A SURVEY ON DEDUPLICATION METHODS

A.FARITHA BANU[*1], C. CHANDRASEKAR[#2]
[*1]*Research Scholar in Computer Science*
[#2] *Assistant professor, Dept. of Computer Applications*
Sree Narayana Guru College
Coimbatore – 641105, TamilNadu, India.

ABSTRACT:

There is an increasing demand for systems that can provide secure data storage in a cost-effective manner. Having duplicate records occupies more space and even increases the access time. Thus there is a need to eliminate duplicate records. This sounds to be simple but requires an tedious work since duplicate records do not share a common key and/or they contain errors that make duplicate matching a difficult task. Errors are also introduced as the result of transcription errors, incomplete information, lack of standard formats, or any combination of these factors. Several approaches are proposed to eliminate duplicate data first at the file level and then at the chunk level to reduce the duplicate-lookup complexity. In this paper,few of the methods are discussed with its advantages and disadvantages. And also a better solution is proposed.

Key words: data storage, managing data, deduplication

I. INTRODUCTION

The quick expansion of information sources has necessitated the users to make use of automated tools to locate desired information resources and to follow and asses their usage patterns. Thus the need for building server side and client side intelligent systems that can mine for knowledge in a successful manner emerged. Deduplication is one such tasks which fastens the search and help producing efficient results.

Deduplication is a task of identifying record replicas in a data repository that refer to the same real world entity or object and systematically substitutes the reference pointers for the redundant blocks; also known as storage capacity optimization. As the volume of data in an organization grows, the amount of repeated data takes a toll on storage availability.

Using deduplication has two big advantages over a normal file system:

- Reduced Storage Allocation - Deduplication can reduce storage needs by up to 90%-95% for files such VMDKs and backups. Basically situations where you are storing a lot of redundant data can see huge benefits.

- Efficient Volume Replication - Since only unique data is written disk, only those blocks need to be replicated. This can reduce traffic for replicating data by 90%-95% depending on the application.

Generally deduplication is done in three ways. They are

1) *Chunking:*

Between commercial deduplication implementations, technology varies primarily in chunking method and in architecture. In some systems, chunks are defined by physical layer constraints or in few systems only complete files are compared, which is called Single Instance Storage or SIS. The most intelligent method to chunk is generally sliding-block in which a sliding block, a window is passed along the file stream to seek out more naturally occurring internal file boundaries.

2) *Client backup deduplication:*

This is the process where the deduplication hash calculations are initially created on the source machines and files that have identical hashes to files already in the target device are not sent. Thus the target device just creates appropriate internal links to reference the duplicated data. The benefit of this is that it avoids data being unnecessarily sent across the network thereby reducing traffic load.

3) *Primary storage and secondary storage:*

Primary storage systems are designed for optimal performance, rather than lowest possible cost. The design criteria for these systems are to increase performance, at the expense of other considerations. Moreover, primary storage systems are much less tolerant of any operation that can negatively impact performance. Also secondary storage systems contain primarily duplicate or secondary copies of data. These copies of data are typically not used for actual production operations and thus they are more tolerant of some performance degradation, in exchange for increased efficiency.

Whenever data is transformed, concerns arise about potential loss of data. By definition, data deduplication systems store data differently from how it was written. As a result, users are concerned with the integrity of their data. The various methods of deduplicating data all employ slightly different techniques. However, the integrity of the data will ultimately depend upon the design of the deduplicating system, and the quality used to implement the algorithms. This paper shares a few of the concepts briefly.

II.STUDIES ON DEDUPLICATION

A. Encrypting data:

Deduplication takes advantage of data similarity in order to achieve a reduction in storage space. Meanwhile the goal of cryptography is to make cipher text indistinguishable from theoretically random data. Thus, the goal of a secure deduplication system is to provide data security, against both nside and outside adversaries, without compromising the space efficiency[4]. First step is to find out the chunks in the documents. After identifying chunks of data both within and between file, they are encrypted using keys. Then deduplication exploits identical content by matching encrypted data. The resulting deduplication can yield cost savings by increasing the utility of a given amount of storage. The drawback is that if the same content is encrypted with two different keys it will results in very different cipher text. Thus it does not sound to be better approach.

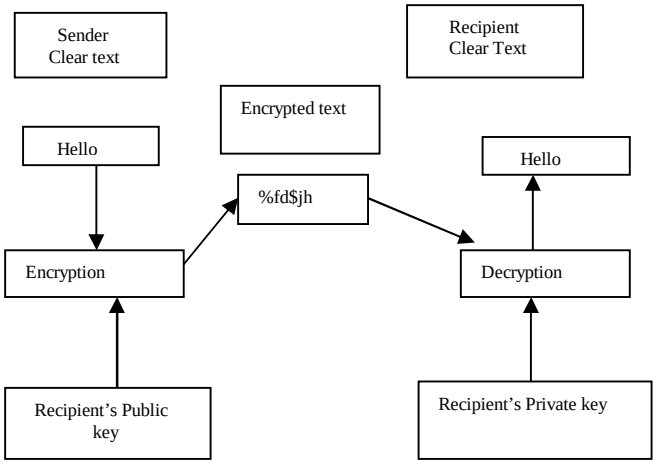

Fig 1: Concept of Encryption

B. Fast Dual-level Fingerprinting:

In this scheme [3], fingerprint datasets both at the file level and at the chunk level is obtained in a single scan of the contents. FDF breaks the dual-level fingerprinting process into task segments and three techniques to optimize the performance are employed.

- FDF utilizes Rabin's fingerprinting algorithm [5] to divide files into variable-sized chunks while capturing and eliminating hot *zero-chunks* simultaneously by judiciously selecting chunk boundaries.
- FDF employs SHA-1 algorithm to generate fingerprints for files and chunks and further defines *hash context* to preserve the intermediate result of the hash algorithm, so that the file-level hashing and the chunk-level hashing can be performed in parallel by sharing the same data cache.
- FDF resolves cache conflicts between different task segments and further pipeline the fingerprinting process by leveraging the computing resources of modern multi-core CPUs, thus the time overhead can be greatly reduced.

The fingerprints are compared to obtain the duplicate data records. Even though it reduces time, computation is more

since different algorithms are used at different steps and the result is to be stored which occupies more space.

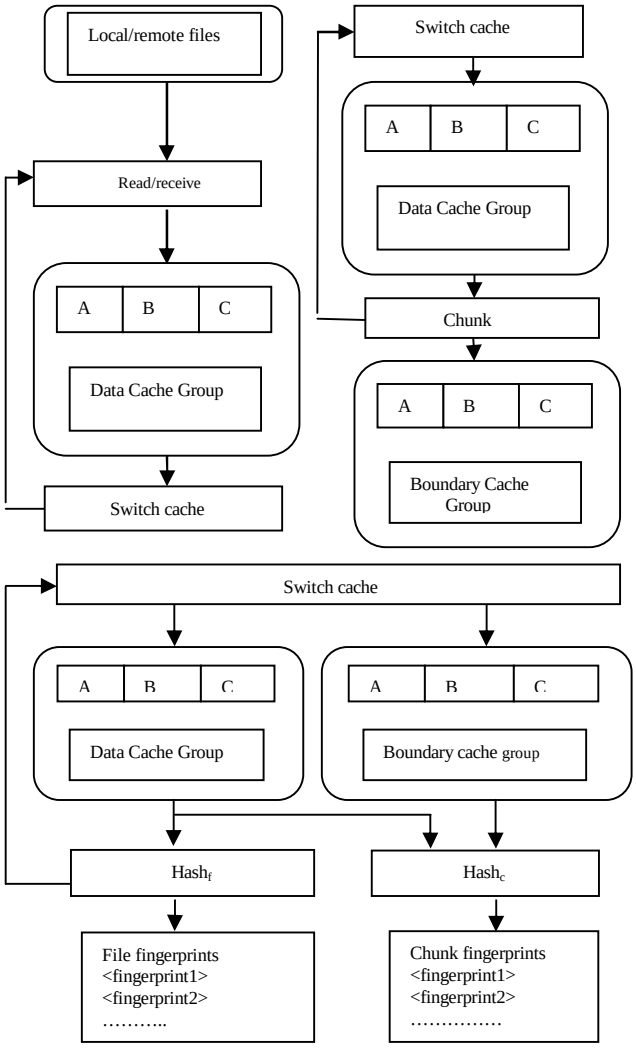

Fig 2: Parallelism of Fingerprinting Task Segments

C. Unsupervised Duplicate Detection:

UDD can effectively identify duplicates from the query result records of multiple Web databases for a given query it uses two classifiers. The WCSS classifier act as the weak classifier which is used to identify "strong" positive examples and an SVM classifier acts as the second classifier. First, each field's weight is set according to its "relative distance," i.e., dissimilarity, among records from the approximated negative training set. Then, the first classifier utilizes the weights set to match records from different data sources. Next, with the matched records being a positive set and the nonduplicate records in the negative set, the second classifier further identifies new duplicates. Finally, all the identified duplicates and non-duplicates are used to adjust the field weights set in the first step and a new iteration begins by again employing the first classifier to identify new duplicates. The iteration stops when no new duplicates can be identified. This method is well suited for only web based data but still it requires an initial approximated training set to assign weight.

Algorithm

1. D=0
2. Set the parameters W of c_1 according to N
3. Use C_1 to get a set of duplicate vector pairs d_1 from P
4. Use C_1 to get a set of duplicate vector pairs f from N
5. $P=P-d_1$
6. While $|d_1| \neq 0$
7. N'=N-f
8. $D=D + d_1+f$
9. Train c_2 using D and N'
10. Classify P using c_2 and get a set of newly identified duplicate vector pairs d_2
11. $P=P-d_2$
12. $D=D+d_2$
13. Adjust the parameters W of c_1 according to N' and D
14. Use C_1 to get a set of duplicate vector pairs d_1 from P
15. Use C_1 to get a set of duplicate vector pairs f from N
16. N=N'
17. Return D

Fig 3: Algorithm for UDD Duplicate Detection

Data Generation Technique:

Publicly available test data [7] with known deduplication or linkage status is used to check the new linkage algorithms and techniques which are latter evaluated and compared. However, publication of data containing personal information is impossible due to privacy and confidentiality issues. So an alternative is to use artificially created data. In these data, content and error rates can be controlled, and the deduplication or linkage status is known. Artificially

generated data seems to be an attractive alternative. It models the content and statistical properties of comparable real world data sets, including the frequency distributions of attribute values, error types and distributions, and error positions within these values.

The first step is to develop a data generator. A data generator creates data sets containing names and addresses, dates, telephone and identifier numbers. Then the original records are created by collecting details from various data sources. These original records are used as reference to identify the duplicate copies. Clearly this technique requires a prior work to develop original records which increases the overhead.

D. Interactive Deduplication Using Active Learning:

The main challenge in deduplication task is to find a function that can resolve when two records refer to the same entity in spite of errors and inconsistencies in the data. A learning based deduplication system ALIAS2 allows automatic construction of the deduplication function by interactively discovering challenging training pairs. The idea is to simultaneously build several redundant functions and the disagreement amongst them to discover new kinds inconsistencies amongst duplicates in the dataset are exploited. An active learner actively picks subsets of instances which when labeled will provide the highest information gain to the learner. The difficult task of bringing together the potentially confusing record pairs is automated by the learner. The user has to only perform the task of labeling the selected pairs of records as duplicate or not.

Architecture:

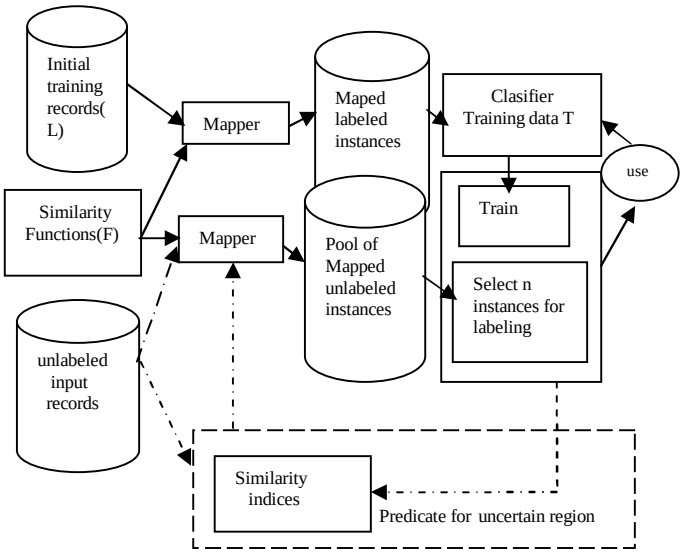

Fig 4: Architecture of Interactive Deduplication

Database of records (D):

The original set D of records in which duplicates need to be detected. The data has d attributes $a_1,\ldots a_d$, each of which could be textual or numeric. The goal of the system is to obtain the subset of pairs in the cross-product D × D that can be labeled as duplicates.

Initial training pairs (L):

An optional small seed L of training records arranged in pairs of duplicates or non-duplicates.

Similarity functions (F):

A set F of n_f functions each of which computes a similarity match between two records r_1, r_2 based on any subset of d attributes.

Steps:

At first map the initial training records in L into a pair format through a mapper module. The mapper module takes as input a pair of records r_1, r_2, computes the n_f similarity functions F and returns the result as a new record with n_f attributes. For each duplicate pair, assign a class-label of "1" and for all the other pairs in L × L, assign a class label of "0". At the end of this step a mapped training dataset L_p ia obtained. These L_p instances are used to initialize the learning component of the system.

1. Input: L, D and F.
2. Create Pairs Lp from the labeled data L and F.
3. Create pairs Dp from unlabeled data D and F.
4. Initial training set T = Lp.
5. Loop until user satisfaction
   - Train classifier C using T.
   - Use C to select a S of n instances from Dp for labeling.
   - If S is empty, exit Loop.
   - Collect user feedback on the labels of S.
   - Add S to T and remove S from Dp.
6. Output classifier C.

Fig 5: Steps Used In Interactive Deduplication

E. Genetic programming:

To deal with the deduplication problem, an approach based on Genetic programming [6] is used. This approach combines several different pieces of evidence extracted from the data content to produce a deduplication function that is able to identify whether two or more entries in a repository are replicas or not. Record deduplication is a time consuming process hence make out duplication function for small repository so that the resulting function can be applied to other areas. The resulting function should be able to efficiently maximize the identification of record replicas.

The steps of Genetic algorithm are the following:

1. Initialize the population (with random or user provided individuals).
2. Evaluate all individuals in the present population, assigning a numeric rating or fitness value to each one.
3. If the termination criterion is fulfilled, then execute
4. The last step. Otherwise continue.
5. Reproduce the best n individuals into the next generation population.
6. Select m individuals that will compose the next generation with the best parents.
7. Apply the genetic operations to all individuals selected. Their offspring will compose the next population. Replace the existing generation by the generated population and go back to Step 2.
8. Present the best individual(s) in the population as the output of the evolutionary process.

The deduplication function obtained using genetic approach is implemented in servers or client side system so that data which is to be stored are verified before they reach the database. The function is automatically generated which reduces the human effort. When a new document is to be stored, function is derived from the evidences collected from it and it is used to check for duplicate availability in data storage. If found a match, the duplicate copy is prevented from storing again. This technique is efficient and produces a outcome.

III.CONCLUSION

Duplicate detection and removal is an important problem in data cleaning, and an adaptive approach that learns to identify duplicate records and removing them has clear advantages. In this paper, surveys of various techniques for deduplication which are previously defined are discussed. The performance of the outcome can be still improved by using PSO algorithm for generating deduplication function. PSO appears similar to GA in term of its selecting strategy of the best child (or the best swarm), but it is really different. It utilizes the intercommunication between each individual swarm with the best one to update its position and velocity. This algorithm operates with randomly created population of potential solutions and searches for the optimum value by creating the successive population of solutions PSO reduces complexity by using simpler terms of computations because its crossover and mutation operation are done simultaneously. Thus a better solution to the deduplication problem will be provided by using PSO generated deduplication function.

IV. REFERENCES

[1]R.A. Baeza-Yates and B.A. Ribeiro-Neto, Modern Information Retrieval. ACM Press/Addison-Wesley, 1999.
[2]R. Bell and F. Dravis, “Is You Data Dirty? and Does that Matter?,” Accenture Whiter Paper, http://www.accenture.com, 2006.
[3]Jiansheng Wei,*1*Ke Zhou, *2*Lei Tian, *1*Hua Wang, Dan Feng,” A Fast Dual-level Fingerprinting Scheme for Data Deduplication“
[4]Mark W. Storer Kevin Greenan Darrell D. E. Long Ethan L. Miller,” Secure Data Deduplication”
[5]Michael O. Rabin, "Fingerprinting by random polynomials", Technical Report, No. TR-15-81, Center for Research in Computing Technology, Harvard University, Cambridge, MA, USA, 1981.
[6]Moise´s G. de Carvalho, Alberto H.F. Laender, Marcos Andre´ Gonc¸alves, and Altigran S. da Silva.” A Genetic Programming Approach to Record Deduplication”
[7]Peter Christen.”Probabilistic Data Generation for Deduplication and Data Linkage”, http://datamining.anu.edu.au/linkage.html.
[8]Weifeng Su, Jiying Wang, and Frederick H. Lochovsky, “Record Matching over Query Results from Multiple Web Databases”, IEEE Transactions On Knowledge And Data Engineering, VOL. 22, NO. 4, APRIL 2010