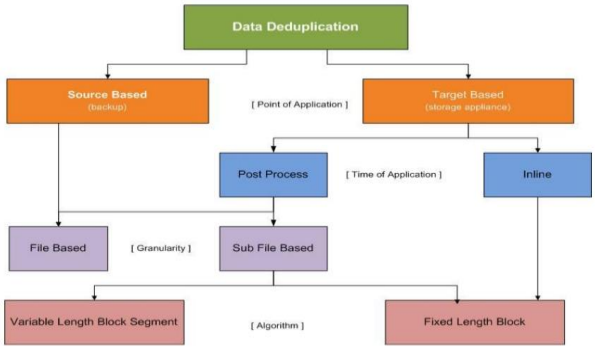

Figure 2-1: Classification of Data De-Duplication Strategies

PREVIOUS WORK

This chapter highlights important contributions to the field of data de-duplication, which are related to the overall context of this thesis. Related work that is only relevant to a part of this thesis will be described in the related part. The research of data de-duplication presently focuses on different aspects. Effectiveness of data reduction is one of them, that is, in order to reduce the storage capacity requirement, remove the duplicate data as much as possible. Data reduction is definitely an important parameter for better data de-duplication architectures. Another aspect is the efficiency of

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

data de-duplication, i.e. to achieve the effectiveness of algorithm what amount of resources are required. Many researchers worked in the field of data de-duplication previously and resulted with different methods for better efficiency. While surveying the recent methods and advancements we can see that most available backup systems uses file-level de-duplication traditionally [4]. But 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 users file system [5]. Because of this file system de-duplication, user faces delay in sending data to backup store, and the rest of the available architectures which support target de-duplication strategy provide single system de-duplication which means at the target side only single system or server handles all the user requests to store data and maintains the hash index for the number of disks attached to it [1].

Name of some previously proposed architectures are VENTI [7], LBFS (lower bandwidth file system) [5], MAD2, SIS (single instance store), CDC (Content defined chunking) [6], INS (Index Name Server) and PASTICHE. VENTI is a network storage system designed for archival data. It uses a unique hash of a blocks content which acts like the block identifier for read and write operations. Such approach enforces a write-once policy for prevention of malicious destruction of data. Venti is a building block for constructing a variety of storage applications such as logical backup, physical backup and Snapshot file systems. It was built with a prototype of the system and presents some preliminary performance results. The system uses magnetic disks as the storage technology, resulting in an access time for archival data that is comparable to non-archival data. VENTI and Single Instance Storage adopt fixed-size file dividing method to partition the file into blocks [7] [8]. LBFS and PASTICHE divide each file into variable sized blocks [5] [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. Zhu ET use 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 [1][10]. Another approach provides a Scalable High Throughput Exact De-duplication Approach for Network Backup Services. In such research it eliminates duplicate data both at the file level and at the chunk level by employing four techniques (Hash Bucket Matrix, Bloom Filter Array, Dual Cache, DHT-based Load-Balance technique) to accelerate the de-duplication process and evenly distribute data. Mad2 supports a de-duplication throughput of at least 100MB/s for each storage component [11]. Some researchers worked in the field of cloud storage and worked with using both fixed size block and variable size blocks. As there are a lot of de-duplication techniques depending on the algorithms chunking of the data blocks. In paper, they had chosen Fixed Block [3] and Rabin's Fingerprint [12] which is the most well-known algorithms as the representatives. Fixed Block algorithm uses fixed size block as a unit of the de-duplication while Rabin's Fingerprint uses variable block size. Tin-Yu Wu, Wei-Tsong Lee, Chia Fan Lin2 proposes a new data management structure named Index Name Server (INS), which integrates data de-duplication with nodes optimization mechanisms for cloud storage performance enhancement. INS manages and optimizes the nodes according to the client-side transmission conditions. By INS, each node can be controlled to work in the best status and matched to suitable clients as possible. It improves the performance of the cloud storage system efficiently and distributes the files reasonably to reduce the load of each storage node [13]. 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. A problem with the available architectures is that the hash algorithm may lead to hash collision, i.e. different blocks produce the same hash codes, which will result in discarding unique block mistakenly. However, LBFS [5], fingerdiff [14] 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. In our architecture we selected

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

SHA-2 hash algorithm because of its strong collision resistant and encryption function. Sengar and Mishra [1] proposed a very scalable and efficient in-line data de-duplication using SHA-1. This algorithm supports bloom filter to reduce the disk access time for segments which are not present in the Disk. It supports load balancing in storage nodes.

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 [3]. 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.

PROBLEM STATEMENT

"A Parallel Architecture for In-Line Data De-Duplication Using SHA-2 Hash" is our proposed architecture. The main aim of designing an algorithm with parallel architecture is to overcome the problem domain with following issues as-

1. Traditional systems uses file level method, because of that more resources are required due to granularity.
2. Hash collision occurred due to less collision resistant encryption function.
3. Sequential implementation leads to more time complexity.
4. It should support to handle high and large number of segments on same time and should be capable of handling billions of segments simultaneously.
5. It should perform the de-duplication at higher speed which means the process terminates in lesser time. With this property the required response will generate at higher speed.
6. To save more capacity and performs efficiently with best efforts.

The proposed architecture uses the hash index for redundancy identification between files so it should fulfill some other features-

1. Resource requirement can be degraded using block level de-duplication.
2. Misjudgment of data blocks can be overcome using fixed size blocks.
3. Use of upgraded hash algorithm leads to lesser probability of hash collision as SHA-256 Provide hash signature up to 2^{128} bytes.
4. Parallel implementation helps reducing time complexity and shows better performance for larger file sizes.
5. Space reclaiming with use of reference count mechanism.
6. Distribution of data among various storage clients with an intelligent data distribution method is required.
7. Good number of database tables so that it can handle the Meta data of files or records.

To decrease the communication overhead it should support better interaction between storage node and server.

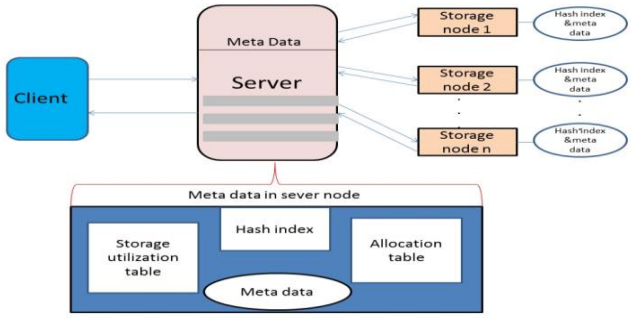

Figure 3-1: Proposed Parallel architecture

CONCLUSION

In our architecture we used proper data distribution of incoming data and already stored data. They are referred as distribution strategy. By using distribution techniques we can improve the load balancing and overall efficiency of the parallel architecture. Hash signature creation is also a very important parameter of any data de-duplication process. In previous researches the use of MD5 algorithm and secured hash algorithm (SHA) was used with 160 bit hash signature type. Previous algorithms for hash creation are less collision resistant in nature. In our architecture we used SHA-256 algorithm for less collision probability. The main advantage of using SHA- 256 algorithm is that, it provides 258 bit hash by which the collision probability is much lesser than previous de duplication methods.

REFERENCES

1. Seetendra Singh Sengar and Manoj Mishra. A parallel architecture for in-line data de-duplication. In Advanced Computing & Communication Technologies (ACCT), 2012

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Second International Conference on, pages 399–403. IEEE, 2012.
2. Deepavali Bhagwat, Kave Eshghi, Darrell DE Long, and Mark Lillibridge. Ex- treme binning: Scalable, parallel deduplication for chunk-based file backup. In Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on, pages 1–9. IEEE, 2009.
3. Qinlu He, Zhanhuai Li, and Xiao Zhang. Data deduplication techniques. In Future Information Technology and Management Engineering (FITME), 2010 International Conference on, volume 1, pages 430–433. IEEE, 2010.
4. Guohua Wang, Yuelong Zhao, Xiaoling Xie, and Lin Liu. Research on a clustering data de-duplication mechanism based on bloom filter. In Multimedia Technology (ICMT), 2010 International Conference on, pages 1–5. IEEE, 2010.
5. Athicha Muthitacharoen, Benjie Chen, and David Mazieres. A low-bandwidth network file system. In ACM SIGOPS Operating Systems Review, volume 35, pages 174–187. ACM, 2001.
6. Xingchen Ge, Ning Deng, and Jian Yin. Application for data de-duplication algorithm based on mobile devices. Journal of Networks, 8(11):2498–2505, 2013.
7. Sean Quinlan and Sean Dorward. Venti: A new approach to archival storage. In FAST, volume 2, pages 89–101, 2002.
8. William J Bolosky, Scott Corbin, David Goebel, and John R Douceur. Single instance storage in windows 2000. In Proceedings of the 4th USENIX Windows Systems Symposium, pages 13–24. Seattle, WA, 2000.
9. Landon P Cox, Christopher D Murray, and Brian D Noble. Pastiche: Making backup cheap and easy. ACM SIGOPS Operating Systems Review, 36(SI):285–298, 2002.
10. Benjamin Zhu, Kai Li, and R Hugo Patterson. Avoiding the disk bottleneck in the data domain deduplication file system. In Fast, volume 8, pages 1–14, 2008.
11. Jiansheng Wei, Hong Jiang, Ke Zhou, and Dan Feng. Mad2: A scalable high- throughput exact deduplication approach for network backup services. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pages 1–14. IEEE, 2010.
12. Deepak R Bobbarjung, Suresh Jagannathan, and Cezary Dubnicki. Improving duplicate elimination in storage systems. ACM Transactions on Storage (TOS), 2(4):424–448, 2006.
13. Tin-Yu Wu, Wei-Tsong Lee, and Chia Fan Lin. Cloud storage performance enhancement by real-time feedback control and de-duplication. In Wireless Telecommunications Symposium (WTS), 2012, pages 1–5. IEEE, 2012.
14. Ma Jiantinga. A deduplication-based data archiving system. International Pro- ceedings of Computer Science & Information Technology, 50, 2012.http://www.wikipedia/Deduplication