Scalable TPTDS Data Anonymization over Cloud using MapReduce

Shital Anup Chintawar
PG Student, Dept. of Computer Science,
Alard College of Engg,Pune
Savitribai Phule University,Pune,India
s.chintawar@gmail.com

Sonali Patil
Prof, Dept. of Computer Science,
Alard College of Engg,Pune
Savitribai Phule University,Pune,India

***Abstract*—** With the rapid advancement of big data digital age, large amount data is collected, mined and published. Data publishing become day today routine activity. Cloud computing is best suitable model to support big data applications. Large number of cloud service need users to share microdata like electronic health records, data containing financial transactions so that they can analyze this data. But one of the major issues in moving toward cloud is privacy threats. Data anonymization techniques are widely used to combat with privacy concerns .Anonymizing data sets using generalization to achieve k-anonymity is one of the privacy preserving techniques. Currently, the scale of data in many cloud applications is increasing massively in accordance with the Big Data tendency, thereby making it a difficult for commonly used software tools to capture, handle, manage and process such large-scale datasets. As a result it is challenge for existing approaches for achieving anonymization for large scale data sets due to their inefficiency to support scalability. This paper presents two phase top down specialization approach to anonymize large scale datasets .This approach uses MapReduce framework on cloud, so that it will be highly scalable and efficient. Here we introduce the scheduling mechanism called Optimized Balanced Scheduling to apply the Anonymization. OBS means individual dataset have the separate sensitive field. Every data set consist of sensitive field and give priority for this sensitive field. Then apply Anonymization on this sensitive field only depending upon the scheduling.

Keywords- *Data Anonymization, Top-Down specialization, Map-Reduce, Cloud, Privacy Preservation*

*****

I. INTRODUCTION

Data sharing and publishing become routine activity for everybody. But this process is complex due to its volume of data. With current trend of big data applications, organizations are moving towards cloud because of its elasticity features. The term big data can be explained with 5V's that are volume, variety, velocity, value and Veracity. Collection, processing of such large and complex data is difficult. Cloud computing is a disruptive trend at present, poses a significant impact on present IT industry and research communities .Cloud computing provides infinite computation power and storage capacity via utilizing a large number of commodity computers together, enabling users to deploy applications in cost effective way without heavy infrastructure investment. Cloud users also can reduce huge open investment cost of IT infrastructure.

But there is always threat of privacy and security while data sharing on cloud .Users hesitates sharing the microdata on cloud due to privacy concerns. If this data is collected and mined, results can be used to take future business decisions effectively. Mining results can offer noteworthy benefits to society. For ex.Microsoft Health Vault is online health service where people gather, store their health information and share with each other. Microdata like personal health data, financial transactions data, data released by government agencies (eg. Census)is extremely sensitive. Privacy of such data can be hindered if there is any loophole in privacy protection measures of cloud data.

Data anonymization techniques and encryption techniques are widely used to combat with privacy threats. Data encryption does not suit for large scale data. Data anonymization is better option to address the scalability and volume of data. Data anonymization is process of masking of identified sensitive data while preserving its original format and data type. Output data can realistic or random sequence of data. Anonymized data set looks same in test environments and produce the same mining results.

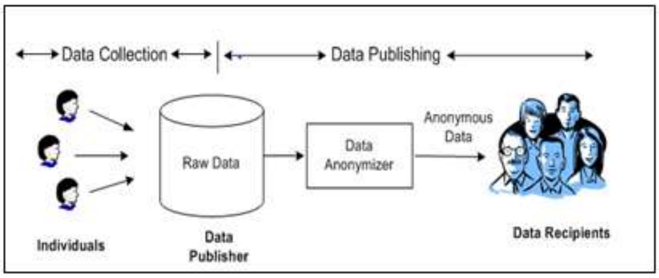


Fig 1 :Overview of Data Anonymization

II. LITEREATURE SURVEY

In Workload-Aware Anonymization Techniques for Large-Scale Datasets ,author KRISTEN LeFEVR[3] has addressed the scalability problem .They introduced two new techniques which are based on scalable decision trees and sampling ,in order to allow anonymization algorithms to be applied to datasets larger than main memory. Also they quantified quality of data is best judged with respect to the workload for which the data will ultimately be used. Proposed algorithms are highly efficient and result will be high quality data. But this approach fails in top down specialization and flops to work in multiple data sets.

In PRISM – Privacy-Preserving Search in MapReduce[14], Prism is privacy preserving search system specially designed for cloud computing. It provides us storage and query processing. PRISM is designed to add features like parallelism, efficiency of MapReduce. To use prism, we doesn't require to modify or change the original system. Use of prism will not lead extra overhead on system. Security of private cloud is not possible using this approach.

Sedic[15] is specially designed to protect data privacy in map reduce operations. Sedic proposed a privacy-aware hybrid computing archetype. Sedic schedules Map's operations such that sensitive data will be handled by jobs on private cloud and non-sensitive data will be handled by jobs on public cloud. It automatically abstracts Combiner's from Reduce functions that allow public clouds to process data.

The HybrEx model in paper "HybrEx Model for Confidentiality and Privacy in Cloud Computing"[7],provides a cohesive way for an organization to use their own infrastructure for sensitive, private data and computation, while incorporating public clouds for nonsensitive, public data and computation. This model uses partitioning of data and computation as a way to provide confidentiality and privacy. Here they discussed how HybrEx can be used in one specific execution environment - MapReduce over Bigtable.

Upper bound privacy leakage constraint-based approach in "A Privacy Leakage Upper-Bound Constraint Based Approach for Cost-Effective Privacy Preserving of Intermediate Datasets in Cloud"[4], to find which intermediate data sets need to be encrypted and which do not. Privacy-preserving cost of intermediate data sets generated can be significantly reduced. But this process is highly complicated, efficient processing of data is quite difficult.

In paper, Privacy- Preserving Data Publishing for Cluster Analysis,[2] B. Fung, K. Wang, L.Wang, P.C.K,Hung, this paper explained about how to Preventing the privacy threats caused by sensitive record linkage and the framework to secure data sharing for the purpose of cluster analysis. It Preserves both individual privacy and information usefulness for cluster analysis and avoids over-masking and improves the cluster quality by preventing the privacy threats caused by sensitive record linkage. But disadvantages are it is highly complicated. Processing on data sets efficiently will be quite a challenging task, performing general operations on encrypted data sets are still quite challenging.

In paper,Airavat: Security and Privacy for MapReduce," Roy I, Setty STV, Kilzer A , Shmatikov V , Witchel E, Airavat enables the execution of trusted and untrusted MapReduce computations on sensitive data, while assuring comprehensive enforcement of data providers privacy policies, Provides end-to-end confidentiality, integrity, and privacy using a combination of mandatory access control and differential privacy. It enables large-scale computation on data items that originate from different sources and belong to different owners. The results produced by this system are mixed with certain type of noise. Airavat cannot limit every computation performed by untrusted code. It. does not protect sensitive data from the public cloud.

Several distributed algorithms are proposed to maintain privacy.by authors Jiang and Mohammed proposed distributed algorithms to anonymize vertically partitioned data from different data sources without disclosing privacy information from one party to another. Jurczyk et al. and Mohammed et al. proposed distributed algorithms to anonymize horizontally partitioned data sets retained by multiple holders. However, the above distributed algorithms mainly aim at securely integrating and anonymizing multiple data sources.

III. PROPOSED SYSTEM AND APPROACH

A. *Objectives:*

- ✓ To present top down specialization approach for data anonymization using MapReduce on cloud framework
- ✓ To present literature review different techniques used for sharing data on cloud
- ✓ To present the design of proposed approach and algorithms.
- ✓ To present the practical analysis proposed algorithms and evaluate its performances.
- ✓ To present the comparative analysis of existing and proposed approach in order to claim the efficiency.

B. *TWO PHASE TOP DOWN SPECIALIZATION APPROACH(TPTDS)*

Two-Phase Top-Down Specialization (TPTDS) approach used to do computation required in TDS in a highly scalable and efficient way. The two phases of this approach are based on the two levels of parallelization provided by MapReduce on cloud. Generally Map Reduce on cloud has two levels of parallelization 1] job level and 2] task level. Job level parallelization means that several MapReduce jobs can be executed concurrently to make full use of cloud infrastructure resources.

Task level parallelization refers to that several mapper/reducer tasks throughout a MapReduce job area unit dead at identical time over information splits. to understand high quality, parallelizing multiple jobs on information partitions at intervals the initial section, but the resultant anonymization levels will not identical. to understand finally consistent anonymous information sets, the second section is crucial to integrate the intermediate results and extra anonymize entire information sets all over again. All intermediate anonymization levels are combined into one at intervals the second section. extra the merging of anonymization levels is finished by merging cuts. Definitely, let in and in be a pair of cuts of Associate in Nursing attribute. There occur domain values that fulfill one in each of the 3 conditions is same to could be a heap of general than could be a heap of specific than. To substantiate that the integrated go-between anonymization level never intrude upon privacy wants, the plenty of common one is chosen as a result of the integrated one, e.g., area unit selected if could be a heap of common than or simply like . For the case of numerous anonymization levels, it'll merge them at intervals identical fashion iteratively.

TDS is continual method that is ranging from the upmost domain values within the arrangement trees of attributes. Every spherical of iteration consists of three main steps. Finding the most effective specialization, playacting specialization and change values of the search metric for future spherical. Such a method of TDS is continual till k-anonymity is desecrated, to description for the utmost knowledge goes to utilize therein. The righteousness of a specialization is measured by a probe metric. therein we have a tendency to accept the info gain per privacy loss (IGPL), a trade-off metric that absorb mind each the privacy and data needs, because the search metric in our approach. A specialization with the very best IGPL worth is thought to be best one and designated of every spherical.

C. MAPREDUCE[16]

MapReduce is a programming model for handing out large data sets with a parallel, and distributed algorithms on a collection or cluster. A MapReduce program is self-possessed of a Map() procedure that performs filtering and sorting (such as sorting Patients by diseases into queues, one queue for each disease) and a Reduce() procedure that performs a summary operation (such as counting the number of patients in each queue, producing name frequencies).

The MapReduce System is coordinates by ordering the distributed servers, executing the different tasks in parallel, handling all communications and data transfers between the different parts of the system, It provides redundancy and fault tolerance, and whole management of the process. MapReduce model is motivated by the map, reduces functions usually used in functional programming. Their purpose in the MapReduce framework is not the identical as their original forms. Besides, the key contribution of the MapReduce framework are not the actual map and reduce functions.. MapReduce libraries have been written in multiple programming languages, with different levels of optimization. But the scalability and fault-tolerance accomplished for a variety of applications by optimizing the execution engine once.

A popular open source implementation example is Apache Hadoop.

1)"Map" step: First the master node takes the input, divides it into smaller sub-problems (like divide and conquer), and distributes them to worker nodes. A worker node may repeat this, so that to a multi-level tree structure can be formed. The worker node processes the smaller sub problems, and returns the results back to master node.

2)"Reduce" step: Later the master node then collects the answers to all the sub-problems and combines them in some way to form the output

Another we can look at MapReduce is as a 5-step process.[16]

1. Prepare the Map() input – the "MapReduce system" defines Map processors, assigns the K1 input key value to each processor would work on and it provides that processor with all the input data associated with that particular key value.

2. Run the Map() code provided by user – Map() is run exactly once for each K1 key value, generating output organized by key values K2.

3. "Shuffle" the Map output to the Reduce processors – the MapReduce system designates Reduce processors, the K2 will be assigned to each processor to work on and provides that processor with all the Map-generated data associated with that key value.

4. Run the user-provided Reduce() code – We make sure that Reduce() should run exactly once for each key value K2 derived by the Map step.

5. Produce the final output –All the Reduce output are collected by MapReduce system , and sorted by key value K2 to get the final output.

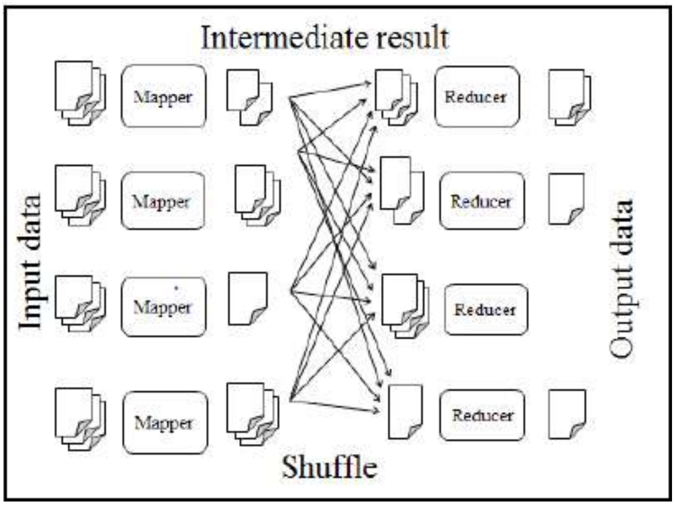


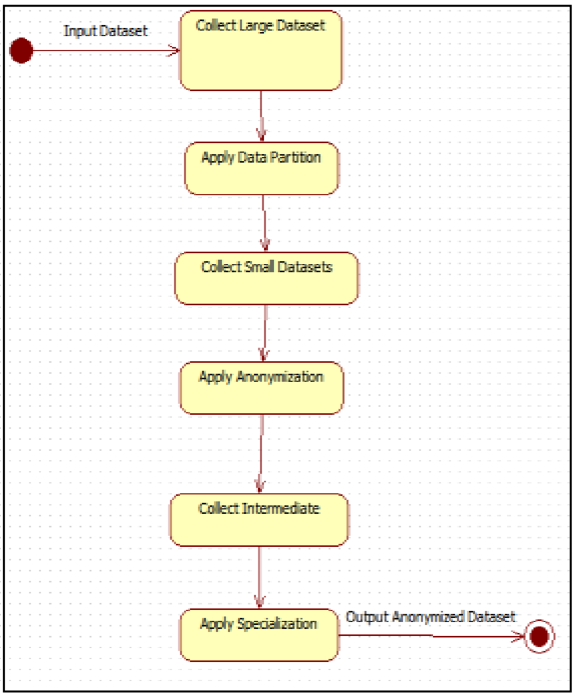
Fig 2 :Flow of MapReduce

D. ALGORITHM

Anonymization Algorithm

```
Anonymization (ID,I,k,m)
1) Scan Input dataset ID and create count tree
2) First initialize Cout
3) For each node n in preorder count-tree traversal do
4) if the item of n is generalized in Cout then  backtrack
5) If n is a leaf node and count is less than k then
6) L: = itemset corresponds to n
 7) find out generalization of items in L that make L k-anonymous
8) Merge generalizations rules with Cout
9) Backtrack to longest prefix of path L, wherein no item is generalized in Cout
10) Return Cout
11) for i :=1 to Cout do
12) Initialize count=0
13) Scan each transaction in Cout
14) Separate each  item in a transaction and store it in p
15) Count=Count+1
 16)For  m:=1 to count do
 17) for all y belongs Cout do
 18) Compare each  item of p with items in Cout
19) if all items of i equal to cout
 20)r=r+1;
21)if ka = r then backtrack to i
22) Else if r >ka then get the index position of the similar transactions
23) make them NULL until ka = r
24) Else update the transactions in db
```

E. STATECHART



F. MODULES AND IMPLEMENTATION

Mainly Our system has following modules and it has following functionalities :

- DATA PARTITION
- ANONYMIZATION
- MERGING
- SPECIALIZATION
- OBS

MODULES DESCRIPTION:

1)DATA PARTITION:

- ✓ In this module the data partition is performed on the cloud.
- ✓ Here we collect the large no of data sets.
- ✓ We are split the large into small data sets.
- ✓ Then we provides the random no for each data sets.

2)ANONYMIZATION:

- ✓ After geting the individual data sets we apply the anonymization.
- ✓ The anonymization means hide or remove the sensitive field in data sets.
- ✓ Then we get the intermediate result for the small data sets
- ✓ The intermediate results are used for the specialization process.
- ✓ All intermediate anonymization levels are merged into one in the second phase. The merging of

anonymization levels is completed by merging cuts. To ensure that the merged intermediate anonymization level ALI never violates privacy requirements, the more general one is selected as the merged one

3)MERGING:

- ✓ The intermediate result of the several small data sets are merged here.
- ✓ The MRTDS driver is used to organizes the small intermediate result
- ✓ For merging, the merged data sets are collected on cloud.
- ✓ The merging result is again applied in anonymization called specialization.

4) SPECIALIZATION:

- ✓ After geting the intermediate result those results are merged into one.
- ✓ Then we again applies the anonymization on the merged data it called specialization.
- ✓ Here we are using the two kinds of jobs such as IGPL UPDATE AND IGPL INITIALIZATION.
- ✓ The jobs are organized by web using the driver.

5) OBS:

- ✓ The OBS called optimized balancing scheduling.
- ✓ Here we focus on the two kinds of the scheduling called time and size.
- ✓ Here data sets are split in to the specified size and applied anonymization on specified time.
- ✓ The OBS approach is to provide the high ability on handles the large data sets.

IV. -Experimental Results and Evaluation

Three sets of experiments are shown in this section to evaluate the effectiveness and efficiency of the approach. In the first one, TPTDS is compared with CentTDS(centralized TDS) from the view point of scalability and efficiency. In the other two, we had tried to examine tradeoff between scalability and data utility via doing adjustments in configurations.

Generally, the execution time and ILoss(privacy loss) are affected by three parameters viz. the size of a data set (S), the number of data partitions (p) and the intermediate anonymity parameter (kI). How the three factors affecting the execution time and ILoss of TPTDS is observed. In the first group, measured the change of execution time TCent and TTP with respect to S when p=1. The size of S varies between 50 MB to 2.5 GB. In the second group, p is set equals to 3. The value of p (p>1) is selected randomly and it does not affect our analysis as what we want to see is the trend of TTP and ILTP with respect to kI. In the third group we set kI as 50,000. The value is selected randomly and it does not affect our analysis because what we want to see is the trend of TTP and ILTP with respect to the number partitions. The no. of partitions varies from 1 to 20.

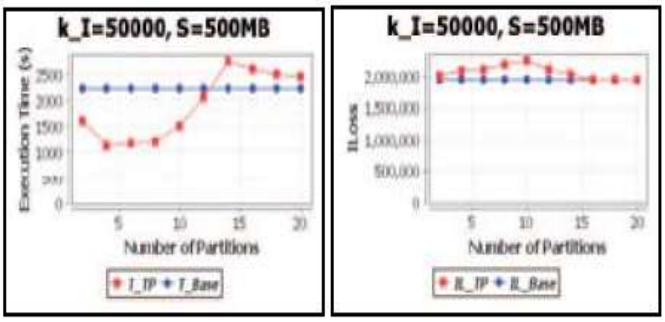


Fig 3:Change in execution time with respect to no.of partitions

V. Conclusion

In this paper, we've got investigated the measurability downside of large-scale information anonymization by exploitation TDS, and projected a extremely climbable two-phase TDS approach exploitation MapReduce on cloud. Information sets area unit 1st divided (random partitioning) and anonymized within the 1st section parallel. The first section can turn out intermediate results. Then these intermediate results area unit united and more anonymized to supply consistent k-anonymous information sets within the second section. Same method perennial till we tend to get k-anonymous records. By- creatively applying MapReduce on cloud to information anonymization and designed a bunch of innovative MapReduce jobs to concretely accomplish the specialization computation in a very extremely climbable manner. We have enforced the optimized equalization programing. OBS is predicated on the time and size of the info sets. Optimized balanced programing ways area unit developed towards overall climbable privacy preservation aware information set programing.

REFERENCES

[1] Xuyun Zhang, Laurence T. Yang, Senior Member, IEEE, Chang Liu, and Jinjun 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, FEBRUARY 2014 363

[2] B. Fung, K. Wang, L. Wang and P.C.K. Hung, “Privacy-Preserving Data Publishing for Cluster Analysis,” Data Knowl.Eng., Vol.68, no.6, pp. 552-575, 2009.

[3] K. LeFevre, D.J. DeWitt and R. Ramakrishnan,“Workload-Aware Anonymization Techniques for Large-Scale Datasets,” ACM Trans. Database Syst., vol.33, no. 3, pp. 1-47, 2008.

[4] X. Zhang, Chang Liu, S. Nepal, S. Pandey and J. Chen, “A Privacy Leakage Upper-Bound Constraint Based Approach for Cost- Effective Privacy Preserving of Intermediate Datasets in Cloud,” IEEE Trans. Parallel Distrib. Syst., In Press, 2012.

[5] L. Wang, J. Zhan, W. Shi and Y. Liang, “In Cloud, Can Scientific Communities Benefit from the Economies of Scale?,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 2, pp.296-303, 2012.

[6] H. Takabi, J.B.D. Joshi and G. Ahn, “Security and Privacy Callenges in Cloud Computing Environments,” IEEE Securityand Privacy, vol. 8, no. 6, pp. 24-31, 2010.

[7] Ko SY, Jeon K, Morales R, “The Hybrex Model for Confidentiality and Privacy in Cloud Computing,”Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'11), 2011;Article 8.

[8] L. Sweeney, “k-anonymity: a model for protecting privacy”, International Journal on Uncertainty,Fuzziness and Knowledgebased Systems, 2002, pp. 557- 570.

[9] A.Machanavajjhala, J.Gehrke, and D.Kifer, et al, “ℓ-diversity: Privacy beyond k-anonymity”, In Proc. Of ICDE, Apr.2006.

[10] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, “Map task scheduling in mapreduce with data locality: Throughput and heavytraffic optimality,”Arizona State Univ., Tempe, AZ, Tech. Rep., Jul. 2012.

[11] N. Cao, C. Wang, M. Li, K. Ren and W. Lou, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,” Proc. 31st Annual IEEE Int'l Conf. ComputerCommunications (INFOCOM'11), pp. 829-837, 2011.

[12] P. Mohan, A. Thakurta, E. Shi, D. Song and D. Culler, “Gupt: Privacy Preserving Data Analysis Made Easy,” Proc. 2012 ACMSIGMOD Int'l Conf. Management of Data (SIGMOD'12), pp. 349- 360, 2012.

[13] P. Jurczyk and L. Xiong, “Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers,” Proc. 23rd Ann. IFIP WG 11.3 Working Conf. Data and Applications Security XXIII (DBSec ’09), pp. 191-207, 2009.

[14] PRISM – Privacy-Preserving Search in MapReduce, Erik-Oliver Blass, Roberto Di Pietro, Refik Molva, Melek Önen in Lecture Notes in Computer Science Volume 7384, 2012, pp 180-200

[15] Sedic: privacy-aware data intensive computing on hybrid clouds K Zhang, X Zhou, Y Chen, XF Wang, Y Ruan - Proceedings of the 18th ACM conference on Computer …, 2011

[16] https://en.wikipedia.org/wiki/MapReduce