International Journal For Technological Research In Engineering
Volume 4, Issue 5, January-2017 *ISSN (Online): 2347 - 4718*

SCALABLE APPROACHES FOR RAPID EXECUTION IN THE DUPLICATE DETECTION

Guntuku. Amani[1], Ms.Shaik Shafia[2]
[1]M. Tech Student, [2]Associate Professor
Department of CSE, Hyderabad Institute of Technology and Management,Gowdavelly(V), Medchal(M), RangaReddy(DIST), Telangana, India.

ABSTRACT: In the Data mining, during the processes of data cleaning, duplicate detection is one phase. At present, users wants to process the larger datasets in the less time and that not possible in the existing system. To find the duplicates in the datasets existing, number of methods are there but those are not time efficient and users cannot get accurate data results. In existing, we used two methods namely, 1) Progressive Sorted neighborhood Method (PSNM), 2) Progressive Blocking (PB) Method. These two methods are providing the good quality in duplicate detection but those are not time efficient. To solve this disadvantage, in this paper we propose a time efficient parallel processing method. This method extended by traditional progressive sorted neighborhood method only. The proposed parallel Processing method can find the duplicate data faster than the existing methods. In experiments, we can observe the processing time of the parallel method and normal method.

KEYWORDS: Duplicate Detection, Magpie Sorting, Parallel Processing, Sorted Neighborhood Method.

I. INTRODUCTION

Whenever the duplicates need to be found from dataset we tend to select data processing. The information mining takes its 'concepts from information Discovery in info (KDD) within the pasture of engineering. Within the recent past, duplication is altering into a significant threat in the majority the domains. As a result of this duplication the information received is additional and therefore memory constraint becomes demanding. Therefore admin finds it troublesome to manage the information sets. The duplicate detection processes are dear. The people keep dynamic their portfolio despite retailers providing several product catalogs. Now-a-days, Databases play a primary role in IT situated economy. Many industries as well as systems rely on the accuracy of databases to carry out operations. As a result, the worth of the data will be saved in the databases; can have significant price suggestions to a system that relies on data to operate and perform business. In an error-free system with exactly clean data, the construction of a comprehensive view of the information contains linking --in relational phrases, joining--two or more tables on their key fields. Unfortunately, information most commonly needs a unique, world identifier that may permit such an operation. Furthermore, the information is neither cautiously controlled for outstanding nor defined in a consistent means throughout distinctive data sources. Accordingly, information quality is frequently compromised by using many causes, together with knowledge entry errors (e.g., studet as an alternative of student), missing integrity constraints (e.g., enabling entries), and more than one conventions for recording information To make things poorer, in independently managed databases not most effective the values, but the constitution, semantics and underlying assumptions about the data could vary as well. The Progressive techniques may method larger dataset in brief span of time and also the quality of knowledge is additionally smart relatively. The Progressive duplicate detection makes it totally different from the normal approach by yielding additional advanced results throughout the first termination; the algorithms of duplicate detection additionally compute the duplicates at a virtually constant frequency however the progressive algorithms increase the time because it finds out the duplicates at the first stage itself. The candidate keys within the record pairs that are identical need to be first discerned. The combine choice techniques of the duplicate detection method exhibits a trade-off between the amounts of your time required to run a reproduction detection rule and also the completeness of the results. This trade-off is created additional efficient by the progressive detection techniques because it computes the leads to shorter quantity of your time. Typically the duplication may even be performed taking under consideration the window size. To avoid a prohibitively dear comparison of all pairs of records, a standard system is to cautiously partition the records into smaller subsets and therefore fitting them to a specific partition. If similar records appear within the same partition and at intervals identical window, then the information is said duplicate. If the window size is chosen too little, some duplicates may well be lost. If the window size is chosen massive enough to search out all duplicates even for the most important cluster, then there are lots of gratuitous comparisons within the space of the smaller clusters. The variability of parameters that need to be set by a user is therefore advanced. The proposed system enhances the strength of duplicate detection even on very massive datasets. The parameterization complexness for duplicate detection is created comfortable generally and contributes to the event of additional user interactive applications. Progressive methods make this exchange-off extra invaluable as they deliver more whole outcome in shorter quantities of time. Revolutionary Sorted nearby procedure take smooth dataset and find some replica files and progressive blocking take dirty datasets and realize significant replica files in databases. Eventually, in this paper we propose parallel processing method and our work extends by these sorting methods.

II. RELATED WORK

The sorted Neighborhood process depends on the assumption that replica records can be close in the sorted record, and accordingly shall be when compared for the duration of the merge step. The effectiveness of the sorted neighborhood strategy is totally dependent upon the contrast key that's selected to sort the records. Typically, no single key shall be plenty to sort the documents in this sort of approach that all the matching files may also be detected. If the error in a file occurs within the unique discipline or element of the subject that's the fundamental a part of the sorting key, there's a very small probability that the file will turn out to be practically an identical record after sorting. To expand the quantity of identical records merged, Herna´ndez and Stolfo carried out a approach for executing a couple of independent runs of the Sorted-Neighborhood Method by means of using yet another sorting key and a slightly small window every time. This process is known as the multi-pass technique. This method is similar in spirit to the multiple-run blocking approach described above. Each impartial run produces a collection of pairs of documents that can be merged. The final outcomes, including the transitive closure of the files matched in extraordinary passes, are due to this fact computed.

A. Map-Reduce Algorithm

A map reduced algorithm was introduced which has high affability for scheduling about responsibilities for dynamic load balancing. The author Oktie, presents the Stringer framework that gives an evaluation arrangement to understanding what hindrances remain towards the objective of really flexible as well as broadly useful duplication recognition calculations. Few unrestrained bunch algorithms are assessed for copy discovery by broad examinations over totally different arrangements of string information with numerous attributes. A theme was introduced to combine multisource data. The results from the preliminary examinations are according that was taken from four card inventory databases that rescale to over ten million records are according within the paper.

B. Sorted Neighborhood Method with Map-Reduce

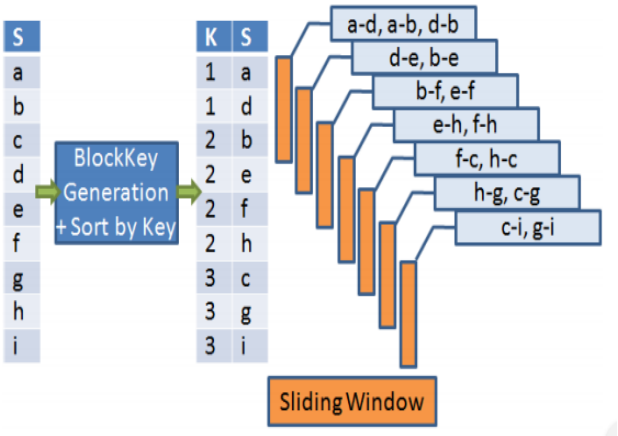


Fig 1. Example Execution of SNM

Sorted Neighborhood Method (SNM) is a popular blocking approach that works as follows;

A blocking key K is determined for each of n entities. Generally the concatenated prefixes of a few attributes form the blocking key. Afterwards the entities are sorted with the aid of this blocking key. A window of a fixed size w is then forward over the sorted records & in each step all entities within the window, i.e., entities inside a distance of w−1, are when put next. Above figure shows a SNM example execution for a window size of w = 3. This is the time consuming process.

Drawbacks of Traditional Methods:

- These adaptive approaches dynamically give a boost to the efficiency of duplicate detection, however unlike our revolutionary procedures, they have got to run for unique durations of time as well as can't exploit the affectivity for any given time slot.
- Wants to method giant dataset in less time
- Quality of dataset turns into increasingly complex

III. FRAMEWORK

A. Duplicate Detection Architecture

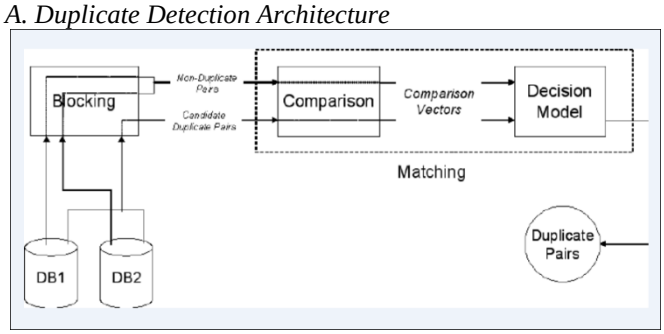


Fig3. Duplicate Detection System

For instance, if we take an online shopping database, in that numbers of catalogues are there and number of employees is enter the data into the database. So, there is possible to enter the same data number of times. That is referred as duplicate data. If this duplicate data is increased in the database then there is no space for other information means here reduces the storage space of the database. This is the major problem of duplicate data. To overcome this problem we have various approaches but those are not efficient as well as they are time consuming approaches. In fig3, first we are collecting the complete data from databases. After that, we need to pair that data and compare those pairs. Which pairs are duplicates those duplicates are clustered into a group. Like this we can detect and remove the duplicate data. The main objective of this paper is to detect duplicate data and count the duplicates in the large datasets within the less time. For that in this paper, we propose new methods to detect the duplicate data as well as count the duplicates in the complete dataset as a parallel. In existing method we got the good quality deduplicated data but it is very time consuming. Hence, we propose progressive and parallel method.

B. Sorting Key Selection

In this project we are sorting the dataset by using the magpie sorting. In these sorting methods, we need to select the sorting key to sort the dataset through that key. Importance of this sorting key is, we are mostly applying these two algorithms on the large datasets means those are in thousands

and lakh of records are stored in the dataset. But, sometimes user needs deduplicate and detect the duplicate count on only particular data. This type of situations, we need a sorting. Without sorting key it is difficult to sort the data from dataset. For selecting the sorting we propose an Attribute Concurrency method. Through this method we can select the best key for sorting. An attribute concurrency method works based on the multi-pass execution method. This multi-pass method executes the multiple keys in each pass. Attribute Concurrency method we apply to the progressive sorted neighborhood method as well as progressive blocking.
To perform this algorithm we have some techniques,

- Window Interval
- Partition Caching
- Magpie Sorting
- Pair Selection and Pair Comparison

a. Window Interval:
In this technique, our proposed system needs to load all the records into the every iteration and load the partition. Here, the partition is nothing but a window. Through this technique, we can define that the how many iterations are PSNM execute on each and every loaded partitions. And the main thing in this is the window size is not constant.
b. Partition Caching:
Through this technique we can reduce the reading burden of the loaded iterations it means, If a partition is read for the first time, the function gathers the requested files from the enter dataset and materializes them to a new, committed cache file on disk. When the partition is later requested next time, the function loads it from this cache file, decreasing the price for PSNM's extra I/O operations.
c. Magpie Sorting:
This is similar to the selection sort but, in selection sort we cannot select any sorting keys and in this magpie sorting we must select the best key as a sorting key.
d. Pair Selection and Pair Comparison:
Pair selection means, in every window it will pair the records and after pairing, compare the pairs. After comparison we can get the duplicate data.

C. Parallel Processing Method
Parallel processing means we execute the number of processes at a time that means parallel; this parallel processing is caused by using some concurrency methods. In this method first we are partition the complete dataset. These concurrency methods execute the all partitions of the dataset at a time to decrease the execution time of the process. This proposed method chooses the sorting key from dataset by using attribute concurrency technique. As well it also takes the window/block size to partition the complete dataset. Basically, our proposed system extended by traditional Progressive Sorted Neighborhood Method (PSNM) as well as Progressive Block (PB) for that reason we require to give the partition size as window size. Based on these sorting key as well as window size, the parallel processing method run the all partitions of the dataset and it also display the parallel processing time of the proposed method.

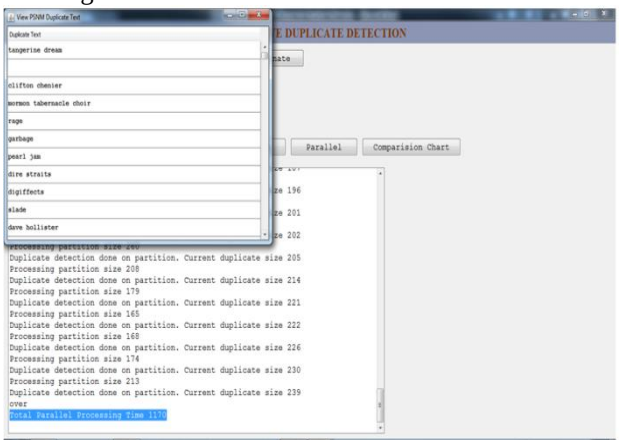
IV. EXPERIMENTAL RESULTS

In our experiments, we are detecting the duplicates on the large dataset by using parallel processing method. The first step is in our experiment are we needs to upload the dataset into the system. After upload dataset, we must select the sorting key and the window/block size. This window/block size used to partition the complete dataset and it is calculated by this formula:

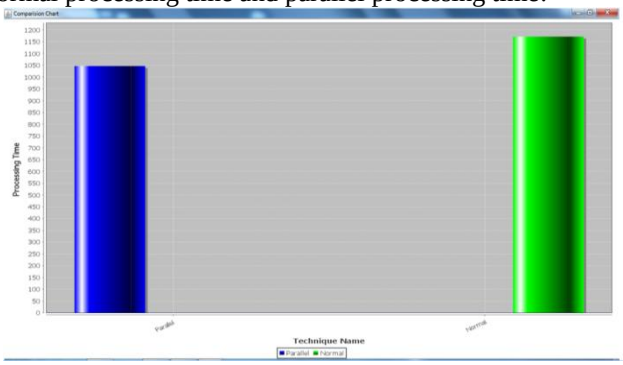
Partitions of Complete Dataset=Dataset Size/

(Window/block size)

By implementing this formula, partitions size will be displayed and every partition size along with it duplicates size we can view in the system. Lastly, we get the processing time of the algorithms also it displays. Here, we perform the traditional PSNM algorithm as well as traditional PB algorithm to verify the processing time our proposed parallel processing method.



The above screen shows that the processing time of the parallel method.
The below screen shows that the comparisons chart for normal processing time and parallel processing time:



From our experiments, we can say that our proposed parallel processing method is a time efficient method to find the duplicates.

V. CONCLUSION

In this paper we proposed a time efficient and parallel processing method. This proposed method inspired by the traditional PSNM as well as PB algorithms. In our proposed method we can get the duplicate detection time, duplicate

count as well as duplicate text. In this experiment we used large dataset & we detect the duplicate count as well as duplicate text within the milliseconds of time. Eventually, we proved that our proposed method is time efficient than the traditional algorithms.

REFERENCES

[1] K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," IEEE Trans. Knowl. Data Eng., vol. 19, no. 1, pp. 1–16, Jan. 2007.

[2] S. E. Whang, D. Marmaros, and H. Garcia-Molina, "Pay-as-you-go entity resolution," IEEE Trans. Knowl. Data Eng., vol. 25, no. 5, pp. 1111–1124, May 2012.

[3] M. Wallace and S. Kollias, "Computationally efficient incremental transitive closure of sparse fuzzy binary relations," in Proc. IEEE Int. Conf. Fuzzy Syst., 2004, pp. 1561–1565.

[4] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," IEEE Trans. Knowl. Data Eng., vol. 24, no. 9, pp. 1537–1555, Sep. 2012.

[5] Kille, F. Hopfgartner, T. Brodt, and T. Heintz, "The Plista dataset," in Proc. Int. Workshop Challenge News Recommender Syst., 2013, pp. 16–23.

[6] M. A. Hernandez and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the merge/purge problem," Data Mining Knowl. Discovery, vol. 2, no. 1, pp. 9–37, 1998.

[7] X. Dong, A. Halevy, and J. Madhavan, "Reference reconciliation in complex information spaces," 2005, pp. 85–96.

[8] O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller, "Framework for evaluating clustering algorithms in duplicate detection," Proc. Very Large Databases Endowment, vol. 2, pp. 1282– 1293, 2009.

[9] S. Ramya and PalaninehruA, "Study of Progressive Techniques for Efficient Duplicate Detection", 2015.

[10] R. Ramesh Kannan, D. R. Abarna, G. Aswini, P. Hemavathy, "Effective Progressive Algorithm for Duplicate Detection on Large Dataset", 2016.

AUTHORS

Ms.Amani Guntuku. B.Tech in Computer science and Engineering from JNTU Hyderabad T.S India in 2013 and M.Tech in Computer Science and JNTU Hyderabad, T.S, India in 2016. She is working at Accenture Solutions Private Ltd.

Ms.Shaik Shafia, B.Tech in Computer Science and Engineering from JNTU Hyderabad,T.S, India in 2006 and M.Tech in Computer Science from JNTU Hyderabad, T.S, India in 2011. She is working presently as Associate Professor in Department of C.S.E in Hyderabad Institute of Technology And Management (HITAM), R.R.Dist, T.S, INDIA. She has 9 years of Experience in teaching. Her research interests include Secure Computing

