A Survey on Data Deduplication in Cloud Storage Environment

Manikantan U.V.[1], Prof.Mahesh G.[2]
[1](Department of Information Science and Engineering, Acharya Institute of Technology, Bangalore)
[2](Department of Information Science and Engineering, Acharya Institute of Technology, Bangalore)

ABSTRACT
With the increasing use of cloud storage platform for storage and processing, there is a growing demand of some mechanism or methodology which will provide the facility of eliminating redundant data and thereby achieving better space and bandwidth requirements of storage services. In other words, it should provide a capability of a better efficiency when applied across multiple users. In this context, Cloud Service Providers normally use Deduplication , which stores only a single copy of each file or block by eliminating redundant data. But providing a secure Deduplication is an uphill task. In this regard, an effort is made to present a survey on the different aspects of Deduplication.

Keywords *– Cloud,Deduplication,Deduplication Storage system,Deduplication Operation, Fingerprinting based Deduplication,Types of deduplication.*

I. INTRODUCTION
The rapid growth of data across multiple platforms has drastically increased the demand of Cloud storage infrastructures. Cloud Storage is a representation of data storage where the digital data is stored in virtualized logical pools, more specifically saying , it is stored in multiple physical storage which spans multiple servers and the physical environment is owned, administered and managed by a hosting company. More specifically saying the Cloud Storage Providers are responsible for keeping track of the data availability and accessibility and the physical security concerning the data.Cloud computing involves both software and hardware which are distributed to users as a service by the service providers. With the rapid growth in the domain of cloud computing ,even more and much versatile services are emerging up.But the basic domains can be illustrated in forms of services such as Paas(Platform as a service), as SaaS(software as a service) and Iaas(software as a service).
Cloud storage can be viewed as a model in which the online storage is networked and records are stored on multiple storage remote servers. The ideology of the cloud storage is derived from cloud computing. In other words, it basically denotes to a storage device accessed over the internet via Web service application program interface (API).HDFS (Hadoop Distributed File System, hadoop.apache.org) is one such example, which is basically a distributed file system that runs on the commodity hardware. The concept was introduced by Apache for managing the rapid growth in the data volume.
In the context of Cloud storage and corresponding technology of user data sharing in multiple platforms, the probability of getting redundant internet data is high.This can be prevented with the introduction of Deduplication. Data Deduplication basically describes a set of methodologies which reduce the storage capacity to store the data and also the amount of data that is being transferred over the network also get reduced [1].

II. DATA DEDPULICATION
Data deduplication refers to methodologies that store only a single copy of redundant data and thereby provide a single copy. By eliminating redundant data both disk space and network bandwidth[2].With respect to service providers, it offers secondary cost savings in power and cooling which is achieved by reducing the number of spindles[3]. It ensures that only one copy of data is stored in the datacenter. Therefore it clearly decreases the size of datacenter. So it basically means that the number of the replicated copies of data that were usually duplicated on the cloud server can be controlled and managed easily to shrink the physical storage space.The recent statistics has found out that deduplication is the most influential storage technology and is predicted to provide 75% of all backups in the next few years.The efficiency of any Data Deduplication application can be effectively measured by the 1. Dedupe ratio where Dedupe Ratio= Size of Actual Data/Size of Data after Deduplication 2. Throughput (Megabytes of Data Deduplicated per sec).Following are the parameters which govern the dedupe ratio and throughput-
1. Nature of data to be deduplicated

2. Where is the Deduplication applied?-either on source device or target device
3. If Data Deduplication is inline or a post processing application
4. Implementation of Data Deduplication[4]

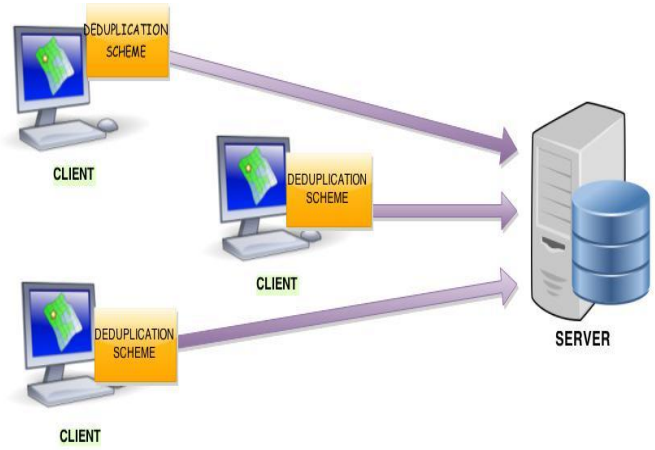

Fig.1. Source Deduplication

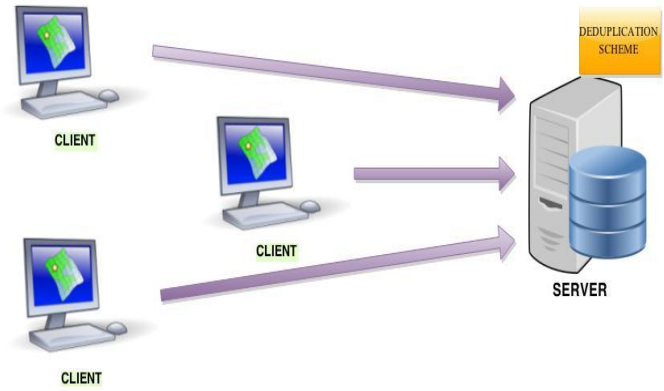

Fig.2. Target Deduplication

The data deduplication in storage systems follow two types of approaches- Finger printing Based and Delta based Approach. Again, the algorithm for fingerprinting involves the following methodology like the sequence of chunking, duplicate detection and storage. The first methodology known as 'Chunking' involves the splitting of data into non-overlapping blocks called "chunks". The processing of one chunk is independent of the other chunk. The following are the two commonly used strategies with respect to chunking- Static Chunking and Content-defined Chunking [5]. The first approach i.e. Static Chunking involves splitting of data into chunks which are always similar in size. The size of Chunk is always a multiple of the disk sector or the block size of the file system. This is also known as Fixed –size Chunking[6, 7, 8, 9] or Fixed block Chunking [10, 11], or Fixed Chunking[12].

The next approach i.e. Content defined Chunking does not involve splitting of data into chunks of similar size. But this is usually preferred more in backup systems because it is not prone to the "boundary- shifting effect" [13], which reduces the redundancy found by the data deduplication systems. This mainly occurs due to the fact that the data gets slightly shifted i.e. other data was inserted into the main data stream. As a result,the conventional approach of static chunking is unable to identify the duplication because the chunks are not similar. The content-defined chunking overcomes this situation by realigning the chunking methodology with the content and the similar chunks are created as before and thus duplication can be identified. Content-defined chunking is found to be give high deduplication ratio[14, 15] with respect to the backup workloads. Hence, it is the most widely used methodology in most deduplication systems for backup workloads [16, 17, 18, 19, 20].

Content-defined chunking is originally related to Brin et al.'s research on copy detection which was used for digital documents [21]. In In Muthitacharoen et al.'s "Low Bandwidth File system" (LBFS) [22], the approach was refined to use Rabin's fingerprinting method [23, 24, 25]. This methodology is still commonly used as a rolling fingerprinting method in content-defined chunking. Policroniades and Pratt were the first to coin the terminology "content-defined chunking". The approach is also called "variable-size chunking" [26, 27, 28, 29] and "Rabin chunking" and also (incorrectly)"Rabin fingerprinting".

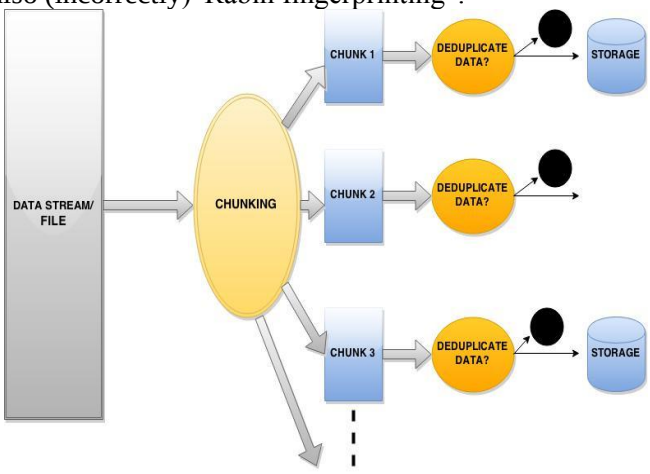

Fig.3.Illustrations of Three step FingerPrinting-Based deduplication

Duplicate classification: In this step, the duplication or the redundancy of a chunk is checked by the deduplication system. The chunk is added to the system only if there is no redundant copies of it. If it is not there ,then its data is stored. Most of the deduplication systems that uses the fingerprinting mechanism use the approach of "compare-by-hash" methodology[Hen03, Bla06]. This mechanism involves the fingerprinting of chunks by cryptographic fingerprinting method. For the purpose of efficient deduplication scheme ,the stored chunks are given a "chunk-index" which contains the fingerprint of all the stored chunks. This is based on the ideology that if the fingerprint of a chunk is already present in the chunk index, it is believed that the data with respect to the new chunk and the already existing chunk are identical and thus it is instantly classified and governed as duplicate. The other widely used names of the chunk index are "segment index", "fingerprint database" , and "fingerprint index" [30].

Storage : The chunks which are classified as new are then stored by the depulication system. One of the methodology which is used with respect to the storage is to group chunks together and store them in a "container" data structure[30]. The other popular data structure is called "arena" which is kind of similar to the container is used in Venti system . To overcome the chunk lookup disk bottleneck ,container played a significant function as the chunks are grouped together to capture a temporal locality. This temporal locality is used to overcome the chunk lookup disk bottleneck as proposed in Zhu et al.'s approach. In addition to the much famous data structure "chunk index", most of the deduplication system has an additional data structure to gather the information that is necessary to recover the file contents- if it is a file system or the block contents –if it is a block device. This additional data structure is known as "file-recipe" in file based deduplication systems[30]. This concept was coined by Tolia et al. [30]. The file recipe includes a list of identifiers commonly known as "chunk identifiers". The chunk identifiers are generated from the cryptographic fingerprint of a chunk. The reconstruction of the original file is possible by reading the chunk data of the fingerprints and followed by concatenating their data in the same order as prescribed by the file recipe.

Delta-Based data dedpulication approach follows the same chunking step, but it is not searching the similar, but necessarily identical data blocks. In this methodology, while writing a new chunk ,they search for similar chunks and it is then followed by storing only a differential encoding between the new data and the already existing chunks which is found to be identical. The most noted example of delta based approach is the design of the IBM ProtecTier backup system [30].

2.1 Implementing Data Deduplication

The first and the foremost requirement is to identify the basic entity of data upon which the deduplication can be performed. There are basically two levels of deduplication based on the operational area-

1. File –level deduplication- This is performed over a single file. In this type of deduplication ,the mechanism of eliminating redundant copies are done on two or more files if they are similar based on the hash values[31].
2. Block-level de-duplication-As the name indicates, the block level deduplication deals with the elimination of redundant data copies with respect to blocks. The procedure involves dividing files into blocks and storing a single copy of each block. It uses either fixed-sized blocks or variable-sized chunks for deduplication at block level.

Data deduplication can be further divided into target based and source based on the basis of the targeted area.

1. Target based de-duplication: The deduplication is performed on the target data storage center and the client or source is totally unaware of the the deduplication mechanism happening at the target. This efficiently improves the storage capability but consumes bandwidth inappropriately.
2. Source based de-duplication: The deduplication is performed on the data at the source before the actual transmission of data to the target. This is performed with the help of a deduplication-aware-backup-agent which is installed on the client, which backs up only unique data. The result is enormous saving in bandwidth as well as storage efficiency. But, it impose extra computational load on the back up client.

2.2. Types of Data Dedpulication

The point in time at which the deduplication algorithm is performed is known as the Deduplication timing[31]. The timing of the algorithm always place a huge constraint on how much time it has to perform data deduplication and the level of knowledge the algorithm know about the new file data. With respect to the timing, the deduplication scenario can be classified into : offline deduplication and online deduplication.

2.2.1. Offline Deduplication

This scenario comes into play when the data deduplication is performed offline. In this case, all data is first written into the storage system first and deduplication is carried out later. The biggest benefit of this orientation is that when the deduplication

methodology is carried on, the system has a static view of the entire file system and has a full knowledge about all the data it has access to and can drastically improve the deduplication efficiency. But the performance can be little slow since it may compare the file data to all data stored on the disk. Again ,the data written to the storage system must be batched until the next scheduled deduplication time. This creates an un-dismissible delay between when the data is written and when space is recreated by removing duplicate data[31].

2.2.2 Online De-duplication

As compared to Offline data deduplication, the online one is performed as the data is being written to disk. The main advantage of this scenario is that this allows for immediate space reclamation. But there will be an increase in write latency- since the write is blocked until all redundant file data is eliminated.

There are a number of existing technologies that have already been created which help deduplication methodology once the timing of deduplication has been set. The most commonly used are, as propounded by Mandagere[31] : WFH-whole file hashing,DE-delta encoding and SFH-sub file hashing.

2.2.3. Whole File Hashing

The methodology used here is a hashing function. An entire file is first sent to this hashing function. The hashing function generally being used is MD-5 and SHA-1. The result of the hashing function is a cryptographic hash which forms the basis for the identification of entire duplicate file. The upside of this methodology is the fast execution with low computation and low metadata overhead. It works with full efficiency for full system backups when whole duplicate files are predominantly more common. But the downside to this methodology is that with growing granularity of duplicate files, it prevents from matching two files which only differ by a byte of data.

2.2.4. Sub File Hashing

As the name indicates, the file initially is divided into a number of smaller pieces before the actual data deduplication. The division of file mainly depends upon the type of SFH that is being used. The SFH methodology is divided into basically two types-namely fixed-size chunking and variable-length chunking.In fixed-size chunking scenario, a file is broken down into a number of static or fixed sized pieces called "chunks".In variable-length chunking scenario, a file is divided into chunks of varying length unlike the fixed size scenario.Methodologies like Rabin fingerprinting [31] are used to determine "chunk boundaries". The broken pieces of file are given to a cryptographic hash function like SHA-1 or MD-5 to calculate the "chunk identifier". The chunk identifier is the key parameter in locating redundant data. As a matter of fact, both these SFH technologies efficiently find the duplicate data at a finer granularity but come with a price. The chunk identifiers should also be kept readily accessible in order to find duplicate chunks. This additional amount of 'additional metadata' is practically dependent on the technique, but is not avoidable.

2.2.5. Delta Encoding

DE is basically derived from the mathematical and scientific symbol- the delta symbol. In these fields, delta is basically used to measure the "change or rate of change" in an object. DE is used to calculate the difference between a target object and a source object. Suppose block A is the source object and B is the target object, then the Delta Encoding of B is the difference between A and B that is unique to B and not A. The storage of the difference between the two blocks depends upon how the DE is applied. The usage of DE happens most often when SFH does not produce results but there is a high similarity between the two items or blocks or chunks and in a way saying storing the difference would take less space than storing the non-duplicate block.

III. FACTORS GOVERNING DATA DEDUPLICATION

The implementation of data dedpulication is done with respect to the following considerations-

1. The methodology to find data duplications in the system.
2. The methodology to remove repetitions of data by the continuous manipulations and maintenance so as to achieve efficient data deduplication in the system.

The system can employ MD5 and SHA-1[31] algorithm to generate the unique fingerprint with respect to each file or blocks of data. This is followed by the creation of the fingerprint index to check for the duplications. The duplications can be discovered by means of the following methodologies:

1. Data blocks or File bit by bit comparison- this is based on the data blocks or file bit by bit comparison by the system. The advantage is that it gives accuracy but with the cost of consuming additional time.
2. Data blocks or File comparison by hash values- the comparison is done with respect to the hash values. This methodology is more efficient , but the chances of

accidental collisions is high. Usage of a combination hash value will greatly reduce the collision probability.

IV. DATA DEDUPLICATION OPERATION

The methodology of deduplication is can be illustrated with the following example. The below figure shows the representations of three files namely A.txt,B.txt and C.txt which are to written into the database. First let's take the case of the first file i.e. A.txt. [31]The Data Depulication system breaks the file into four namely A,B,C, and D. The number of the partitions depend upon the deduplication scheme. The partition of the file into different segments is followed by the addition of the hash identifier to all segments for reconstruction[31]. It is then stored in the database separately. The second file B.txt is again divided into four segments just like the way we did the first file. It is similarly divided into A,B,C and D. Both of the file A.txt and B.txt have the same set of segments. It means that the deduplication system will not store it. This is done by the deletion of the new copy segments and providing a link to the last stored file segments. Now let us take a different file C.txt. The deduplication system will break this file into several parts,which is E,B,C,D ,as per our example. But here the only part which is not common is the 'E' segment whereas the other segments are already stored in the database. Hence the system will only keep a copy of E segment and provide a direct link for other segments which is already stored in the database. As a result only five out of the total 12 blocks or segments are stored in the database as explained in the diagram. Hence it is conviently clear that it drastically reduces the overall storage space. In other words if the data size of one segment is 1MB then the total amount of space needed will be 12MB. So with the help of Data dedpulication scheme ,the system now only need to 5MB instead of the 12MB. So we saved 7MB.

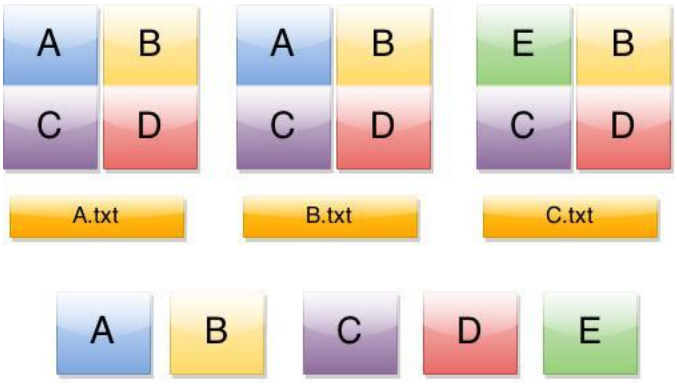

Fig. 4. Sample files data and segments

V. DE-DUPLICATION STORAGE SYSTEM

There are several deduplication storage system that is being used with respect to different storage purpose. The following are the few:

Venti[31]: It is a type of network storage system. The deduplication scheme inside this storage system is based on the identification of identical hash values of the data blocks so that it reduces the overall usage of the storage space. Venti follows the concept of 'write-one' policy to avoid collision of the data and is often associated with the generation of blocks for huge storage applications. The disadvantage with this system is that it is not suitable to deal with a vast amount of data and it does not offer scalability.

HYDRAstor[31]:This system is highly scalable secondary storage solution, which offers a decentralized hash index for the grid of storage nodes, which act as the back end and a traditional file interface as the front end. The back end of the HYDRAstor is able to organize large scale, variable size and content addressed ,immutable and highly resilient data blocks with the help of Directed Acyclic Graph.

Extreme Binning[31]: This system gives scalability with the help of a paralleled deduplication approach which aims at a non tradional backup workload. This normally involves the compostion of low-locality individual files. It prefers the usage of file similarity over the locality and allows only one disk access for blocks lookup for file. The mechanism involves arranging similar data files into bins and removing duplicated chunks from each bin. The deduplication is done with respect to the different bins. One more speciality of the Extreme Binning is that it keeps only primary index in memory so that it reduces RAM consumption.

MAD2[31]: The main feature of this system is its accuracy. It provides an accurate deduplication back up service which primarily works on both at the file level and at the block level. The methodology involves the following techniques- a hash bucket matrix, a bloom filter array, a distributed Hash Table based load balancing and dual cache, in order to achieve the desired performance.

Duplicate Data Elimination (DDE)[31]: It works on the combination of copy-on-write, lazy updates and content hashing to identify and coalesce identical blocks of data in SAN system. The main difference between DDE and the other methodologies is that it exactly deduplicates and calculates the corresponding hash values of the data blocks right at the client side itself, before the actual transmission of data.

VI. BENEFIT

The main benefits from data deduplication can be classified as follows:

1. The drastic decrease in the amount of storage space required for a given set of data. This can be mainly attributed to the scenario where the application involves so many repeated duplications of data that are stored on a single storage system.
2. The drastic decrease in the amount of network bandwidth consumed while transferring the number of data bytes between two end points.
3. Deduplication also has a profound impact on the virtual servers as it permits nominally distinct system files of each virtual server to collaborate into a single storage system. Even if one sever alters a file, the dedpulication scheme will not alter the content on the remaining servers.

VII. CONCLUSION

The paper primarily focuses on the Data Deduplication methodology and terminologies with respect to the storage system. Data Deduplication is the new data compaction technology which removes duplicates in data. It differs from the compression techniques by working on the data at sub-file level where as compression encodes the data in the file to reduce its storage requirement. However , compression can be used to augment data deduplication to provide higher dedupe ratio (Size of data to be written / Size of data on disk) . Small to large enterprises have been adopting this new technology as it gives significant Return on Investment by: Reducing the storage capacity required to store the data and reducing network bandwidth required to transfer the data.

REFERENCES

[1] Dongfang Zhao, Kan Qiao, Ioan Raic,y ,"HyCache+: Towards Scalable High-Performance Caching Middleware for Parallel File Systems", Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357, 2014.

[2] Dirk Meister, Jürgen Kaiser," Block Locality Caching for Data Deduplication". In Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST). USENIX, February 2013.

[3] Danny Harnik-IBM Haifa Research Lab,Benny Pinkas-Bar Ilan University ,Alexandra Shulman-Peleg-IBM Haifa Research Lab "Side channels in cloud services, the case of deduplication in cloud storage" Issue No.06 - November/December (2010 vol.8)

[4] Jan Stanek, Alessandro Sorniotti,Elli Androulaki, and Lukas Kencl "A Secure Data Deduplication Scheme for Cloud Storage" Published in: RZ3852 in 2013

[5] Calicrates Policroniades and Ian Pratt. Alternatives for detecting redun-dancy in storage systems data. In Proceedings of the 2004 USENIX Annual Technical Conference (ATC), pages 73–86. USENIX, 2004.

[6] Keren Jin and Ethan L. Miller. The effectiveness of deduplication on virtual machine disk images. In Proceedings of the 2nd Israeli Experimental Systems Conference (SYSTOR). ACM, 2009.

[7] Jaehong Min, Daeyoung Yoon, and Youjip Won. Efficient deduplication techniques for modern backup operation. IEEE Transactions on Computers, 60(6):824–840, 2011

[8] Stephanie N Jones. Online de-duplication in log-structured file system for primary storage. Master's thesis, University of California, Santa Cruz, 2010.

[9] Cornel Constantinescu, Joseph Glider, and David Chambliss. Mixing dedu-plication and compression on active data sets. In Data Compression Confer-ence (DCC), 2011, pages 393–402. IEEE, 2011.

[10] Dutch T. Meyer and William J. Bolosky. A study of practical deduplication. Trans. Storage, 7(4):14:1–14:20, February 2012.

[11] Nagapramod Mandagere, Pin Zhou, Mark A Smith,and Sandeep Uttam-chandani. Demystifying data deduplication.InProceedings of the Middleware Conference Companion, 2008.

[12] Vasily Tarasov, Amar Mudrankit, Will Buik, Philip Shilane, Geoff Kuenning, and Erez Zadok.Generating realistic datasets for deduplication analysis. In Proceedings of the 2012 USENIX Annual Technical Conference (ATC). USENIX,2012.

[13] Kave Eshghi and Husi Khuern Tang. A framework for analyzing and im-proving content-based chunking algorithms. Hewlett-Packard Labs Technical Report TR, 30, 2005.

[14] Dirk Meister and André Brinkmann. Multi-level comparison of data dedu-plication in a backup scenario. In Proceedings of the 2nd Israeli Experimental Systems Conference(SYSTOR), pages 8:1–8:12. ACM,2009.

[15] Calicrates Policroniades and Ian Pratt. Alternatives for detecting redun-dancy in storage systems data. In Proceedings of the 2004 USENIX Annual Technical Conference (ATC), pages 73–86. USENIX, 2004.

[16] Benjamin. Zhu, Kai Li, and Hugo Patterson. Avoiding the disk bottleneck in the Data

Domain deduplication file system.In Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST).USENIX, 2008.
[17] Jiansheng Wei, Hong Jiang, Ke Zhou, and Dan Feng. MAD2:A scalable high-throughput exact deduplication approach for network backup ser-vices. In Proceedings of the 26th IEEE Conference on Mass Storage Systems and Technologies (MSST),pages 1–14. IEEE, May 2010.
[18] Dirk Meister and André Brinkmann. dedupv1: Improving deduplication throughput using solid state drives. In Proceedings of the 26th IEEE Confer-ence on Mass Storage Systems and Technologies (MSST). IEEE,May 2010.
[19] Deepavali Bhagwat, Kave Eshghi, Darrell D. E. Long, and Mark Lillib-ridge. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. In Proceedings of the 17th IEEE International Symposium on Mod-eling, Analysis, and Simulation (MASCOTS), pages 1–9. IEEE, 2009.
[20] Mark Lillibridge, Kave Eshghi, and Deepavali Bhagwat. Improving restore speed for backup systems that use inline chunk-based deduplication. In Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST). USENIX, February 2013.
[21] Sergey Brin, James Davis, and Hector Garcia-Molina.Copy detection mechanisms for digital documents.In ACM SIGMOD Record, volume 24, pages 398–409.ACM,1995.
[22] Athicha Muthitacharoen, Benjie Chen, and D. Mazières. A low-bandwidth network file system. SIGOPS Oper. Syst. Rev.,35(5):174–187, October 2001.
[23] Michael O. Rabin. Fingerprinting by random polynomials. Technical report, Center for Research in Computing Technology, Harvard University, 1981.
[24] Andre Z. Broder. Some applications of rabin's fingerprinting method. In Sequences II: Methods in Communications, Security, and Computer Science, vol-ume 993,page 143152, 1993.
[25] Calvin Chan and Hahua Lu. Fingerprinting using polynomial (rabin's method). Faculty of Science, University of Alberta, CMPUT690 Term Project, 2001.
[26] Mark Lillibridge, Kave Eshghi, Deepavali Bhagwat, Vinay Deolalikar, Greg Trezise, and Peter Camble. Sparse indexing: Large scale, inline deduplica-tion using sampling and locality. In Proceedings of the 7th USENIXConference on File and Storage Technologies (FAST). USENIX, 2009
[27] Keren Jin and Ethan L. Miller. The effectiveness of deduplication on virtual machine disk images. In Proceedings of the 2nd Israeli Experimental Systems Conference (SYSTOR). ACM, 2009.
[28] Jaehong Min, Daeyoung Yoon, and Youjip Won. Efficient deduplication techniques for modern backup operation.IEEE Transactions on Computers, 60(6):824–840, 2011.
[29] Erik Kruus, Cristian Ungureanu, and Cezary Dubnicki. Bimodal content defined chunking for backup streams. In Proceedings of the 8th USENIX Conference on File and storage technologies (FAST), page 18.USENIX, 2010.
[30] Dirk Meister: Advanced Data Deduplication Techniques and their Application D77 Dissertation Johannes Gutenberg University Mainz
[31] Deepak Mishra ,Dr. Sanjeev Sharma-Comprehensive study of data de-duplication International Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV
[32] Stephanie N. Jones Online De-duplication in a Log-Structured File System for Primary Storage Technical Report UCSC-SSRC-11 03 May 2011