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.5, Special Issue 2, April 2017*

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

Privacy Preservation of Big Data In Cloud Using Two-Phase And K-Anonymization

Amrutha B, Prajna Sharma K S, Manjunath K, Renuka G.B

Lecturer, Department of Computer Science Badra Institute of management and Studies, Bangalore, India

ABSTRACT— This paper discusses approach to anonymizing large data sets. A large number of cloud services require users to share private data like health records for data analysis. Now a days, the scale of data in many cloud application increases tremendously in accordance with big data trends, thereby making it a challenge for commonly used software tools to capture, manage and process such large-scale data within a elapsed time. So to achieve privacy preservation on privacy-sensitive large-scale data sets due to their insufficient of scalability. Anonymizing datasets through generalization, suppression and other algorithms to satisfy k- anonymity are widely used technique for privacy preserving So we propose a scalable two-phase top-down generalization and specialization approach on cloud to anonymize large-scale data sets using Hadoop offering such as HDFS, MapReduce framework and k-anonymity. Our approach results in high scalability with respect to data volume and efficiency in processing and anonymizing over the existing approaches. Performance evolution demonstrate that our approach scalability and efficiency of can be significantly improved over existing approaches.

KEYWORDS: Data Anonymization, Privacy Preserving, cloud, Hadoop, Map reduce..

I. INTRODUCTION

Information sharing has become part of the routine activity of many individuals, companies, organizations, and government agencies. Such Information sharing is subject to constraints imposed by privacy of individuals or data subjects as well as data confidentiality of institutions or data providers and also with the wide adoption of online cloud services and proliferation of mobile devices, the privacy concern about processing on and sharing of sensitive personal information is increasing. To reduce these risks various proposals have been designed for privacy preserving in data publishing. In this survey we will briefly review recent research on data privacy preservation and privacy protection in MapReduce and cloud computing environments, and survey current existing techniques, and summarize the advantage and disadvantage of these approaches.

.

Existing technical approaches for preserving the privacy of data sets stored in cloud mainly include encryption and anonymization. On one hand, encrypting all data sets, a straightforward and effective approach. However, processing on encrypted data sets efficiently is quite a challenging task, because most existing applications only run on unencrypted data sets. Although recent progress has been made in homomorphic encryption which theoretically allows performing computation on encrypted data sets, applying current algorithms are rather expensive due to their inefficiency. On the other hand, partial information of data sets, e.g., aggregate information, is required to expose to data users in most cloud applications like data mining and analytics. In such cases, data sets are anonymized rather than encrypted to ensure both data utility and privacy preserving.

Cloud systems provides massive computation power and storage capacity that enable users to deploy applications without infrastructure investment. Because of its salient features, cloud is promising for users to handle the big data processing pipeline with its elastic and economical infrastructural resources. For instance, MapReduce , extensively studied and widely adopted large-scale data processing paradigm, is incorporated with cloud infrastructure to provide more flexible, scalable, and cost-effective computation for big data processing. A typical example is the Amazon Elastic MapReduce service.

Data anonymization refers to hiding identity and/or sensitive data for owners of data records. The privacy of an individual can be effectively preserved while certain aggregate information is exposed to data users for diverse analysis and mining. A variety of anonymization algorithms with different anonymization operations have been proposed. Large-scale data processing frameworks like MapReduce have been integrated with cloud to provide powerful computation capability for applications. So, it is promising to adopt such frameworks to address the scalability problem of anonymizing large-scale data for privacy preservation. In this paper the MapReduce, a widely-adopted parallel data processing framework, to address the scalability problem of the Top-Down Specialization (TDS) approach for large-scale data anonymization and propose a highly scalable two-phase TDS approach for data anonymization based on MapReduce on cloud.

Hadoop MapReduce is a software framework adopted to process vast amounts of data (multi-terabyte data-sets) in-parallel to address scalability, fault-tolerant manner[5]. To store the data onto the cloud there should be some file system in order to store and retrieve the data, since the traditional file systems cannot hold large data sets and due to its redundancy Hadoop Distributed file system(HDFS) has been introduced. HDFS is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. The MapReduce framework and the HDFS are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

Top-down specialization (TDS), is for information and privacy preservation that is done by classifying the user data and generalization. Both centralized and distributed TDS are available, centralized TDS leads to inadequacy in handling large set of data and the distributed TDS preserve privacy of multiple data sets[6].

II.RELATED WORK

LeFevreet al. [8] addressed the scalability problem of anonymization algorithms via introducing scalable decision trees and sampling techniques. Iwuchukwu et al. [9] proposed an R-tree index-based approach by building a spatial index over data sets, achieving high efficiency. However, the above approaches aim at multidimensional generalization, thereby failing to work in the Top- Down Specialization (TDS) approach. Fung et al. [10, 11] proposed the Centralized 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 the approach is centralized, leading to its inadequacy in handling large-scale data sets[7].

Several distributed algorithms are proposed to preserve privacy. Jiang et al. [13] and Mohammed et al. [11] proposed distributed algorithms to anonymize vertically partitioned data from different data sources without disclosing privacy information from one party to another. Jurczyk et al. [6] and Mohammed et al. [10] 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. Our research mainly focuses on the scalability issue of TDS anonymization, and is therefore orthogonal and complementary to them.

III.METHODOLOGY

In this approach we are anonymizing the data using k-anonymity and processing the data using MapReduce. The top down specialization is an iterative process used for information and privacy preservation. In the first phase data sets are partitioned into smaller units, these units are run on TPTD function. Here all the data sets partitioned are run parallel using MapReduce. TPTD anonymizes the all the partitioned units to generate the intermediate anonymization levels without violating the k-anonymity during specialization. In the second phase, all the intermediate anonymization levels are merged. The whole merged set is again run on TPTD to achieve k-anonymity. The resultant merged and anonymized data is the final output of our approach.

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.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

Algorithm 1: Two Phase Top-Down Approach

Input: Data set X, Number of partition p, anonymity parameter a, a'

Output: Anonymization data X*

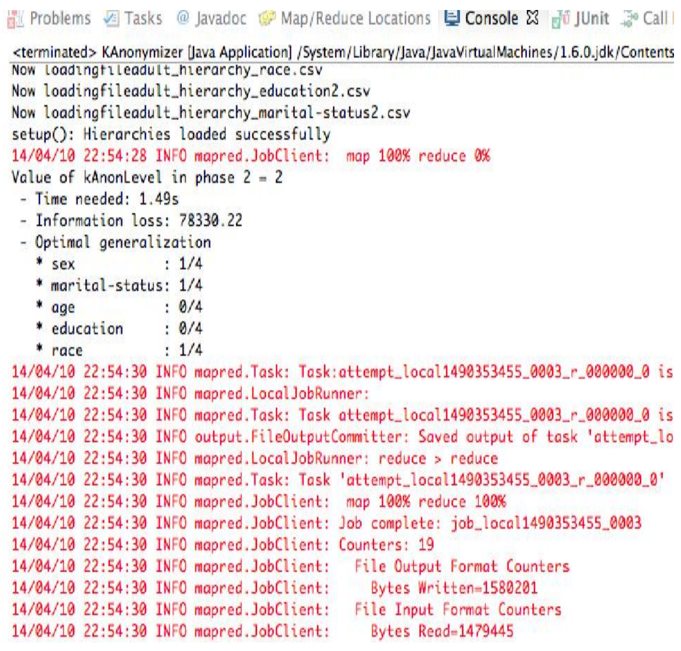
```
1. Partition X into $X_i$ , $1 \leq i \leq p$
2. Run TPTD($X_i$,$a^I$,$AL^0$) ->$AL_i'$,$1 \leq i \leq p$ in parallel as multiple MapReduce jobs.
3. Merge all intermediate anonymization levels into one, merge ($AL'_1$,$AL'_2$… $AL'_p$) ->$AL^I$
4. Run TPTD($X_i$,a,$AL^I$) ->$AL^*$ to achieve k-anonymity
5. Specialize X w.r.t $AL^*$
```

3.1 K- anonymity

K-anonymity is one of the most emerging techniques for privacy preserving. K-anonymity details with each tuple in a table are indistinguishable from the other tuples i.e.

"If RT(A_1,A_2…A_n) be a released table, and QI be a quasi-identifier associated with it. RT is said to satisfy k-anonymity with respect to QI if each sequence of values in RT[QI] appears at least with k occurrences in RT[QI][1]".

Quasi-identifiers are the tuples other than the personal identifying information(Name, DOB, phone number, email) and the sensitive attributes(Disease, type of transaction) such as ZIP, Age, Gender etc. For achieving k-anonymity we need to do generalization and suppression. Generalization is done by replacing quasi-identifiers with less specific, but semantically consistent value. Suppression is done when generalization causes too much information loss.



```
<terminated> KAnonymizer [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents
Now loadingfileadult_hierarchy_race.csv
Now loadingfileadult_hierarchy_education2.csv
Now loadingfileadult_hierarchy_marital-status2.csv
setup(): Hierarchies loaded successfully
14/04/10 22:54:28 INFO mapred.JobClient:  map 100% reduce 0%
Value of kAnonLevel in phase 2 = 2
 - Time needed: 1.49s
 - Information loss: 78330.22
 - Optimal generalization
   * sex           : 1/4
   * marital-status: 1/4
   * age           : 0/4
   * education     : 0/4
   * race          : 1/4
14/04/10 22:54:30 INFO mapred.Task: Task:attempt_local1490353455_0003_r_000000_0 is
14/04/10 22:54:30 INFO mapred.LocalJobRunner:
14/04/10 22:54:30 INFO mapred.Task: Task attempt_local1490353455_0003_r_000000_0 is
14/04/10 22:54:30 INFO output.FileOutputCommitter: Saved output of task 'attempt_lo
14/04/10 22:54:30 INFO mapred.LocalJobRunner: reduce > reduce
14/04/10 22:54:30 INFO mapred.Task: Task 'attempt_local1490353455_0003_r_000000_0'
14/04/10 22:54:30 INFO mapred.JobClient:  map 100% reduce 100%
14/04/10 22:54:30 INFO mapred.JobClient: Job complete: job_local1490353455_0003
14/04/10 22:54:30 INFO mapred.JobClient: Counters: 19
14/04/10 22:54:30 INFO mapred.JobClient:   File Output Format Counters
14/04/10 22:54:30 INFO mapred.JobClient:     Bytes Written=1580201
14/04/10 22:54:30 INFO mapred.JobClient:   File Input Format Counters
14/04/10 22:54:30 INFO mapred.JobClient:     Bytes Read=1479445
```

3.2 Hadoop MapReduce

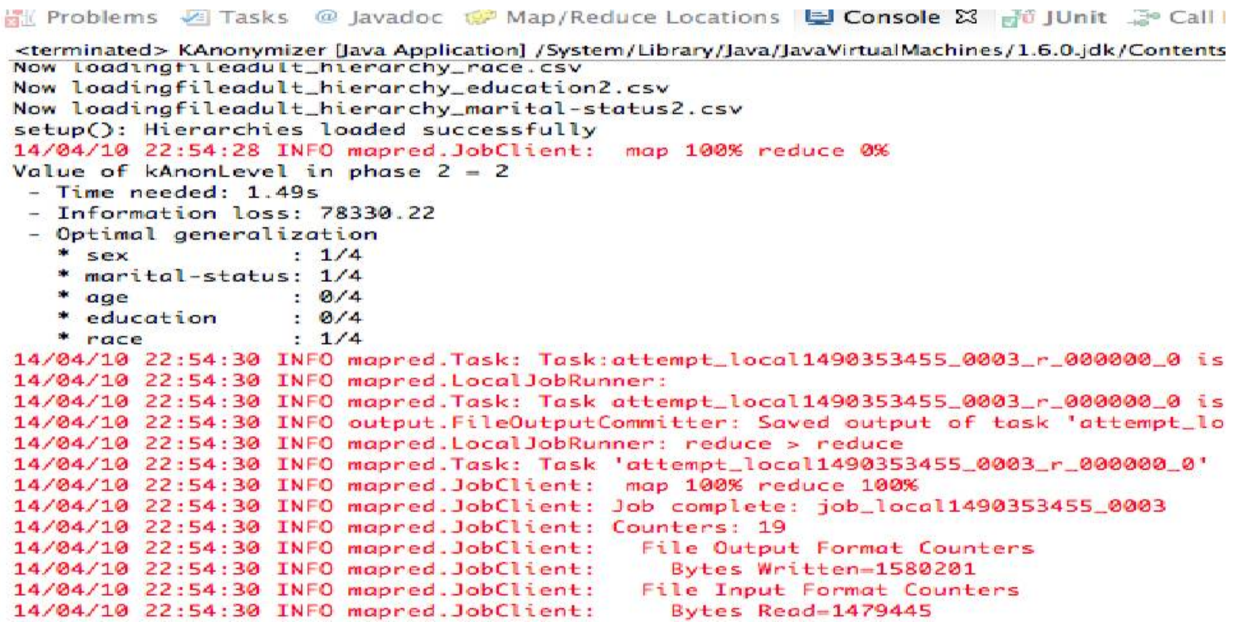
Hadoop is used as distribute cloud computing services, to efficiently process very large data volumes by linking many systems together and work in parallel. The Hadoop MapReduce framework is based on centralized master/slave architecture. A single master server (JobTracker) and several slave servers (TaskTracker) are present in this framework. The JobTracker keeps track of the slave nodes, and provides an interface infrastructure for job submission. The TaskTracker executes on each of the slave nodes where the actual data is normally stored. Users submit MapReduce jobs to the JobTracker, which inserts the jobs into the pending jobs queue and executes them on a FIFO basis. The JobTracker manages the map and reduce task assignments with the TaskTracker's. The TaskTracker's execute the jobs based on the instructions from the JobTracker and handle the data movement between the map and reduce phases, respectively.

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.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

Algorithm 2: MapReduce for data partitioning
Input: Data set (IX_r ,r), r € X , Number of partition p, anonymity parameter a, a'
Output: X_i ,$1\leq i\leq p$
Map: Generate random number R, where $1\leq R\leq p$;
Emits(R, r)
Reduce: For each R, emit(null,list(r)).

Algorithm 3: MapReduce for data specialization
Input: Data set (IX_r ,r), r € X;Anonymization level AL^*.
Output: Anonymization data (r^*, count)
Map: Construct anonymous record $r^*=(p_1,p_2\ldots p_m,sv)$, p_i,
$1\leq i\leq m$, is the parent of a specialization in current AL and is also an ancestor of v_i in r; emit(r^*,count)
Reduce: For each r^*, sum<- ∑count; emit(r^*, sum).



```
<terminated> KAnonymizer [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents
Now loadingfileadult_hierarchy_race.csv
Now loadingfileadult_hierarchy_education2.csv
Now loadingfileadult_hierarchy_marital-status2.csv
setup(): Hierarchies loaded successfully
14/04/10 22:54:28 INFO mapred.JobClient:  map 100% reduce 0%
Value of kAnonLevel in phase 2 = 2
 - Time needed: 1.49s
 - Information loss: 78330.22
 - Optimal generalization
    * sex           : 1/4
    * marital-status: 1/4
    * age           : 0/4
    * education     : 0/4
    * race          : 1/4
14/04/10 22:54:30 INFO mapred.Task: Task:attempt_local1490353455_0003_r_000000_0 is
14/04/10 22:54:30 INFO mapred.LocalJobRunner:
14/04/10 22:54:30 INFO mapred.Task: Task attempt_local1490353455_0003_r_000000_0 is
14/04/10 22:54:30 INFO output.FileOutputCommitter: Saved output of task 'attempt_lo
14/04/10 22:54:30 INFO mapred.LocalJobRunner: reduce > reduce
14/04/10 22:54:30 INFO mapred.Task: Task 'attempt_local1490353455_0003_r_000000_0'
14/04/10 22:54:30 INFO mapred.JobClient:  map 100% reduce 100%
14/04/10 22:54:30 INFO mapred.JobClient: Job complete: job_local1490353455_0003
14/04/10 22:54:30 INFO mapred.JobClient: Counters: 19
14/04/10 22:54:30 INFO mapred.JobClient:   File Output Format Counters
14/04/10 22:54:30 INFO mapred.JobClient:     Bytes Written=1580201
14/04/10 22:54:30 INFO mapred.JobClient:   File Input Format Counters
14/04/10 22:54:30 INFO mapred.JobClient:     Bytes Read=1479445
```

3.3 Two Phase Top-Down Architecture
Figure below represents the two phase top down architecture. Where the input is the patient data and the out is the its anonymized data so that the data miners shouldn't identify the patient.

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

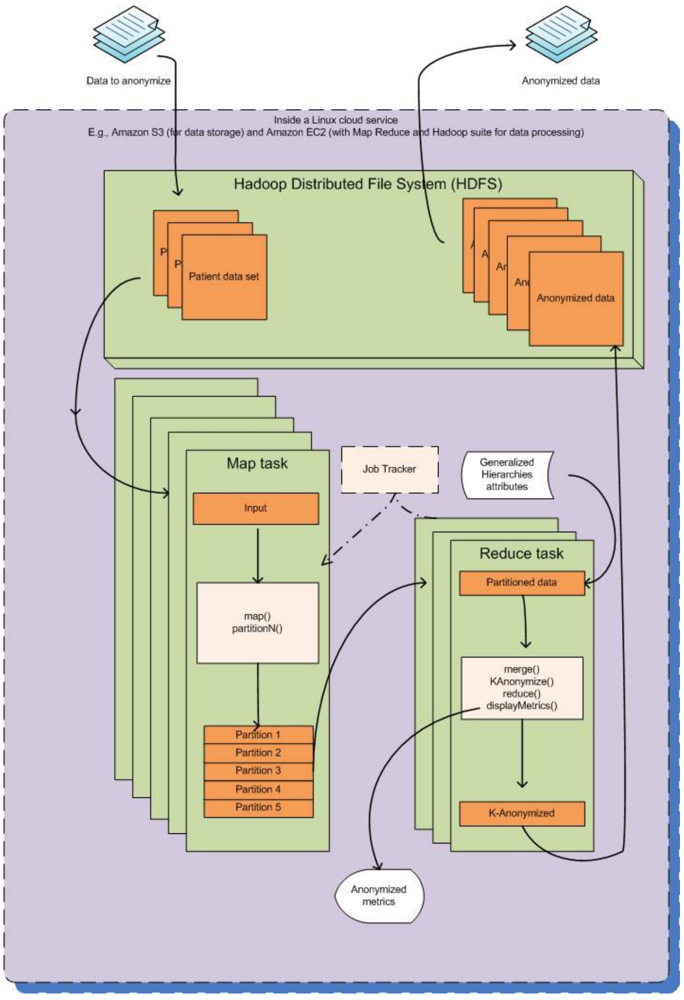
International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India



Initially the patients data is given to the Map tasker where the data is partitioned, all the partitioned data is given as input to the reducer task. The reducer will merge the data and then anonymize, the merged data is rearranged into a metrics. The resultant will be the anonymized data.

IV.EXPERIMENT EVULUATION

A. Apparatus

We adopt Hadoop, an open-source implementation of MapReduce, to implement TPTD. Hadoop provides the mechanism to set simple global variables of Mappers and reducers. The Hadoop configuration is done in fedora and the data set is taken from the DataBase(like Cloud, SQL, MongDB or any unstructured data) using the Hadoop commands. The resultant data will be the anonymized data which can be either returned to the cloud or any vendor whoever require the data.

B. Evaluation process and results

We are conducting two or three groups of experiments. Firstly the execution time of our approach and the centralized top-down specialization w.r.t' data size. Secondly we are monitoring the scalability and effectiveness of our approach by changing the data size X, number of data partitions p and the anonymity parameter k Generally, 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*)..

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

International Journal of Innovative Research in Computer and Communication Engineering

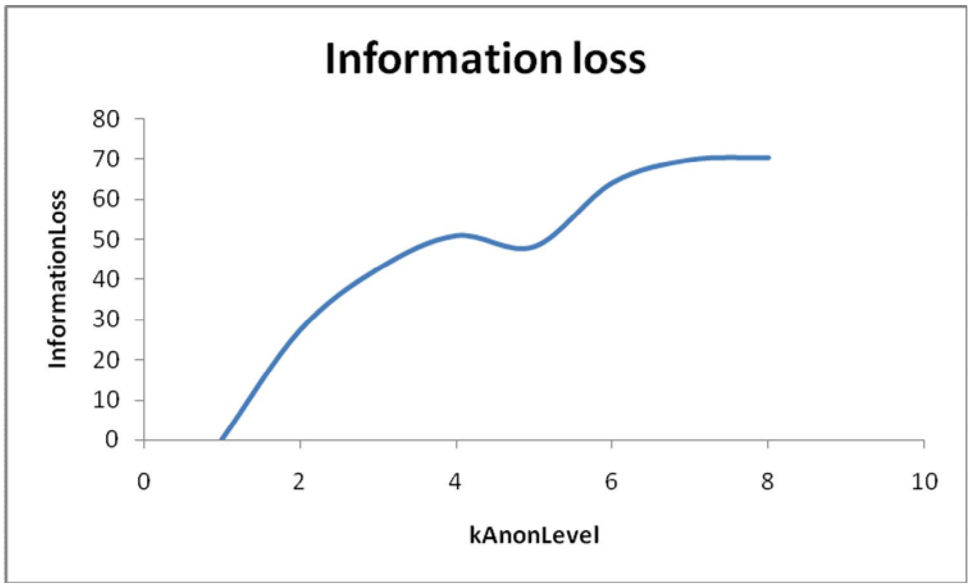
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017

An International Conference on Recent Trends in IT Innovations - Tec'afe 2017

Organized by

Dept. of Computer Science, Garden City University, Bangalore-560049, India

kAnon Level	Information loss
8	70.58
7	70.02
6	64.25
5	48.29
4	51.02
3	42.74
2	27.37
1	0



As a conclusion, all the experimental results demonstrate that our approach significantly improves the scalability
and efficiency of TDS over existing TDS approaches

V.CONCLUSION

We have investigated the scalability problem of large-scale data anonymization by Top-Down Specialization (TDS), and proposed a highly scalable two-phase TDS approach using MapReduce on cloud. Datasets 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 datasets 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 datasets, 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 datasets. Optimized balanced scheduling strategies are expected to be developed towards overall scalable privacy preservation aware dataset scheduling.

Future Enhancement

a. Evaluate the scalability and security across other sample data sets
b. Alternate anonymization techniques can be applied and evaluated
c. Data sets can be partitioned and anonymized in parallel in the first phase (Map), producing intermediate results. Then, the intermediate results can be merged and further anonymized to produce consistent data sets in the second phase (Reduce).

REFERENCES

[1] L. Sweeney, “k-anonymity: a model for protecting privacy”, International Journal on Uncertainty, Fuzziness and Knowledgebased Systems, 2002, pp. 557-570.
[2] A.Machanavajjhala, J.Gehrke, and D.Kifer, et al, “ℓ-diversity: Privacy beyond k-anonymity”, In Proc. of ICDE, Apr.2006.
[3] N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-anonymity and l-Diversity”, In Proc. of ICDE, 2007, pp. 106-115.
[4]. Dean J, Ghemawat S. Mapreduce: a flexible data processing tool. Communications of the ACM 2010; 53(1):72–77. DOI: 10.1145/1629175.1629198.
[5]. 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.
[6]. 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.
[7]. 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.
[8] 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.
[9]. Roy I, Setty STV, Kilzer A, Shmatikov V, Witchel E, “Airavat: Security and Privacy for Mapreduce,” Proceedings of 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10), 2010; 297–312.
[10]. Blass E-O, Pietro RD, Molva R, Önen M, “PRISM-Privacy-Preserving Search in Mapreduce,” Proceedings of the 12th International Conference on Privacy Enhancing Technologies (PETS'12), 2012; 180–200.
[11]. 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.
[12]. Zhang K, Zhou X, Chen Y, Wang X, Ruan Y. Sedic: “Privacy-Aware Data Intensive Computing on Hybrid Clouds,” Proceedings of 18th ACM Conference on Computer and Communications Security (CCS'11), 2011; 515–526.
[13]. Wei W, Juan D, Ting Y, Xiaohui G, “Securemr A Service Integrity Assurance Framework for Mapreduce,” Proceedings of Annual Computer Security Applications Conference (ACSAC '09), 2009; 73–82