IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Volume-6, Special Issue-2, March 2018 E-ISSN: 2347-2693

Deduplicates In Big Data: A Technical Survey

A.Sahaya Jenitha[1], V.Sinthu Janita Prakash[2]

[1*]Department of Computer Applications, Cauvery College for women, Trichy-18, India
[2]Department of Computer Applications, Cauvery College for women, Trichy-18, India
Available online at: www.ijcseonline.org

Abstract- Deduplication is a task of identifying one or more records in repository that represents same object or entity. The problem is that the same data may be represented in different way in every database. While merging the databases, duplicates occur despite different schemas, writing styles or misspellings. They are called as replicas. Removing replicas from the repositories provides high quality information and saves processing time. With the development of cloud computing through virtualization technology, creation of VMs rapidly increasing, this in turn increases data centres. Backup in virtualized environments takes the snapshot of VM called VM image and moved to backup device. Data is duplicated by VMs for many purposes like backup, fault tolerance, consistency, disaster recovery, high availability, etc., these results in unnecessary consumption of resources, such as network bandwidth and storage space. Data Deduplication is a process of detecting and removing duplicate data thus the amount of data, energy consumption and network bandwidth is reduced. This paper describes Deduplication methods for large scale databases (Big data) and several Deduplication techniques like Extreme Binning, MAD2, and Multi-level Deduplication where Deduplication is performed in backup services. The paper also describes Cloud spider, Liquid Deduplication techniques for VM images in Big Data extracted from cloud environment, their comparison based on several factors.

***Keywords*:** Deduplication, Big data, Cloud, Live Virtual Machine Migration, Cloud spider, MAD2, Extreme Binning, Liquid, SAFE, Multi-Level Deduplication.

I. INTRODUCTION

Cloud Computing is defined as "Dynamic provision of hardware and software services as a utility on demand through interconnected virtualized computers" [1] and is achieved through virtualization. Virtualization technology makes the creation of many virtual machines in the cloud environment. Running virtual machines there will be lot of data duplicated for backup, achieving consistency, achieving high availability, disaster recovery etc, Copying and storing the state of VM is called VM image. VM image is used in VM migration, backup and restore operations [2]. Moving one VM image from source host to target without suspending the source host is called as Live virtual machine migration [3]. Virtual Machine information like VM internal state and external state data is of huge size. In Live VM Migration, Migrating entire information from source machine to target machine needs utilization of high bandwidth. Hence it is necessary to deduplicate the redundant data before migration. Deduplication consists of identification and elimination of redundant data [4]. When redundant data is detected, the data is discarded and the respective pointer to data is created for the migrated data and is transferred to target using Deduplication technology. Thus, the available bandwidth is only utilized for transferring the entire VM image to target. Data Deduplication is generally used in back up methods. Backing up duplicate data results in more storage and network bandwidth. By using data Deduplication technology in back up methods helps to reduce network burden and allows backups on the disk [5]. Less time is taken to backup the data

In this paper section II describes the basic Deduplication process. Section III gives the detailed description of various techniques. Section IV gives comparison of mentioned Deduplication techniques. Section V discusses the findings from the comparison in section IV. Conclusion is given in section VI based on the survey.

II. METHODS OF DEDUPLICATION

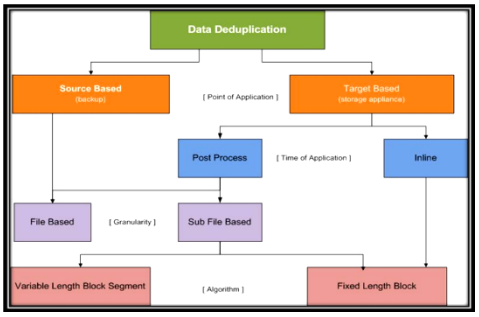

Fig 1 Based on data deduplicated there are two methods in Deduplication

A. File Level Deduplication:

This method first detects the identical files and is removed. One copy of file is stored. A Pointer is used to point the original file for the subsequent copies. This method doesn't consider the contents present inside the file. For example two document files with simple title change are stored as two different files. The advantage of this method is simple and fast. This method is also known as *Single Instance Storage [4]*.

B. Block or Sub file Deduplication:

The File is divided into few chunks called blocks and duplicate blocks are detected using specialized hash algorithm. If the data is unique written into disk else only pointer is used to point the disk location. According to the size of block there are two processes in block Deduplication.

a) Fixed-Length block Deduplication breaks the data into fixed size blocks. The disadvantage of this method is it fails to find the redundant data as a small change in the block result rewritten of all subsequent blocks to the disk. But this method is fast, simple and minimum CPU overhead.

b) Variable-Length block Deduplication breaks the data into variable size blocks. The advantage of this method is if any change occurs the boundary of that block is changed and no change in subsequent blocks and it saves more storage space when compared with fixed-length block Deduplication. This method requires more CPU cycles to identify block boundaries and for scanning entire file.

B. Based on implementation methods there are two methods in Deduplication.

- **Source/Client based Deduplication:**

The complete Deduplication process is done at source/client side before sending the data to a backup device. Only unique data is transferred to the backup device with the minimum available band width and it needs less space.

- **Target based Deduplication:**

The Deduplication process is done at back up device, when it receives the data with all its redundancy. This method needs more network bandwidth and it offloads backup client from Deduplication process. Based on when the Deduplication is done in back up device there are two methods [4].

Inline Deduplication

Allows Deduplication immediately after receiving the data at backup device. This method requires less storage capacity needed for backup.

Post-process Deduplication

Allows Deduplication after the received data is written into disk i.e., Deduplication is scheduled later. This method requires more storage space to store backup data.

C. Based on how Deduplication is done, there are two methods.

- **Hash based Deduplication:**

In the above mentioned methods i.e., file and block level Deduplication this technology is used to identify whether two files or blocks are same. A Hash is generated by using algorithms like MD5, SHA-1for files or blocks. For any two files or blocks it generates same hash value then the two files or blocks are identical and is not stored, if the generated hash value is different, then they are different and is stored in disk [5].

- **Content or Application-aware Deduplication:**

It divides the data into large segments by knowing the content of the objects like files, database objects, application objects. Then it finds the redundant segments and stores only the bytes changed in the two segments, hence known as byte level Deduplication [5].

III. OTHER TECHNIQUES

A. **Extreme Binning:**

The file is divided into variable size chunks using sliding window chunking algorithm [15] using Rabin fingerprints. Using MD5 [16] and SHA [17] a cryptographic hash or chunk ID is calculated. Chunk IDs are used to detect the duplicates. New chunks are written to disk and are updated in index along with their chunk IDs. Metadata about each chunk like size and retrieval information present in index. The percentage of Deduplication based on content overlapping in the data, chunking method [18] and granularity of chunks [19]. Deduplication is better for smaller chunks.

In Extreme Binning chunk index is divided into two tiers. The first tier is primary index present in RAM. Per file one chunk ID entry in primary index. This is called representative chunk ID of that file which it also contains pointer to bin. The Second tier is mini secondary index called bin, where the entire remaining chunk IDs of the file and their chunk size are Border's Theorem [20]. Minimum chunk ID is the minimum element of H(S1) and H(S2) where H is the set of hashes of the elements of S1 and S2 and is the representative chunk ID. Minimum chunk IDs are same with high probability when two files are similar. When a file is arrived for back up, chunking is done, representative index is find, hash for file is calculated and checked whether the representative chunk ID is present in it or not and if matched and whole file hash is not matched then secondary index or bin is created and all unique chunk IDs are added to this bin and are written to the disk. Now primary index is updated. If matched the corresponding bin is loaded in RAM and all the

remaining chunk IDs are retrieved if chunk IDs are not present they are added to the bin and again written to the disk. The primary index is not updated. If whole file hash is matches Deduplication is complete and references are updated.

B. MAD2:

Jiansheng et al. [7] proposed an approach, which is used for network backup services to perform Deduplication which addresses the following challenges.

Duplicate-look up disk bottleneck:

1) In traditional Deduplication approaches detection of duplicates add an index instead of writing the duplicate file to the disk. The index becomes larger than RAM as the volume of data grows which degrades the performance.

Storage node island effect:

2) Deduplication is success in multiple storage nodes not in multiple servers.

This approach uses the following four techniques.

Locality-Preserved Hash Bucket Matrix:
Detection of duplicates process is speed up by preserving locality of files and chunks using Hash Bucket Matrix (HBM). Fingerprints are computed for all files or chunks by using MD5 or SHA-1 hash algorithms. A range of fingerprints is called fingerprint space which is partitioned into *n super buckets* [7] of each size. Each super bucket contains *buckets* of same size. Collection of every bucket of each super bucket is called a *tanker*. If duplicate fingerprints are found they are not added to HBM, otherwise added. Upon completion of one tanker HBM size is increased by adding a new tanker. If the amount of fingerprints in each tanker is same approximately with high probability then the probability of consecutive fingerprints in the same tanker being stored is also high, preserving fingerprint locality which accelerates the Deduplication process.

Using Bloom Filter Array as Quick Index:
Fast index which identifies unique content is achieved by Bloom Filter (BF) [8], a probabilistic data structure which recognizes whether duplicate data is present or not in a tanker. Usage of one BF causes problems like rebuilt of BF as false positive [10] rate grow vastly as the BF capacity becomes less than the number of fingerprints and if an item in BF is removed. Location of duplicates is also ineffective. Counting filter solves the above problems [9], but it increases RAM size and degrades the performance. So, Bloom Filter Array (BFA) with same hash function by all bloom filters is used in each tanker to note the membership information of fingerprints members. If a positive is returned then there is a duplicate in the tanker which is identified by prefix of fingerprint. If negative is returned then a unique fingerprint. Bloom Filter has to be increased with the increase of data.

Dual Cache Mechanism:
If a unique fingerprint arrives it is appended to an appendable tanker which maintains reference count for duplicates. Existing Unique fingerprints are maintained by Reference-only state tankers. There are two caches direct –mapped cache(DMC) which maps the appendable tankers in the bucket and set-associative cache which maps the reference-only state tankers to the bottom the respective bucket by which the hit rate is maintained for duplicate fingerprints to achieve fingerprint locality. In DMC all tankers will be changed to reference only states if current imbalance is greater than assigned threshold and are moved to the on-disk HBM and this is periodic rebalancing policy which is used in MD2. MD2 uses LRU replacement policy inside each bucket set to minimize the cost of disk access resulted by false positive of BFA and it uses batch write back policy where all buckets which are cached are written to disk when time to replace the dirty buckets [7] (change of reference counts by insertion and deletion operations logically) to achieve fingerprint locality and disk access locality, thus the SAC access efficiency is improved.

DHT-based Load Balancing:
Data is partitioned into dissimilar groups by using Distributed Hash Table (DHT) [11] a dynamic load balancing using SHA-1 hash algorithm and distributes the load among multiple storage components (SC) and are responsible for Deduplication. The first three techniques addresses the challenge *i)* and the last technique addresses the *ii)* challenge mentioned above.

C. Cloud Spider:

C.Policroniades et al. [18] have proposed this technique where Replication and Scheduling methods are integrated to reduce the latency associated in Live VM Migration over WAN. Deduplication methods are also used along with these methods to deduplicate VM images as it occupies more disk space and need more network band width [6]. This technique work as follows:

(1) Multiple replicas of VM images are created.

(2) The cloud sites with less average cost of computation are identified, and are called as *eligible sites* [6].

(3) Replicas of VM images are created in all eligible sites.

(4) One of the VM images are considered as primary copy and changes in the primary copy are transferred to the replicas using

(5) Incremental back up.

(6) Here Deduplication technique called Content-Based Redundancy is used while transferring the changes to eliminate redundant data.

a) Content Based Redundancy:

The file is divided into variable size blocks using Rabin finger prints [14]. Rabin Finger Print is a rolling hash function which uses 48-byte sliding window. Thus blocks are created based on content of the window not by fixed size. Cryptographic hash function is used to create the check sum of each block. If two blocks check sum is same then the two blocks are identical and they are eliminated.

In Cloud Spider Design there are three components. They are:

(1) i) and ii) steps performed by *Enterprise Cloud Manager* (ECM).

(2) is done by *Site Manager (SM)*

(3) Allocation of necessary computing resources is ensured by Application Manager (AM) to maintain SLA. By this method 80% of latency in migration is reduced [6].

D. Optimization of Deduplication Technique:

In this technique three stages are present in Deduplication i) chunking ii) fingerprint generation iii) detection of redundancy

Chunking:

- This process includes chunking module uses *context aware chunking* where portioning of file into fixed size or variable size chunks is performed based on type of file i.e., for multimedia files fixed size chunking and for text files variable size chunking is done. Variable size chunking is done by using sliding window.

Fingerprint Generation:

- Fingerprint for each is generated by using Rabin's algorithm [26] by *Fingerprint Generator* and again *incremental Modulo-K* is used to get efficient fingerprint and it is handled by *fingerprint manager*. It is responsible for redundancy detection. Collection of fingerprints called *tablet.* An array of pointers to tablets is managed by fingerprint manager. Fingerprint tables are maintained by both client and server where redundancy elimination and location of data chunk is performed respectively.

Detection of Redundancy:

In this LRU Based Index Partitioning is used which in turn consists of Filter-based fingerprint lookup i.e., Bloom filter is used to check whether fingerprint is present in fingerprint table or not. Searching time for the fingerprint in fingerprint table based on its size. So, it incorporates table based index partitioning where the physical distance between the fingerprints in a tablet is shorter [25]. Tablet management is done based on LRU [25].

By this technique by the increase of chunk size from 4KB to 10 KB, 34.3 % chunking time increases, 0.66% Deduplication ratio decreases, and overall back up increases from 51.4 MB/sec to 77.8 MB/sec i.e., by 50%

E. Multi-level Selective Deduplication :

B.Zhu et al. [27] have proposed that, to increase the reliability in virtualized environment backing up of VM images, but the cost is high because of their huge storage. By using Backup service with full Deduplication [27] redundant data is eliminated but increases cost and compete computing resources. VM data duplication is done in two phases *i) inner-VM-* Most of the data is duplicated between VM's snapshots while backing up. *ii) Cross-VM-* Because of software and libraries like Linux and MySQL, back up of huge amount of high similar data different VMs. So, Deduplication should be at these levels.

VM image is partitioned into segments and every segment in turn contains many blocks which are formed by using variable size chunking algorithm [28]. Block hashes and data pointers are recorded by segment Meta data. Segment Modification is recorded by dirty bit in virtual disk drive. Thus reuse of metadata and filtering of unmodified data is done in *level 1*. In *level 2* is Block fingerprint comparison [29] is done i.e., if segment is modified it is compared with parent snapshot and duplicates are removed. The length of segment is restricted to page boundary of each virtual image file. As fingerprints of modified segments should be loaded, it requires little amount of space. In *level 3* Deduplication, duplicate data blocks are identified using common data set (CDS) [29] among multiple VMs. Pointers to block fingerprint and location of real content present in content block store which helps to get the original data.

F. SAFE: Structure-Aware File and Email Deduplication:

Daehee et al.[31] have proposed that this approach deduplicates MS docx, pptx, pdf, emails, structured files redundant objects. It performs file level Deduplication and object level Deduplication. File level Deduplication [31] eliminate parsing of duplicate files by file parser. File parser converts all structured as objects and these are managed by object manager and store manager. Emails are passed to email parser, perform chunking, finds hashes and Deduplication. Object level Deduplication [31] checks the existence of objects by using object index table and store manager stores unique objects into storage.

G. On-Protocol-Independent Data Redundancy Elimination:

This technique is used in wire line, wireless and cellular networks [12] and contains the following steps:

(1) Fingerprints are calculated for every incoming data packets by using hash function (Fingerprinting)

(2) Subset of fingerprints is finding and is called representative finger prints. Fingerprint table stores representative fingerprints and the pointers to the locations of

respective chunks in packet cache. (Indexing and Lookup & Storing)

(3) If for any representative fingerprint in fingerprint table checked for a match in a packet cache

(4) If a match found the data without redundancy is encoded with metadata like fingerprint, matched data chunk description, number of duplicate bytes at the beginning and ending of the data packet (Data Encoding) and compressed [13].

(5) The receiver receives the compressed packet decompresses it and the original data is reconstructed with the available metadata by using the fingerprint table and packet store. It performs just reverse operations of data encoding called data decoding [12].

H. Liquid:

K.Jin et al. [21] have proposed that,this is the latest technique for Deduplication of VM Image files. In VM Migration storage of VM images is on NAS which is a shared network storage is an issue. This issue is resolved by Deduplication techniques [22]. Liquid [23] is a distributed File System which deals with scalability and storage issues of VM images. In Liquid Fixed Size chunking is used for VM images as Deduplication Ratio is better [21] and is performed in client side. Block Size is in multiples of 4KB from 256 KB to 1MB to have better IO performance and Deduplication Ratio. Choosing too small size or large size of blocks didn't give better Deduplication ratio. Fingerprint calculation is performed for modified blocks of file using MD5 or SHA-1which determines the duplicates. Fingerprint calculation is expensive hence it maintains two caches one is *shared cache* where read only blocks are contained and is replaced by LRU [24] when it is full by which reading performance of VM images is improved. The Second one is *private cache* where it contains only modified block and is present in individual VM. If modified block is present in shared cache then it is removed and is added to private cache and it finds private fingerprint which is a global unique number. When the private cache is full or hypervisor executes POSIX flush () then the modified block is replaced by LRU policy. Fingerprint is calculated for modified blocks by multiple threads for fast calculation. If two fingerprints are same then the two blocks are identical and redundant data is removed. Storing of Deduplicated blocks in data servers and fingerprints in a Meta server is done. Three files are used to store the deduplicated blocks information i) *extent file* which contains all data blocks ii) *index file* which maps fingerprints to corresponding data blocks iii) *bit map file* which indicates the slot in extent file is valid. To access a VM image the client downloads finger prints from Meta server and data blocks from data servers and integrated VM is exported to hypervisors.

IV. COMPARISON

S.No	Name Of The Deduplication Techniques	Year	Chunking Algorithm Used	Techniques Used And Proposed	Issues Concentrated	Achievements	Application Scenario
1.	*Extreme Binning*	2009	Virtual Size	Rabin Fingerprints	-Storage -Scalability -Parallel Deduplication	-Only 35.82 GB for chunk index,167 GB of RAM -Maximum parallization by one file-one backup node -Less Deduplication loss	Backup Systems
2.	*MAD 2*	2010	Virtual Size	Rabin Fingerprints Bloom Filter Array	-Duplicate look up disk -Storage node island effect -Load Balancing Storage	-High Throughput -Load Balancing among Multiple Storage nodes using DHT. -Better than Extreme Binning -10GB RAM	Network Backup Services
3.	*Cloud Spider*	2011	Variable Size	Rabin Fingerprints	-VM Migration - Storage	- 80% of latency reduced in VM Migration -Less storage when compared to RandomMin and RandomMax Strategies	VM Migration
4.	*Optimization of Deduplication Technique*	2011	Fixed Size and Variable Size based on type of file.	LRU-index Partitioning & Incremental Modulo-K (INC-K)	-Backup Speed -Deduplication Ratio - Chunking -Fingerprint Lookup	-Backup Speed -Deduplication ratio Chunking -Fingerprint lookup	Backup Operation
5.	*Multi-level Selective Deduplication*	2012	Variable size	Multilevel	-Storage -Reliability	-70% of Global Deduplication	Back up of virtual disks in cloud computing

				duplication	-Parallelism of Deduplication	-Reduces two-third of storage cost	
6.	*SAFE*	2013	Fixed Size	Structure-Aware File and Email Deduplication	-Storage -Network Bandwidth	- 10-40% Storage Savings -Reduces 20% data traffic	Structured files and Emails
7.	*On-Protocol Independent Data Redundancy Elimination*	2014	Fixed Size	Fingerprint	- Bandwidth Utilization	DRE techniques are effective in application Scenario	Wire line, wireless, Cellular Networks
8.	*Liquid*	2014	Fixed Size	Fingerprints, Private Fingerprints	-Scalability -Storage -Parallel Deduplication -Network Bandwidth	-High IO Performance -Little network bandwidth consumption by using Bloom Filter.	VM images in cloud environment

V. FINDINGS

From the comparison in section IV based on application scenario the Deduplication technique is chosen. For Backup services Multilevel Selective Deduplication allow 70% Deduplication and two third reduction in storage. As per backup speed constraint, optimization of Deduplication technique increase speed by 50% as it uses LRU index partitioning and incremental modulo-K methods for detection of duplication. For VM Migration scenario Cloud spider reduces latency by 80% which helps in

reducing migration time. DRE techniques are
suitable for wire-line, wireless and cellular networks. Liquid is better for backup of VM images. For files and Emails SAFE is used. Parallelization of Deduplication is performed in some techniques. This can be implemented for other techniques as future work.

VI. CONCLUSION

Deduplication is very important in live virtual machine migration to transfer of VMs with the available bandwidth and less migration time. It is also important for back up services, wire, wireless, cellular networks etc., to reduce the amount of data in storage and to speed up the backup process. All Techniques have steps like use of chunking algorithm either fixed or variable size based on application scenario, fingerprint generation for every chunk by using MD5 or SHA1, detection of the duplicates by using fingerprint table lookup and elimination of duplicates. Original data is reconstructed by using the fingerprint and data pointers that are stored in fingerprint table. Parallelization of Deduplication process by using multiple threads or by multiple nodes is discussed in some of the techniques to speed up the Deduplication process. A Detailed survey on Deduplication techniques is presented in this paper. From section IV and V it is concluded that based on application scenario and constraints the best Deduplication technique is chosen.

REFERENCES

[1] T.Y.J.Naga Malleswari, D.Rajeswari, Dr.V. Jawahar Senthil Kumar, "*A Survey of Cloud Computing, Architecture & Services Provided by Various Cloud Service Providers*", in proceedings of *Interenational Conference on Demand Computing*, 978-93-5087-502-5 201, Bangalore, 2012.

[2] http://www.techopedia.com/definition/16821/virtual-machine-snapshot-vm-snapshot.

[3] K. Parimala1 G. Rajkumar, A. Ruba, S. Vijayalakshmi, "Challenges and Opportunities with Big Data", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.5, pp.16-20, 2017

[4] EMC2, "*Information Storage and Management*", in *Wiley India Edition*, 2nd ed.USA, 2012, pp.249-251.

[5] Qinlu He, Zhanhuai Li, Xiao Zhang, "*Data Deduplication Techniques*", in International Conference on Future Information *Technology and Management Engineering*, China, 2010, pp.43-433.

[6] Priyanka Sethi, Prakash Kumar, "*Leveraging Hadoop Framework to develop Duplication Detector and analysis using Map Reduce, Hive and Pig*", in IEEE 978-1-4799- 5173-4/14, 2014.

[7] Sumit Kumar Bose, Scott Brock, Ronald Skeoch, Nisaruddin Shaikh, Shrisha Rao, "*Optimizing Live Migration of Virtual Machines Across Wide Area Networks Using Integrated Replication and Scheduling*" in IEEE 978-1-4244-9493-4/11, 2011.

[8] Jiansheng Wei, HongJiang,Ke Zhou, Dan Feng, "*MAD2: A Scalable High- Throughput Exact Deduplication Approach for Network Backup Service*", in IEEE, 978-1-4244- 7153-9/10, China, 2010.

[9] B.H.Bloom, "*Space/Time trade-offs in hash coding with allowable errors*", *Communications of the ACM*, vol.13, no.7, p.422-426, July 1970.

[10] Z.Broder andM.Mitzenmacher, "*Network Applications of Bloom Filters: A Survey*", *Internet Mathematics*, vol.1, pp.485-509,2005.

[11] Guohua Wang, Yuelong Zhao, Xiaoling Xie, Lin Liu, "*Research on a clustering data de-duplication mechanism based on Bloom Filter*, in IEEE, 978-1-4244 7874-3/10, 2010.

[12] Marcin Bienkowski, Miroslaw Korzeniowski , Friedhelm Meyer auf der Heide, "*Dynamic Load Balancing in Distributed Hash Tables*", International Graduate School of dynamic intelligent systems, Germany, [Online]

[13] Yan Zhang, "*on Protocol-Independent Data Redundancy Elimination*" , IEEE Communicaitons Surveys &Tutorials, vol 16, No.1, First Quarter 2014.

[14] M.Al-laham, and I.M.M.E.Emary, " *Comparitive Study between various algorithms of data compression techniques*", *International Journal Computer Science and Network Security*, vol. 7, no.4, April 2007.

[15] M.O.Rabin, "*Fingerprinting by random polynomials*", Center for Research in Computing Technology, Harvard University, Tech.Rep. TR-15-81, 1981.

[16] Deepavali Bhagwat, Kave Eshghi, Darrell D.E.Long, Mark Lillibridge, "*Extreme Binning: Scalable, Parallel Deduplication for Chunk- Based File Backup*" in IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09.

[17] R.Rivest, "*The MD5, message-digest algorithm*", IETF, Request For Comments (RFC) 1321, Apr. 1992, [Online]

[18] National Institute of Standards and Technology, "*Secure hash Standard*,", FIPS 180-1, Apr. 1995. [Online]
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[19] C.Policroniades and I.Pratt, "*Alternatives for detecting redundancy in storage systems data*", in Proceedings of the General Track: 2004 USENIX Annual Technical Conference, 2004, pp. 73-86. Deduplication Techniques: A Technical Survey (IJIRST/ Volume 1 / Issue 7 / 062) All rights reserved by www.ijirst.org 325

[20] M.Dutch, "*Understanding data Deduplication ratios*", *SNIA Data Management Forum*, June 2008.

[21] A.Z.Broder, "*On the resemblance and containment of documents*", in *SEQUENCES'97: Proceedings of the Compression and Complexity of Sequences* 1997, pp. 21-29.

[22] K.Jin and E.L.Miller, "*TheEffectiveness of Deduplication on Virtual Machine Disk Images*", in *Proc. SYSTOR,Israeli Exp. Syst. Conf.*, New York, NY, USA, 2009, pp.1-12.

[23] A.Liguori and E.Hensbergen, "*Experiences with Content Addressable Storage and Virtual Disks*," *in Proc. WIOV08*, SanDiego, CA, USA, 2008, p.5.

[24] XunZhao, Yang Zhang, Yongwei Wu, Kang Chen, Jinlei Jiang, Keqin Li, " *Liquid: A Scalable Deduplication File System for Virtual Machine Images*", *IEEE Transactions on Parallel and Distributed Systems*, vol.25, No.5, May 2014.

[25] A.V.Aho, P.J. Denning, and J.D. Ullman , "*Principles of Optimal Page Replacement*", *J.A.C.M*, vol.18, no.1, pp. 80-93, Jan. 1971.

[26] Jaehong Min, Daeyoung Yoon, and Youjip Won, "*Efficient Deduplication Techniques for Modern Backup Operation*", IEEE Transactions on Computers, vol.60. June 2011.

[27] A.Muthitachareon, B.Chen, D.Mazieres, " *A Low bandwidth Network File System*", *SIGOPS Operating Systems Rev*., vol.35, no.5, pp.174-187,2001.

[28] B.Zhu, K.Li, H.Patterson, "*Avoiding the disk bottleneck in the data domain Deduplication file system*" in *FAST'08: Proceedings of the 6th USENIX Conference on File and Storage Technologies*, pp. 1-14, Berkely, CA, USA, 2008.

[29] U.Manber. *Finding similar files in a large file system,* in proceedings of the USENIX Winter 1994 Technical Conference, pp. 1-10, 1994.

