A Survey on Two-Phase Top-Down Specialization for Data Anonymization using Map Reduce on Cloud

Monali S. Bachhav
University of Pune,
Department of Computer Engineering,
SITRC College of Engineering,
Nashik-422213

Amitkumar Manekar
Assistant Professor,
Department of Computer Engineering,
SITRC College of Engineering,
Nashik-422213

ABSTRACT

Most cloud services require users to share personal data like electronic health records for analysis of data or mining, bringing privacy concerns. In many cloud applications at present the scale of data increases in accordance with Big Data, thereby making it a complicated to commonly used software tools to handle and process a large-scale data within a tolerable elapsed time. It is challenging for previous annonymization approaches to achieve privacy preservation on large scale data sets due to insufficiency. The proposed a scalable two-phase top-down specialization (TDS) approach uses MapReduce architecture on cloud to annonymized large scale datasets finally deliberately design a group of innovative MapReduce jobs to particularly accomplish specialization computation in a highly scalable way. So the ability of TDS and efficiency of TDS can be significantly improved over existing approaches.

Keywords

Data anonymization, top-down specialization, MapReduce, cloud, privacy preservation

1. INTRODUCTION

CLOUD computing, is a disruptive trend at present, poses a significant effect on IT industry and research communities[1],[3]. Cloud computing provides computation power and storage capacity via utilizing a large number of commodity computers together, enabling users for the deployment of applications cost-effectively without heavy infrastructure investment. Cloud users decreases huge upfront investment of IT infrastructure, and give total concentration on their own core business. However, numerous potential of customers are still confusingt to take advantage of cloud due to privacy and security concerns.

One of the most important issue in cloud computing is privacy of the data, and the concern with context of cloud computing although some privacy issues are not new [3]. Private information as electronic health records and financial transaction records are usually deemed very sensitive although these data can offer significant human benefits if they are explain and mined by organizations such as disease research centers. For example, an online health services of cloud, collect data from users and share data with institutes of research. Data privacy 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 economic loss or severe social reputation impairment to data owners. Before distributing or analyzing of the data sets on the cloud data privacy issues need to be addressed urgently. Anonymizatio of the data has extensively studied and widely adopted for data privacy preservation in noninter-active data publishing and sharing scenarios [10]. Data anonymization refers to hide identity and/or sensitive data for owners of data records. Then, privacy of individual can be effectively preserved while certain aggregate information is exposed to data users for diverse analysis and mining. Different forms of anonymization algorithms with different anonymization operations have been proposed [10], [19], [20]. However, the scale of data sets that need anonymizing in some cloud applications increases tremendously in accordance with the cloud computing and Big Data trends [19]. Data sets have become so large that anonymizing such data sets is becoming a considerable challenge for traditional anonymization algorithms. The researchers have started to search the scalability problem of large-scale data anonymization [20].

2. RELATED WORK

Recently, data privacy preservation has been extensively investigated [9]. The scalability problem of anonymization algorithms via introducing scalable decision trees and sampling techniques. An R-tree index-based approach by building a spatial index over data sets, for achieving high efficiency. However, the approaches aim at multidimensional generalization [20], thereby failing[6], [9], [10] to work in the TDS approach.The [8] propoesd TDS approach that produces anonymous data sets without the data exploration problem. A data structure Taxonomy Indexed PartitionS (TIPS) is exploited to improve the efficiency of TDS. But these approach is centralized, leading to its inadequacy in handling data sets of large scale.

Several distributed algorithms are proposed to preserve privacy of multiple data sets retained by multipleparties. The proposed [6], distributed algorithms to anonymize vertically partitioned data from different data sources without disclosing privacy information from one party to another. distributed algorithms to anonymize horizontally partitioned data sets retained by multiple holders. However, the given distributed algorithms mainly aim at securely integrating and anonymizing multiple data sources. The research mainly focuses on the scalability issue of TDS anonymization, and is, therefore, orthogonal to them. As to MapReduce relevant to protection, the data privacy problem caused by MapReduce and presented a system named Airavat in-corporating mandatory access control with differential privacy. Further, leveraged MapReduce[1] to automatically partition a computing job in terms of data levels of security protecting data privacy in hybrid cloud. The research exploits MapReduce itself to anonymize large-scale data sets before data are processed by other MapReduce jobs, arriving at preservation of privacy.

2.1 Problem Analysis

When handling large-scale data sets on cloud the problem of scalability problem of existing TDS is analyze. The centralized TDS approaches in [6], [9], and [10] exploits the data structure TIPS to increase the scalability and efficiency by indexing anonymous records of data and retaining information like statistical in TIPS. The data structure speed up the specialization process because indexing structure avoids frequently scanning entire data sets and storing statistical results circumvents recomputation over-heads. On the other side, the amount of metadata retained to maintain the statistical information and linkage information of record partitions is relatively large compared with data sets themselves, thereby consuming necessary memory. Moreover, overheads incurred by maintaining the linkage structure and updating the statistic information will be huge when date sets become large. Due to this, centralize approaches probably suffer from low efficiency and scalability when handling large-scale data sets. There is an assumption that all data processed should fit in memory for the centralized approaches [10].

A distributed TDS approach [8] is proposed to address the distributed anonymization problem which mainly concerns privacy protection against other parties, rather than scalability issues. This approach only employs information gain, rather than its formation with privacy loss, as search metric when determining the best specializations. As pointed out in [10], a TDS algorithm without considering privacy loss probably chooses a specialization that leads to a quick violation of anonymity requirements. Hence, the distributed algorithm not able to produce anonymous data sets exposing the same data utility as centralized ones. Besides, the issues such as communication protocols and fault tolerance must be kept in mind when designing such distributed algorithms. As such, it is not appropriate to leverage existing distributed algorithms to solve the scalability problem of TDS. There is assumption that all processed data should fit in memory for centralize approach [10].This assumption fails sometimes to hold in most data-intensive cloud applicatios. In cloud environment, computation provisioned in the type of virtual machines. The cloud offers several several forms of virtual machines. But centralized approaches difficult in handling data-sets of large scale.

3. PROPOSED ARCHITECTURE

Basic notations for convenience. Let F indicates a set of data which contains data records. A record b ϵ F has the form b =(v_1,, v_2,,. . ., v_m) $_{\mathrm{Mv}}$, where m is the number of attributes, v_i,1≤ i ≤ m, is an attribute value and sensitive value is like diagnosis. The sensitive values are denoted as $_{\mathrm{Mv}}$. An attribute of a record is denoted by A, and the taxonomy tree of this attribute is denoted as T . Let DO represent the set of all domain values in T. The quasi-identifier of a record is denoted by q =(q_1, q_2, . . ., q_m), where q_iϵDOi. Quasi-identifiers, shows groups of anonymous records, can lead to privacy if they are too specific that only a small group of people are linked to them. The set of quasi-identifier is indicated as qid = ($_{\mathrm{A1,A2,\ldots.Am}}$). The set of the records with q is defined as QI-group, denoted by (QIG_q). QI acronym for quasi -identifier. In this k-anonymity [16] used as a privacy model. The anonymity parameter k is specified by users according to their privacy requirements.

Using Specialization approach the data is anonymized. A specialization is use to replace a domain value with all its child values. Domain values of an attribute form a "cut" through its taxonomy tree. The cut of attribute $_{A_i}$, indicated as cut_i, 1 ≤ i ≤ m, is a subset of values in $_{\mathrm{doi}}$. Cut contains exactly one value in each root to leaf path in taxonomy tree T_i. The cut of all attributes searches the anonymity of a data set. To gain the degree of anonymization during the specialization process, we give the proper definition.

TDS is an iterative process starting from topmost domain value in taxonomy tree. Each round of iteration consists of three main steps searching the best specialization, performing specialization and updating values of the search metric for the next round[10]. Such a process is repeated until k-anonymity is changed, to expose the maximum data utility. The correctness of a specialization is measured through a search metric. The information gain per privacy loss (IGPL), a tradeoff metric that considers both the privacy and information requirements, as the search metric in this approach. A specialization with the greater IGPL value is regarded as the best one and selected in each round.

3.1 Two Phase Top Down Specialization

Basically the TPTDS works on three basic component, data partition, anonymization level merging, and data specialization.

TPTDS is proposed to conduct computation required in TDS. The approach is based on parallelization i.e. job level and task level. In this job level parallelization means multiple job MapReduce job can be executed to use cloud infrastructure resource. For example Amazon Elastic MapReduce service[15].

Algorithm 1. Sketch of two phase TDS

```
Input: Data set F, anonymity parametersg, $k^I$ and the number of partition p.

Output: Anonymous data set $F^*$.

1: Partition F into $F_i$, 1≤ i ≤ p.

2: Execute MRTDS($F_i$, $k^I$, $AL^0$)→$AL_i^{'}$, 1≤ i ≤ p in
    Parallel as multiple MapReduce jobs.

3: Merge all intermediate anonymization levels into one,
    Merge($AL_1^{'}$, $AL_2^{'}$, ….$AL_p^{'}$)→ $AL^I$

4: Execute MRTDS(F,k, $AL^I$)→ $AL^*$ to achieve k-anonymity.

5: Specialize F according to $AL^*$, Ouput $F^*$.
```

3.1.1 Data Partition

In this for the dividation of data the random sampling technique is used. Specifically a random number rand, 1≤rand≤p, is generated for each data record.

In this the important thing is the number of reducer should be equal to p, so each reducer handle one value of rand exactly producing p resultant files.

3.1.2 Anonymization Level Merging

All anonymized levels are merged into one.By merging the cuts anonymization level is formed. All anonymized level satisfy K-anonymity.

3.1.3 Data Specialization

The original data set F is specialized for anonymization in a MapReduce job. In this Map function emits anonymous records and its count. The Reduce function simply aggregates anonymous records and counts their number.

Algorithm 2. Data Specialization Map & Reduce

Input: Data record, Anonymization level AL^*

Output: Anonymous record.

Map: Construct anonymous record using sensitive value and partition.

Reduce:emit sum.

3.2 MapReduce Version of Centralised TDS

Usually, a single MapReduce job is insufficient to accomplish a difficult task in a driver program to achieve such an objective. MRTDS consists of Drivers of MRTDS and two types of jobs, i.e., IGPL Initialization in many applications. A group of MapReduce jobs are orchestrated and IGPL Update. The job execution arranges by the drivers.

3.2.1 IGPL Initialization Job

The main goal of IGPL Initialization is to initialize information gain and privacy loss of all specialization in the initial anonymization level.

3.2.2 IGPL Update Job

The IGPL Update job dominates the scalability and efficiency of MRTDS, since it is executed iteratively .The iterative MapReduce job have not been well supported by MapReduce framework like Hadoop [14].IGPL Update Job requires less computation and consumes less network bandwidth. Thus current is more efficient than latter.

4. MRTDS FRAMEWORK

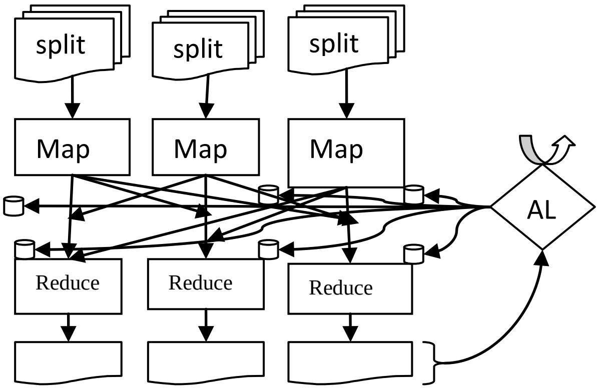

Figure 1. MRTDS Overview

For the explanation of how data sets are being processes in MRTDS, the framework based on standard MapReduce is explained in fig1. The solid arrow shows data flows in canonical MapReduce framework. The iteration of the MapReduce controlled by the AL driver. For handling the iterations the data flows shown by the curve arrow. AL is dispatched from Driver to all workers including Mappers and Reducers via the distributed cache mechanism. The value of AL varies in Driver according to the output of the IGPL Initialization or IGPL Update jobs. The amount of such data is extremely small compared with data sets that will be anonymized, the data can be efficiently transmitted between Driver and workers Hadoop [14] used as an open-source implementation of MapReduce, for the implementation of MRTDS. Since most of Map and Reduce functions need to access current anonymization level AL, distributed cache mechanism is use to pass the content of AL to each Mapper or Reducer node as shown in Fig. 1. Hadoop provides the mechanism to set simple global variables for Mappers and Reducers. The division of hash function in shuffle phase is modified because the two jobs require that the key-value pairs with the same key:p field rather than entire key should go to the same Reducer.To reduce communication traffic, MRTDS exploits combiner mechanism that aggregates the key-value pairs with the same key into one on the nodes running Map functions. To further decrease the traffics, MD5 (Message Digest Algorithm) is employed to compress the records transmitted for anonymity.

5. PERFORMANCE ANALYSIS

Method	Advantages	Limitation
Two Phase Top Down Specialization	Parallelization & solves scalabilty problem.	Scalability problem
MRTDS Framework	Reduce Communication Traffic	Data splitting cause transmission overhead.

6. CONCLUSION

The investigation of scalability problem of large-scale data anonymization by TDS, and presents a scalable two-phase TDS approach using MapReduce on cloud. The data set are divided and anonymized in parallel in the first phase, producing intermediate results. Map Reduce applied on cloud for data anonymization and deliberately designed a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. With this approach the scalability and efficiency of TDS are improved significantly over existing approaches.

The future scope of this work in cloud surrounding, preserving privacy for analysis of data, mining and distribution is very challenging issues for research due to huge data sets requires search. For data anonymization investigation of adoption approach to generalization of bottom-up algorithm.

7. REFERENCES

[1] Xuyun Zhang, Laurence T. Yang, Chang Liu, Jinjun Chen,“A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud,“ IEEE Trans. Parallel and Distributed Systems, vol. 25, No. 2, Feb 2014.

[2] OpenStack, http://openstack.org/, 2013.

[3] S.Chaudhari,“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.

[4] 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.

[5] 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.

[6] N. Mohammed, B.C. Fung, and M. Debbabi, "Anonymity Meets Game Theory: Secure Data Integration with Malicious Partici-pants," VLDB J., vol. 20, no. 4, pp. 567-588, 2011.

[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] 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] B. 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.

[10] 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.

[11] 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.

[12] KVM, http://www.linux-kvm.org/page/Main_page, 2013.

[13] UCI Machine Learning Repository, ftp://ftp.ics.uci.edu/pub/machine-learnng-databases/,

[14] Apache, "Hadoop,"http://hadoop.apache.org , 2013.

[15] Amazon Web Services, "Amazon Elastic MapReduce," http://aws.amazon.com/elasticmapreduce/, 2013.

[16] L.Sweeney, "k-Anonymity: A Model for Protecting Privacy,"int'l J. Uncertainty, Fuzziness and Knowledge-Based System, vol. 10,no. 5, pp. 557-570, 2002.

[17] Y. Bu, B. Howe, M. Balazinska, and M. Ernst, "The Hadoop Approach to Large-Scale Iterative Data nalysis," VLDB J., vol. 21, no. 2,pp. 169-190, 2012.

[18] W. Jiang and C. Clifton, " A Secure Distributed Framework for Achieving k-Anonymity," VLDB J., vol. 15, no. 4, pp. 316-333, 2006.

[19] V. Borkar, M.J. Carey, and C. Li, "Inside 'Big Data Database Technology (EDBT '12), pp. 3-14, 2012. Management': Ogres, Onions, or Parfaits?," Proc. 15th Int'l Conf. Extending Database Techonology (EDBT '12), pp.3-14, 2012.

[20] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional K-Anonymity," Proc. 22nd Int'l Conf. Data Eng. (ICDE '06), 2006.