Date of Publication: 30 May 2018| Volume 6, Issue 2 | Pages: 24-29

Storage-Effective Deduplication by Improved Hashing Hybrid Computations SHA3 and Firefly

Karmjeet Singh, M-Tech Scholar, Doaba Institute of Engineering and Technology (PTU) Punjab, karmjeetsingh216@gmail.com
Ramanjot Kaur, Assistant Professor, Doaba Institute of Engineering and Technology (PTU) Punjab, raman0803@gmail.com

ABSTRACT: Today's modern generation is the world of digitization and cloud assesses, it begins to be exceedingly paramount to bestow prodigious specifics. Prolong excessive traffic of raw material which upload to storage space upshots data de-duplication in cloud storage and also responsible for network congestion many times. Necessity comes with a new invention to fulfill the future requirement of high definition raw material which continued to inject to the cloud. To overcome data redundancy and security issuance, the number of techniques instigates which helps to get the traits of same raw material, but dwindled and becomes obsolete against the new obstacle of server-storage and data delicacy, so necessitate of up to date competence inevitably inflated. To engineer this target in this paper I arbitrate to club together the consequence of different hashing, compression computations, so that performance, accuracy, and throughput to be revamped, I have achieved with the help of SHA family and firefly algorithms.

Keywords: Data-De-duplication, Secure Hashing-chunking algorithm, Cloud data optimization, and de-duplication secure optimized algorithms, SHA3 and firefly algorithms.

1. INTRODUCTION

The World is going to be computerized completely, in every second new concept introduced that helps to computer problems and to minimize the labor. Documentation is needed for the future record also for reuse, which continuous uploads to server storage, a million records updated every day which tend to raise high data duplication rate, later pioneer cloud computing in 1960 and feat on ARPANET. A cloud storage which acts as evolution for data warehouse triggers statics security in case of backup server failure and theft. Ease of usage makes cloud network most popular as well as high in demand. Cloud computing caper vital role in computing which is on an eminent visitation footing in the phrase of attainable, freely access worldwide and ascendible. Cloud computing infrastructure get by framework by physical parameters [13]. According to the recent survey, stats show that cloud storage filled with 2400 GB daily, that is responsible for heavy traffic and network congestion.

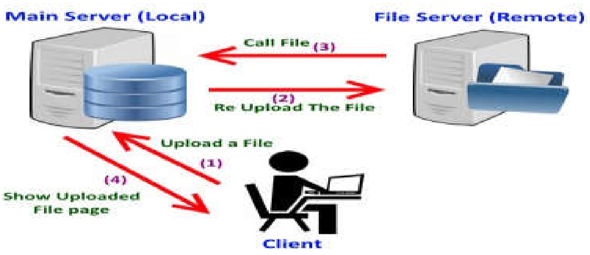

Figure 1: Data uploading to a server

Beyond this unique service, web-server are regularly filled with a raw material like organizations' records, folder - files with different extensions, audio video facts, image files gif png etc. with unique location code in a web server. But many times possibility occurs of same data bits (0100) exist at different places or at disparate server space and this concept is called deduplication. Sometimes same information filled repeated in database multiple times that is known as redundancy. Deduplication can be understood by front-end and Hadoop file method [12]. To optimize these problems number of computations introduced like hashing function (SHA), chunking algorithm, Bees Colony rule etc, for security encryption LZ77 LZ78 -decryption cipher-text plan text, public-private cryptography so on, to identify identical parameter of different as well as of same raw facts to find out repeated statics on could space and these programs played giant role. But algorithms become obsolete to get all similar parameter of new high-quality raw material which uploaded to cloud because latest technologies inject data with new parameters. I am clubbing together previous designed hashing algorithms to enhance the performance parameters like time, efficiency, throughput, and speed for better adjudication.

2. RELATED WORK

Schneier Bruce et al. [1] Hash function helps to map data of variable length to data of fixed sized length. The output generated by a hash function is called hash value, hash code etc, A cryptographic hash function is the subcategory of hash function that has the certain feature which suitable for use in cryptography.

Harsha Nagarajaiah et al. [2] the relatively low performing embedded processors are capable of providing the need computational provision if they were to holder security functions in the field. When likened to the algorithmic presentation on the extraordinary end scheme.

Hui Wang et al. [3] Firefly Algorithm played an important role to fix identical bits. On the basis of Adaptive parameter flies energy of attraction calculated. Two flies attract each other directly depends on light emitted by them. Higher the light energy results high will be the attraction between two flies. The same phenomenon applied on the data duplication.

Meixia Miao et al. [4] securities of datum play a crucial role in cloud deduplication. Algorithms play a role to maintain a data safety while transferring data online. It allowed a secret sharing of data between sources to the destination with the aid of server key and threshold signature key but difficult to fully block brute force attack on deduplication.

Mitchell et al. [14] Hash value can be obtained by processing the data. A hash index chart is generated which have addresses, Hash size can be varied hinge on stored data. Hash apparatus helps to generate hash which helps process unique id of each record.

3. DEDUPLICATION TECHNIQUES

A. Based on Deduplication process, there are two methods.

(i) Hash-based De-duplication: - In the above-mentioned methods, i.e. file and block level De-duplication 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 binary files or chunks [15] it produces same hash value, then the two files or blocks is indistinguishable and is not stored, if the generated hash value is different, then they are different and are stored in disk [7].

(ii) Content or Application-aware De-duplication: - It divides the data into large segments by knowing the contents of the substances like files, database objects, presentation objects. Then it inventions the redundant divisions and stores only the bytes changed in the two segments, hence known as byte level De-duplication.

Graphically representation:-

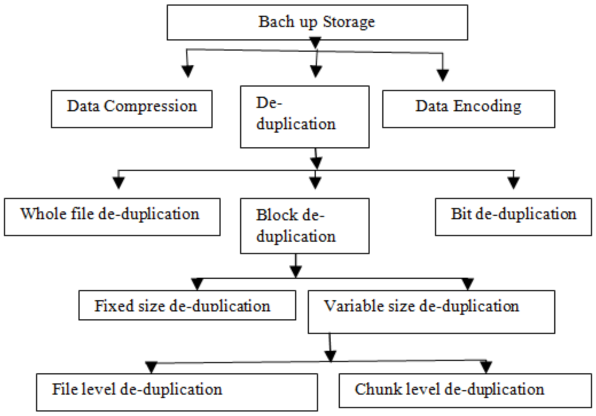
Bach up Storage
Data Compression
De-duplication
Data Encoding
Whole file de-duplication
Block de-duplication
Bit de-duplication
Fixed size de-duplication
Variable size de-duplication
File level de-duplication
Chunk level de-duplication

Figure 2: deduplication levels

B. Based on footage two methods

i. *File extent duplication*: - Duplication can be short out by removing the same parameter document. Backup have one copy of original file while other are removed by pointer detecting duplicity. This method is also known single instance storage. It doesn't affect the actual content of dossier. Used for simplicity and fastest work.

ii. *Chunk duplication:* - Duplication detected at chunk or at blocking way. Each value of record allocates a chunk with computation. If statics is distinctive than written to memory else discarded. This method is simple and less CPU usage. Further it divides into two categories.

iii. *Stable footage*: - facts are split into stable size, easy to process but fail to search redundant data.

iv. Shifting footage: - it more efficient then stable it change stack size while processing, able to scan complete file.

C. A Transmission-based method in deduplication

i. Sender/ receiver deduplication: - Before transmission each packet of data analyzed. Only checked bits transfer to the receiver. It helps to improve network traffic and less bandwidth.

ii. *Destination deduplication:* - in this method optimization done later, after transmission, it helps to save the time of preprocessing but waste bandwidth over the network.

4. ALGORITHM EXPLANATION

Improved Hashing Hybrid Computations SHA3 and Firefly Hybrid algorithm designed to obtained better efficiency:- Steps

```
1. Start.
2. Initial default parameters
            Buffer_size =0;
            Frame_size=0;
            Counters = 0;
            Padding bits=initial cursor.
3. Select and upload image
4. begin  for till file_byte.end
            Read bytes(uploaded image.bytes);
            End for.
5. Generate frames of upload bytes and add padding bits.
            while (current< length of file)
            if ((hash_current_file & mask_content) == 0)
                        if (firstChunk_content == 1)
                                    chunk_content = new byte[(int)
                        current];
                                    firstChunk_content = 0;
                        else
                                    chunk_content = new
                        byte[frames];
                         if (fsChunk_file.read(chunk_content) !<1)
                                    hashvalue =
            combine.chunking(chunk_content);
end while
7. Optimize hashing buffers
            If buffer.content<threshold
                        Repeat step 6
            If(hash_found==0)
                        Upload file bytes
             Else
                        Share file bytes
8. Stop.
```

- *SHA-3 ALGORITHM Explanation*

SHA-3 is a member of the Secure Hash Algorithm family. The SHA-3 typical was unconfined by NIST on August 5, 2015. The reference implementation source code was dedicated to public domain via CC0 waive. That move was determined by uncertainties which so far haven't come to pass-that SHA-2 might be vulnerable to being cracked. Hashing algorithms are an important information security tool and used to confirm messages, as well as digital signatures and documents. "A noble hash algorithm has a few vital features," giving to NIST. "Any change in the unique message, however minor, must cause a change in the condensation, and for any given file and digest, it must be infeasible for a forger to create a diverse file with the same digest."

- *Firefly Algorithm Explanation*

Firefly computation played a vital role in optimization, is proposed by Xin-She Yang by analyzing the reaction of Firefly (one in the family of insects) which is based on the attraction of sexual charm and brightness. Flies generate unique flashing pattern which helps in attraction. If brighter flashing light between them, then more bonding power will be and vice versa. A Light coalition with the functioning of finding the same pattern exists nearly.

Algorithm firefly:

- Arbitrary originate the populace with N startup possibilities.
- {Yi}i = 1,2,...,N};
- Analyse the fitness value of Yi;
- Firefly iteration Fi = N;
- **While** Fi <= Maxi Fn **do**
- **for** i = 1 to N **do**
- **For** k = 1 to N **do**
- **If** f(Yk) < f(Yi) **then** Yk <<1 to Yi;
- Compute the flash value of a new Yi;
- Fi++;
- end; end; end;

5. PROBLEM FORMULATION

It makes a system, network efficiency and storage optimized. Today, in the context of user data allocation platforms the contests for large scale [8, 9], highly redundant internet data storage is high. Due to this redundancy storage cost is decreased. Storage of this gradually centralized Web data can be gotten by its de-duplication. If we consider a case in which user updates one same file on multiple time, it takes space a lot of server memory [10].

- If the server has a large amount of data than searching technique become slow.
- Unwanted space consumption is very costly when users are in billions.

- Current hashing function or searching technique is not much better. It is a more time-consuming process to search for any of the records.

Data is the most important thing in the system so we need an accurate fingerprint generator algorithm which finds files fast and accurate and current system having this type of functions but isn't proposed 100% accuracy.

6. OBJECTIVES

This thesis encompasses a set of objectives that are associated with a set of objectives that are associated with the milestone of this process. The objectives are mentioned below.

- To study various methods to store data online.
- To propose hybrid approach expending De-Duplication Approach for Reducing Memory Consumption in Cloud Computing using an Improved SHA3 algorithm with firefly optimized hash code generation.

To compare the proposed method in terms of various parameter (Processing speed, storage space, accuracy etc.) matrixes with existing de-duplication schemes.

7. Methodology

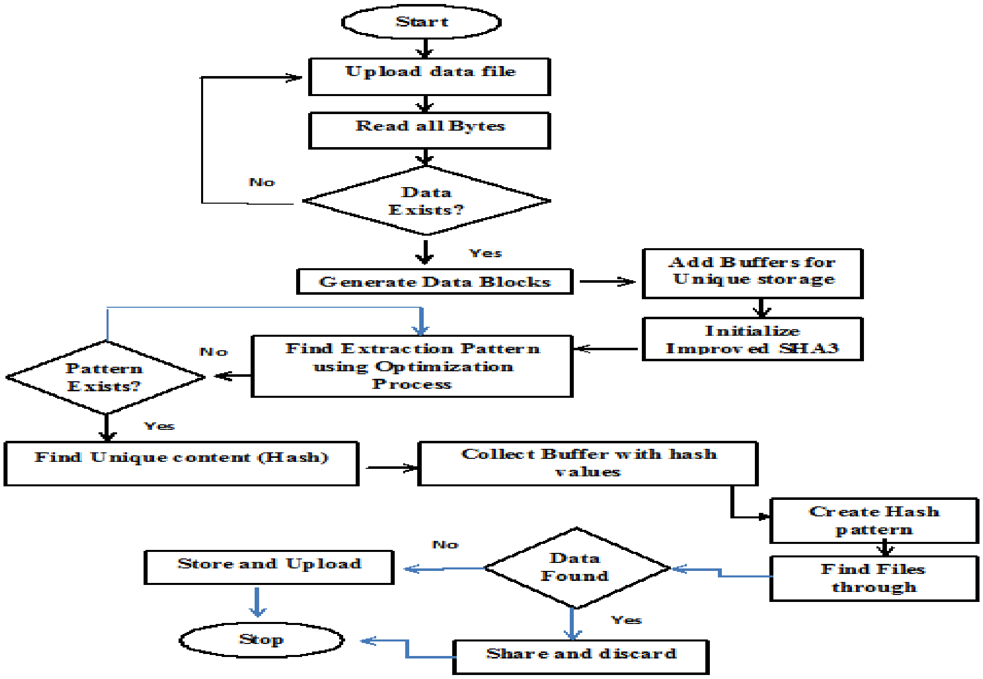

Figure 3: data flow diagram explanation

8. PROPOSED WORK

The proposed algorithms are the combination of two approached which create a secure and optimized way to find the unique content from a file. This approach improves the existing hashing algorithm's performance with the help of optimization for generating hashing pattern of input data. All the content bytes in this approach divides into various sub data block to compute with the help of multi-threading policy. These blocks are a part of actual content and system process the blocks with the help of buffers to collect the unique bytes from them. All the buffers connected with proposed algorithm are refined to form a unique hash pattern. After the collection of all buffers and their refinement, the proposed approach going to match the upload content with already stored data in the form of a unique hash. It makes the comparison process faster as the comparison is just in bytes instead of whole data. If the content exists in the database then it shares the content and discards uploading

Date of Publication: 30 May 2018| Volume 6, Issue 2 | Pages: 24-29

process in the database. This process saves storage space along with faster searching of data from large datasets.

9. RESULT & DISCUSSION

Comparison between the memory consumption of existing and proposed working technique is covered in this figure. The main issue in the overall process is to save the storage space while working with media file requests from the cloud users. This figure shows the better storage management and shows more space than the traditional approach.

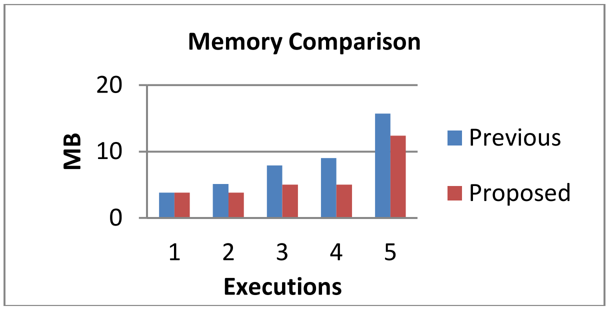

Figure 4: Memory consumption

Another main issue after the storage is to compute execution time of processing algorithms and check the efficiency in terms of their working capability. The various existing approaches are used to find the unique properties of the uploaded content and compare with the stored data. This approach provides high-speed processing in this area and finds the unique properties from the content in less time.

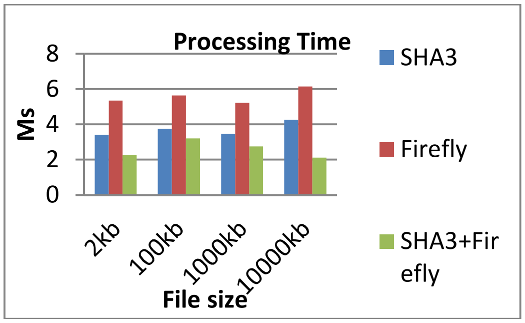

Figure 5: Processing time

As the cloud servers are working in terms of networks. Here another computation is shown in the figure to find the various operations on various upload on the cloud network. The five duplicators handle the operations like uploading, deleting and updating the content. As the storage structure become a little complex with de-duplication process so this parameter needs to find the saving of time while performing the operations.

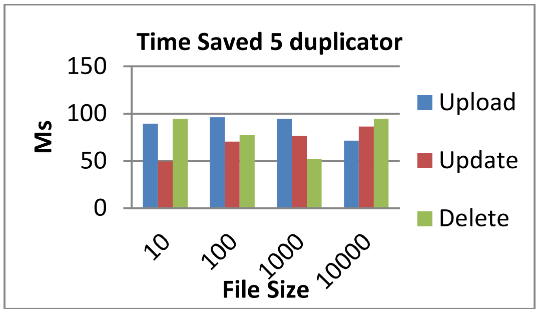

Figure 6: Performance with five duplicators

As the cloud servers are working in terms of networks. Here another computation is shown in the figure to find the various operations on various upload on the cloud network. The ten duplicators handle the operations like uploading, deleting and updating the content. As the storage structure become a little complex with de-duplication process so this parameter needs to find the saving of time while performing the operations.

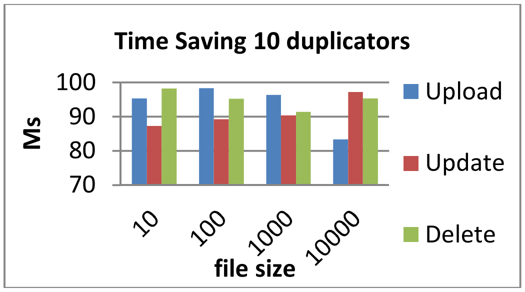

Figure 7: Performance with ten duplicators

All the parameters are working better, but if the accuracy is not up to mark then it might be harmful to the storage. Here various different comparisons are used to find the stability of the accuracy parameter. In given approaches, the proposed architecture provides high accuracy factor to store, find and perform other operations on the files.

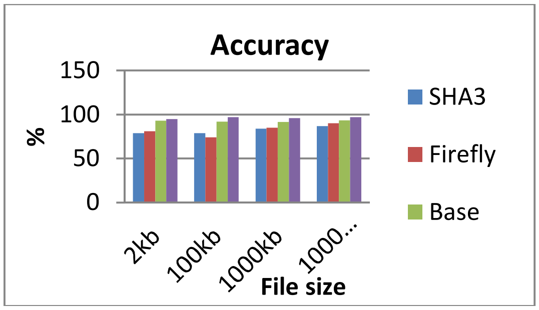

Figure 8: Accuracy

Date of Publication: 30 May 2018| Volume 6, Issue 2 | Pages: 24-29

10. CONCLUSION & FUTURE SCOPE

This paper is responsible for clubbing different computation to get the desired corollary that comes in towering throughput and helps to improve cost. This fresh research comes with a fruitful outcome to minimize the de-duplication and reduced server maintenance cost. For instance, consider the use of a cryptographic hash function to find out duplicate segments of data. If there are two different files output same hash value then that maximum chance of collision occurrence.

Collision proximity depends hash function used, although the probabilities are small, they nearly non zero. But the matter of data corruption and hash collision has to face. Even encryption helps to remove any perceptible pattern in data. But encrypted data cannot be de-duplication. Security concern arose when de-duplication occur data security and access validation breaches. The Improved Secure hash function works according to updated steps designed to find out repetitive data. It helps to maintain searching the speed in a large database and in search engine, offered an optimal solution which previous algorithms wane to achieve.

11.REFERENCES

[1] Schneier, Bruce. "Cryptanalysis of MD5 and SHA: Time for a New Standard". Computerworld. Retrieved 2016-04-20. Much more than encryption algorithms, one-way hash functions are the workhorses of modern cryptography.

[2] H. Nagarajaiah, S. Upadhyaya, and V. Gopal, "Data de-duplication and event processing for security applications on an embedded processor," Proc. IEEE Symp. Reliab. Distrib. Syst., pp. 418–423, 2012.

[3] Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H. and Cui, L., 2017. Firefly algorithm with adaptive control parameters. Soft computing, 21(17), pp.5091-5102. (kasu chola)

[4] Secure multi-server-aided data deduplication in Cloud Computing, Meixia Miaoa, Jianfeng Wangb, Hui Li b, Xiaofeng Chenb.

[5] Debnath, B.K., Sengupta, S., and Li, J., 2010, June. ChunkStash: Speeding Up Inline Storage Deduplication Using Flash Memory.

[6] M. O. Rabin fingerprinting by random polynomials. Center for Research in Computing Technology Harvard University Report TR-15-81 (1981)

[7] Bhagwat, D., Eshghi, K., Long, D.D. and Lillibridge, M., 2009, September. Extreme binning: IEEE International Symposium on (pp. 1-9).

[8] Khajeh-Hosseini, A., Greenwood, D., Sommerville, I., (2010). Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS. Submitted to IEEE CLOUD 2010.

[9] Rashmi Rao, Pawan Prakash, "Improving security for data migration in cloud computing using randomized encryption technique", IOSR, Volume.11, pp. 39-42, 2013.

[10] Jyoti Malhotra, Priya Ghyare2, "A Novel Way of De-duplication Approach for Cloud Backup Services Using Block Index Caching Technique", Vol. 3, Issue 7, July 2014 DOI: 10.15662/ijareeie.2014.0307040.

[11] Leesakul, W., Townend, P. and Xu, J., 2014, April. Dynamic data deduplication in cloud storage. In Service-Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on (pp. 320-325). IEEE.

[12] Sun, Z., Shen, J. and Yong, J., 2011, June. DeDu: Building a deduplication storage system over cloud computing. In Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on (pp. 348-355). IEEE.

[13] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. and Zagorodnov, D., 2009, May. The eucalyptus open-source cloud-computing system. In Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on (pp. 124-131). IEEE.

[14] Mitchell, G.R., and Houdek, M.E., International Business Machines Corp, 1980. Hash index table hash generator apparatus. U.S. Patent 4,215,402.

[15] Chang, S.J., Perlner, R., Burr, W.E., Turan, M.S., Kelsey, J.M., Paul, S. and Bassham, L.E., 2012. Third-round report of the SHA-3 cryptographic hash algorithm competition. NIST Interagency Report, 7896.