IJRSET NOVEMBER 2016 Volume 3, Issue 11 Pages: 18-27

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

AN EFFICIENT DEDUPLICATION PROCESS OF LOAD BALANCING METHOD IN CLOUD STORAGE

[1] R. Shalini, [2] P. Shanthi
[1] M.Phil Research Scholar, [2] Assistant Professor
[1, 2] SRI JAYENDRA SARASWATHY MAHA VIDYALAYA COLLEGE OF ARTS AND SCIENCE,
[1, 2] Coimbatore, Tamilnadu, India

ABSTRACT: In the real Cloud Computing Environment, the centralized cloud management was developed an efficiency and cost inflection point, and offers simple offsite storage for disaster recovery, which is always a critical concern for data backup. To prepare one alternative, Cloud backup service is a best choice which is cost-effective and provides protection to personal computing devices or cloud storage. The proposed method will focus on the following factors, user provisioning cost, security to storage, load balancing, and avoid redundancy. For providing these factors, the functions to perform are de-duplication process, chunking of data, hash function, load balancing and Application aware. The de-duplication process performs the functions local de-duplication and global de-duplication to check the local cloud storage and the global cloud storage, which gives more effectiveness and latency of de-duplication. An intelligent data chunking method and Hash functions are performed, which splits the files into the chunks of data and apply hash functions to those chunks of data, which results in minimal computational overhead and high security to the cloud storage. Load balancing function is used to balance the load while storing the files into the resources of cloud.

Keywords: [Deduplication, chunking, Back-up.]

1. INTRODUCTION

Cloud computing is the new emerging trends in the new generation technology. Every user has huge amount of data to share to store in a quickly available secured place. The concept of deduplication is arrived here to efficiently utilize the bandwidth and disk usage on cloud computing. To avoid the duplication copies of the same data on cloud may cause lose of time, bandwidth utilization and space.

Cloud computing is internet-based, a network of remote servers connected over the Internet to store, share, manipulate, retrieve and processing of data, instead of a local server or personal computer[1]. The benefit of cloud computing are enormous. It enables us to work from anywhere. The most important thing is that customer doesn’t need to buy the resource for data storage. When it comes to Security, there is a possibility where a malicious user can penetrate the cloud by impersonating a legalize user, there by affecting the entire cloud thus infecting many customers who are sharing the infected cloud. There is also big problem, where the duplicate copies may upload to the cloud, which will lead to waste of band width and disk usage. To improve this problem there should be a good degree of encryption provided, that only the customer should be able to access the data and not the legitimate User. Yan Kit Li et al.[1] shown To formally solve the problem of authorized data deduplication. Data deduplication is a data compression technique for removing duplicate copies of identical data, and it is used in cloud storage to save bandwidth and to reduce the amount storage space. The technique is utilized to enhance the storage use and can likewise be applied to network data exchange to reduce the amount of bytes that must be sent. Keeping multiple data copies with the identical content, de-duplication removes redundant data by keeping only one copy and referring other identical data to that copy. De-duplication occurs either at block level or at file level. In file level de-duplication, it removed duplicate copies of the identical file. Deduplication can also take place in the block level that eliminates duplicate blocks of data that is occurred in non identical files [2].

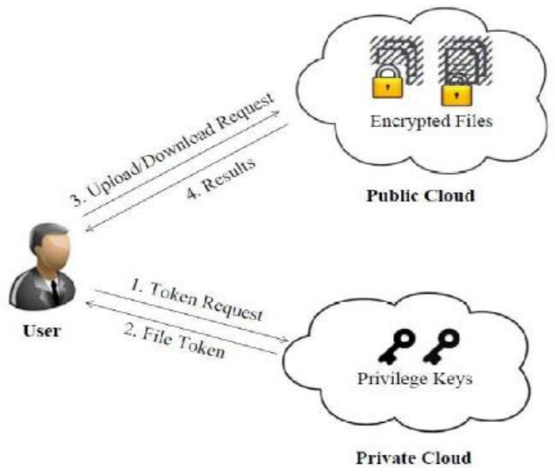

Figure 1-Architecture of Authorized deduplication

2. LITERATURE SURVEY

1. A Hybrid Cloud Approach for Secure Authorized Deduplication (G.Prashanthi et al, 2015)

From this paper, we referred- Several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct tested experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

2. A Hybrid Cloud Approach for Secure Authorized Deduplication (Gaurav Kakariya et al, 2014)

`From this paper, we referred- Cloud computing has reached a maturity that leads it into a productive phase. This means that most of the main issues with cloud computing have been addressed to a degree that clouds have become interesting for full commercial exploitation. This, however, does not mean that all the problems listed above have actually been solved, only that the according risks can be tolerated to a certain degree. Cloud computing is therefore still as much a research topic, as it is a market offering. For better confidentiality and security in cloud computing, we have proposed new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. The proposed system includes proof of data owner so it will help to implement better security issues in cloud computing.

3. A Hybrid Cloud Approach for Secure Authorized Deduplication (Jin Li et al, 2010)

From this paper, we referred-In this paper, the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

4. Secure Deduplication And Data Security With Efficient and Reliable CEKM (N.O.Agrawal et al, 2014) From this paper, we referred- The basic idea is that we can limit the damage of stolen data if we decrease the value of that stolen information to the attacker. We can achieve this through a 'preventive' disinformation attack. We posit that secure deduplication services can be implement given additional security features insider attacker on Deduplication and outsider attacker by using the detection of masquerade activity. The confusion of the attacker and the additional costs incurred to distinguish real from bogus information, and the deterrence effect which, although hard to measure, plays a significant role in preventing masquerade activity by risk-averse attackers. We posit that the combination of these security features will provide unprecedented levels of security for the deduplication.

5. Implementation of Hybrid Cloud Approach for Secure Authorized Deduplication (Jadapalli Nandini et al, 2015)

From this paper we referred- The notion of authorized data de-duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de-duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test-bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

6. A Hybrid Cloud Approach for Secure Authorized Deduplication (Jagadish et al, 2012)

From this paper, we referred- In this project, the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. In this project, we perform several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. As a proof of concept in this project, we implement a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments on our prototype. From this project, we show that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

7. A Study on Authorized Deduplication Techniques in Cloud Computing (Bhushan Choudhary et al, 2014)

From this paper, we referred- The thought of authorized information deduplication was proposed to ensure the information security by counting differential benefits of clients in the duplicate copy check. The presentation of a few new deduplication developments supporting authorized duplicate copy in hybrid cloud architecture, in that the duplicate check tokens of documents are produced by the private cloud server having private keys. Security check exhibits that the methods are secure regarding insider and outsider assaults detailed in the proposed security model. As an issue verification of idea, the developed model of the proposed authorized duplicate copy check method and tested the model. That showed the authorized duplicate copy check method experience minimum overhead comparing convergent encryption and data transfer.

8. Secure Authorized Deduplication on Cloud using Hybrid Cloud Approach (Ankita Mahajan) From this paper, we referred- We also presented several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

9. Secured Authorized Deduplication based Hybrid Cloud (Rajashree Shivshankar, 2014)

From this paper, we referred- Data deduplication is an important technique for eliminating redundant data.Instead of taking no. of same files, it stores only single copy of the file. In most organizations, storage system contains many pieces of duplicate data. . For example, the same file may be saved in several different places by different users. Deduplication eliminates these extra copies by saving just one copy of the data and replacing the other copies with pointers that lead back to the original copy. It is data compression technique for improving the bandwidth efficiency and storage utilization. Data deduplication most widely used in cloud computing. It makes data management scalable and storage problem in cloud computing. Data deduplication protects the confidentiality of sensitive data. Data deduplication works with convergent encryption technique to encrypt the data before uploading. . Companies frequently use deduplication in backup and disaster recovery applications.

3. TECHNOLOGY CLASSIFICATION OF DEDUPLICATION

Post-process deduplication (PPD):

It is also known as asynchronous deduplication or offline deduplication. It involves the removal of redundant data after a backup is completed and data has already been written to storage. The benefit of this method is that backup data is straightforward and takes very less time because the calculations of hash values and lookup takes place only after all of the data is stored.

In-line deduplication:

This process involves the calculation of hash values as the data enters the system. The benefit of this process over post-process is that it will take very less space because the calculation of hash values and the lookup process is completed before the data enters the database. So only one instance of a particular data is stores and the duplicate data is reference to the data present in the server.

Source deduplication:

This type of deduplication is the best suited to use at remote offices for backup to the cloud. The deduplication takes place typically within the system by regularly scanning new files creating hashes and compares them to the hashes of existing files. It offers a number of benefits, including the reduction of bandwidth and the amount of data that has to be sent to the cloud server.[3].

Target deduplication:

This is best suited for the use in the data center for the reduction of massive data sets. In this case, the client is unmodified and is not aware of any deduplication. Target deduplication requires that the target backup server or dedicated Hardware target appliance handles all of the deduplication. This process requires more network resources compared to source deduplication because the original data, with all its redundancy, must go over the network.

File Level and Sub-file Level Deduplication:

The full file level duplicates easily can be eliminated by calculating single checksum of complete file data and comparing it against existing checksums of the already backed up files. This method of deduplication is simple and fast, but the extent of deduplication is less, as this process does not address the problem of duplicate files or data-sets. The sub-file level deduplication breaks the file into smaller fixed or variable size blocks, and uses hash based algorithm to compare these blocks and find similar blocks.

Fixed-Length Blocks:

A fixed-length block approach divides the files into fixed size length blocks and uses a simple checksum-based approach (MD5/SHA etc.) to find the duplicates. This process has a limited effectiveness. The reason for this is that the primary opportunity for data reduction is in finding duplicate blocks in two transmitted datasets that are mostly- but not completely of same data segment.

Variable Length Data Segment technology:

This technique divides the data stream into variable-length data segments using a methodology that can find the same block boundaries in different locations and contexts. This allows the boundaries to float within the data stream so that changes in one part of the dataset have little or no impact on the boundaries in other location of the dataset.

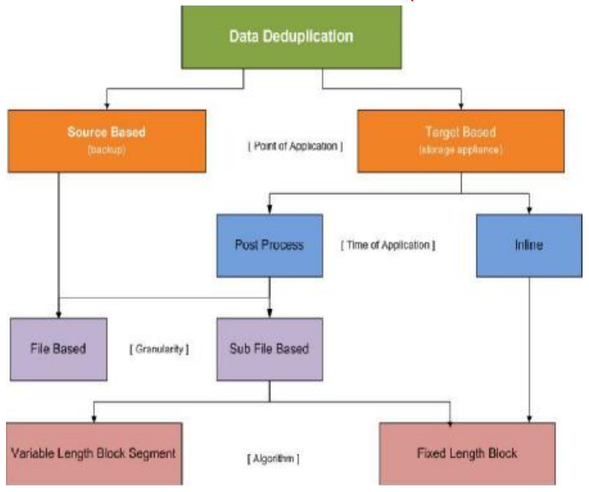

Figure2- DEDUPLICATION PROCESS

4. ANALYSES OF CHUNKING ALGORITHMS

Data De-duplication can be performed in two different ways, either Hash based where the fingerprint of the chunk is used in de-duplication of data or Content based, where the de-duplication is done by byte by byte comparison. Following section gives a brief study on such algorithms.[4,5,6]

4.1 Hash Based Chunking

Hash Based De-duplication involves using a hashing algorithm to identify the chunks of the data. The hash algorithm takes the chunk as the input and produces a cryptographic hash value for the chunk. The most commonly used hashing algorithms are SHA-1 [22] and MD-5[7]. The hash value is known as the fingerprint of the chunk. The chunks can either be of fixed length or variable length. If the fingerprint already exists in the chunk index, then this chunk is termed as duplicate and it is not stored into the disk, else if the chunk was not found in the chunk index, then this unique chunk is stored into the disk. Following are the two ways of chunking the data file.

4.2 Fixed length or Fixed blocks Chunking

Here the evaluation of data includes a fixed reference window used to look at segments of data during de-duplication process. It provides a fixed block boundary e.g. 4KB, or 8KB. Fixed length chunking is used most often when general purpose hardware is involved for carrying de-duplication. Nevertheless the fixed length chunking algorithm achieves significantly very less reduction than a variable length approach. The reason is because the duplicates are usually found between any two transmitting data set or any two consequent backup data sets, the two data sets with a small amount of difference are likely to have very few identical chunks. Advantage is that it requires the minimum CPU overhead, and it is fast and simple. Because of the block size or block boundaries being fixed, it results in boundary shifting problem, where if the data in the file is shifted, then it affects all the data following it, and the duplicates are not detected as a result of this.

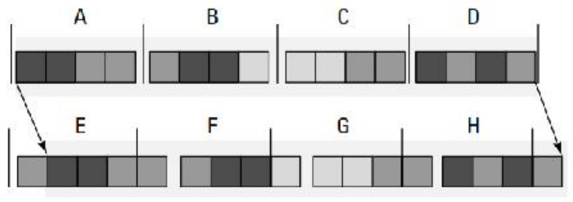

Figure 3- Fixed Length Chunking

Figure 3 illustrates the boundary shifting problem due to fixed size chunking, where chunks A, B, C and D are similar to chunks E, F, G and chunks H respectively. But due to the addition of some text in the beginning before the chunk E affects all the chunks following it and the duplicates are not detected due to the fixed window size.

4.3 Variable length or Variable block Chunking

Here the evaluation of data uses a variable length window to find duplicate data in stream or value of data processed. It divides the data stream into variable length data segments using a data dependent methodology that can find the same data block boundaries in different locations and contexts. Here the window size varies based on what algorithm is being used with average window size as 4KB. The most frequently used variable length chunking algorithm is TTTD Figure 2 illustrates the variable length chunking. Even after adding some data before the chunk E, neither the chunk E nor the chunks following it are affected. This way of creating variable length blocks makes the data to float inside the data file and helps in finding maximum number of duplicates[8].

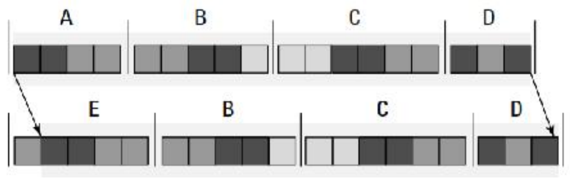

Figure 4- Variable Length Chunking

T. T. Thwel et.al. [2] have used the TTTD algorithm for chunking the data files. This paper has clearly specified about the procedure involved in the variable length chunking. It uses a minimum size and maximum size threshold for setting the maximum and minimum values of every chunk. Two divisor values namely main divisor and second divisor are also used for finding the boundary of the chunk. Main divisor finds the breakpoint and if it unsuccessful in doing so, then the backup breakpoint found using the second divisor acts as the breakpoint. But TTTD has a limitation due to the second divisor which mostly produces breakpoints which are near to the maximum threshold. This results in larger sized chunks where a lot of time is wasted in performing unwanted calculations and comparisons [9,10].

T. S. Moh et.al. [1] has proposed TTTD-S algorithm to eliminate the disadvantage of TTTD algorithm. It uses a new parameter called average threshold which is the average of maximum and minimum threshold. When this algorithm reaches this parameter, the original values of main divisor and second divisor is halved. These values are switched back to the original values once the breakpoint is found. This avoids unnecessary comparisons and calculations.

4.4 Content or Application Aware Based Chunking

F. Douglis et.al. [5] used the content aware de-duplication which is performed in a different way. Here the data is considered as an object. It takes the objects and compares it with the other objects for finding the duplicates in an efficient manner. Here the data is divided into large data segments and by using the knowledge of the content of the data, similar segments are determined and only the changed bytes between the objects are saved. This is a byte level comparison.[11,12]

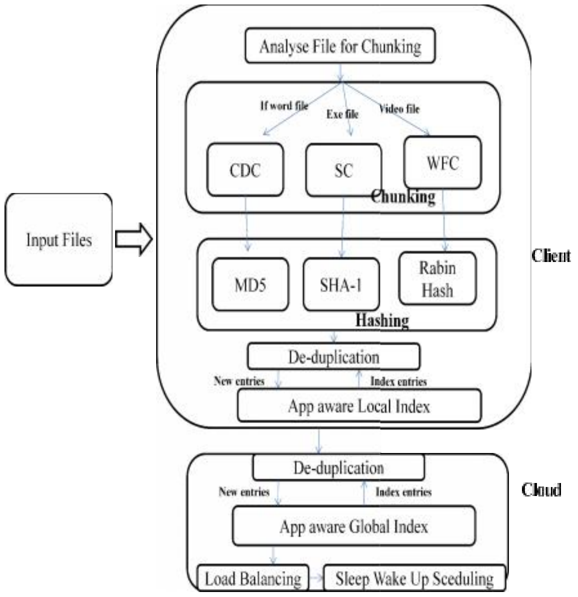

Figure 5- SYSTEM ARCHITECTURE OF CHUNKING ALGORITHMS

5. LOAD BALANCING FUNCTION IN DEDUPLICATION

The proposed method will focus on the following factors, user provisioning cost, security to storage, load balancing, and avoid redundancy. For providing these factors, the functions to perform are de-duplication process, chunking of data, hash function, load balancing and Application aware[13]. The de-duplication process performs the functions local de-duplication and global de-duplication to check the local cloud storage and the global cloud storage, which gives more effectiveness and latency of de-duplication. An intelligent data chunking method and Hash functions are performed, which splits the files into the chunks of data and apply hash functions to those chunks of data, which results in minimal computational overhead and high security to the cloud storage. Load balancing function is used to balance the load while storing the files into the resources of cloud. The main contributions of Deduplication is

1. Chunking of data with the file type
2. Hash function based on the file priorities
3. De-Duplication process to avoid duplicate files
4. Load Balancing to reduce overhead

The goal of this is to provide a scheme for balancing theLoad in the personal cloud computing which is categorizedinto several number of chunks which, for every cloud serveri, returns an estimate ni of the total number of chunk files forEach cloud server, so that each ni is within a constant factorof n, with high probability.[14,15,16] A DHT network is an over layon the application level. The logical proximity abstraction derived from the DHT does not necessarily match the physical proximity information in reality. That means a message traveling between two neighbours in a DHT overlay may travel along physical distance through several physical network links. In the load balancing algorithm, light nodes n may rejoin as a successor of are heavy node j. Then, the requested chunks migrated from j to i need to traverse several physical network links, thus generating Considerable network traffic and consuming significant network resources (i.e., the buffers in the switches on a communication path for transmitting a file chunk from a source node to a destination node).[17] We improve our proposal by exploiting physical network locality. Basically, instead of collecting as in vector per algorithm i around, each light node i gathers NV vectors. Each vector is built using the method introduced previously. From the NV vectors, the light node I seeks NV heavy nodes by invokingAlgorithm1 (i.e., SEEK) for each vector and then selects the physically closest heavy node based on the message round-trip delay.The pseudocodes for all types of intervals are depicted[18,19,20]

Pseudocode 1: Load balancing using distributed hashtable (DHT)

```
Short the number of files in the chunk with
number of user
State:=staying
If(predecessor is short) // number of files and
task is short
With probability ½ change state to leaving
If (state =leaving and predecessor .state
=staying)
{
P:=random(0.1)
P:=the node responsible for P
Contact consecutively the node P and its
6.log(u.ni)
successor on the ring
If ( a node R accepts)
Leave and rejoin in the middle of R
}
```

At any time, if any node contacts reject imbalanced Middle At any time, if any node contacts reject imbalanced Long Wait for contacts Id any node contacts accept the current load task and balanced In distributed file systems (e.g., Google GFS and HadoopHDFS), a constant number of replicas for each file chunk are maintained in distinct nodes to improve file availability with respect to node failures and departures. Our current load balancing algorithm does not treat replicas distinctly. It is unlikely that two or more replicas are placed in an identical node because of the random nature of our load rebalancing algorithm. More specifically, each under loaded node samples a number of nodes, each selected with a probability of 1/n, to share their loads (where n is the totalnumber of storage nodes).

CONCLUSION

This paper discusses the information about data deduplication for the cloud based systems. It includes the methods that are used to achieve cost effective storage and effective bandwidth usage by deduplication. The core concept involves eliminating the duplicate copies of the repeated data by using hashing algorithms . However, reliability and speed are at stake. However, data deduplication is the most crucial element for improving efficiency of the cloud system. An intelligent data chunking method and Hash functions are performed, which splits the files into the chunks of data and apply hash functions to those chunks of data, which results in minimal computational overhead and high security to the cloud storage. Therefore Load balancing function is used to balance the load while storing the files into the resources of cloud.

REFERENCES

[1]. Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang and Yang Xiang, "Secure Distributed Deduplication Systems with Improved Reliability" 0018-9340 (c) 2015 IEEE.

[2]. Jingwei Li, Jin Li, Dongqing Xie and Zhang Cai, "Secure Auditing and Deduplicating Data in Cloud [10].1109/TC.2015. 2389960, IEEE 2015 Transactions on Computers.

[3]. Prof. N.B. Kadu, Mr. Amit Tickoo, Mr.Saurabh I. Patil, Mr. Ganesh B. Divte, "A Hybrid Cloud

Approach for Secure Authorized Deduplication" International Journal of Scientific and Research Publications, April 2015

[4]."A SURVEY: DEDUPLICATION ONTOLOGIES, International Journal of Computer Applications (0975 – 8887) Volume 109 – No. 1, January 2015.

[5]. Yufeng Wang, Chiu C Tan, Ningfang Mi "Using Elasticity to Improve Inline Data Deduplication Storage Systems" 2014

[6]. Waraporn Leesakul, Paul Townend, Jie Xu, "Dynamic Data Deduplication in Cloud Storage" 2014 IEEE 8th International Symposium.

[7]. Jin Li, Xiaofeng Chen, Mingqiang Li, and Wenjing Lou. "Secure De-duplication with Efficient and Reliable Convergent Key Management". In IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 6, June 2014.

[8]. Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu," Application-Aware Local-Global Source Deduplication for Cloud Backup Services of Personal Storage" IEEE ,May 2014

[9]. Quanlu Zhang, Shenglong Li, Zhenhua Liy, Yuanjian Xingz, Zhi Yang, and Yafei Dai, "CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability" 10.1109, IEEE 2014 Transactions on Cloud Computing

[10] Chazelle, B., Kilian, J., Rubinfled, R. and Tal, A. 2004. The Bloomier Filter: an efficient data structure for static support lookup tables.

In Proceedings of the 15th annual ACM-SIAM symposium on Discrete Algorithms, 30-39.

[11] Wei, J., Jiang, H., Zhou, K. and Feng, D. 2013. Efficiently Representing Membership for Variable Large Data Sets. In Proceedings of the IEEE Transactions on Parallel and Distributed Systems, Vol.25, 960-970.

[12] https://www.wikipedia.org

[13] Bhagwat D., Eshghi, Long D. D. E., and Lilibridge M., "Extreme binning: Scalable, parallel deduplication for chunk-based file backup", Proceedings of the 7th IEEE International Symposium on Modelling, Analysis and Simulation (MASCOTS), 1-9, 2009.

[14] Zhu, B., Li, k., and Patterson, H. 2008. Avoiding the disk bottleneck in the Data Domain deduplication file system. In Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST), 269-282.

[15] Lillibridge, M., Eshghi, K., Bhagwat, D., Trezise, G. and Camble, P. 2009. Sparse indexing: Large scale, inline deduplication using sampling and locality. In Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST), 111-123.

[16] Can, W., Qin, Z. G., Yang, L. and Juan, W. 2012. A Fast Duplicate Chunk Identifying Method Based on Hierarchical Indexing Structure. In Proceedings on IEEE International Conference on Industrial Control and Electronics Engineering, 624-627.

[17] Wildani, A., Miller, E. L., and Rodeh, O. 2013. HANDS: A Heuristically arranged non-backup inline
deduplication system. In Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE), 446-457.

[18] Sengar, S.S. and Mishra, M. 2012. E-DAID: An Efficient Distributed Architecture for inline data deduplication. In Proceedings of the IEEE International Conference on Communication Systems and Network Technologies (CSNT), 438-442.

[19] Xia, W., Jiang, H., Feng, D. and Hua, Y. 2011. SiLo: a similarity-locality based near exact deduplication scheme with low RAM overhead and high throughput. In Proceedings of the USENIX Annual Technical Conference (ATC), 26-28.

[20] Andre, B., Dirk, M. and Kaiser, J. 2013. Block locality caching for data deduplication system. In Proceedings of the 6th ACM International Systems and Storage Conference, 19-24.