ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using Map Reduce On Cloud

Shweta Sunil Bhand, Prof.J.L.Chaudhari

PG Scholar, Department of Computer Engineering, JSPM'S BhivarabaiSawant Institute of Technology & Research, Pune, Maharashtra, India

Department of Computer Engineering, JSPM'S BhivarabaiSawant Institute of Technology & Research, Pune, Maharashtra, India

ABSTRACT: Releasing person-specific data in its most specific state poses a threat to individual privacy. This paper presents a practical and productive algorithm for determining a abstract version of data that masks sensitive information and remains useful for standardizing organization. The classification of data is implemented by specializing or detailing the level of information in a top-down manner until a minimum privacy requirement is compromised. This top-down specialization is practical and efficient for handling both definitive and continuous attributes. Our method exploits the scenario that data usually contains redundant structures for classification. While generalization may remove few structures, other structures emerge to help. Our results show that standard of classification can be preserved even for highly prohibitive privacy requirements. This work has great applications to both public and private sectors that share information for mutual advantage and productivity.

KEYWORDS: Data anonymization; top-down specialization; MapReduce; cloud;privacy preservation

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. Dataanonymization 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.

A highly scalable two-phase TDS approach for data anonymization based on MapReduce on cloud. To make use of the parallel capability of MapReduce on cloud, classification required in an anonymization process is split into two phases. In the first one, original datasets are partitioned into a group of small datasets, and those datasets are anonymized in parallel, creating intermediately results. In the second one, the intermediate results are aggregated into one, and further anonymized to achieve consistent k-anonymous data sets. It leveragesMapReduce to accomplish the concrete computation in both phases. A group of MapReducejobs are deliberately designed and coordinated to perform specializations on data sets collaboratively. It evaluate the approach by conducting experiments on real-world data sets. Experimental results show that with the approach, the scalability and efficiency of TDS can be improved. It evaluate the approach by conducting experiments on real-world data sets. Experimental results demonstrate that with the approach, the scalability and efficiency of TDS can be improved significantly over existing approaches. The major contributions of the research are threefold. Firstly, it creatively applyMapReduce on cloud to TDS for data anonymization and deliberately design a group of innovative MapReduce jobs to concretely accomplish the specializations in a highly scalable fashion. Secondly, it propose a two-phase TDS approach to gain high scalability via allowing specializations to be conducted on multiple data partitions in parallel during the first phase.

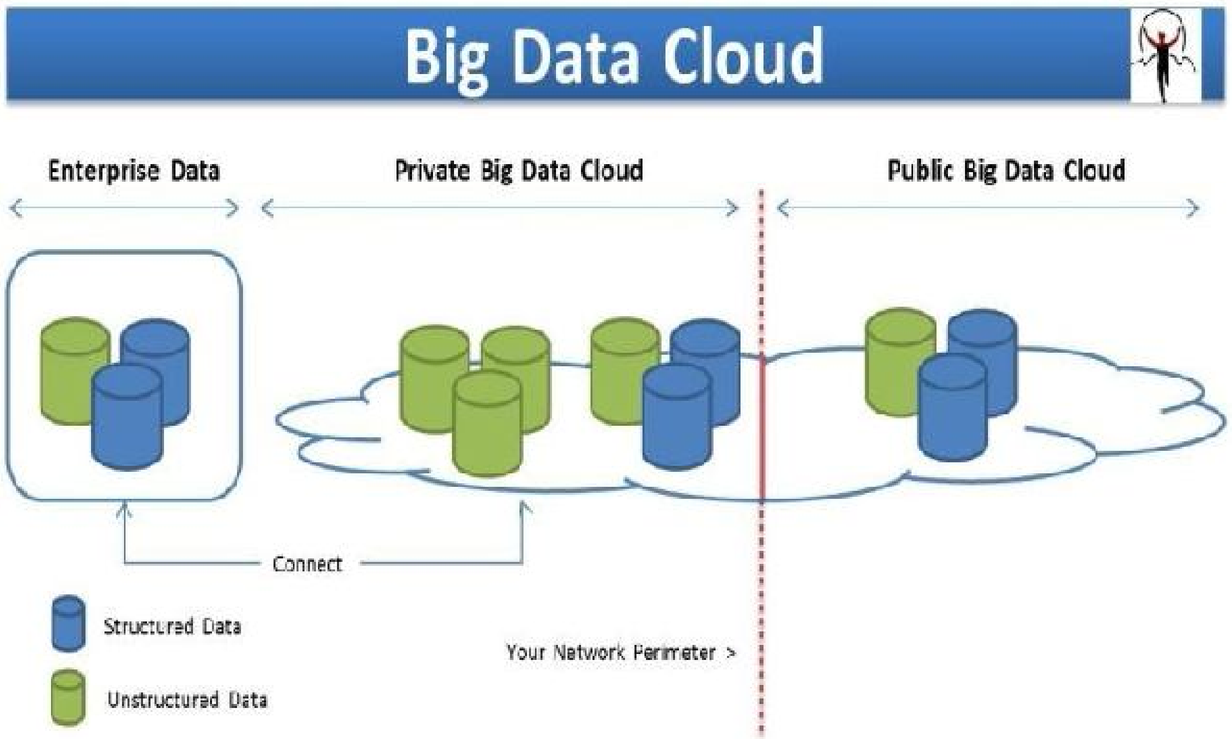

Fig1.BigData on cloud

II. RELATED WORKS

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 lwuchkwu et.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.

III. PROPOSED SYSTEM

Two Phase Top DownSpecialization: A top-down approach (also known as stepwise design and in some cases used as a synonym of decomposition) is essentially the breaking down of a system to gain insight into its composite sub-systems.

In a top-down method an overview of the system is devised, specifying but not detailing any first-level subsystems. Individual subsystem is then refined in yet major detail, sometimes in many additional subsystem levels, until the entire specification is deduced to base elements. A top-down model is specified with the assistance of "black boxes", these make it easier to update. However, black boxes may fail to elucidate elementary mechanisms or be detailed enough to realistically verify the model. Top down approach begins with the big picture. It breaks down from there into smaller elements.

Two-Phase Top-Down Specialization (TPTDS) approach to conduct the computation required in TDS in a highly scalable and efficient fashion. The two phases of the approach are based on the two levels of parallelization provisioned by MapReduce on cloud. Actually, MapReduce on cloud has two levels of parallelization, 1) job level and 2) task level. Job level parallelization means the multiple MapReduce jobs can be executed simultaneously to make full use of cloud infrastructure resources. Aggregated with cloud, MapReduce becomes more powerful and elastic as cloud can offer infrastructure resources on demand, e.g., Amazon Elastic MapReduce service. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data splits. To achieve high scalability, parallelizing multiple jobs on data partitions in the first phase, but the resultant anonymization levels are not similar. To obtain ultimate consistent anonymous data sets, the second phase is required to integrate the intermediate results and further anonymize entire data sets. Details are formulated as follows. All intermediate anonymization levels are merged into one in the second phase. The merging of anonymization levels is done by merging cuts. Precisely, let in and in be two cuts of an attribute. There are domain values and that satisfy one of the three conditions is identical to is more general than is more specific than. To ensure that the merged intermediate anonymization level never breaks privacy requirements, the more general one is chosen as the merged one, e.g., will be selected if is more general than or identical to . For the case of multiple anonymization levels, it can merge them in the same way iteratively.

I. Advantages:

Top-down (aka symbolic) approach

- Hierarchically organized (top-down) architecture
- All the necessary knowledge is pre-programmed, i.e. already present - in the knowledge base.
- Analysis/computation includes creating, manipulating and linking symbols (hence propositional and predicate-calculus approach).
- "Serial executive" can be looked as the natural rule-interpreter which acts on the parallel-processing unconscious intuitive processor.
- Thus the program behaves better at relatively high-level tasks such as language processing aka NLP - it is consistent with currently accepted theories of language aquisition which assume some high-level modularity.
- Disadvantages:
- The solution gives limited coverage in the first phases.
- A average percentage of user accounts are managed in the first phases.
- You can develop custom adapters at an early stage.
- The support and business will not realize the benefit of the solution as rapidly.
- The implementation cost is likely to be greater.

II. Algorithm:

```
Algorithm TDS
Initialize every value in T to the top most value.
Initialize Cuti to include the top most value.
while some x ∈∪Cuti is valid and beneficial do
Find the Best specialization from ∪Cuti.
Perform Best on T and update ∪Cuti.
Update Score(x) and validity for x ∈∪Cuti.
end while
```

```
9     return Generalized T and ∪Cuti.
```

III. Direct Anonymization Algorithm DA (D,I,k,m)

```
1. Scan D and create count-tree
2. InitializeCout
3. For each node v in preorder count-tree tranversal do
4.        If the item of v has been generalized in Cout then
5.        backtrack
6.        if v is a leaf node and v.count<k then
7.        J:=itemset corresponding to v
8.        Find generalization of items in J that make J k-anonymous
9.        Merge generalization rules with Cout
10.       Backtrack to longest prefix of path J,wherein no item has   been generalized in Cout
11. ReturnCout
12.       for i :=1 to Count do
13.       Initialize count=0
14.       scan each transactions in Cout
15.       Separate each item in a transaction and store it in p
16.       Increment count
17.       for j:=1 to count do
18.       For all g belongs Cout do
19.       Compare each item of p with that of Cout
20.       If all items of i equal to cout
21.       Increment the r
22.       Ifka equal to r then backtrack to i
23 else if r greater than ka then get the index position of the similar transactions
24.       make them NULL until ka equal to r
25.       else update the transactions in database
```

IV. METHODOLOGY

A MapReduce program is consists of a Map() procedure that performs filtering and sorting (such as sorting students by email into queues, one queue for each email) and a Reduce() procedure that performs a summary operation (such as counting the number of students in every queue, resulting name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, executing the various tasks in parallel, keeping all communications and data transfers between the various parts of the system, and giving for redundancy and fault tolerance.

The model is inspired by the map and reduces functions commonly used in programming, even though their purpose in the MapReduce framework is not the same as in their original forms. The main contributions of the MapReduce framework are not the actual map and reduce functions, but the extensibility and fault-tolerance gained for a variety of applications by optimizing the execution engine once. A single-threaded implementation of MapReduce will usually not be faster than a traditional implementation. When the optimized distributed shuffle operation (which reduces network communication cost) and fault tolerance features of the MapReduce framework come into play, is the use of this model beneficial. MapReduce libraries have been written in multiple programming languages, with separate levels of optimization. A famous open-source implementation is Apache Hadoop. The name MapReduce originally referred to the proprietary Google technology but has since been genericized.

The Hadoop distributed file system (HDFS) is a scalable, distributed and portable file-system written in Java for the Hadoop framework. A Hadoop cluster has typically a single namenode plus a cluster of data nodes, redundancy options are available for the namenode due to its importance. Each datanode serves blocks of data over the network using a block protocol specific to HDFS. The file system makes use of TCP/IP sockets for communication. Clients use RPC(remote procedure call) to communicate between each other. HDFS stores large files (typically in the range of gigabytes to terabytes) across no of machines. It achieves reliability by replicating the data across hosts, and hence theoretically does not need RAID storage on hosts (but to improve I/O performance some RAID configurations are still useful). With default replication value, 3, data is stored on three nodes: two on the same rack, and one on a separate rack. Data nodes can communicate with each other to adjust data, to move copies around, and to keep the replication of data. HDFS is not POSIX-compliant, since the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The advantage of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.

First, many existing clustering algorithms (e.g., k- means) requires the calculation ofthe “centroids”. But there is no notion of “centroids” in our setting where each attribute for msa data point in the clustering space. Second, k-medoid method is so reliable to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the kmedoidmethod. Existing anonymization algorithms can be used for column generalization e.g. Mondrian. The algorithms can be verified on the subtable containing only attributes in one column to ensure the anonymity requirement. Existing data analysis (e.g., query answering) methods can be easily used on the sliced data. Current privacy measures for membership disclosure protection include differential privacy and presence.

V. CONCLUSION

Privacy preserving data analysis and data publishing are becoming serious problems in today’s ongoing world. That’s why different approaches of data anonymization techniques are proposed. To the best of our knowledge, TDS approach using MapReduce are applied on cloud to data anonymization and deliberately designed a group of innovativeMapReduce jobs to concretely accomplish the specialization computation in a highly scalable way.

VI. FUTUTRE SCOPE

There are possible ways of data anonymization in which the current situation may be improved and next generation solutions may be developed. As future work a combination of top-down and bottom up approach generalization is contributed for data anonymization in which data Generalization hierarchy is utilized for anonymization.

REFERENCES

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

[3]. 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.

[4]. 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.

[5]. D. Zissis and D. Lekkas, “Addressing Cloud Computing Security Issues,” Fut. Gener. Comput.Syst., vol. 28, no. 3, pp. 583-592, 2011.

[6]. 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.

[7]. L. Hsiao-Ying and W.G. Tzeng, “A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding,” IEEETrans. Parallel Distrib.Syst., vol. 23, no. 6, pp. 995-1003, 2012.

[8]. 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.

[9]. 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.

[10].Microsoft HealthVault, http://www.microsoft.com/health/ww/ roducts/Pages/healthvault.aspx, accessed on: Jan. 05, 2013.