Cross User Bigdata Deduplication

Yash Karanje

Ankita Jadhav

Nikhita Biradar

Ketaki Kadam

Prof. M. P. Navale

Department of Computer Engineering

NBN Sinhgad School of Engineering, Pune, India

***Abstract*– Today's world has been digitalized to a large extent. The total amount of data generated per day is more than 2.5 exabytes out of which social media fuels up with maximum contribution along with business transactional data, sensor-generated data. Such a huge amount of data must be managed properly to use it for certain business domain-specific decision-taking purposes. It is very confronting to store and manage such huge amounts of data which is mostly redundant in nature and that too present over multiple cloud platforms for multiple users; it requires high resources including the cost required to store, backup time, processing time; which results into reduction of system throughput. So, Data Deduplication is the most preferable way that we propose here considering the above issue. We propose a model that will perform deduplication of data for multiple users to achieve the uniqueness of textual data (only) uploaded by multiple users; data access must be efficient though, maintain the privacy of data against brute-force attacks. This intension will be achieved by employing certain algorithms like a Fixed-size blocking algorithm & Encryption algorithm and effective data organization. It will not only preserve the space by means of reducing storage allocation but also effectively manage network bandwidth.**

Keywords – Hashkey, Secure hashing algorithm, Brute-force attack, Inter-User deduplication, Intra-User deduplication, Fixed size blocking algorithm, Cryptokeys.

1. INTRODUCTION

A tremendous amount of data is getting produced from different resources per unit time; its proper handling, processing & storage must be done effectively. Here, we're concerned with the data privacy, size of that data and redundancy is the main cause behind that. This results in improper knowledge discovery, inefficient decision making & data may get exposed to brute-force attacks; redundancy also leads to inefficient storage space utilization. To deal with the above issues, we use data deduplication.

We propose a model that will perform deduplication of data for multiple users to achieve the uniqueness of textual data (only) uploaded by multiple users; data access must be efficient though, maintain the privacy of data against brute-force attacks. This purpose will be achieved through a certain hashing algorithm and effective data organization. This application mainly focuses on efficient data access, preserve the privacy of textual data owned by multiple users by avoiding redundancy of textual data.

In the beginning, we have a text input file and this file assumed to go under the Blocking/Chunking algorithm to produce blocks of data. Then every block will go through an Advanced Encryption standard (AES) to generate cryptokeys. To make efficient use of storage space, clusters will be created to store those generated hash keys so that the search operation becomes relatively more simple and optimal to carry out. Subsequent checking will take place in two phases specifically Intra User and Inter User deduplication checking.

Consider an email system where individual attachment (text document) size of 5kb is supposed to be sent to 20 various users. If sent individually, we will have 21 copies in total resulting in 105kb of data. As far as the document is assumed to be in read-only format, it is always a better option to make it a shared entity among all users so that we will have only a single copy of it sized 5kb only accessible for all users. This will preserve storage space because we have avoided data redundancy by maintaining a single and unique copy of it.

2. LITERATURE SURVEY

At this time and age, because of the exponential increase in the use of arising technology like big data and cloud computing, the speed of data expansion is also increasing quickly, so data deduplication technique [12] has been commonly utilized in cloud storage because it can considerably diminish storage costs by keeping unique copy of duplicate data**.**

In the existing system, [12] initially private keys are generated for all the users available and it also generates related system parameters. The data uploading phase consists of 4 sections; those are generation of tags, intra-deduplication, inter-deduplication and encryption of data and key recovery. In tag generation, the User desires to upload data that chooses a intra-tag randomly together with the private key. In the intra-deduplication phase has only verified the duplicate within the outsourced data from the identical domain. Within the inter-deduplication phase, checks the duplicate from the root node by matching and comparing the length of data. In the end, data encryption is performed so that confidential information can only accessed and decrypted by that specific user having the accurate and expected encryption key.

Distributed Deduplication for Big Data Storage in the Cloud [9] by Shengmei Luo, Guangyan Zhang, Chengwen Wu. They have used techniques such as Hadoop Distributed File System, Map reduce and Fixed-size blocking algorithm. It uses effective data routing algorithm which relies on the concept of data similarity, Therefore reduces the network overhead to recognize required storage location. Their system uses several storage data nodes for the sake of parallel deduplication.

A Study on Data Deduplication Techniques for Optimized Storage [3] by E. Manogar, S. Abirami used Chunk based deduplication, Deduplication based on Location and Deduplication based on Time. They have particularly examined the above three techniques of data deduplication. Among them, it has been declared that variable size data deduplication is better in comparison with other techniques by checking the hash of every and each chunk.

A Distributed Solution of Data Deduplication [13] by Yongwei Wu, Guangwen Yang and Yang Zhang used Deduplication for virtual machines. It accomplishes maximum deduplication ouput by storing all fingerprint index within RAM, completely avoid access to any disk when doing fingerprint lookup.

Hadoop Based Scalable Cluster Deduplication for Big Data [8] by Qing Liu, Guiqiang Ni, Yinjin Fu Used techniques such as Map reduce, Fixed size blocking algorithm and Hadoop Distributed File System. Map reduce technique is used for parallel deduplication framework. The index table is distributed among the nodes which are maintained in one of the lightweight local databases that is MySQL.

Improved Deduplication through Parallel Binning [14] by Zhang, Z., Schwarz, Litwin W. Long, and Bhagwat, D. used Extreame Binning and Parallel Binning techniques. Extreme binning uses file similarity. They compared the logical results of Extreme Binning were we access a greater number of bins. Their findings show the advantages of those extensions when a new file is a moderate alteration of pre-existing file.

Bucket Based Data Deduplication Technique for Big Data Storage System [7] by Naresh Kumar, S. C. Jain, and Rahul Rawat. They have used Fixed Size blocking Algorithm and Bucket based technique. A Fixed Size blocking Algorithm is used to divide an input text file into number of blocks. Every block will goes under Hashing algorithm to generate unique hash values. These hash values of each block are stored in buckets. Map reduce technique compares the hash value of incoming hash of block with hashes of blocks those are previously stored in buckets.

Authorized Data Deduplication Using Hybrid Cloud Technique [6] by Mane VIdya Maruti, Mininath K.Nighot used Content level Deduplication, file-level Deduplication. Duplication is checked in an authenticated way. The very basic requirement for checking file duplication is proof of ownership. The user needs to submit the input text file and proof of ownership of the file before sending the request for analysing duplicate to the cloud.

Sr. No.	Title	Authors	Technique Used	Methodology
1	Distributed Deduplication for Big Data Storage in the Cloud	Shengmei Luo, Chengwen Wu, Guangyan Zhang.	Block level chunking(Super chunk) data routing algorithm based on similarity index.	It uses effective data routing algorithm which relies on the concept of data similarity, Therefore reduces the network overhead to recognize required storage location. Their system uses several storage data nodes for the sake of parallel deduplication
2	A Study on Data Deduplication Techniques for Optimized Storage	E. Manogar, S. Abirami	Chunk based deduplication, Deduplication based on Location and Deduplication based on Time	They have particularly examined the above three techniques of data deduplication. Among them, it has been declared that variable size data deduplication is better in comparison with other techniques by checking the hash of every and each chunk.
3	A Distributed Solution of Data Deduplication	Yang Zhang, Guangwen Yang and Yongwei Wu	Deduplication for virtual machines	It accomplishes high deduplication ouput by storing all fingerprint index within RAM, completely avoid access to any disk when doing fingerprint lookup.
4	Improved Deduplication through Parallel Binning	Zhang, Z., Schwarz, Bhagwat, D., Litwin, W., and Long, D.	Parallel Binning, Extreame Binning	Extreme binning uses file similarity. They compared the logical results of Extreme Binning were we access a greater number of bins. Their findings show the advantages of those extensions when a new file is a moderate alteration of pre-existing file.
5	Authorized Data Deduplication Using Hybrid Cloud Technique	Mininath K. Nighot, Mane VIdya Maruti,	Content level Deduplication, file level Deduplication	Duplication is checked in an authenticated way. The very basic requirement for checking file duplication is proof of ownership. The user needs to submit the input text file and proof of ownership of the file before sending the request for analysing duplicate to the cloud.

3. MATHEMATICAL MODEL

Sr. No.	Description	Observation
1.	Problem Description and System	
	Let S be Closed system defined as, S = { Ip, Op, A, U, DB} Store Ip from U and perform various actions from the set of actions A so that uniqueness of data can be maintained.	System

	U is set of users. DB is set of clusters Ip is input Text File A is set of Actions Op is output Ack.	
	Output set : Op1 = { D | D is set of data blocks } Op2 = { H | H is a set of Crypto keys for data blocks } Op31 = { T | T is an ack if available in local space } Op32 = { F | F is an ack if NOT available in local space } Op = { Y | Y is an ack about availability in global space } Set of actions = A={F1,F2,F3,F4} Where, Function 1 = F1 = Blocking Function 2 = F2 = Hashing Function 3 = F3 = Local cluster availability Function 4 = F4 = Global cluster availability	Intermediate outputs obtained to reach final state System goes through a set of different states

2.	**Mapping Functions f(x)->y**	**X (Input)**	**Y (Output)**
	F1(Ip) → d Where, d is a data block & d € D	Ip	d
	F2(d) → H	d	H
	F3(H) → T	H	T
	F3(H) → F	H	F
	F4(T) → Y	T	Y
	F4(F) → Y	F	Y

4. METHODOLOGY

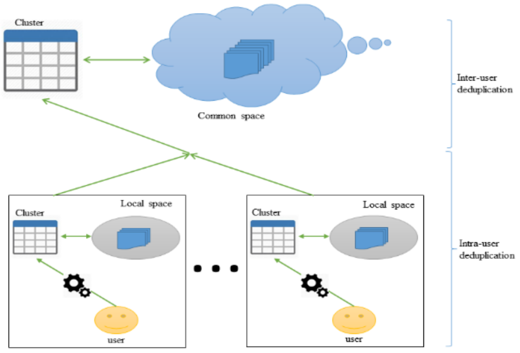
- In this proposed system, deduplication is employed for efficient use of storage space and it is a better option to handle the duplicate data. There are 2 phases of deduplication checking; first is intra-user deduplication and second one inter-user deduplication.

- Multiple users upload text files in their respective local space. For each file uploaded, the blocking algorithm will be applied to divide the file into multiple blocks. Every block of data i.e. object will go under the Advanced Encryption standard (AES) and generate the cryptokey. The generated cryptokey will be unique for that block of data.

- If intra-user checking fails to locate the record, then only that particular block will be stored in that local space & hence inter-user checking will take place. Clusters will be maintained on both local & global side to store cryptokeys & the same will be used while deduplication checking.

- Clusters are created to store cryptokeys and every cluster will maintain certain unique & independent characteristics shown by cryptokeys. Those clusters will be the persistent storage systems. The test crypto key will be given to every individual function representing those clusters and will be compared in that cluster only whose function it satisfies. If no match found then only that record will be added to the current cluster and hence the data. Functions representing every cluster must be unique and independent.

  That's how two level deduplication checking will take place.



Architecture diagram

Cross user Bigdata Deduplication system implemented in following steps:

4.1 Blocking :

Basically, there are 2 varieties of data deduplication techniques, those are file-level deduplication and block-level deduplication. In our system, we have performed block-level deduplication more specifically variable block-level deduplication. Initially, when the user wishes to upload a text file, that input text file is divided into ‘n’ several blocks having a variable size. Blocking of the input text file is performed based on punctuation marks present in this file. We considered punctuation marks like a full stop, semicolon, exclamation mark, question mark, etc. as criteria to separate out a sentence that result into variable-sized blocks. Now each of the blocks is given to Encryption algorithm to generate cryptokey that is unique.

4.2 Keystore:

The Java KeyStore class functions as a key database where private keys, public keys, certificates, and secret keys can be stored. The keys to be stored in the keystore and the keystore itself can be encrypted with the same or different key. In our application, we're maintaining the same key to be used for encryption of keys as well as the keystore. For a specific user, the user id is used to access the keystore and the user id itself is used to encrypt and decrypt the secret keys stored in the keystore. By using the user id, the keys will not be accessible unless the user is logged in. Thus we can use these keys as and when required for uploading and downloading purposes.

4.3 Encryption :

We have used symmetric encryption to generate cipher text for each of the block. Symmetric encryption uses one key for encryption as well as for decryption of confidential information. Samples of Symmetric Encryption Algorithm are Data Encryption Standard (DES), Advanced Encryption Standard (AES), Rivest cipher 4 (RC4).

We have used Advanced Encryption Standard (AES) for encrypting each of the block. A number of the feature of AES are mentioned below:

- AES type of symmetric block cipher
- Block size of 128 bits

- Keys may be size of 128 bits or 192 bits or 256 bits

AES consists of the following four steps:

AddRoundKey : The AddRoundkey operation is that the only operation that’s performed directly on AES round key. In the AES algorithm, the key’s successively expanded into ten keys under the operation called key round schedule. The round key is nothing but the output of every round of key schedule.

SubBytes : In SubBytes section, each block is divided into bytes. Each of the bytes is substituted with the help of a fixed table that is substitution box (s-box). At the end of this phase, we got the matrix containing rows and columns.

ShiftRows : Within the ShiftRows section of AES, each of the rows is shifted. The first row remains unchanged and each byte of the second row shifted by one byte and so on.

MixColumns : MixColumns phase is somewhat similar to the ShiftRows section of AES algorithm. MixColumns section performs operation of dividing the matrix using columns rather than of rows.

In this way security is achieved by converting each of the blocks into cipher text.

4.4 File record:

A file record is a directory of all the available files in the database. It also maintains the sequence of the blocks in every particular available file. Whenever we perform operations like uploading and downloading every time we check if the file is available in file record or not. According to the status of file availability, further operations will take place.

4.5 Checking for duplicate files:

The file uploaded by the user goes through blocking followed by encryption then clustering phase. That ciphertext is compared with the previously stored one. It will not upload if matched with previously stored cyphertext.

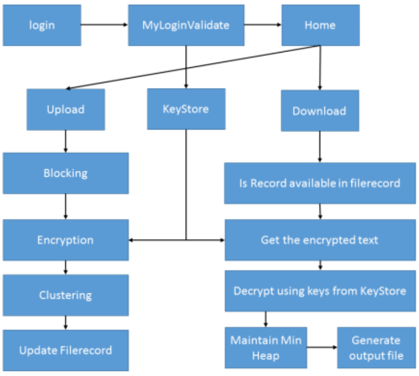
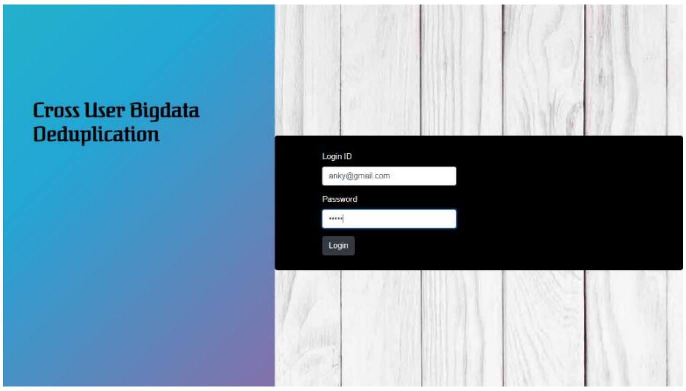


Fig. Flow diagram

5. RESULT


Cross User Bigdata Deduplication

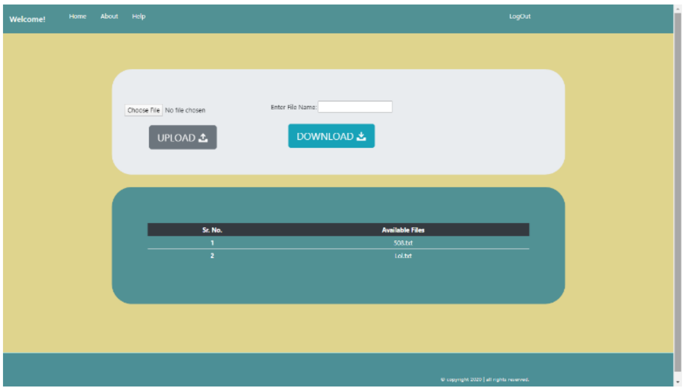
Login ID

anky@gmail.com

Password

•••••

Fig. Login page



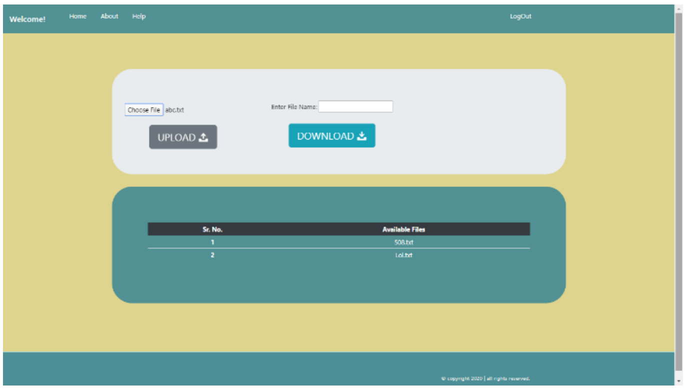
No file chosen

Enter File Name:

Sr. No.	Available Files
1	508.txt
2	Lol.txt

© copyright 2020 | all rights reserved.

Fig. Home page



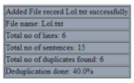
Choose File abc.txt

Enter File Name:

Sr. No.	Available Files
1	508.txt
2	Lol.txt

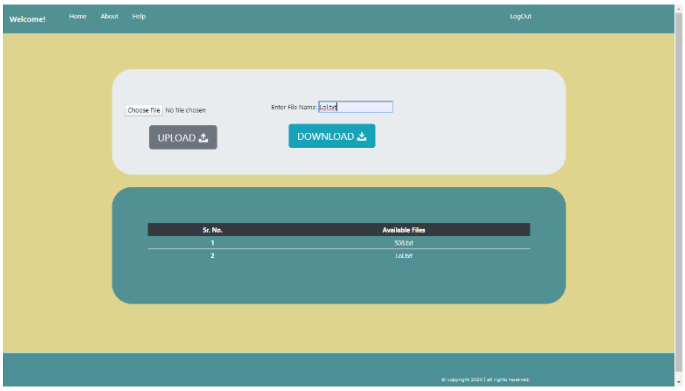
© copyright 2020 | all rights reserved.

Fig. Upload



Added File record Lol.txt successfully
File name: Lol.txt
Total no of lines: 6
Total no of sentences: 15
Total no of duplicates found: 6
Deduplication done: 40.0%

Fig. Upload Success



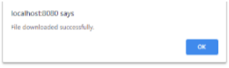
Choose File No file chosen

Enter File Name: Lol.txt

Sr. No.	Available Files
1	508.txt
2	Lol.txt

© copyright 2020 | all rights reserved.

Fig. Download



localhost:8080 says
File downloaded successfully.

Fig. Download Success

6. CONCLUSION

Here, we’ve proposed a cross user bigdata deduplication system to maintain the uniqueness of textual data generated by multiple sources & owned by multiple users. Since the amount of data getting generated is tremendous; hence it becomes challenging to handle & process such huge data for knowledge discovery purposes. This system mainly focuses on redundancy issues; moreover, it also achieves efficient data organization & access, preserves privacy. For that, we’ve made use of the Fixed-size blocking algorithm and Hashing algorithm to generate cryptokeys. This is how the paper summarises existing technologies & proposed methodologies for bigdata deduplication.

7. REFERENCES

[1] C. Liu, Y. Lu, C. Shi, et al., ”ADMAD: Application-driven metadata aware deduplication archival storage System”, in Proc. 5th IEEE Int. Workshop Storage Netw. Archit. Parallel I/Os, 2008, pp. 29–35.7.

[2] Deepavali Bhagwat, Kave Eshghi,Darrell D. E. Long, ”Extreme Binning: Scalable, Parallel Deduplication for Chunkbased File Backup”,in Proc. IEEE Int. Symp. Modell. Anal. Simulation Comput. Telecommun. Syst., 2009, pp. 1–9.

[3] E. Manogar and S. Abirami,”A Study on Data Deduplication Techniques for Optimized Storage”,2014 Sixth International Conference on Advanced Computing(lCoAC), IEEE 2014, pp. 161-166.

[4] Hyungjune Shin, Dongyoung Koo†, Youngjoo Shin, and Junbeom Hur, “Privacy-preserving and Updatable Block-level Data Deduplication in Cloud Storage Services”, IEEE 11th International Conference on Cloud Computing, 2018.

[5] M. Miao, J. Wang, H. Li, and X. Chen, “Secure multi-server-aided data deduplication in cloud computing,” Pervasive and Mobile Computing, vol. 24, pp. 129–137, 2015

[6] Mane VIdya Maruti, Mininath K.Nighot, “Authorized Data Deduplication Using Hybrid Cloud Technique”, 2015 International Conference on

Energy Systems and Applications (ICESA 2015), 2015

[7] Naresh Kumar, R. Rawat, and S. C. Jain, ”Bucket Based Data Deduplication Technique for Big Data Storage System”,5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, 2016, pp. 267-271

[8] Q. Liu, Y. Fu, G. Ni, R. Hou, ”Hadoop Based Scalable Cluster Deduplication for Big Data”, IEEE 36th International Conference on Distributed Computing Systems Workshops, 2016.

[9] Shengmei Luo, Guangyan Zhang, Chengwen Wu,” Boafft: Distributed Deduplication for Big Data Storage in the Cloud”,IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 4, NO. X, XXXXX 2016.

[10] Supriya Milind More, Kailas Devadkar, “A Comparative Survey on Big Data Deduplication Techniques for Efficient Storage System”, IJIACS, ISSN 2347-8616, Volume 7, Issue 3, 2018.

[11] Xingyu Zhang, Jian Zhang, “Data Deduplication Cluster Based on Similarity-Locality Approach”, 2013.

[12] Xue Yang, Rongxing Lu, Ali A. Ghorbani, Jun Shao, Xiaohu Tang, “Achieving Efficient and Privacy-Preserving Multi-Domain Big Data Deduplication in Cloud”, 8.2881147, IEEE Transactions on Services Computing, 2018.

[13] Yang Zhang, Yongwei Wu and Guangwen Yang, “Droplet: a Distributed Solution of Data Deduplication,” 2012 ACM/IEEE 13th International Conference on Grid Computing, Beijing, 2012, pp. 114-121.

[14] Zhang, Z., Bhagwat, D., Litwin, W., Long, D. and Schwarz, S.T. (2012) Improved Deduplication through Parallel Binning. 2012 IEEE 31st Int. Performance Computing and Communications Conf. (IPCCC), pp. 130–141. IEEE.