A Heuristic method to Enhance data deduplication process in cloud Group Members

Upendra A. Joshi[1],AshishU. Kulkarni[2], Yuvraj V. Mogal[3], Prof.B.L. Dhote[4]

[1,2,3,4]Department of Computer Engineering, Sinhgad Institute of Technology, Pune, India

Abstract—*The use of cloud storage is increasing rapidly over a decade and hence maintaining efficiency of these cloud storage becomes extremely important issue. Main issue that hinders the cloud efficiency is the data redundancy. Hence eliminating data redundancy is very necessary. There are many methodologies which could tackle the issue but most effective technique is to deploy an effective deduplication scheme over the cloud data which could eliminate duplication amongst the files stored on the cloud, hence only unique data can be stored on cloud thereby improving its space complexity.There are many methodologies and theories which highlight on use of deduplication in cloud like- Data Deduplication over unencrypted data, Application aware data deduplcation schemes, etc. Most of these systems for deduplication have some performance issues that can lead to lower accuracy of the technique.*

This paper proposes a novel deduplication scheme over the data in which every unique file on the cloud will generate a unique hash key which will be maintained by the mechanism called bloom filter. The deduplication will be done on basis of the hash key generated and later the data will be encrypted and stored on the cloud.

Keywords—*deduplication, Hash key generation, cloud storage, RCC encryption, Bloom filter.*

I. INTRODUCTION

The use of mobile computing devices like laptops, cell phones, tablets, etc. is rapidly increasing main issue with this devices is its limited storage capacity. To catalyse this issue use of the cloud storage becomes a necessity. To enhance the performance of cloud it becomes essential to eliminate data redundancy in the storage. For this purpose many models are in use. Most effective one is the data deduplication of the files stored on the cloud[1].

Hashing is the technique which is used to create the string of fixed length to represent set of many strings. Using hashing the set of strings can be identified or indexed with the fixed length hash key generated by the hashing algorithm. As a hashing technique this system uses MD5 hash key generation algorithm[2]. The hash key generated by the algorithm is of 32bit value. The MD5 Hash key generation algorithm is performed into two steps known as Padding and Compression consisting three main operations viz. Bitwise Boolean Operation, Modular Addition, Cycle shift Operation. In Padding the set of strings is converted into the blocks of 512-bit (sixteen 32-bit words). Then the entire string is padded to convert the length of the string into multiple of 512. The algorithm then operates on the 32-bit state, divided into four 8-bit words. The processing of each block is done individually. MD5 hash key algorithm gives fixed size hash value as an output regarding the size of the input string.

The Bloom Filter (BF) is the data structure introduced in 1970 by Burton Bloom. Initially it was used in data base applications, last few years they have been considered inmany networking applications like overlay and peer to peer networking systems[3].The BF is an effective data structure which can be used for hash key management as its working is based on hash functions, thereby allowing false positive; the memory saving capability of BF is very impressive and thus it outweigh its minute drawbacks making its use very tempting. More accurately, a Bloom Filter represents a set X of t elements from a universe U. It uses an array of n bits, denoted by B[1],...,B[n], which are initially all set to 0. A number of r independent hash functions h1,...,h(r) are used, with $\log_2(n)$ bits long output;the hash functions separately map each element of the universe to a random number which is uniformly distributed over the range. For each element s in X, the bits B[hi(s)] are set to 1, for $1 \le i \le r$ (a bit can be set to 1 many times). To answer a question of the form "Is y in X?", one checks whether all bits B[hi(y)] are set to 1. If not, y is not a member of X, while if all B[hi(y)] are set to 1, it is considered as y is in X. As mentioned above, there is a probability of false positives f, that can be properly tuned by varying the values n and r. It is a known result [3] that the optimal value of r is r =(n/a)log2; in this configuration, all bits B[1],...,B[n] are set with probability p =1 /2 (thus, roughly, the same number of ones and zeros occur in the BF) and f is minimized.

In general for a standard bloom filter with size n, the maximum number of elements a(max) that can be accumulated can be given as follows:

$$a(max)= -(n/r) \log(1-f^{(1/r)})$$

The technique used for tagging the multiple users to the same file for deduplication of the data is the Subset Vector Creation. This technique can be used for allowing the access of the same file to the users who have uploaded it. In Subset Vector Creation each file is identified by its unique Hash Key and for that hash key the Subset Vector is created. The each entity in the Subset Vector contains two values, the first one is username (the user who uploaded the file) and the second one is filename (the name of the file he uploaded). The first entry in the Subset Vector is always the first user who uploads the file. Then the multiple users and their respective file name is added to the same set if the hash key of the files they are uploading is same based on the content of the file. The example would be if the user 'ABC' uploads the file '123.txt' with the unique hash key 'KEY' then the subset vector for that unique hash key is created, then if user 'XYZ' uploads the file '987.txt' with the same hash key generated based on the content then the Subset Vector for that hash key would be as follows:

{'KEY'} = {(ABC, 123.txt), (XYZ, 987.txt)}
Where, KEY = unique Hash Key for that particular file

The advantage of this Subset Vector Creation technique is that if even multiple users uploads the same file with different file name still those users can be tagged to the previously uploaded file because it creates the vector based on the hash key.This technique as a file tagging tool is proven effective due to the use of data structure called vector. Though the understanding of vectors is slightly complicated its use is application oriented in the proposed system.

Cloud data being exposed to world via internet, its security is likely to be tampered and data stored on cloud becomes liable to lose its integrity. To avoid this, the proposed work will store only encrypted data on the cloud to guarantee the security and integrity of data stored on cloud. For encryption purpose the use of Reverse Circle Cipher is being made in the methodology. The RCC is a very effective encryption technique [4] and its scope is proved to be vast. The algorithm divides the plain text into N blocks, each block consisting 10 characters. In each block rotate 1 to N number of characters in clockwise direction and substitute each character in the block with randomly chosen character. This process is applied to all blocks in the entire plaintext.

In this paper, the section 2 represents the related work and section 3 elaborates proposed technique in detail. The performance of the system is analysed in section 4 and finally this paper concludes with future extension traces in section 5.

II. LITERATURE SURVEY

The work described in [1] is based on the deduplication by using the secure auditing. The system which is described in this paper have two servers for deduplication. The first server is used normally for all user but second server gives the more protection and data integrity by performing the deduplication on the already encrypted files. The advantage of this system is that it helps clients to generate the data tags for the file before uploading on cloud, and uses proof of ownership protocol for data deduplication. The main drawback of this system is that the data can be plain or encrypted and respect to that the server must be selected. So the two servers always should be integrated to each other for mutual understanding for preventing the duplication of the data.

The technique in [2] uses the MD5 Hashing algorithm used for verification of the data uploaded on cloud. The file uploaded in this system is encrypted at the user end first with RSA Partial Homomorphic Algorithm. The generation of the Hash Key of the uploaded file is done at the server end of the system and the generated key is sent to the client for future use and stored at users end. When user wants to verify his data he has to request for verification and then the server calculates the Hash Key for the user previously uploaded file with same MD5 Hash Key algorithm and sends it to the user. After receiving the hash key, user checks it to the previously stored hash key (generated at the time of the uploading) and if they match that means the data is safe and no modification is done on it. The technique is used for only verification of the data integrity but it doesn't provide the information that the data is accessed by other user or not, means data is exposed or remained safe.

The methodology used in [3] is Bloom Filter which is used in this system for managing the high speed packet classification application. The hash key is generated for every packet and it is stored in the hash table. The bloom filter is used to manage and to perform operations on the hash table. It is proven in this work that Bloom Filter is very effective in terms of space occupancy and complexity. This system has non zero error probability.
The technique in [4] test the Bloom Filter for its applications and to find out its statistical performance. The proposed work uses the NIST test suite, which is used widely to test randomness in the cryptography. In this first the Bloom Filter is isolated by using the randomness in the hash functions. After isolating the Bloom Filter the randomness of the Bloom Filter is tested. This system does not create any new type of the Bloom Filter, it only test the BF for its statistical behaviour.

The new encryption technique is described in [5] called as Reverse Circle Cipher (RCC). The RCC is a simple block cipher scheme which is used to effectively reduce time as well as space complexity. This technique can be used for personal data security and also for network security. The RCC uses arbitrarily variable key length which may consist only two to three bits or it can be of the size of the plain text. The RCC scheme is very cost effective because it can be incorporated within the application layer. This algorithm or scheme is only used on text file. If even single character in the cipher text of the file generated by this algorithm is disposition the entire file becomes the erroneous.
The new technique Performance Oriented I/O Deduplication (POD) is introduced in the proposed scheme of [6]. This new methodology is based on performing the deduplication on the clouds primary storage to give more space and time complexity. The POD takes two approaches towards the deduplication likewise Select-Dedupe and iCache. The Select-Dedupe is used as deduplication selective technique based on the request. The iCache is the adaptive memory management methodology used to alleviate the data fragmentation. The POD scheme is performed on the I/O path so it may take the additional energy for performing it on the I/O path. It does not perform deduplication on the cloud storage directly.

The Application aware Local – Global source Deduplication (ALG-Dedupe) is the technique which is used for the personal cloud storage [7]. It can perform the deduplication by combining the deduplication schemes of the server end and user end to achieve the maximum result. By using application awareness this scheme minimizes computational overhead and deduplication effectiveness. It can only be used for the personal cloud storage not for public cloud storage.
The methodology described in [8] is Deduplication with Dynamic Ownership Management. In Dynamic Ownership Management the multiple owners of the same file are managed by creating the group of the owners and giving them the decryption key of the file. When the owner changes or deletes its file then the key for the decryption of the file also changes and then new key is provided to remaining members from group for data integrity. The use of Dynamic Ownership is that for single file, if the multiple owners are presents then also the file will not be duplicated on the cloud storage. The identification of the file is based on the content of the file. For every instance when group member changes the decryption key is also changes.

III. HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Specification:
Required minimum of 3 machines of following configuration

- CPU : 2.9 Ghz (C2D)
- RAM : DDR 1 GB
- HDD : 100 GB
- Motherboard : Intel 945 GLX
- Monitor, Key Board, Mouse, UPS, DVD Writer ,D-link Network Switch , CAT 5/6 Ethernet cable etc,.

Software Specification:

- Coding Language : Java,
- Development Kit : JDK 1.6, JRE.
- Front End : Java Swing
- Server : Apache Tomcat
- Development IDE : Netbeans 6.9.1
- Data Base : My SQL 5.0

IV. PROPOSED METHODOLOGY

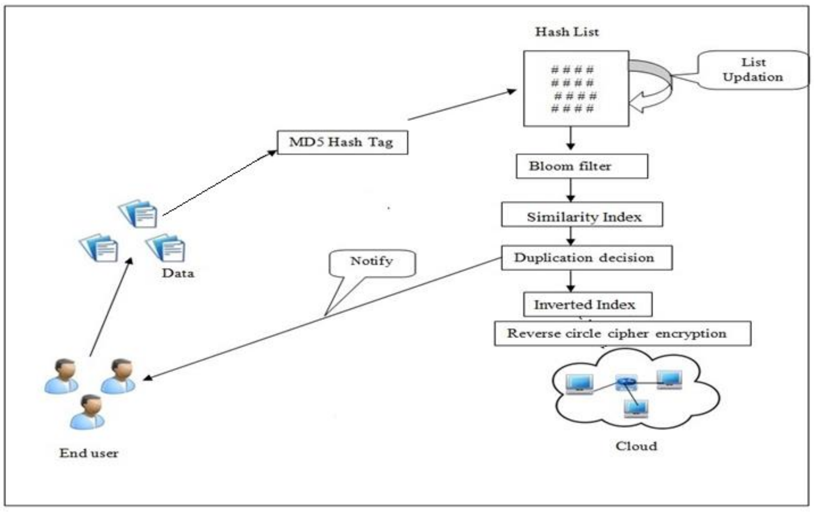


Fig 1: Proposed System Architecture for Data Deduplication

Below Section Details complete Methodology used in Data deduplication.
Phase 1: Inthis phase,Data is to be uploaded by authenticated user. Deduplication System applies Reverse circle Chiper Encryption Algorithm and Data is being sent to Next Phase.
Phase 2: All Encrypted Content is being Hashed using MD5 Algorithmic procedure and file to hash list is maintained.
Phase 3: Once the document hash is created it experiences through the procedure of deduplication. In this procedure sprout channel accumulates all the hash estimation of past records. Arrangement of past qualities, new hash esteem is analyzed. When match is discovered, this hash esteem is bolstered to the similarity index relationship calculation for location of % connection. In the event that the esteem returned by similarity index is 1 then the record is copied and it won't spare to the cloud.

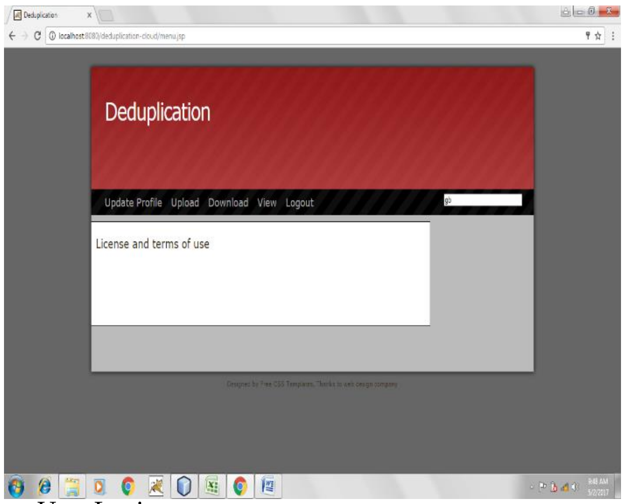
Phase 4: When duplication is recognized, every one of the references of this record with past documents are kept up for the future utilize.
Phase 5: In above case if File is not duplicate system flow comes to this step.
Phase 6: All Information of file is saved using Inverted Index. Cloud plugin is used to deduplicate cloud data. The Proposed framework utilizes switch circle to figureout an encryption calculation for forcing the solid security approach. Turn around circle figure is secured contrasted with other in light of the fact that it makes utilization of private key for encryption reason. Once the info string is acquired it is partitioned into pieces of 10 characters. At that point these individual pieces are turned by their particular file and later nourished to the encryption module. The Encryption module acknowledges the pivoted string and the ASCII estimation of each of the character encryption is performed. Detail usage technique for turn around circle figure calculation is clarified in beneath calculation.

ALGORITHM 2: REVERSE CIRCLE CIPHER

```
Input:  File Text T and Key K
Output: Encrypted Text  $T_E$
Step 0: Start
Step 1:  Create a vector called DIV and initialize count=0, initialize String B to empty
Step 2: FORi=0 to length of T
Step 3: Keep joining characters from T into String B, and count++
Step 4: If count =10
Step 5: Add B to DIV, set count=0 and empty B
Step 6: End FOR
Step 7: FORi=0 to Size of DIV
Step 8: String $B_s$= DIV[i]
Step 9: rotate $B_s$ by one character, initialize sum =0
Step 10: FOR j=0 to length of K
Step 11: sum =sum+ASCII of K[j]
Step 12:  END FOR
Step 13: Val=sum%20
Step 14: FOR j=0 to length of $B_s$
Step 15:   ASCII of $B_s$[j] + Val
Step 16: Replace a new character
Step 17: End FOR
Step 18: Concatenate $B_s$ to a string $T_E$
Step 19: return $T_E$
Step 19: End FOR
Step 20: Stop
```

Phase7: Deduplication System provides feature to download file which was upload by him.
Phase 8: This phase user is able to share his file with trusted users and one who are authorized
.

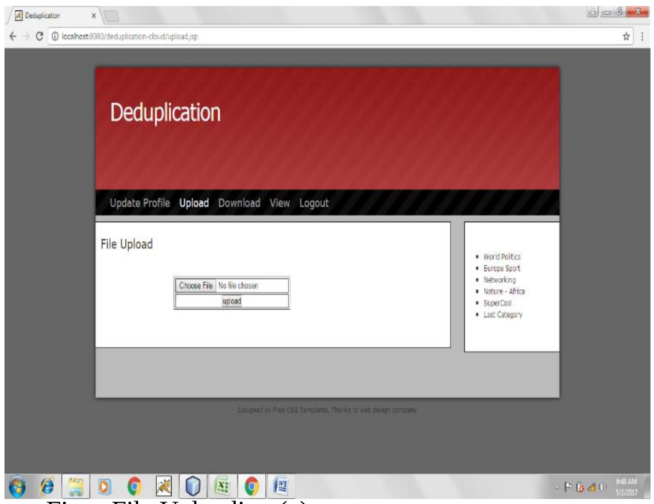
V. SIMULATION



License and terms of use

Designed by Free CSS Templates. Thanks to web design company

Fig:- User Login



File Upload

- World Politics
- Europa Sport
- Networking
- Nature - Africa
- SuperCool
- Last Category

Designed by Free CSS Templates. Thanks to web design company

Fig:- File Uploading (a)

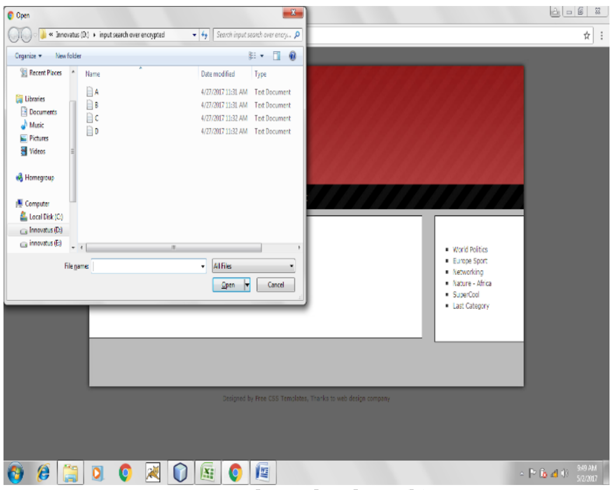
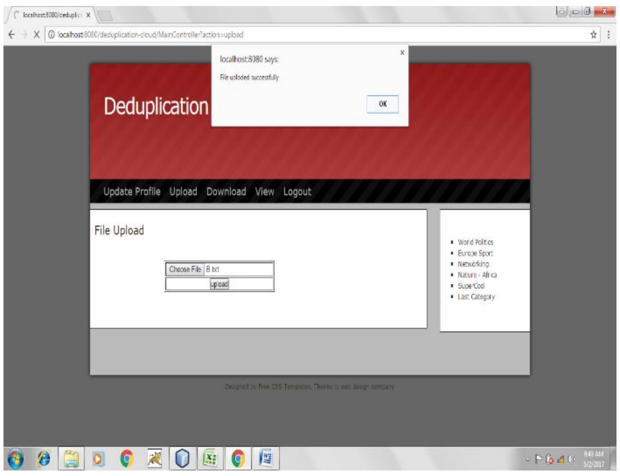
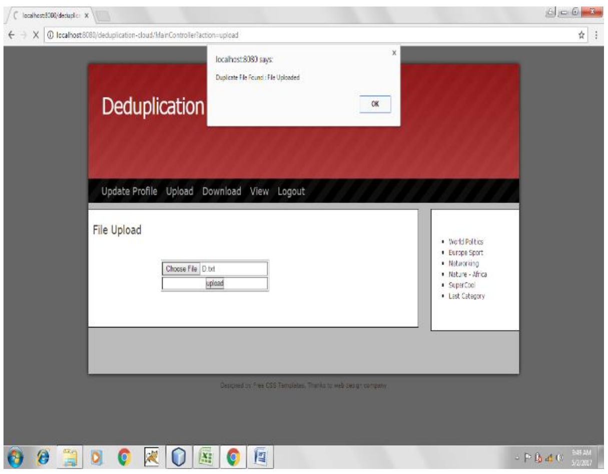


Fig:- File Uploading (b)


localhost:8080 says:

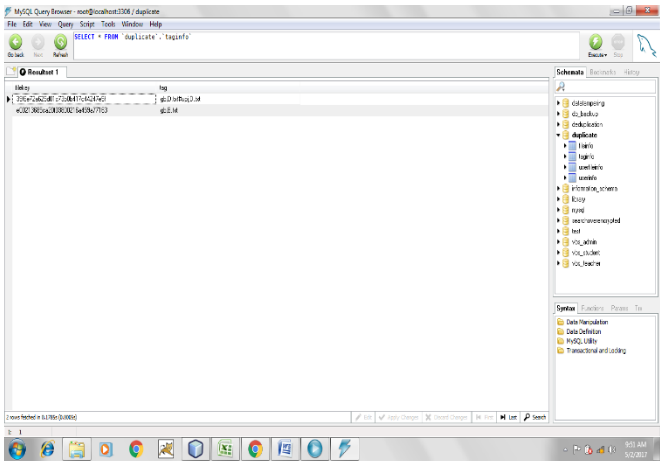
File uploded successfully

Fig:- Successful Uploading


localhost:8080 says:

Duplicate File Found : File Uploaded

Fig:- Duplicate File Up


MySQL Query Browser - root@localhost:3306 / duplicate

```
SELECT * FROM `duplicate`.`taginfo`
```

filekey	tag
396e72a625d01c73b0b417c44247e5f	gb.D.txt#upj.D.txt
e0021368 5ca20038 0021 6a459a77163	gb.B.txt

Fig:- Tag schema in database for Similar files

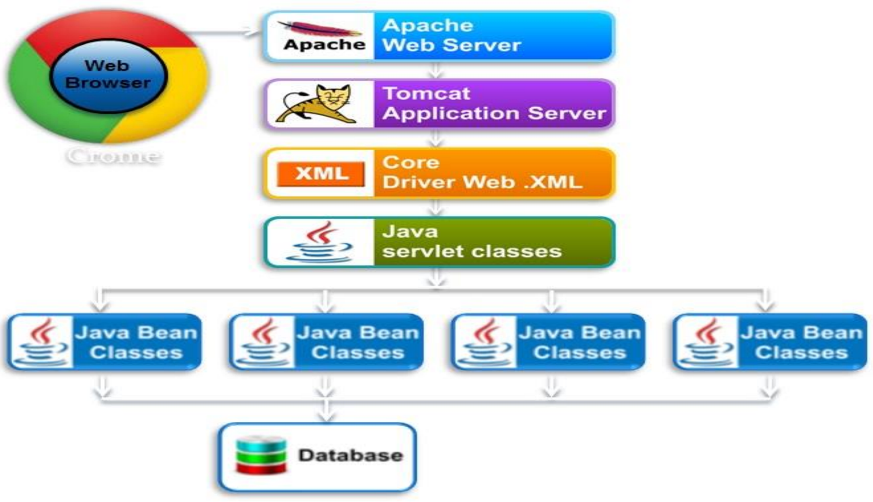


Fig:- WEB DEPLOYMENT DIAGRAM

VI. RESULTS AND DISCUSSIONS

Proposed system of Deduplication isdeployed as a web application using Apache Tomcat and developed using J-creator IDE.

Performance is evaluated based on the precision and recall parameters. So precision can be defined as the ratio of the number of relevant images and text files are identified as duplicates to the total number of irrelevant and relevant images and text files are identified as duplicates.

- ❑ P = The number of relevant images and text files are identified as duplicates,
- ❑ R=The number of irrelevant images and text files are identified as duplicates
- ❑ So, Precision = (P/ (P+ R))*100

It is observed that the tendency of average precision for the images and text files are identified as duplicates is more than the average of the other de-duplication techniques.

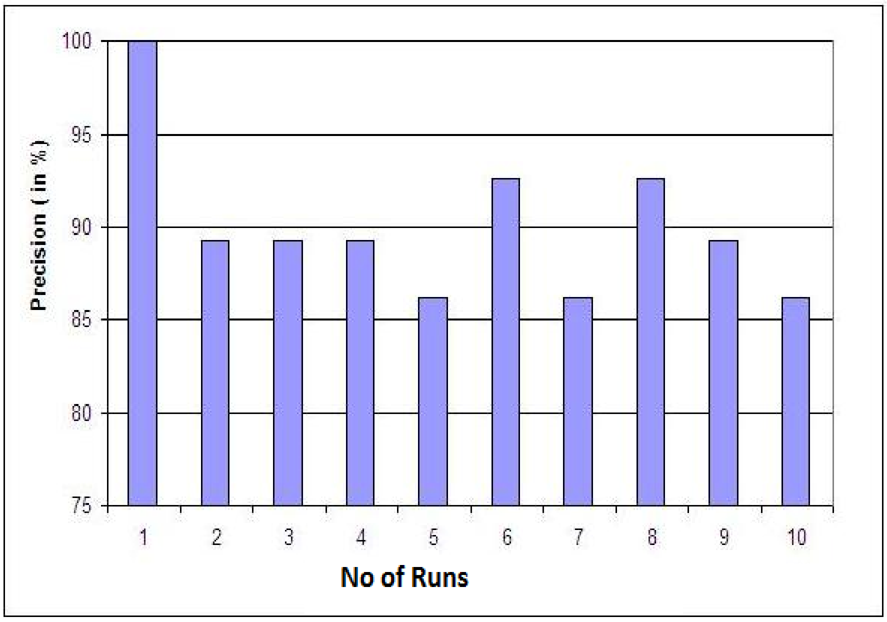


Fig 2: Average precision of the proposed approach

Recall is the ratio of the number of relevant images and text files is identified as duplicates to the total numbers of relevant images and text files are not identified as duplicates and it is usually expressed as a percentage.

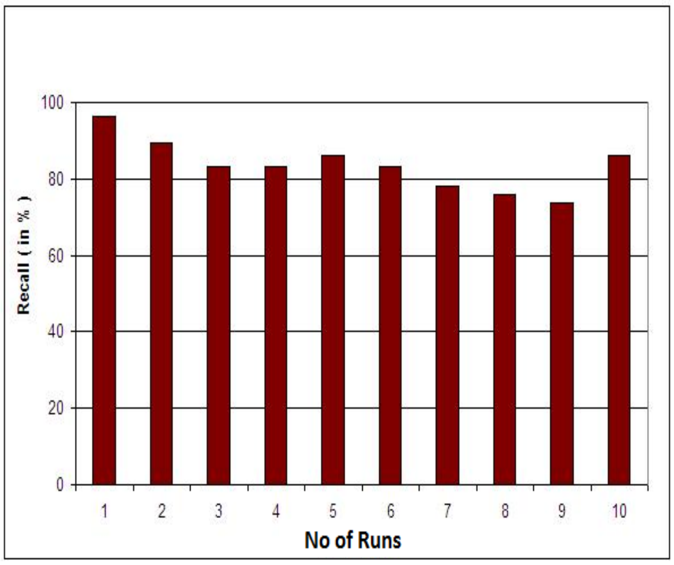


Fig 3: Average Recall of the proposed approach

It is been observe that the tendency of average Recall for the images and text files are identified as duplicates is more than the average of the other De-Duplication techniques. So this shows that our proposed system is achieving high accuracy than any other method.

VII. CONCLUSION AND FUTURE SCOPE

The developing necessity for secure cloud storage services and better Encryption Decryption Lead to combine them, thus, defining an innovative solution to data Management and storage in cloud.Numerous Deduplication Schemes exists but fail to provide a complete reliable solution to duplication.

Above proposed System is scalable and achieves better Performance as compared to other deduplication approaches. System can enhance to implement in Internet of things paradigm. In future the scheme proposed could be enhanced to work in all format of data including audio, video.

REFERENCES

[1] Jingwei Li, Jin Li, DongqingXie and Zhang Cai, "Secure Auditing and Deduplicating Data in Cloud", IEEE Transactions on Computers, 2015

[2] PriyankaOra, Dr.P.R.Pal, "Data Security and Integrity in Cloud Computing Based On RSA Partial Homomorphic and MD5 Cryptography", IEEE International Conference on Computer, 2015

[3] N. Bonelli, C. Callegari, S. Giordano, and G. Procissi, "A Bloom Filter Bank Based Hash Table for High Speed Packet Processing", IEEE International Conference on High Performance Computing and Communications, 2014

[4] VenkateshwarKottapalli, Sunil Khatri, "A Practical Methodology to Validate the Statistical BehaviorofBloomFilters" IEEE, 2016

[5] Ebenezer R.H.P. Isaac, Joseph H.R. Isaac and J. Visumathi, "Reverse Circle Cipher for Personal and Network Security", IEEE, 2013

[6]Bo Mao, Hong Jiang, Suzhen Wu and Lei Tian, "Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud", IEEE Transactions on Computers, 2015

[7]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 TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 5, MAY 2014

[8]JunbeomHur, Dongyoung Koo, Youngjoo Shin, and Kyungtae Kang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage", IEEE Transactions on Knowledge and Data Engineering, 2016