ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

Privacy-Preserving Data Publishing for Two-Phase TDS approach using Mapreduce on Cloud

SHERIFA.G [1]

PG student, Department of Computer Science and Engineering, Parisutham Institute of Technology and Science,

Thanjavur, TamilNadu, India [1]

Abstract: A large number of cloud services require users to share private data like electronic health records for data analysis or mining, bringing privacy concerns. Unavowed data sets via generalization to satisfy certain privacy requirements such as k-anonymity is a widely used category of privacy conserving techniques. At present, the tensile of data in many cloud applications increases tremendously in accordance with the big data trend, thereby making it a challenge for frequently used software tools to capture, manage, and process such vast-scale data within a tolerable pass by time. As a result, it is a challenge for existing unavowed approaches to achieve privacy preservation on privacy-sensitive large-scale data sets due to their insufficiency of scalability. In this paper, we put forward a scalable two-phase top-down specialization (tds) approach to anonymize large-scale data sets using the mapreduce framework on cloud. In both phases of our start to deal with, we consciously design a group of innovative mapreduce jobs to concretely accomplish the specialization computation in a highly scalable way.

Keywords: Data anonymization, top-down specialization, mapreduce, cloud, privacy preservation

I. INTRODUCTION

CLOUD computing, a disruptive trend at nowadays, constitute a significant impact on current IT industry and research centre. Cloud computing provides large computation power and storage capacity via utilizing a large number of commodity computers together, enabling users to put application cost-effectively without heavy environment investment. Cloud users can reduce huge upfront contribution of IT infrastructure, and focus on their own nucleus business. In whatever way, few dormant customers are still hesitant to take advantage of cloud due to privacy and security concerns .

Privacy is one of the most concerned issues in cloud computing. Personal data like electronic health records and financial transaction records are usually deemed extremely sensitive although these data can offer significant human benefits if they are analyzed and mined by organizations such as disease research centre. Like, Microsoft HealthVault, an online cloud health centre, accumulate data from users and shares the data with research institutes. Data preservation can be divulged with less effort by malicious cloud users or providers because of the failures of some traditional privacy protection measures on cloud .This can bring considerable economic loss or severe social reputation impairment to data owners. Hence, data privacy controversy need to be addressed urgently before data sets are analyzed or shared on cloud.

Data anonymization has been extensively studied and widely adopted for data privacy preservation in noninteractive data publishing and sharing scenarios. Data anonymization refers to hiding identity and/or sensitive data for owners of data. Then, the privacy of an data owner can be effectively preserved while certain aggregate information is exposed to data users for diverse experimenting and quarring. A variety of anonymization algorithms with different anonymization operations have been propose. In whatever way, the scale of data sets that need anonymizing in some cloud applications increases tremendously in accordance with the cloud computing and Big Data trends. Data have become so massive that hiding such data sets is becoming a considerable challenge for traditional anonymization methods. The analyst have start to examine the scalability problem of large-scale data anonymization.

II. SYSTEM ARCHITECTURE

The bellow architecture based on cloud data delivers for efficient technologies using Map-Reduce procedures. The Two-Phase Top-Down Specialization approach will filter into two process first process data cluster into big data and that data will apply for k-anonymous and create intermediate data after publish data to cloud. That intermediate data apply for map-reduce knowledge will deliver efficiently.

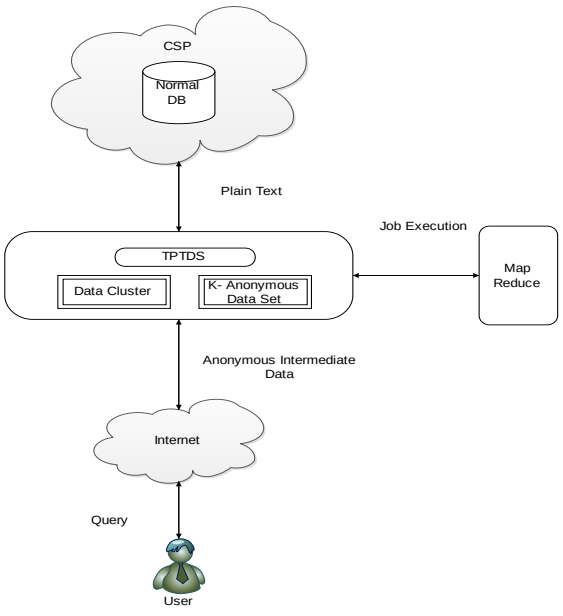

Fig. 1 System architecture

DOI 10.17148/IJARCCE.2015.4216

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

Large-scale data processing frameworks like MapReduce have been integrated with cloud to provide powerful computation capability for applications. So, it is assuring to adopt such frameworks to address the scalability dispute of anonymizing large-scale data for privacy conservancy. In our research, we bargaining chip MapReduce, a widely adopted parallel data processing framework, to address the scalability issue of the top-down specialization (TDS) approach for large-scale data anonymization. The TDS approach, contribution a good tradeoff between data utility and data consistency, is extensively applied for data anonymization .Most TDS algorithms are centralized, resulting in their inadequacy in handling large-scale data sets. Despite, some distributed algorithms have been proposed , they mainly focus on secure anonymization of data sets from several parties, rather than the scalability.

III. RELATED WORK

Recently, data privacy preservation has been extensively investigated .LeFevre et al. addressed the scalability problem of anonymization algorithms via introducing scalable decision trees and sampling methods. Some analyst proposed an R-tree index-based approach by building a spatial index over data sets, attaining high skill. However, the above methods aim at multidimensional generalization, thereby failing to work in the TDS approach. Fung et al., proposed the TDS approach that produces anonymous data sets without the data exploration dispute. A data structure Taxonomy Indexed PartitionS (TIPS) is exploited to improve the efficiency of TDS. But the approach is rationalized, leading to its deficiency in handling large-scale data sets.

IV. PROBLEM ANALYSIS

We analyze the scalability problem of existing TDS approaches when handling large-scale data sets on cloud. The centralized TDS approaches exploits the data structure TIPS to improve the scalability and efficiency by indexing anonymous data records and retaining statistical information in TIPS.

V. PRELIMINARY

1..Basic Notations

We describe several basic notations for convenience. Let D denote a data set containing data records. A record r 2 D has the form r ¼ hv1; v2; . . . ; vm; svi, where m is the number of attributes, vi, 1 _ i _ m, is an attribute and sv is a sensitive value. The record of sensitive values is denoted as SV . An attribute of a data record is denoted as Attr, and the taxonomy tree of this attribute is represented as TT . The quasi-identifier of a record is denoted as qid ¼ hq1; q2; . . . ; qmi, here qi 2 DOMi. Quasi-identifiers, denoting groups of anonymous data, can lead to privacy breach if they are too specific that only a small group of people are linked to them. Quasi-identifier set is denoted as QID ¼ hAttr1; Attr2; . . . ; Attrmi. The set of the records with qid is defined as QI-group, denoted by QIGðqidÞ. QI is the acronym of quasi-identifier.

2.Top-Down Specialization

Generally, TDS is an iterative process starting from the topmost domain values in the taxonomy trees of facet. Each round of iteration consists of three main steps, such as, finding the best specialization, executing specialization and updating values of the search metric for the next round . Such a process is repeated until k-anonymity is disobey, to expose the maximum data utility. The generosity of a specialization is measured by a search metric. We accept the information gain per privacy loss (IGPL), a tradeoff metric that considers both the privacy and information needed, as the search metric in our method. A specialization with the highest IGPL value is regarded as the best one and selected in each round.Given a specialization spec : p ---->Child(p), the IGPL of the specialization is calculated by

$$IGPL(spec)=IG(spec)/(PL(spec)+1)$$

The term IP(spec) is the information gain after performing spec, and PL(spec) is the privacy loss. IG(spec) and PL(spec) can be computed via statistical information derived from data sets. Let Rx denote the set of original records containing attribute values that can be generalized to x. |Rx| is the number of data records in Rx. Let I(Rx) be the entropy of Rx. Then, IG(spec) is calculated by

$$IG(spec)=I(Rp)- \sum_{c \in chilp(p)} (|Rc|/|Rp|)\, I(Rc),$$

Let |Rx,sv| denote the number of the data records with sensitive value sv in Rx. I(Rx) is computed by

$$I(Rx)= -\sum_{sv \in SV} (|Rx,sv|/|Rx|).\log2(|Rx,sv|/|Rx|)$$

Let Ap(spec) denote the anonymity before performing spec, while Ac(spec) be that after performing spec. Privacy loss caused by spec is calculated by

$$PL(spec) =Ap(spec) - Ac(spec)23$$

VI. TWO-PHASE TOP-DOWN SPECILAIZATION (TPTDS)

The sketch of the TPTDS approach is elaborated. Three components of the TPTDS approach, such as, data partition, anonymization level combining, and data specialization are amplified in later section.

1.Sketch of Two-Phase Top-Down Specialization

We propose a TPTDS approach to conduct the computation required in TDS in a highly scalable and excellent fashion. The two phases of our approach are based on the two levels of parallelization provisioned by MapReduce on cloud. Normally, MapReduce on cloud has two levels of parallelization, i.e., job level and task level. Job level parallelization defines that multiple MapReduce jobs can be executed simultaneously to make full use of cloud environment resources. Integrating with cloud, MapReduce becomes more powerful and elastic as cloud can offer infrastructure resources on demand, for instance, Amazon Elastic MapReduce service [29]. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data

DOI 10.17148/IJARCCE.2015.4216

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

segments. To attain high scalability, we parallelizing multiple jobs on data partitions in the beggining phase, but the resultant anonymization levels are not same. To obtain finally consistent anonymous data sets, the next phase is necessary to add the intermediate results and further anonymize entire data sets.

2.Data Partition

When D is partitioned into Di, 1 _ i _ p, it is required that the disposal of data records in Di is similar to D. A data record here can be treated as a point in an m-dimension space, where m is the number of attributes. Thus, the transitional anonymization levels derived from Di, 1 _ i _ p, can be more similar so that we can get a better merged anonymization level.

Random sampling technique is affiliated to partition D, which can satisfy the above requirement. Especifically, a random number rand, 1 _ rand _ p, is generated for each data record. A record is assigned to the partition Drand. It shows the MapReduce program of data partition. Refer that the number of Reducers should be equal to p, so that each Reducer handles one value of rand, exactly producing p resultant files. Each file contains a random sample of D.

3.Anonymization Level Merging

All intermediate anonymization levels are merged into one in the second section. The merging of anonymization levels is concluded by blending cuts. Especifically, let Cuta in AL0a and Cutb in AL0b be two cuts of an attribute. There exist domain values qa 2 Cuta and qb 2 Cutb that satisfy one of the three conditions: qa is identical to qb, qa is more general than qb, or qa is more distinct than qb.

To ensure that the merged intermediate anonymization level ALI never violates privacy demand, the more common one is tabbed as the merged one, for example, qa will be selected if qa is more general than or identical to qb. For the case of multiple anonymization levels, we can integrate them in the same way iteratively. The below lemma ensures that ALI still complies privacy requirements.

4.Data Specialization

An original data set D is concretely specialized for anonymization in a one-pass MapReduce job. After obtaining the merged intermediate anonymization level AL^{I} , we run MRTDSðD; k; AL^{I} Þ on the entire data set D, and get the final anonymization level AL–. Then, the data set D is anonymized by replacing original attribute values in D with the responding domain values in AL^{*}.

Details of Map and Reduce functions of the data specialization MapReduce job are described in Algorithm 3. The Map function emits anonymous records and its count. The Reduce function simply aggregates these anonymous records and counts their number. An anonymous record and its count represent a QI-group.

VII. DFD

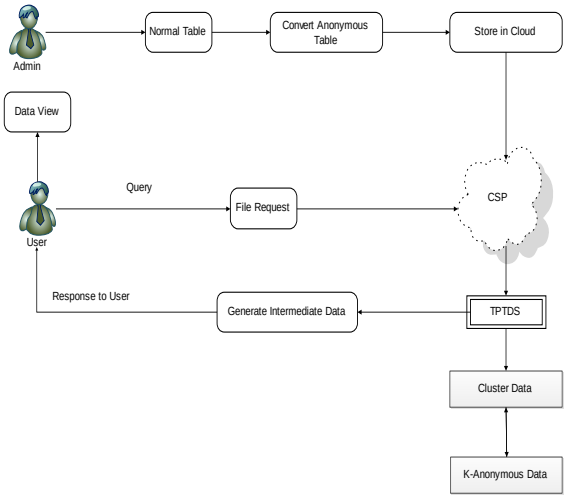

Fig. 2 Dataflow diagram for two phase TDS approach

VIII. MAP REDUCE VERSION OF CENTRALIZED TDS

We considerably the MRTDS in this section. MRTDS plays a main role in the two-phase TDS method, as it is invoked in both phases to concretely conduct process. Basically, a practical MapReduce program consists of Map and Reduce functions, and a Driver that agrees the macro computation of jobs. We describe the MRTDS Driver.

1.MRTDS Driver

Usually, a single MapReduce job is inadequate to accomplish a complex task in many applications. Thus, a cluster of MapReduce jobs are orchestrated in a driver program to achieve such an unbiased. MRTDS consists of MRTDS Driver and two types of jobs, such as., IGPL Initialization and IGPL Update. The driver classify the execution of jobs.

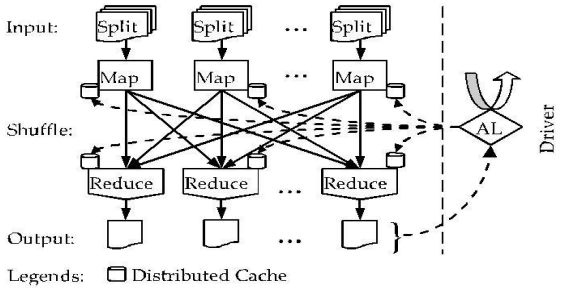

Fig. 3 Map reduce architecture

2.IGPL Initialization Job

The Map and Reduce functions of the job IGPL Initialization are described in following algorithm. The main task of IGPL Initialization is to initialize information gain and privacy loss of all specializations in the initial anonymization level AL.The statistical information jRpj, jðRp; svÞj, jRcj, and jðRc; svÞj is required for each specialization to calculate information gain.The number of records in each current QI-group needs figure out, so does the number of records in each QI-group after potential specializations.

DOI 10.17148/IJARCCE.2015.4216

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

3.IGPL Update Job

The IGPL Update job dominates the scalability and readiness of MRTDS, since it is executed iteratively as described in Algorithm. So far, iterative MapReduce jobs have not been well supported by standard MapReduce framework like Hadoop. Accordingly, Hadoop variations like Haloop and Twister have been proposed recently to support efficient iterative MapReduce computation. Our method is based on the standard MapReduce framework to facilitate the discussion herein.

4.Implementation and Optimization

To elaborate how data sets are handled in MRTDS, the execution framework based on standard MapReduce is detail in The solid arrow lines represent the data flows in the canonical MapReduce framework.We can see that the repetitive of MapReduce jobs is controlled by anonymization level AL in Driver. The data flows for handling repetitive are denoted by dashed arrow lines. AL is dispatched from Driver to all workers including Mappers and Reducers via the distributed cache mechanism. The value of AL is modified in Driver according to the output of the IGPL Initialization or IGPL Update jobs. As the amount of such data is small compared with data sets that will be anonymized, they can be effectively transmitted between Driver and workers.

IX. PERFORMANCE EVALUATION

1.Overall Comparison

To evaluate the effectiveness and efficiency of our two-phase approach, we inspect it with the centralized TDS approach proposed, denoted as CentTDS. CentTDS is the state-of-the-art approach for TDS anonymization. Scalability and dossier utility are considered for the effective-ness. For scalability, we check whether both approaches can still work and scale over large-scale data sets. Data fitness is measured by the metric ILoss, a general purpose data metric propose.

Actually, ILoss means information loss caused by data anonymization. Basically, higher ILoss pinpoints less data utility. How to calculate ILoss can be found, which is available in the online supplemental material. The ILoss of CentTDS and TPTDS are represented as ILCent and ILTP , respectively. The execution time of CentTDS and TPTDS are denoted as TCent and TTP , respectively.

2.Experiment Evaluation

2.1.Experiment Settings

Our experiments are conducted in a cloud environment named U-Cloud. U-Cloud is a cloud computing environment at the University of Technology Sydney (UTS). The system overview of U-Cloud has been depicted. The computing facilities of this system are located among several labs at UTS. On top of hardwares and Linux operating system (Ubuntu), we build in KVM virtualization software that virtualizes the infrastructure and provides unified computing and storage resources. To create virtualized data centers, we inaugurate OpenStack open source cloud environment for global management, resource arranging and interaction with users. Else, Hadoop clusters are built based on the OpenStack cloud platform to facilitate large-scale data processing.

2.2.Experiment Process and Results

We conduct three groups of experiments in this section to evaluate the effectiveness and efficiency of our method. In the first one, we compare TPTDS with CentTDS from the perspectives of scalability and efficiency. In the other two, we consider on the tradeoff between scalability and data utility via adjusting configurations. Commonly, the execution time and ILoss are affected by three factors, namely, the size of a data set (S), the number of data partitions (p), and the intermediate anonymity parameter (kI). How the three factors influence the execution time and ILoss of TPTDS is observed in the following experiments.

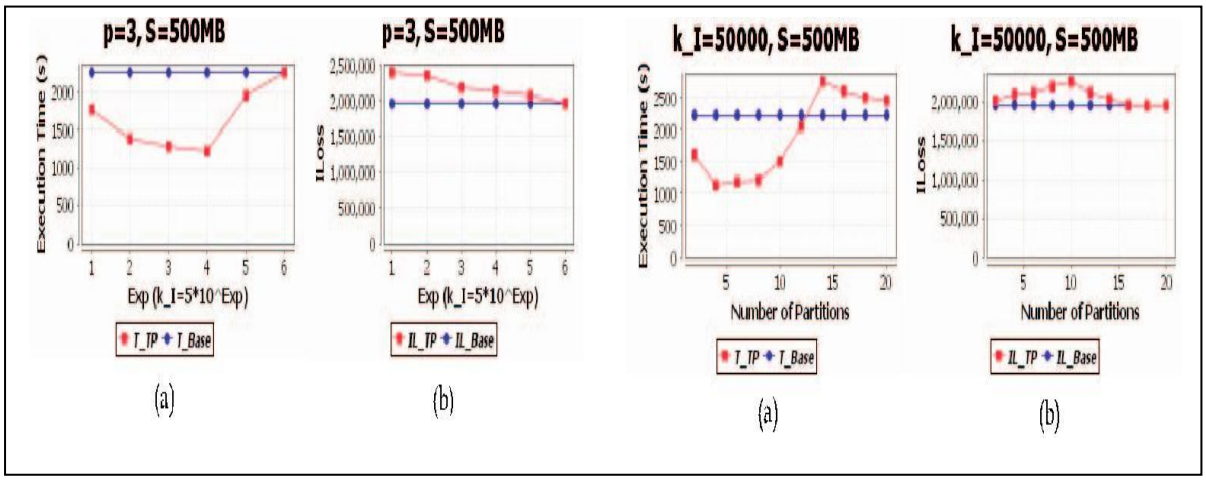

Fig. 4 Change of execution time and ILOSS w.r.t intermediate anonymity parameter

Fig. 5 Change in execution time and ILOSS w.r.t number of partitions

DOI 10.17148/IJARCCE.2015.4216

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

X. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the scalability problem of large-scale data inominate by TDS, and proposed a highly scalable two-phase TDS approach using MapReduce on cloud. Data sets are segmented and anonymized in parallel in the first section, producing mean results. Then, the mean results are merged and further anonymized to produce consistent k-anonymous data sets in the second section. We have creatively applied MapReduce on cloud to data anonymization and deliberately designed a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. Results on real-world data sets have demonstrated that with our method, the scalability and efficiency of TDS are improved significantly over existing approaches.

In cloud environment, the privacy preservation for data scrunity, share and mining is a challenging research issue due to increasingly larger volumes of data sets, thereby requiring allout investigation. We will examine the adoption of our approach to the bottom-up generalization algorithms for data anonymization. Depend on the contributions herein, we plan to further analyse the next step on scalable privacy preservation aware analysis and scheduling on large-scale data sets. Optimized balanced scheduling system are expected to be developed towards overall scalable privacy preservation aware data set scheduling.

ACKNOWLEDGEMENT

My sincere thanks to my guide Mr.A.Rajarajan Asst.Professor, Department of Computer Science and Engineering, Parisutham Institute of Technology and Science, Thanjavur for his help and guidance to enable me to propose this system.

REFERENCES

[1] S. Chaudhuri, "What Next?: A Half-Dozen Data Management Research Goals for Big Data and the Cloud," Proc. 31st Symp. Principles of Database Systems (PODS '12), pp. 1-4, 2012.

[2] H. Takabi, J.B.D. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Nov. 2010.

[3] 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, 2012.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," Proc.

[5] 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.

[6] D. Zissis and d. Lekkas, "Addressing cloud computing security issues," future generation computer systems, vol. 28, no. 3, pp. 583- 592, 2011.

[7] K. 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.

[8] 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, 2010.

[9] L. Hsiao-Ying and W.G. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 995-1003, 2012.

[10] 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.

BIOGRAPHY

G.Sherifa received B.E (CSE) from Periyar Maniammai University in 2013. I am currently persuading M.E (Computer Science and Engineering) in Parisutham Institute of Technology and Science Thanjavur.