RESEARCHERID
THOMSON REUTERS

[Bhagoriya* *et al.*, 6(4): April, 2017]
ICTM Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

SURVEY AND PROPOSED PARALLEL ARCHITECTURE FOR INLINE DATA De-DUPLICATION USING SHA-2 ALGORITHM

Nupur Bhagoriya*, Ashlesha Panse, Parv Khatri

[*] Computer science & engineering, acropolis institute of technology & research, Indore (m.p.), India

DOI: 10.5281/zenodo.569945

ABSTRACT

In this digital world of internet, data storage and server storage use is often and every home user, enterprises, several organizations are using email and online storage as a storing node. Online backup storage is an easy option for everyone to store digital data, files and other multimedia files. Due to huge requirement of data storage online it is found that so many users stored the same data repeatedly. This makes the storage servers loaded as well as more disk storage is required to save a large amount of same data. Due to the same reason the search operation takes more time to found a specific file and time complexity to give positive search result and acknowledgement is increased. This problem was overcome with a mechanism known as Data de-duplication. This process is used for removing duplicate data and to reduce redundancy at server node.

Data de-duplication method is usually applied at the time of storing the data (Inline) or after storing the data (Post process) and it can be applied over primary data or secondary data. De-duplication process can be accomplished at File, Block or Byte level. In previous modifications of de-duplication algorithms it is found that some issues like time complexity, unbalanced load, hash collision etc. has occurred. In this thesis report we have studied previous and recent work on de-duplication and proposed a solution which is a Parallel architecture for inline data de-duplication which uses the Secure Hash Algorithm (SHA) 256 for performing data de-duplication task in order to overcome the issues of time complexity and to reduce hash collision. In this architecture write and delete operations are performed for efficiency and time evaluation. The time taken is much lesser for redundant data during write procedure. This decrement in time is a result of using de-duplication process. This architecture is useful for storage servers where a huge amount is stored every day and software industries always looks for new developments so that they can keep their storage systems up to date and free for efficient utilization of the server nodes.

KEYWORDS:, Parallel Architecture, Data De-Duplication, SHA-2.

INTRODUCTION

Nowadays, online backup storage, content delivery networks, blog sharing, news broadcasting and social networks as an ascendant part of Internet services are data centric. Hundreds of millions of users of these services generate petabytes of new data every day. For instance, as of April 2011, an online file-sharing and backup services called drop boxes, has more than 25 million 2GB drop boxes (total of 50 petabytes). A large portion of internet service data is redundant for the following reasons.

1. Now a day's, a person tends to save data at multiple times for data safety reasons and avoids purchasing storage for high cost. This leads to more redundant data.
2. One another reason is, while incremental (or differential) data backups or disk image files for virtual desktop tend not to have duplicated whole-file copies, but still there is large ratio of duplicated data portion from the modifications and revisions of the files.

In this digital world the amount of data is increasing with every day. By previous studies it is concluded that the data and digital information added every year is increased with 57%. This amount of growth in information is definitely putting the strain and load on data storage systems. The terror attacks of the 9/11 events and the data loss 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. To reduce such problems it is necessary to have a better back-up system so that the data can be back up periodically for the purpose of data availability and integrity [1].

Enterprise data consists of pictures, audio, video, email conversations, scanned documents etc. Every organization archives this data for business and legal issues. Rapidly increasing data arises many challenges to the existing storage systems. The large amount of data requires more storage medium to be used [2]. As the data increases, more data is for backup cite. Due to increment in storage data it is found that it brings some difficulties in backu p systems. 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 amount of data is redundant and slight changed to another chunk of data. The identification of these duplicate chunks is fundamental to improve the quality of information retrieval [1].

There are many techniques exists for eliminating redundancy from the stored data. Now a days, data de-duplication is a hot emerging technique that gaining popularity in research community. Data de-duplication is a specialized data compression technique for eliminating redundant data and to increase availability of data for better storage utilization. In the de-duplication process, redundant data is left and not stored, leaving single copy of the data chunk to be stored, and a pointer to the unique copy of data [2]. The cost of the storage media has decreased, but the main problem is to manage number of disks in the back-up systems.

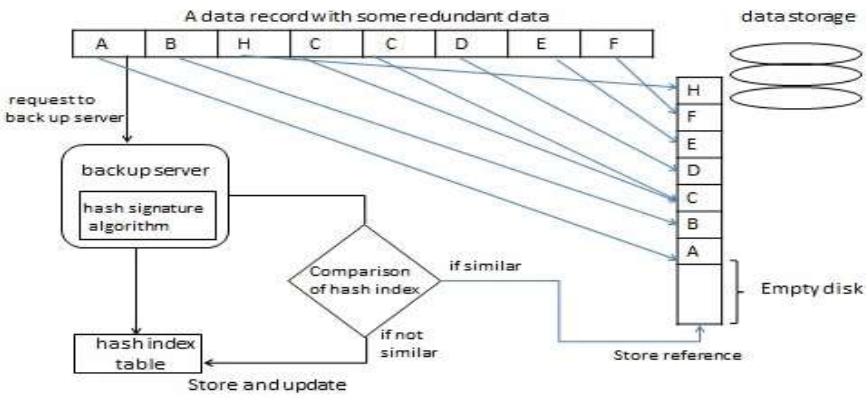

Figure 1-1: De Duplication Process

In fact, in storage archives a large amount of data is redundant and slight changed to another chunk of data. There are many techniques exists for eliminating redundancy from the stored data. De-duplication is a method to reduce the required Storage capacity since only the unique data is store (Refer Figure 1-1).

RELATED WORK

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 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

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

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 major 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 is known as the primary index, it 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 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.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

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.

PROPOSED ARCHITECTURE

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

Our proposed architecture for inline data de-duplication is given below:

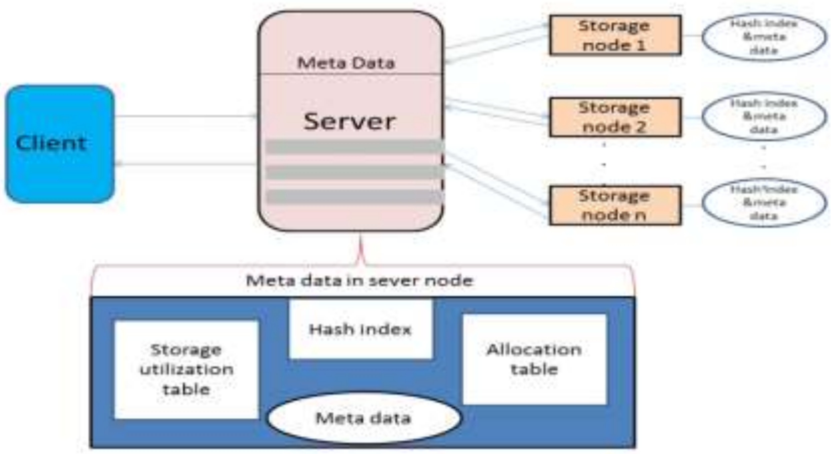

Figure 3-1: Proposed Parallel architecture

Proposed inline parallel architecture makes use of various concepts given below-

Client

The node that contains or need back up for data is client. When client require to store any data, it sends that data to server node.

Server

After a client request a file to backup, server first receives that file at backup store and after accepting the file, server divides it in fixed size blocks (example 1024 KB) and group these divided blocks into super block and these super blocks are distributed among nodes of available storage using strategies of data distribution. Now storage nodes and server create hash signature of distributed parts and a sequential search of hash signature is performed. This search is performed in parallel way at the maintained hash table.

Meta Server

All the database tables are collected at Meta data server. The meta data tables contain the file name, number of parts in each file, parts path information, number of references to each file part, hash signature of each files part and storage utilization of each storage node information.

Storage Server

Every storage server has its hash table and information related to it. Storage node first receives data for backup and performs hash calculation of data block and compares this with its hash table which is maintained by server itself. The data is stored in disk storage if hash signature found unique. If its signature is found

Hash Index

Each data block's hash signature is contained in the hash index. It also contains the information regarding its location in the disk.

REFERENCES

[1] Seetendra Singh Sengar and Manoj Mishra, "A Parallel Architecture For In-line Data De-duplication," IEEE seccond International Conference on Advanced Computing & Communication Technologies (ACCT), pp. 399–403 2012

[2] Deepavali Bhagwat, Kave Eshghi, Darrell DE Long, and Mark Lillibridge, "Ex- treme binning: Scalable, Parallel Deduplication for Chunk-based File Backup," IEEE International Symposium on, Modeling, Analysis & Simulation of Computer and Telecommunication Systems, MASCOTS'09. pp. 1–9, 2009.

[3] Qinlu He, Zhanhuai Li, and Xiao Zhang, "Data Deduplication Techniques," IEEE International Conference on Future Information Technology and Management Engineering (FITME), vol. 1, pp. 430–433, 2010.

[4] Guohua Wang, Yuelong Zhao, Xiaoling Xie, and Lin Liu, "Research on a Clustering Data De-duplication Mechanism Based on Bloom Filter, " IEEE International Conference on Multimedia Technology (ICMT), pp. 1–5, 2010.

[5] Athicha Muthitacharoen, Benjie Chen, and David Mazieres, "A Low-Bandwidth Network File System," In ACM SIGOPS Operating Systems Review, vol. 35, pp. 174–187, 2001.

[6] Xingchen Ge, Ning Deng, and Jian Yin, "Application for Data De-duplication Algorithm Based on Mobile Devices," In Journal of Networks, vol. 8, no. 11, pp. 2498–2505, 2013.

[7] Sean Quinlan and Sean Dorward, "Venti: A new approach to archival storage," In FAST, vol 2, pp. 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 Seattle WA, pp. 13–24, 2000.

[9] Landon P Cox, Christopher D Murray, and Brian D Noble, "Pastiche: Making Backup Cheap and Easy," ACM SIGOPS Operating Systems Review, vol. 36 no. 1, pp. 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, vol. 8, pp. 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," IEEE 26th Symposium on, Mass Storage Systems and Technologies (MSST),pp. 1–14, 2010.

[12] Deepak R Bobbarjung, Suresh Jagannathan, and Cezary Dubnicki, "Improving Duplicate Elimination in Storage Systems," ACM Transactions on Storage (TOS), vol. 2, no. 4, pp. 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," IEEE, In Wireless Telecommunications Symposium (WTS), pp.1–5, 2012
[14] Ma Jiantinga, "A Deduplication-based Data Archiving System," International Proceedings of Computer Science & Information Technology, 2012.
[15] Available at http://www.wikipedia/Deduplication
[16] C. Ungureanu, B. Atkin, A. aranya, S. Gokhale, S. Rago, G. Calkowski, C. Dubnicki and A. Bohra. HydraFS, "A High Throughput File System for the content-addressable storage system," In FAST'10: Proceedings of the 8th Conference on File and Storage Technologies, February 2010.
[17] Santos Walter, Thiago Teixeira,Carla Machado, Wagner Meira, Altigran S. Da Silva, Renato Ferreira, Dorgival Guedes, "A Scalable Parallel Dedulication Algorithm," 19th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), pp.79-86, 24 Oct, 2007.
[18] J F Gantz, "The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010," IDC, March 2007.
[19] Purushottam Kulkarni, Fred Douglis, Jason LaVoie, and John M. Tracey, "Redundancy Elimination within Large Collection of Files," In Proceedings of the 2004 USENIX Annual Technical Conference, Boston, MA, pp. 59-72, June 2004.
[20] Available at http://www.wikipedia/Sha
[21] Filipe, Ricardo, and Joao Barreto, "Towards Full on-line Deduplication of the Web," Proceedings of INFORUM. 2010.
[22] Sun, Zhe, Jun Shen, and Jianming Yong, "A Novel Approach to Data Deduplication over the Engineering-oriented Cloud Systems," Integrated Computer-Aided Engineering vol. 20, no. 1, pp. 45-57, 2013.
[23] Karunakaran, Deepa, and Rangarajan Rangaswamy, "Optimization Techniques To Record Deduplication," Journal of Computer Science vol. 8, no. 9,pp. 1487,2012.
[24] Kim, Chulmin, Ki-Woong Park, KyoungSoo Park, and Kyu Ho Park, "Rethinking Deduplication in Cloud: From data profiling to blueprint," IEEE, 7th International Conference on, Networked Computing and Advanced Information Management (NCM), pp. 101-104,2011.
[25] Jiantinga, Ma, "A Deduplication-based Data Archiving System," International Proceedings of Computer Science & Information Technology vol .50,2012.
[26] De Carvalho, Moisés G., Marcos André Gonçalves, Alberto HF Laender, and Altigran S. da Silva "Learning to Deduplicate," In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, pp. 41-50. 2006.

CITE A JOURNAL:

Bhagoriya, N., Panse, A., & Khatri, P. (2017). SURVEY AND PROPOSED PARALLEL ARCHITECTURE FOR INLINE DATA De-DUPLICATION USING SHA-2 ALGORITHM. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*,*6*(4), 700-705. doi:10.5281/zenodo.569945