Volume 5 Issue I, January 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Efficient Approach towards Duplicate Detection System

Miss. Ruchira Dhananjay Deshpande[1], Sonali Bodkhe[2]
[1]M. Tech Student, [2]Assistant Professor
Department of Computer Science & Engineering,G. H. Raisoni Academy of Engineering and Technology, Nagpur, Maharashtra, India

Abstract: Information on the web is very huge in size and the tasks of search engines have become more and more complex as a single entity on the web have two or more representations in databases. The duplicate detection is the process of identifying the entities who has multiple representation of the same real world entity, as the duplicate detection methods has to process large datasets, the identification of duplicate document in a large database is a issue significantly with wide-spread applications. In this paper a review on various approaches of duplicate detection will be presented. Proposed system will compare two Duplication detection methods, the first is based on two novel progressive duplicate detection algorithms that significantly increase the efficiency of finding duplicates if the execution time is limited. The second is based on Secure Hashing Algorithm which will detect and delete duplicate data, the secure hash algorithm will perform data de-duplication task in order to overcome the issues of time and to reduce hash collision.
Keywords: De-duplication; Progressive sorted neighbourhood method; Progressive blocking; Hash; Parallel de-duplication.

I. INTRODUCTION

In computing, data deduplication has become a very important process of data mining, it is a specialized process of data compression which eliminating duplicate copies of repeating data. Related and somewhat synonymous terms are intelligent (data) compression and single-instance (data) storage, large database storage, content delivery networks, blog sharing, news broadcasting and social networks as ascendant part of Internet services are data centric.

Hundreds of millions of users of these services generate petabytes of new data every day. Databases play an important role in today's IT based economy. Many industries and systems depend on the accuracy of databases to carry out operations.

Therefore, the quality of the information stored in the databases, can have significant cost implications to a system that relies on information to function and conduct business.

In an error-free system with perfectly clean data, the construction of a comprehensive view of the data consists of linking in relational terms, joining two or more tables on their key fields. Unfortunately, data often lack a unique, global identifier that would permit such an operation.

Furthermore, the data are neither carefully controlled for quality nor defined in a consistent way across different data sources.

Thus, data quality is often compromised by many factors, including data entry errors (e.g.,studant instead of student), missing integrity constraints , and multiple conventions for recording information To make things worse, in independently managed databases not only the values, but the structure, semantics and underlying assumptions about the data may differ as well.

Progressive duplicate detection algorithms 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.

In progressive duplicate detection algorithms, two dynamic progressive duplicate detection algorithms, PSNM and PB, will be implemented which expose different strengths.

Introduces 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.

The following architecture shows the generalize scenario for de-duplication.

Volume 5 Issue I, January 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

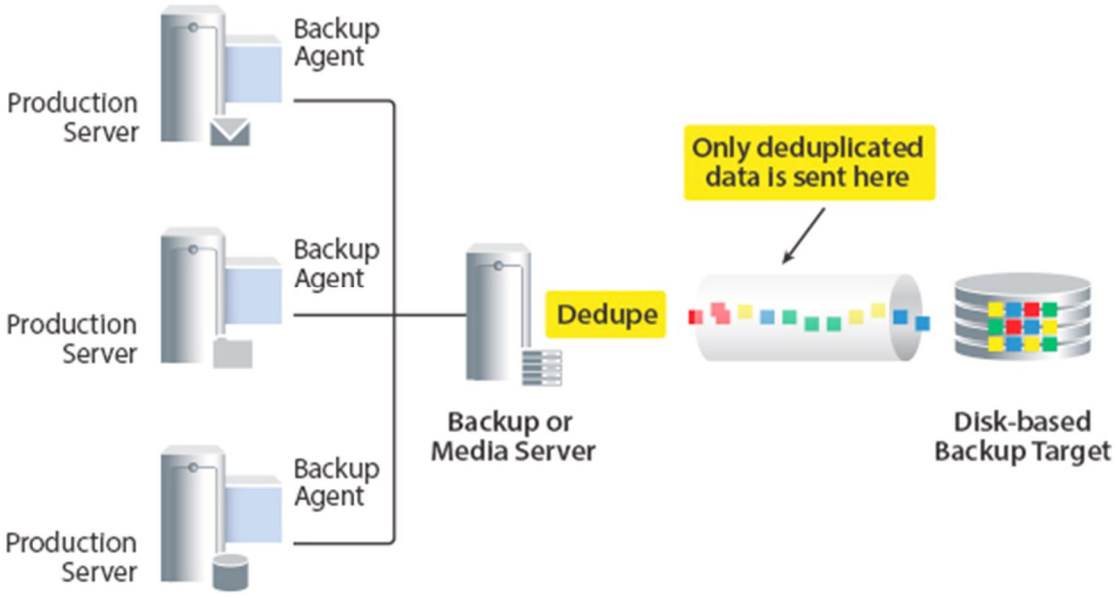

Figure1. Architecture for De-duplication

In contrast to progressive duplicate detection the cryptographic hashing is another concept in detection and deleting redundant data. In backup servers hash is used for finding the duplicate data. Hash is a fixed length representation of any arbitrary length message. The complexity of comparisons can be reduced by using hash as the original length of data is much more than the hash size. In de-duplication process whenever any record comes for server, it calculates the hash signature for the record using secure hash algorithm (SHA). Once hash signature is generated server checks this signature in hash index, which is already maintained in the system. While searching for the signature in hash index if the server finds its entry in the hash index (record already exists) then rather storing it again server creates a reference for this. This reference will point to the location of block on the disk. In second case if server does not find the entry of record in hash index table it will store the record on the disk and adds an entry for its hash signature in hash index.

II. RELATED WORK

Two novel, progressive duplicate detection algorithms 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].

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.

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 € U* in the graph represents a record in the data and an edge (*u; v*) *€ V* exists only if the two records are deemed similar. In these join techniques, two records are 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 deduplication 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 deduplication 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.

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.

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 analyzed 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

III. PROPOSED WORK

Proposed work is based on comparative study between previous progressive duplicate detection algorithms and cryptographic

Hashing for duplication Detection and deletion.

A. Progressive duplicates detection algorithms

Implementing two dynamic progressive duplicate detection algorithms, PSNM and PB, this will expose different strength**.** Progressive duplicate detection algorithms 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. It introduces a concurrent progressive approach for the multi-pass method and adapts an incremental transitive closure algorithm that together forms the first complete progressive duplicate detection workflow.

Progressive sorted neighborhood method is based on traditional sorted neighborhood method. PSNM sorts the input data by using the sorting key. It compares record only within a window which is in sorted order. The main intention is that records that are close in sorted order are more likely to be duplicates than the records which are far apart.

In contrast to windowing algorithms, blocking algorithms assign each record to a fixed group of similar records (the blocks) and then compare all pairs of records within these groups. Progressive Blocking (PB) is a novel approach that builds upon an equidistant blocking technique and the successive enlargement of blocks. Like PSNM, it also pre-sorts the records to use their rank-distance in this sorting for similarity estimation. Based on the sorting, PB first creates and then progressively extends a fine-grained blocking.

B. Cryptographic hashing

It is another concept in detection and deleting redundant data. In backup servers hash is used for finding the duplicate data. Hash is a fixed length representation of any arbitrary length message. The complexity of comparisons can be reduced by using hash as the original length of data is much more than the hash size.

In de-duplication process whenever any record comes for server, it calculates the hash signature for the record using secure hash algorithm (SHA). Once hash signature is generated server checks this signature in hash index, which is already maintained in the system. While searching for the signature in hash index if the server finds its entry in the hash index (record already exists) then rather storing it again server creates a reference for this. This reference will point to the location of block on the disk. In second case if server does not find the entry of record in hash index table it will store the record on the disk and adds an entry for its hash signature in hash index.

IV. CONCLUSION

The Study will help to choose the best duplicate detection Algorithm for designing a data deduplication framework that will help to improve early quality. The problem of data deduplication on large datasets is addressed in an effective manner .The Secure Hash Algorithm is proposed to outperform the previously proposed algorithm that is namely PSNM and PB which dynamically adjust their behaviour by automatically detect and delete duplicate data gives broader view towards solving the problem of deduplication on large datasets.

REFERENCES

[1] Thorsten Papenbrock, Arvid Heise, and Felix Naumann, Progressive Duplicate Detection, IEEE Transactions on Knowledge And Data Engineering, Vol. 27, NO. 5, MAY 2015
[2] 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
[3] 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.
[4] F. Naumann and M. Herschel, An Introduction to Duplicate Detection. Morgan & Claypool, 2010.
[5] 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.
[6] O. Hassanzadeh and R. J. Miller, Creating probabilistic databases from duplicated data, VLDB Journal, vol. 18, no. 5, 2009.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Transac

[13] tions on Knowledge and Data Engineering (TKDE), vol. 24, no. 9, 2012.
[14] 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.
[15] 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.
[16] A Parallel Architecture for Inline Data De-duplication SHA-2 Hash by Neha Kurav, Preeti Jain,Using Volume 5, Issue 4, April 2015 ISSN:2277 128X ijarcsse,2015.