|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774

INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

Distributed Bottom up Approach for Data Anonymization using MapReduce framework on Cloud

R. H. Jadhav [1]

P.E.S college of Engineering, Aurangabad, Maharashtra, India[1]

rjadhav377@gmail.com

ABSTRACT: ***Many numbers of users require sharing private data like electronic health records, financial transaction records or college student's records for data analysis and mining. Therefore anonymity is one of the most important privacy preserving techniques used for privacy concerns. Currently, the scale of data in many applications increases rapidly in accordance with the Big Data trend. It is a big challenge for existing data anonymization approaches to achieve privacy preservation on private or sensitive data sets due to their lack of efficiency. Here we introduce data anonymization for processing large scale data using Distributed Bottom up approach. In Bottom up approach we start process from bottom element of the tree that is child nodes and they are replaced with its Parent node. Distributed data anonymization improves the scalability and efficiency of Bottom up approach over existing approaches using MapReduce framework and it is executed until k-anonymity is violated. MapReduce increases parallelization capability of data anonymization on large scale data and it addresses the scalability problem of anonymizing large scale data for privacy preservation.***

Keywords — *Anonymization, Bottom up approach, MapReduce framework, Cloud, Privacy Preservation.*

I INTRODUCTION

Data Anonymization means hiding identity of Person or sensitive data from records. It is the process of encrypting or removing personal identifiable information from data sets [1]. It is technology that converts text data into non human readable form. The privacy of an individual Person or things can be well preserved by data Anonymization. After performing Anonymization on these records, these records are used for data analysis and mining. Various Anonymization algorithms with different Anonymization techniques have been used for Privacy preservation. Example is private data like Electronic health records of Patients are required for data analysis or mining on cloud [11].

There are several approaches available for performing data Anonymization like Generalization, Specialization and Suppression. Data sets have become so large that anonymizing such data sets is becoming a considerable challenge for traditional anonymization algorithm [3]. There is main challenge for improving the scalability for anonymizing big data. Big data processing frameworks like MapReduce have been integrated with cloud to provide powerful computation capability for applications. MapReduce effectively solves the scalability problem of Bottom Up (BU) Approach occurs in Generalization.

Data is said to have the *k*-anonymity if each record in the table cannot be distinguished from at least (k-1) records. Following example shows the *K*-anonymity for Health records of Patients. In First table there are non anonymized health records of patient. We apply *K*-anonymity property on this table by using Suppression on Patient's name attribute means replacing the name with asterisk symbol (*) and in Generalization we give appropriate range to age attribute. The second table has the 2-anonymity.

TABLE I
HEALTH RECORD OF PATIENTS BEFORE *K*-ANONYMITY

Name	Age	Sex	State	Disease
Ragini	29	F	Tripura	Ebola
Amruta	24	F	Kerala	Swine Flu
Saniya	29	F	Tripura	Cancer
Karan	28	M	Karnataka	TB
Jaya	24	F	Kerala	Heart Disease
Bharat	22	M	Karnataka	TB
Ramesh	18	M	Kerala	Viral Infection
Krishna	28	M	Karnataka	Ebola
Jayesh	16	M	Kerala	Cancer
Jayesh	19	M	Kerala	Swine Flu

TABLE II
HEALTH RECORD OF PATIENTS AFTER *K*-ANONYMITY

Name	Age	Sex	State	Disease
*	20< Age ≤30	F	Tripura	Ebola
*	20< Age ≤30	F	Kerala	Swine Flu
*	20< Age ≤30	F	Tripura	Cancer
*	20< Age ≤30	M	Karnataka	TB
*	20< Age ≤30	F	Kerala	Heart Disease
*	20< Age ≤30	M	Karnataka	TB
*	Age ≤20	M	Kerala	Viral Infection
*	20< Age ≤30	M	Karnataka	Ebola
*	Age ≤20	M	Kerala	Cancer
*	Age ≤20	M	Kerala	Swine Flu

II LITERATURE SURVEY

Now days, data privacy preservation has been frequently investigated like LeFevre shown the scalability problem of data anonymization algorithms via introducing scalable decision trees and sampling techniques [13]. Iwuchukwu and Naughton proposed an R-tree index-based approach by creating a spatial index over data sets, achieving more efficiency [4]. However, objective of the above approaches are multidimensional generalization, thereby failing to work in the TDS approach. Fung proposed the TDS approach that produces anonymous data sets without the data distortion problem. A data structure Taxonomy Indexed Partitions (TIPS) is used to increase the efficiency of TDS. But the Top down approach is centralized, fails in handling big data sets.

Several distributed algorithms are proposed to preserve privacy of multiple data sets maintained by multiple parties. Jiang and Clifton and Mohammed showed distributed algorithms to anonymize vertically partitioned data from different data sources without disclosing private information from one party to another [6]. Jurczyk and Xiong and Mohammed proposed distributed algorithms to anonymize horizontally partitioned data sets maintained by multiple owners. However, the above distributed algorithms have main objective is that securely integrating and anonymizing multiple data sources [17]. Research mainly focuses on the scalability and efficiency problems of TDS approach anonymization.

As to MapReduce-relevant privacy protection, Roy proposed the privacy problem occurred by MapReduce and presented a system named Airavat incorporating mandatory access control with differential privacy. Further, Zhang used MapReduce to increase parallel execution capability of anonymization process. They use two phase top down approach for improving scalability and efficiency of data anonymization. In first phase splits the input as large data then apply MapReduce on it. In second phase merge the intermediate results and apply anonymization again [8].

From existing work MapReduce to accomplish the intensive computation required in big data anonymization via TD [14, 23]. But TD probably performs slower than BU when *k*-anonymity parameter is small. Scalability and efficiency of anonymization algorithms for privacy preservation has drawn attention of researchers. R-tree indexing, scalable decision trees and sampling techniques are introduced to achieve high scalability and efficiency. However, the proposed approaches aim at multidimensional scheme, thereby failing to work for sub-tree generalization. MapReduce has been widely adopted in various data processing applications to boost scalability and efficiency.

III PROPOSED WORK

Before dealing with the proposed system directly we will get through relevant objectives.

A. *RELEVANT OBJECTIVES*

1. Improve the scalability and efficiency of Data anonymization

MapReduce framework has integrated with Cloud to increase computation capability for applications. This framework addresses the scalability problem of anonymizing large scale data for privacy preservation. Efficiency of Data anonymization can be improved by using MapReduce framework because such framework has parallelization capability.

2. Increase the data utility.

It is utilization of data after anonymization will be used for mining and analysis. If the rate of information gains is higher than privacy preservation then data utility is high.

3. Maintain the consistency of privacy preservation.

Consistency maintained by using efficient data anonymization approach like distributed Top down specialization approach.

Proposing an algorithm to deal with these privacy problems like patient health records or college student database are required for analysis. Here we use the Distributed Bottom Up (DBU) approach for data anonymization, this approach is better for small data sets therefore in the first module we split data into small data sets then apply DBU those data. For each subset of large scale data apply separate DBU approach and each small data set executed parallel by using MapReduce framework.

B. ***PROPOSED SYSTEM***

In Existing approaches do not perform scalable and efficient way variable size of data (Large scale or small scale) anonymization over Cloud. For solving this problem we propose the scalable and efficient Distributed Bottom Up (DBU) approach using MapReduce framework. DBU approach is better when size of input data set is large and these data set splits then apply anonymization technique on small data sets by using MapReduce framework. Anonymized intermediate results are integrated by performing reduce function. DBU approach performed until *k*-anonymity is violated. Efficiency of distributed anonymization is better than centralized anonymization because, MapReduce increases parallelization capability.

The following architecture diagram contains different steps,

1. Data partition
2. MapReduce program.
3. Anonymization.
4. Merging.

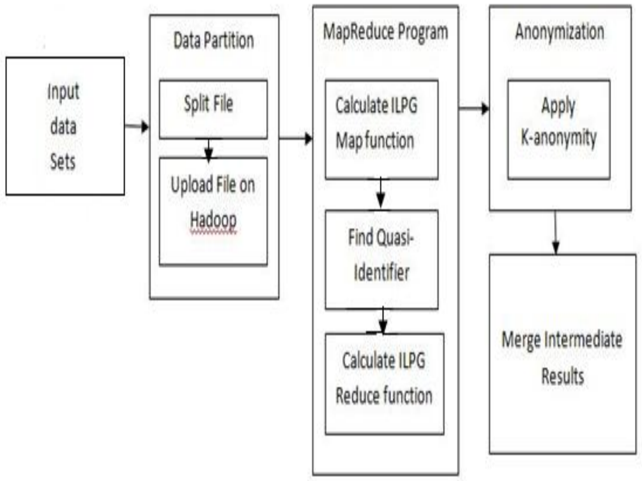

Figure .1 Architecture diagram

Once the system gets started firstly it loads the input data sets, then split large data into small data sets. The data partition is performed on the cloud. Then we provide the random no for each data set. MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map procedure that performs filtering and sorting. Reduce procedure that performs a summary operation. MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel though in practice it is limited by the number of independent data sources and the number of CPUs near each source.

Here in MapReduce perform the Information Loss per Privacy Gain (ILPG) initialization and update of Map function by finding Cut of each attribute. A cut of a tree is a subset of values in the tree that contains exactly one value on each root to leaf path. Then calculate ILPG for Reduce function, it aggregate the records and emits their count. In last phase it applies k-anonymity and merges the anonymized data.

IV EXPERIMENTAL RESULTS

Following table shows the experimental result of distributed bottom up approach applied on college student data sets. In first table we compare the performance of DBU with centralized bottom up (CBU) approach by using execution time in seconds and number of records parameter in data sets. TDBU indicate time required to execute DBU approach and TCBU time required to execute CBU approach with respect to number of records in lakhs.

TABLE III
EXECUTION TIME OF CBU AND DBU APPROACH

TCBU(Seconds)	TDBU(Seconds)	Number of records (Lakhs)
7	5	5
10	7	10
13	10	15
15	12	20
17	13	25
20	15	30
14	17	35
26	20	40

In the following figure 2, curves of the graph DBU and CBU respectively shows execution time in seconds with respect to number of records in lakhs. Here graph shows DBU require minimum time to process the input data sets than CBU.

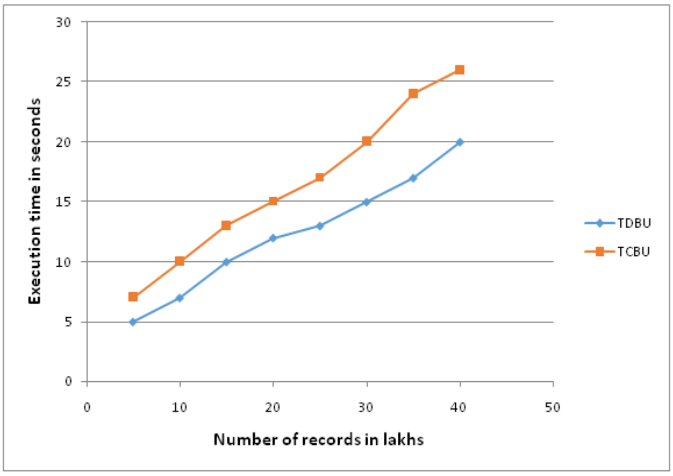

Figure .2 Change of Execution Time With Respect To Number of Records

We compared the performance of both centralized and distributed bottom up approach for data anonymization using MapReduce program. Above table values shows the DBU require minimum time to process private data than CBU approach because DBU executed in parallel. From the graph TDBU curve shows better performance as compared to TCBU with respect to number of records. In the above graph Y-axis indicate change of execution time for data anonymization and X-axis indicate number of records, we tested the performance on various number of records. When records are minimum both approaches required minimum execution time, then we increase number of records TCBU takes more time than TDBU because in DBU we split the input data for distributed approach and process the small data sets in parallel. For increasing the parallelization capability of data anonymization we have used MapReduce program. Therefore DBU approach is more efficient than existing approaches.

V CONCLUSION

There is scalability problem of large-scale data anonymization if we use centralized Bottom up approach for privacy preservation of private data. Therefore data sets are partitioned into small data sets and processed in parallel by using MapReduce, and then it produces intermediate results. Intermediate results are anonymized to produce consistent *K*-anonymous data sets.

In DBU approach the quasi-identifier is used to generate anonymous attributes and then apply k-anonymity on each intermediate result. Bottom up approach gives better efficiency when size of data set is small. Therefore we split the input data sets then apply Bottom up approach in parallel on each small data set. After that collect the intermediate results apply anonymization on every partitioned data set. In last step collect anonymized data and merge them. Experimental results on private data sets have demonstrated that with distributed approach, the Scalability and Efficiency of Bottom Up approach are improved significantly over existing approaches.

REFERENCES

1. W. Jiang and C. Clifton, “A Secure Distributed Framework for Achieving k-Anonymity,” *VLDB J.*, vol. 15, no. 4, pp. 316-333, 2006.
2. L. Sweeney, “k-Anonymity: A Model for Protecting Privacy,” *Int’l J. Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557-570, 2002.
3. LeFevre, D.J. DeWitt, and R. Ramakrishnan, “Workload-Aware Anonymization Techniques for Large-Scale Data Sets,” *ACM Trans. Database Systems*, vol. 33, no. 3, pp. 1-47, 2008.
4. T. Iwuchukwu and J.F. Naughton, “K-Anonymization as Spatial Indexing: Toward Scalable and Incremental Anonymization,” *Proc. 33rd Int’l Conf. Very Large Data Bases (VLDB ’07)*, pp. 746-757, 2007.
5. B.C.M. Fung, K. Wang, and P.S. Yu, “Anonymizing Classification Data for Privacy Preservation,” *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 5, pp. 711-725, May 2007.
6. N. Mohammed, B. Fung, P.C.K. Hung, and C.K. Lee, “Centralized and Distributed Anonymization for High-Dimensional Healthcare Data,” *ACM Trans. Knowledge Discovery from Data*, vol. 4, no. 4, Article 18, pp. 530-562, 2010
7. J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, “Twister: A Runtime for Iterative Mapreduce,” *Proc. 19th ACM Int’l Symp. High Performance Distributed Computing (HDPC ’10)*, pp. 810-818, 2010.
8. X. Zhang, C. Liu, S. Nepal, S. Pandey, and J. Chen, “A Privacy Leakage Upper-Bound Constraint Based Approach for Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud,” *IEEE Trans. Parallel and Distributed Systems, to be published*, Vol. 24, no. 6, pp. 1192-1202, 2013.
9. L.T. Yang, X. Zhang, C. Liu and J. Chen, “A scalable two phase top-down specialization approach for data anonymization using MapReduce on cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 363-373, 2014.
10. Fung, K. Wang, L. Wang, and P.C.K. Hung, “Privacy-Preserving Data Publishing for Cluster Analysis,” Data and Knowledge Eng., vol. 68, no. 6, pp. 552-575, 2009.
11. B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, “Privacy-Preserving Data Publishing: A Survey of Recent Developments,” ACM Computing Surveys, vol. 42, no. 4, pp. 1-53, 2010 .
12. X. Xiao and Y. Tao, “Anatomy: Simple and Effective Privacy Preservation,” Proc. 32nd Int’l Conf. Very Large Data Bases (VLDB ’06), pp. 139-150, 2006.
13. K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, “Incognito: Efficient Full-Domain K-Anonymity,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’05), pp. 49-60, 2005.
14. J. Dean and S. Ghemawat, “Mapreduce: Simplified Data Processing on Large Clusters,” Comm. ACM, vol. 51, no. 1, pp. 107-113, 2008.
15. W. Ke, P.S. Yu and S. Chakraborty, “Bottom-up generalization: A data mining solution to privacy protection,” Proc. 4th IEEE International Conference on Data Mining (ICDM’04), pp.249-256, 2004.
16. Palit and C.K. Reddy, “Scalable and parallel boosting with mapreduce,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1904-1916, 2012.

17. X. Xiao and Y. Tao, “Personalized Privacy Preservation,” *Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’06)*, pp. 229-240, 2006.
18. X. Xiao and Y. Tao, “Personalized Privacy Preservation,” *Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’06)*, pp. 229-240, 2006.
19. L. Hsiao-Ying and W.G. Tzeng, “A Secure Erasure Code Based Cloud Storage System with Secure Data Forwarding,” I*EEE Trans. Parallel and Distributed Systems*, vol. 23, no. 6, pp. 995-1003, 2012.
20. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,” *Proc. IEEE INFOCOM*, pp. 829-837, 2011.
21. P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, “Gupt: Privacy Preserving Data Analysis Made Easy,” *Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’12)*, pp. 349-360, 2012.
22. Microsoft Health Vault, http://www.microsoft.com/health/ww/products/Pages/healthvault.aspx.
23. Amazon Web Services, http://aws.amazon.com /elasticmapreduce. 2013.
24. Apache Hadoop, http://hadoop.apache.org.