A Scalable Two Phase Top Down Specialization Approach For Data Anonymization Using Mapreduce On Cloud

Sameesha Vs/Cse [#1] ,

Anna University Chennai, Veerammal Engineering College, Dindigul(Dt),Tamilnadu,India

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. Anonymizing data sets via generalization to satisfy certain privacy requirements such as k anonymity is a widely used category of privacy preserving techniques. At present, the scale of data in many cloud applications increases tremendously in accordance with the Big Data trend, thereby making it a challenge for commonly used software tools to capture, manage, and process such large-scale data within a tolerable elapsed time. As a result, it is a challenge for existing anonymization approaches to achieve privacy preservation on privacy-sensitive large-scale data sets due to their insufficiency of scalability. In this paper, we propose 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 approach, we deliberately design a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. Experimental evaluation results demonstrate that with our approach, the scalability and efficiency of TDS can be significantly improved over existing approaches.

I. INTRODUCTION:

Cloud computing is one of the most pre-dominant paradigm in recent trends for computing and storing purposes. Data security and privacy of data is one of the major concern in the cloud computing. Data anonymization has been extensively studied and widely adopted method for privacy preserving in data publishing and sharing methods. Data anonymization is preventing showing up of sensitive data for owner’s data record to mitigate unidentified Risk. The privacy of individual can be adequately maintained while some aggregate information is shared to data user for data analysis and data mining. The proposed method is generalized method data anonymization using Map Reduce on cloud. Here we Two Phase Top Down specialization. In First phase, original data set is partitioned into group of smaller dataset and they are anonymized and intermediate result is produced. In second phase, intermediate result first is further anonymized to achieve persistent data set. And the data is presented in generalized form using Generalized Approach

SCOPE OF THE PROJECT

Recently data privacy preservation has been extensively studied and investigated. Le Fever et.al has addressed about scalability of anonymization algorithm via introducing scalable decision tree and the sampling technique, and lwuchkwuet.al proposed R-tree based index approach by building a spatial index over data sets, achieving high efficiency. However the approach aim at multidimensional generalization which fail to work in Top Down Specialization[TDS]. Fung et.al proposed some TDS approach that produce anonymize data set with data exploration problem. A data structure taxonomy indexed partition [TIPS] is exploited to improve efficiency of TDS but it fails to handle large data set. But this approach is centralized leasing to in adequacy of large data set. Several distributed algorithm are proposed to preserve privacy of multiple data set retained by multiple parties, Jiang et al proposed distributed algorithm to anonymization to vertical portioned data. However, the above algorithms mainly based on secure anonymization and integration. But our aim is scalability issue of TDS anonymization. Further, Zhang et al leveraged Map Reduce itself to automatically partition the computation job in term of security level protecting data and further processed by other Map Reduce itself to anonymize large scale data before further processed by other Map Reduce job, arriving at privacy preservation.

I. PROBLEM DEFINITION:

We analyze the scalability problem of existing TDS approaches when handling large-scale data sets on cloud. The centralized TDS approaches in [12],

[20], and [21] exploits the data structure TIPS to improve the scalability and efficiency by indexing anonymous data records and retaining statistical information in TIPS. The data structure speeds up the specialization process because indexing structure avoids frequently scanning entire data sets and storing statistical results circumvents recomputation overheads. On the other hand, 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 considerable memory. Moreover, the overheads incurred by maintaining the linkage structure and updating the statistic information will be huge when date sets become large. Hence, centralized 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 [12].Unfortunately, this assumption often fails to hold in most data-intensive cloud applications nowadays. In cloud environments, computation is provisioned in the form of virtual machines (VMs). Usually, cloud compute servicesoffer several flavors of VMs. As a result, the centralized approaches are difficult in handling large-scale data sets well on cloud using just one single VM even if the VM has the highest computation and storage capability. A distributed TDS approach [20] is proposed to address the distributed anonymization problem which mainly concerns privacy protection against other parties, rather than scalability issues. Further, the approach only employs information gain, rather than its combination with privacy loss, as the search metric when determining the best specializations. As pointed out in [12], a TDS algorithm without considering privacy loss probably chooses a specialization that leads to a quick violation of anonymity requirements. Hence, the distributed algorithm fails to produce anonymous data sets exposing the same data utility as centralized ones. Besides, the issues like communication protocols and fault tolerance must be kept in mind when designing such distributed algorithms. As such, it is inappropriate to leverage existing distributed algorithms to solve the scalability problem of TDS.

II. PROPOSED SYSTEM

In this system, we propose 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 approach, we deliberately design a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. This approach get input data"s and split into the small data sets. Then we apply the ANONYMIZATION on small data sets to get intermediate result. Then small data sets are merge and again apply
the ANONYMIZATION. We analyze the each and every data set sensitive field and give priority for this sensitive field. Then we apply ANONYMIZATION on this sensitive field only depending upon the scheduling..

III. MODULE DESCRIPTION:

The Project mainly focuses on five modules, which are completely inter-related to each other. The descriptions about the modules are given below

1.DATA PARTITION
2. ANONYMIZATION
3. MERGING
4. SPECIALIZATION
5. OBS

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 getting 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. METHODS AND ALGORITHMS:

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.

```
Algorithm 1. Sketch of two phase TDS
    Input: Data set F, anonymity parameters g, and the number of partition p.
    Output: Anonymous data set F
    1: Partition F into Fi, 1≤ i ≤ p.
    2: Execute MRTDS(Fi, , )→ , 1≤ i ≤ p in Parallel as multiple MapReduce jobs.
    3: Merge all intermediate anonymization levels into one, Merge( , )→
    4: Execute MRTDS(F,k, )→ to achieve k-anonymity.
    5: Specialize F according to , Ouput F
```

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.

Anonymization Level Merging

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

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
    Output: Anonymous record.
    Map: Construct anonymous record using sensitive value and partition.
    Reduce:emit sum.
```

MAPREDUCE VERSION OF CENTRALIZED 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.

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.

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. IGPL Update Job requires less computation and consumes less net work bandwidth. Thus current is more efficient than latter.

MRTDS FRAMEWORK

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 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.

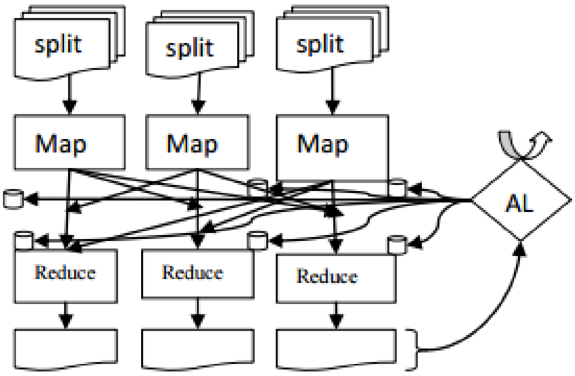

Figure 1. MRTDS Overview

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

CONCLUSION:

In this paper, we have investigated the scalability problem of large-scale data anonymization by TDS, and proposed a highly scalable two-phase TDS approach using MapReduce on cloud. Data sets are partitioned and anonymized in parallel in the first phase, producing intermediate results. Then, the intermediate results are merged and further anonymized to produce consistent k-anonymous data sets in the second phase. 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. Experimental results on real-world data sets have demonstrated that with our approach, the scalability and efficiency of TDS are improved significantly over existing approaches. In cloud environment, the privacy preservation for data analysis, share and mining is a challenging research issue due to increasingly larger volumes of data sets, thereby requiring intensive investigation. We will investigate the adoption of our approach to the bottom-up generalization algorithms for data anonymization. Based on the contributions herein, we plan to further explore the next step on scalable privacy preservation aware analysis and scheduling on large-scale data sets. Optimized balanced scheduling strategies are expected to be developed towards overall scalable privacy preservation aware data set scheduling.

REFERENCES:

[1] X. Zhang, L.T. Yang, 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, In press, 2013.

[2]. K. LeFevre, D.J. DeWitt and R. Ramakrishnan, “Workload-aware anonymization techniques for large-scale datasets,” ACM Transactions on Database Systems, vol. 33, no. 3, pp. 1-47, 2008.

[3]. T. Iwuchukwu and J.F. Naughton, “K-anonymization as spatial indexing: Toward scalable and incremental anonymization,” Proc. the 33rd International Conference on Very Large Data Bases (VLDB'07), pp. 746- 757, 2007.

[4]. J. Dean and S. Ghemawat, “Mapreduce: A flexible data processing tool,” Communications of the ACM, vol. 53, no. 1, pp. 72-77, 2010.

[5]. K.-H. Lee, Y.-J. Lee, H. Choi, Y.D. Chung and B. Moon, “Parallel data processing with mapreduce: A survey,” ACM SIGMOD Record, vol. 40, no. 4, pp. 11-20, 2012.

[6]. 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.

[7]. Amazon Web Services, “Amazon elastic mapreduce(amazon emr),” http://aws.amazon.com/ elasticmapreduce/, accessed on 10 June, 2013.

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

[9]. B.C.M. Fung, K. Wang, R. Chen and P.S. Yu, “Privacy-Preserving Data Publishing: A Survey of Recent Developments,” ACM Comput. Surv., vol. 42, no. 4, pp. 1-53, 2010.

[10]. Geherke, J. 2006. Models and methods for privacy-preserving data publishing and analysis. Tutorial at the 12th ACM SIGKDD.

[11]. Chaum, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Comm. ACM 24, 2, 84–88.

[12] T. Bozkaya and Z.M. O ¨ zsoyoglu, “Indexing Large Metric Spaces for Similarity Search Queries,”