ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Analysis of an Efficient Approach for Duplicate Detection System

Ruchira Deshpande[#1], Sonali Bodkhe[#2]

[#1,2]Department of Computer Science & Engineering, Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharashtra, India

ABSTRACT: In computing, there are multiple representations of same data. Finding out those duplicated data is a difficult task. Duplicate detection is the process of identifying those redundant data to reduce storage utilization and avoid the confusion. Today, duplicate detection methods need to process larger data in shorter time period. The proposed system will be having efficient approaches for finding duplicate data. The first is based on Novel progressive duplicate detection algorithms that will significantly increase the efficiency of finding duplicate files. The second is based on secure hashing algorithm which will find duplicated content by dividing the data into chunks. The best algorithm will be analysed for finding duplicate data with higher accuracy and also in a shorter time span.

KEYWORDS*:* Progressiveness; Duplicate file detection; De-duplication; Duplicate Content; Secure Hashing.

I. INTRODUCTION

In many Organizations, data are among the most important assets of a company. But due to data changes and sloppy data entry, errors such as duplicate entries, making data cleansing and in particular duplicate detection is indispensable. However, the pure size of today's datasets renders duplicate detection processes is expensive. Duplicate detection is the process of identifying multiple representations of same real world entities. Today, duplicate detection methods need to process ever larger datasets in ever shorter time, maintaining the quality of a dataset becomes increasingly difficult. A large portion of internet service data is redundant because now a day's people tend to save data at multiple times for data safety reasons and avoids purchasing storage for high cost; this leads to more redundant data.

The identification of these duplicate data is fundamental to improve the quality of information retrieval. Progressive duplicate detection algorithms namely progressive sorted neighbourhood method (PSNM), enhances the efficiency of duplicate detection even on very large data. These introduce a concurrent progressive approach for the multi-pass method and adapt an incremental transitive closure algorithm that together forms the first complete progressive duplicate detection workflow.

Data de-duplication has become a terribly necessary method of knowledge mining, it is a specialized method of information compression that eliminating duplicate copies of continuation data. Related and somewhat synonymous terms are unit intelligent (data) compression and single-instance (data) storage, large info storage, content delivery networks, blog sharing, news broadcasting and social networks as ascendant part of web services are unit information central. Hundreds of several users of those services generate peta bytes of latest information. Databases play the necessary role in today's IT based mostly economy. Many industries and systems rely on the accuracy of databases to hold out operations. Therefore, the quality of the knowledge stored within the databases, have important value implications to a system that depends on info to run and conduct business. In an error-free system with absolutely clean information, the construction of a comprehensive view of the info consists of linking in relative terms, joining 2 or a lot of tables on their key fields. Unfortunately, data typically lack a distinctive, global symbol that would allow such associate operation. Furthermore, the information area unit neither rigorously controlled for quality nor outlined in an exceedingly consistent means across completely different data sources.

Progressive duplicate detection algorithms namely progressive sorted neighborhood technique (PSNM), which performs best on little and almost clean datasets, and progressive blocking (PB), which performs best on giant and terribly dirty datasets. Both enhance the potency of duplicate detection even on terribly giant datasets. In progressive

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

duplicate detection algorithms, two dynamic progressive duplicate detection algorithms, PSNM and PB, will be enforced that expose completely different strengths. Introduces a concurrent progressive approach for the multi-pass technique associated adapt a progressive transitive closure algorithmic rule that along forms the first complete progressive duplicate detection progress.

Hash is a fixed length representation of arbitrary length message. Hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called as hash values. A hash value is a numeric value of a fixed length uniquely identifies data .hash value represent large amount of data.

In distinction to progressive duplicate detection the cryptographic hashing is another thought in detection and deleting redundant information. Secure hash Algorithm is a Collision free and impossible to re-create the same message.Hash function produce hash value called as message digest.

Below given are the architecture diagrams of file based de-duplication and content based duplicate detection in figure 1 and 2 respectively. This gives the brief idea of the architectures for duplicate detection system for both the files and the content inside the files. Different algorithms are applied to evaluate the parameters of de-duplication.

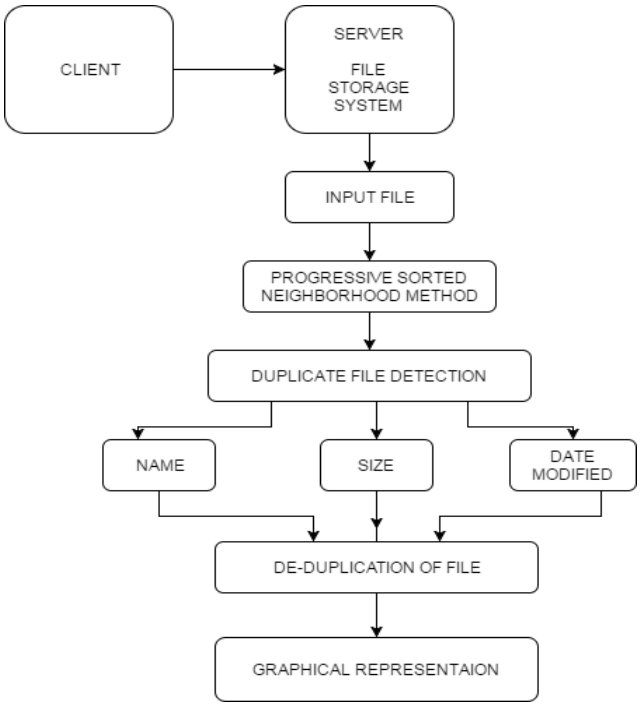

Figure 1.1: Architecture for File De-Duplication System

Above architecture gives the brief idea of file based duplicate detection system in client server environment that how exactly the duplicate files are identified by giving efficient results. Progressive sorted neighborhood method is used to identify duplicate files with mainly three parameters and that are name of the file, size of the file and last date modified of the file.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

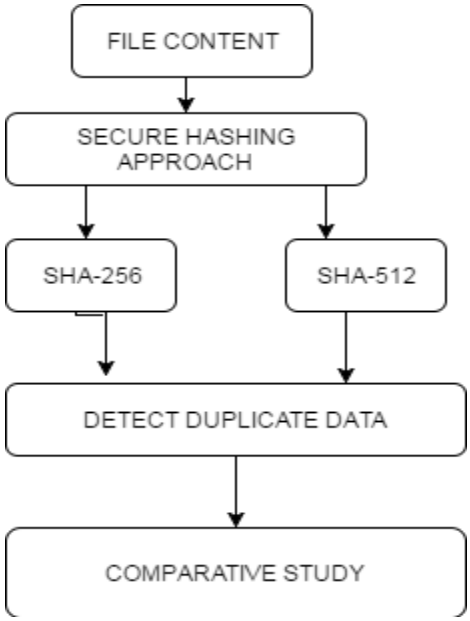

Figure 1.2: Architecture for Detecting Duplicate content

Architecture in figure 2 gives the brife idea of duplicate content detection using secure hashing techniques.

II. RELATED WORK

All active methods and non identical duplicate entries present in the records of the database are investigated [2]. It works for both the duplicate record detection approaches: Distance Based technique that measures the distance among the individual fields, by using distance metrics of all the fields and later computing the distance among the records. Rule based technique that uses rules for defining that if two records are same or different. Rule based technique is measured using distance based methods in which the distances are 0 or 1. The techniques for duplicate record detection are very essential to improve the extracted data quality.

Much research on duplicate detection [1], [5], also known as entity resolution and by many other names focuses on pair-selection algorithms that try to maximize recall on the one hand and efficiency on the other hand. The most prominent algorithms in this area are Blocking and the Sorted Neighbourhood Method. Previous publications on duplicate detection often focus on reducing the overall runtime. Thereby, some of the proposed algorithms are already capable of estimating the quality of comparison candidates. The algorithms use this information to choose the comparison candidates more carefully. For the same reason, other approaches utilize adaptive windowing techniques [2], which dynamically adjust the window size depending on the amount of recently found duplicates. These adaptive techniques dynamically improve the efficiency of duplicate detection, but in contrast to our progressive techniques, they need to run for certain periods of time and cannot maximize the efficiency for any given time slot.

Two novel, progressive duplicate detection algorithms[1] namely progressive sorted neighborhood method (PSNM), which performs best on small and almost clean datasets, and progressive blocking (PB), which performs best on large and very dirty datasets. Both enhance the efficiency of duplicate detection even on very large datasets. Which expose different strengths and outperform current approaches. They exhaustively evaluate on several real world datasets testing own and previous algorithms [1].

To ensure scalability, [3] clustering approaches are considered which can use as input the new state-of-the-art scalable approximate join techniques to find similar items. The input to the clustering is the output of the approximate join which can be modelled as a similarity graph G $(U; V)$, where a node $u \in U$ in the graph represents a record in the data and an edge $(u; v) \in V$ exists only if the two records are deemed similar. In these join techniques, two records are

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

deemed similar if their similarity score based on a similarity function is above a specified threshold μ. The similarity graph is often weighted, i.e., each edge $(u; v)$ has a weight w $(u; v)$ which is equal to the similarity score between the records corresponding to nodes u and v. But a key point is that these approximate join techniques are extremely proficient at finding a small and accurate set of similar items. This feature permits the effective use of clustering techniques on the output of the join, including the use of techniques that would not scale to graphs over the original input relations.

The word record is used to mean a syntactic designator of some real-world object [4], such as a tuple in a relational database. The record matching problem arises whenever records that are not identical, in a bit-by-bit sense or in a primary key value sense may still refer to the same object. For example, one database may store the first name and last name of a person (e.g. "James Pit"), while another database may store only the initials and the last name of the person (e.g. "J. K. Pit"). The record matching problem has been recognized as important for at least 50 years. Record matching algorithms vary by the amount of domain-specific knowledge that they use.

The pair wise record matching algorithms [4], [7] used in most previous work have been application-specific. Many algorithms use production rules based on domain-specific knowledge. The process of creating such rules can be time consuming and the rules must be continually updated whenever new data is added to the mix that does not follow the patterns by which the rules were originally created. Another disadvantage of these domain-specific rules is that they answer whether or not the records are or are not duplicates, there is no in between. In computing, data de-duplication is a specialized data compression technique for eliminating duplicate copies of repeating data. Related and somewhat synonymous terms are intelligent (data) compression and single-instance (data) storage. [6]This technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. In the de-duplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis.

In hash based data de-duplication process [8] it uses cryptographic hash to detect redundant copy of any record. In the general process storage server maintains a hash table, which contains two fields. One is hash signature and other is its real address. It calculates the hash signature for each record requesting for backup by using secure hash algorithm. Now it searches for this hash signature in hash table. If signature not found, that means record is unique, and do an entry for this in hash table.

Entity Resolution [12] which is used for determining entities associated to similar object of the real world. It has significant importance in data integration and data quality. They proposed Map Reduce for SN blocking execution. Both blocking methods and methods of parallel processing are used in the implementation of entity resolution of huge datasets. [5] Introduced a Duplicate CountStrategy, which become accustomed to the window size depending on the count of duplicates detected. Blocking and windowing methods [5] used to reduce the time taken to detect duplicates. Sorted Blocks are also analysed which denotes a generalization of these two methods. Blocking divides the records to disjoint subsets and windowing slides a window on the sorted records and then comparison is made between records within the window.

Data de-duplication is a specific data firmness method which makes all the data owners, who upload the same data, share a particular copy of duplicate data and removes the duplicate copies in the storage.[12] When users upload their data, the cloud storage server will check whether the uploaded data have been deposited or not. If the data have not been stored, it will be really written in the storage; otherwise, the cloud storage server only stores a pole, which points to the first stored copy, instead of storing the whole data. Hence, it can avoid the same data being stored recurrent.

Backup is an effective measure to protect data. Data can be restored using backed up copies in case of data loss. Full backup, incremental backup, and differential backup are three common backup strategies. The traditional sliding blocking (TSB) [9] algorithm is a typical chunk level duplicate detection algorithm. It divides the files into chunks and introduces a block-sized sliding window to move along the detected file and to find redundant chunks. In order to enhance the duplicate detection precision of the TSB algorithm, Wang et al. proposed a novel improved sliding blocking algorithm, called SBBS. For matching-failed segments, SBBS continues to backtrack the left/right quarter and half sub-blocks.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

III. PROBLEM DEFINATION

1. A user has only limited, maybe unknown time for data cleansing and wants to make best possible use of it. Then, simply start the algorithm and terminate it when needed. The result size will be maximized.
2. A user has little knowledge about the given data but still needs to configure the cleansing process.
3. A user needs to do the cleaning interactively to, for instance, find good sorting keys by trial and error. Then, run the progressive algorithm repeatedly; each run quickly reports possibly large results.
4. All presented hints produce static orders for the comparisons and miss the opportunity to dynamically adjust the comparison order at runtime based on intermediate results.
5. Progressive duplicate detection works on single machine so all the parameters are not taken into consideration.
6. Scalability has been neglected so far.
7. It takes more time to find duplicate for larger datasets.

IV. PROPOSED SYSTEM

Proposed work relies on study of progressive duplicate detection algorithm and secure hashing algorithm for duplicate Detection.

4.1 Progressive Duplicate Detection Algorithm.

Implementing progressive duplicate detection algorithm namely PSNM which may expose completely different strength. Progressive duplicate detection algorithms specifically progressive sorted neighborhood methodology (PSNM) that performs best on tiny and virtually clean datasets, and progressive obstruction (PB), that performs best on massive and extremely dirty datasets. It introduces a coincidental progressive approach for the multi-pass methodology Associate in Nursing adapts progressive transitive closure algorithmic program that along forms the primary complete progressive duplicate detection work flow.

Progressive sorted neighborhood methodology relies on ancient sorted neighborhood methodology. PSNM kinds the computer files by exploitation the sorting key. It compares record solely among a window that is in sorted order the most intention is that records that are pass on sorted order are a lot of doubtless to be duplicates than the records that are way apart. PSNM, it additionally pre-sorts the records to use their rank-distance during sorting for similarity estimation.

Algorithm 1 :

Progressive Sorted Neighborhood Method

Require: dataset reference D, sorting key K, window size
W, enlargement interval size I, number of records N

1: procedure PSNM(D, K, W, I, N)
2: pSize calcPartitionSize(D)
3: pNum dN=(pSize □ W + 1)e
4: array order size N as Integer
5: array recs size pSize as Record
6: order sortProgressive(D, K, I, pSize, pNum)
7: for currentI 2 to dW=Ie do
8: for currentP 1 to pNum do
9: recs loadPartition(D, currentP)
10: for dist 2 range(currentI, I, W) do
11: for i 0 to jrecsj □ dist do

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

12: pair hrecs[i]; recs[i + dist]i
13: if compare(pair) then
14: emit(pair)
15: lookAhead(pair)

4.2 Secure Hashing Algorithm

In duplicate detection method whenever any record comes for server, it calculates the hash for the record exploitation secure hash algorithm (SHA). Hash is a fixed length representation of arbitrary length message. Hash function is any function that can be used to map data of arbitrary size to data of fixed size. The values returned by a hash function are called as hash values. A hash value is a numeric value of a fixed length uniquely identifies data .hash value represent large amount of data.

Secure hash Algorithm is a Collision free and impossible to re-create the same message.Hash function produce hash value called as message digests.

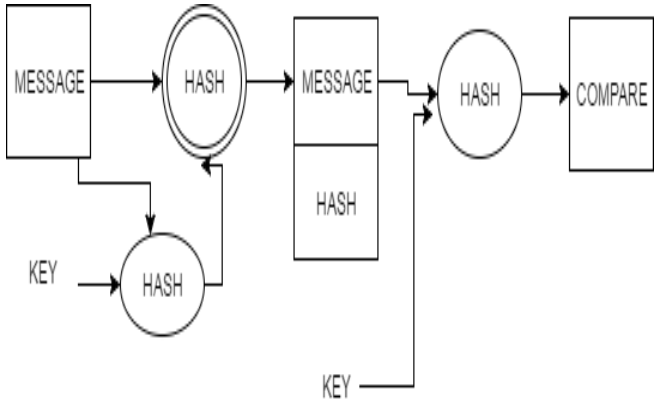

Figure4.1: Basic hash function diagram

Algorithm 2 :

Secure Hashing Algorithm

SHA-256

SHA-256 operates in the manner of SHA-1: The message to be
hashed is first
(1) Padded with its length in such a way that the result is a multiple of 512 bits long, and then
(2) Parsed into 512-bit message blocks $M^{(1)}; M^{(2)}; \ldots ; M^{(N)}$
The message blocks are processed one at a time;
Beginning with a fixed initial hash value $H^{(0)}$;
Sequentially compute

$$H^{(i)} = H^{(i-1)} + C_M{}^{(i)} (H^{(i-1)});$$

Where C is the SHA-256 compression function and + means word-wise mod 2^{32} addition.
$H^{(N)}$ is the hash of M.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

SHA-512

SHA-512 is a variant of SHA-256 which operates on eight 64-bit words. The message to be hashed is first-:

(1) Padded with its length in such a way that the result is a multiple of 1024 bits long, and then
(2) parsed into 1024-bit message blocks $M^{(1)}; M^{(2)}; \ldots; M^{(N)}$.
The message blocks are processed one at a time,
Beginning with a fixed initial hash value $H^{(0)}$,
Sequentially compute

$$H^{(i)} = H^{(i-1)} + C_M{}^{(i)}(H^{(i-1)});$$

Where C is the SHA-512 compression function and + means word-wise mod 2^{64} addition.
$H^{(N)}$ is the hash of M.

A cryptographic hash function is a special class of hash function that has certain properties which make it suitable for use in cryptography. It is simply a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size (a hash function) which is designed to also be a one-way function, that is, a function which is infeasible to invert. The only way to recreate the input data from an ideal cryptographic hash function's output is to attribute of possible inputs to see if they produce a match, The input data is often called the message, and the output (the hash value or hash) is often called the message digest or simply the digest.

The ideal cryptographic hash function has five main properties:

1. It is deterministic so the same message always results in the same hash
2. It is quick to compute the hash value for any given message
3. It is infeasible to generate a message from its hash value except by trying all possible messages
4. A small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value
5. It is infeasible to find two different messages with the same hash value

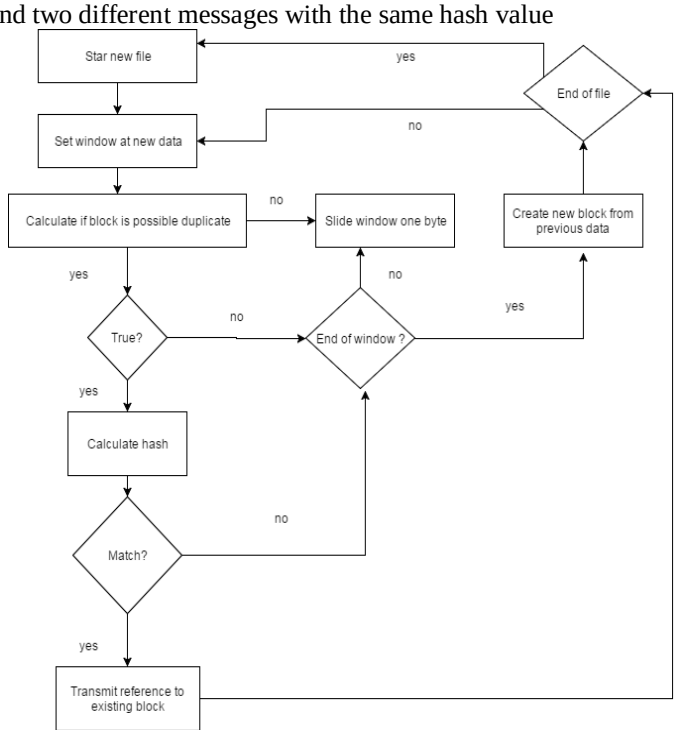

Figure 4.2: Flow Diagram for calculating Hash function

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Two documents are regarded as duplicates if they comprise identical document content. Documents that bear small dissimilarities and are not identified as being "exact duplicates" of each other but are identical to a remarkable extent are known as near duplicates.

- Files with a few different words - widespread form of near-duplicates

The most challenging from the technical perspective, is small differences in content.

V. RESULT AND ANALYSIS

In the previous sections, progressive sorted neighborhood method for detecting and deleting duplicate files in client server environment and secure hashing technique for finding the duplicate content inside the file is presented. In this section, the performance of proposed approach is evaluated.

5.1 File based duplicate detection

Comparing to the related work progressive sorted neighborhood method applies best for finding duplicate data of different datasets. Files inside any folder or drive are taken into consideration for finding the duplicate files and deleting those files to reduce storage space of system and to increase the reliability.

Tabular and graphical representation is given below for file based approach for finding duplicate files inside the folder or drive.

Total Files	Duplicate Files
911	327

Table 5.1: Duplicate Files Found

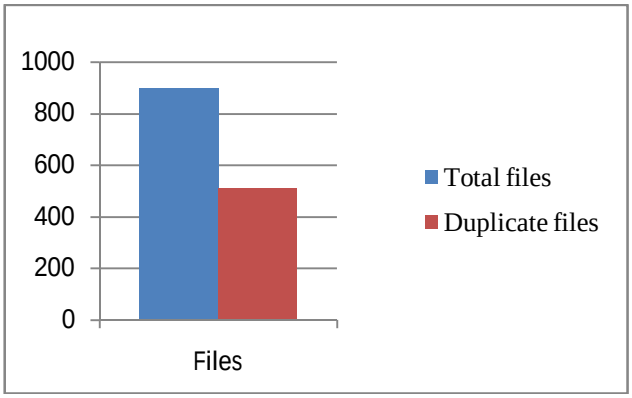

Figure 5.1: File based approach

On the basis of NAME, SIZE and MODIFIED DATE duplicated files are identified and deleted. Time required to scan the drive is in milliseconds and is negligible compared to the efficient results for finding out the duplicated files in client server system.

5.2 Content based duplicate detection

Second task is to identify duplicate content inside the file which is done by using the technique that is secure hashing. SHA-256 and SHA-512 these two highly efficient algorithms are taken into consideration for better results. Comparative analysis is shown with graphical representation below.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Files	SHA-256	SHA-512
File 1	73 %	75 %
File 2	81 %	83 %
File 3	81 %	77 %

Table 5.2: Original Content found

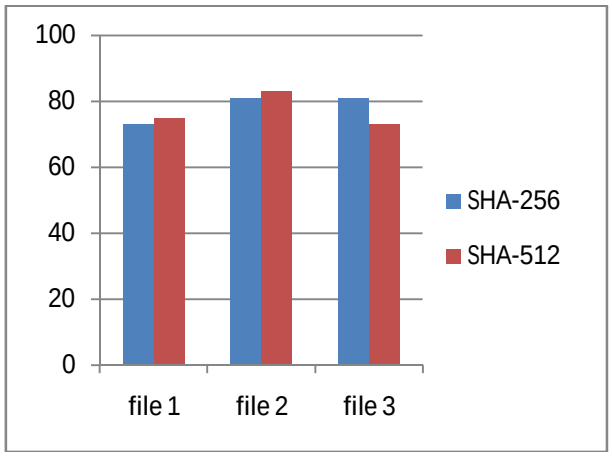

Figure 5.2: % of original content

Lower % of original content found gives the higher % duplicate content found. SHA-512 hashes the data in 1024 block size which gives better performance for larger file size. Where, SHA-256 divides the chunk into 512 block size which gives better performance for smaller file size.

Files	SHA-256	SHA-512
File 1	27 %	25 %
File 2	19 %	17 %
File 3	19 %	23 %

Table 5.3: Duplicate Content found

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

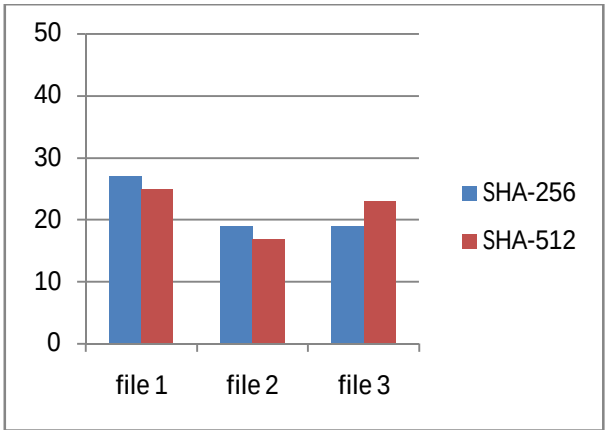

Figure 5.3: % of duplicate content

Both the algorithms SHA-256 and SHA-512 can be used to detect duplicate content inside the file according to the file sizes smaller and larger.

Files	SHA-256	SHA-512
File 1	1400	1050
File 2	1100	9450
File 3	8400	9440

Table 5.4: Hashing Time (ms)

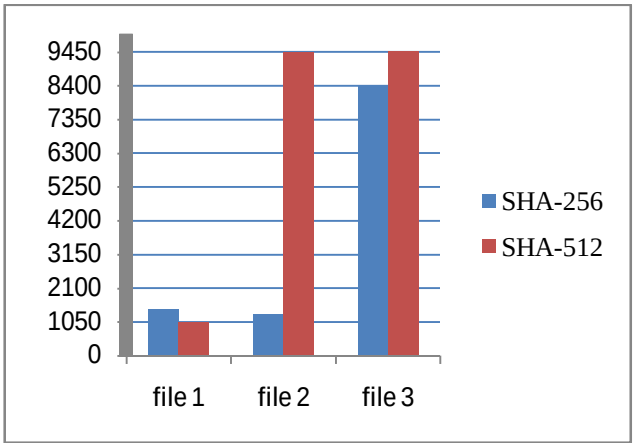

Figure 5.4: Hashing time (ms)

The SHA-256 algorithm takes as an input a message with maximum length of less than 2^{64} bits and produces as output a 256-bit message digest. The input is processed in 512-bit blocks. The probability of collision is more in the case of SHA-256 algorithm with 2^{128} in case of security.

The SHA-512 algorithm takes as an input a message with maximum length of less than 2^{128} bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks. The probability of collision is very less in the case of SHA-512 algorithm with 2^{256} in case of security.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

SHA-256 hash the data faster for small files and SHA-512 hash the data faster for large files as shown in figure 5.4

Experimental Result:

Proposed duplicate detection system is tested on client-server environment and following is the Execution Environment

1. Windows 7(64 bit)
2. Intel Dual core i3 processor (2.53GHz)
3. Windows server 2008
4. 3 GB RAM
5. Router
6. Crimping LAN cable

Sometimes, the results may vary depending upon Execution Environment.

VI. CONCLUSION AND FUTURE WORK

The proposed system introduces progressive sorted neighborhood method and secure hashing techniques for detecting duplicate data effectively. The PSNM method significantly increases the efficiency of finding duplicate files in shorter time with higher accuracy. It maximizes the gain of overall process within the time available by reporting most results much earlier than traditional approaches.

According to the result and analysis, for content based approach SHA-256 gives better results for smaller files and SHA-512 gives better results for larger files based on the % of duplicate content found and the hashing time required to hash the data divided into the chunk size of 512 and 1024 respectively.

In future work, there is scope for finding out duplicate data in multimedia storage that is audio, video. For finding duplicate data combination of progressive approach with hashing techniques can be taken into consideration for better results using this comparative analysis of hashing techniques, which can also help to increase the scalability of the duplicate detection system.

REFERENCES

1. Miss. Ruchira Dhananjay Deshpande, Sonali Bodkhe, An Efficient Approach towards Duplicate Detection System, in International Journal for Research in Applied Science & Engineering Technology, 2017
2. Ruchira Deshpande, Sonali Bodkhe. Implementation of an Efficient Approach for Duplicate Detection System in International Conference On Emanations in Modern Engineering Science and Management (ICEMESM-2017) Volume: 5 Issue: 3.
3. Thorsten Papenbrock, Arvid Heise, and Felix Naumann, Progressive Duplicate Detection, IEEE Transactions on Knowledge And Data Engineering, Vol. 27, NO. 5, MAY 2015
4. L. Kolb, A. Thor, and E. Rahm, Parallel sorted neighborhood blocking with mapreduce, in Proceedings of the Conference Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), 2011.
5. S. E. Whang, D. Marmaros, and H. Garcia-Molina, Pay-as-you- go entity resolution, IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 25, no. 5, 2012.
   A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, Duplicate record detection: A survey, IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 19, no. 1, 2007.
6. F. Naumann and M. Herschel, An Introduction to Duplicate Detection. Morgan & Claypool, 2010.
7. O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller, Framework for evaluating clustering algorithms in duplicate detection, in Proceedings of the International Conference on Very Large Databases (VLDB), 2009.
8. O. Hassanzadeh and R. J. Miller, Creating probabilistic databases from duplicated data, VLDB Journal, vol. 18, no. 5, 2009.
9. U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg, "Adaptive windows for duplicate detection," in Proceedings of the International Conference on Data Engineering (ICDE), 2012.
10. S. Yan, D. Lee, M. yen Kan, and C. L. Giles, Adaptive sorted neighborhood methods for efficient record linkage, in International Conference on Digital Libraries (ICDL), 2007.
11. J. Madhavan, S. R. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy, Web-scale data integration: You can only afford to pay as you go, in Proceedings of the Conference on Innovative Data Systems Research (CIDR), 2007.
12. S. R. Jeffery, M. J. Franklin, and A. Y. Halevy, Pay-as-you-go user feedback for dataspace systems, in Proceedings of the International Conference on Management of Data (SIGMOD), 2008.
13. C. Xiao, W. Wang, X. Lin, and H. Shang, Top-k set similarity joins,in Proceedings of the International Conference on Data Engineering (ICDE), 2009.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

14. P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 24, no. 9, 2012.B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz, The Plis
15. B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz, The Plista dataset, in Proceedings of the International Workshop and Challenge on News Recommender Systems, 2013.
16. Neha Kurav, Preeti Jain, A Parallel Architecture for Inline Data De-duplication SHA-2 Hash in Volume 5, Issue 4, April 2015 ISSN:2277 128X ijarcsse,2015.
17. Guiping Wang, Shuyu Chen, Jun Liu. A Network Differential Backup and Restore System based on a Novel Duplicate Data Detection algorithm in WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS.
18. Yueguang Zhu, Xingjun Zhang, Runting Zhao, Xiaoshe Dong. Data De-duplication on Similar File Detection in 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.
19. Deepali Choudhari, R. W. Deshpande. Deduplication Techniques in Storage System using International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 11, November 2015.
20. Wen Xia, Hong Jiang, Senior Member, IEEE, Dan Feng, Member, IEEE, and Yu Hua, Senior Member, IEEE . Similarity and Locality Based Indexing for High Performance Data De-duplication in ieee transactions on computers, vol. 64, no. 4, april 2015.