International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.41 (2015)

OPTIMIZED SCHEDULING FOR DATA ANONYMIZATION IN CLOUD USING TOP DOWN SPECIALIZATION

[1]M.Karthik Prabu,
PG Scholar,
Department of CSE,
SNS College of Technology,
Coimbatore.

[2]B.Santhosh Kumar,
Assistant professor,
Department of CSE,
SNS College of Technology,
Coimbatore.

[3]Dr.S.Karthik,
Professor and Dean,
Department of CSE,
SNS College of Technology,
Coimbatore.

ABSTRACT:**Cloud computing is a way of providing on demand network access in the shared data computing resources which are accessed by millions of users. The main concern of cloud computing is to provide the privacy and security concerns for the cloud resource user who share their data in the public cloud. Data Anonymization is a method that makes data worthless to anyone except the owner of the data. As the number of users is increased, the amount of data collected is also increased. In big data scalability is the big issue for maintenance. Map reduce technique is used to handle the big data efficiently which will partition the data's into sub levels. In the previous work top down specialization approach is used individually to anonymize the partitioned data. This approach gives poor performance at the certain value of k-anonymity. To overcome this problem, in this work particle swarm optimization technique is introduced to optimize the load balancing by selecting the optimized anonymous solution.**

Keywords: **Cloud, Data Anonymization, Map Reduce, top-down specialization**

1. INTRODUCTION:

Big data is a popular term used to describe the exponential growth and availability of data, both structured and unstructured. The big data is the term of datasets so large and complex that it becomes difficult to process using hand database management tools or traditional data processing applications. Big Data processing is performed through a programming paradigm known as Map reduces. Typically, implementation of the Map Reduce paradigm requires networked attached storage and parallel processing. The fact is that with so much data being generated by so many organizations and users, storage and security simply have to become critical business issues. Ninety per cent of the total data in the world today has been created in the past two years ,and 2014 and beyond will see us generating exponentially larger levels of data .So with more data comes greater threat of attack and greater need for security . Privacy is one of the most concerned issues in cloud computing, and the concern aggravates in the context of cloud computing although some privacy issues are not new to adopt such frameworks to address the scalability problem of anonym zing large-scale data for privacy preservation. In our research, we leverage Map Reduce, a widely adopted parallel data processing framework, to address the scalability problem of the top-down specialization (TDS) approach for large-scale data Anonymization. The TDS approach, offering a good tradeoff between data utility and data consistency, is widely applied for data Anonymization. Most TDS algorithms are centralized, resulting in their inadequacy in handling large-scale data sets. Although some distributed algorithms have been proposed they mainly focus on secure Anonymization of data sets from multiple parties, rather than the scalability aspect. As the Map Reduce computation paradigm is relatively simple, it is still a challenge to design proper Map Reduce jobs for Top-Down Specialization .Data Anonymization has been extensively studied and widely adopted for data privacy preservation in non interactive data publishing and sharing scenarios .Data Anonymization refers to hiding identity and/or sensitive data for owners of data records. Then, 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. However, the scale of data sets that need anonym zing in some cloud applications increases tremendously in accordance with the cloud computing and Big Data trends. Data sets have

become so large those anonyms zing such data sets is becoming a considerable challenge for traditional Anonymization algorithms. The researchers have begun to investigate the scalability problem of large-scale data Anonymization. In this paper, we propose a highly scalable two-phase TDS approach for data Anonymization based on Map Reduce on cloud. To make full use of the parallel capability of Map Reduce on cloud, specializations required in an Anonymization process are split into two phases. In the first one, original data sets are partitioned into a group of smaller data sets, and these data sets are anonym zed in parallel, producing intermediate results. In the second one, the intermediate results are integrated into one, and further anonym zed to achieve consistent k-anonymous data sets. We leverage Map Reduce to accomplish the concrete computation in both phases. A group of Map Reduce jobs is deliberately designed and coordinated to perform specializations on data sets collaboratively. We evaluate our approach by conducting experiments on real-world data sets. Experimental results demonstrate that with our approach, the scalability and efficiency of TDS can be improved significantly over existing approaches.

2. RELATED WORK AND PROBLEM ANALYSIS:

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.A wide verity of privacy models and Anonymization approaches have been put forth to preserve the privacy sensitive informational data sets. Data privacy is one of the most concerned issues because processing large-scale privacy-sensitive data sets often requires computation power provided by public cloud services for big data applications. We studied the scalability issues of existing BUG approaches when handling big data-data sets on cloud. In the proposed work, top down methods are used in order to reach and best anonym zed level. The top down approaches are individually lacks in some parameters which will not give a better accurate result. In our proposed approach are to generate a better optimized output with better accuracy.

3. METHODOLOGY:

3.1Top-Down Specialization

Generally, TDS is an iterative process starting from the topmost domain values in the taxonomy trees of attributes. Each round of iteration consists of three main steps, namely, finding the best specialization, performing specialization and updating values of the search metric for the next round. Such a process is repeated until k-anonymity is violated, to expose the maximum data utility. The goodness of a specialization is measured by a search metric. We adopt the information gain per privacy loss (IGPL), a trade-off metric that considers both the privacy and information requirements, as the search metric in our approach. A specialization with the highest IGPL value is regarded as the best one and selected in each round.

Input: Data set D, anonymity parameters k, k^I and the number of partitions p.

Output: Anonymous data set D*.

1: Partition D into Di, $1 \leq i \leq p$.

2: Execute MRTDSBUG (D_i, k^I,AL^0) -> AL_i , $1 \leq i \leq p$ in parallel as multiple Map Reduce jobs.

3: Merge all intermediate Anonymization levels into one, merge (AL_1, AL_2 , ….AL_p) ->AL^I .

4: Execute MRTDSBUG (D, k, AL^I) -> AL* to achieve k-anonymity.

5: Specialize D according to AL*, Output D*.

3.2 Map Reduce

Hadoop Map Reduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. A Map Reduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. Typically the compute nodes and the storage

nodes are the same, that is, the Map Reduce framework and the Hadoop Distributed File System (see HDFS Architecture Guide) 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 band width across the cluster. The Map Reduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs 'component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. Minimally, applications specify the input/output locations and supply *map* and *reduce* functions via implementations of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the *job configuration*. The Hadoop *job client* then submits the job jar/executable etc.) And configuration to the Job Tracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client. Although the Hadoop framework is implemented in Java TM, Map Reduce applications need not be written in Java.

3.3Data mapping using PSO

The partioned data's will be allocated to the VM in which best optimized anonymization level can be achieved. The optimized allocation of resources is done in our work by using the Particle Swarm Optimization method which is an bio-logically inspired approach. In PSO, each and every possible allocation of data partition into VM will be considered as an particle.

3.4Data anonymization using TDS

Data anonymization is done by using the top down specialization approach. TDS is an iterative process starting from the top most domain values in the taxonomy trees of attributes. Each round of iteration consists of three main steps, namely, finding the best specialization, performing specialization and updating values of the search metric for the next round. Such a process is repeated until k-anonymity is violated, to expose the maximum data utility. The goodness of a specialization is measured by a search metric. We adopt the information gain per privacy loss (IGPL), a tradeoff metric that considers both the privacy and information requirements, as the search metric in our approach. 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 IG (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.

4.CONCLUSION:

This approach gives effective way of preserving privacy information of the user by hiding user's sensitive information. It reduces the computation cost considerably by selecting the optimal Feature set for specialization. Delay can be also reduced. The specialization is applied just for optimal feature set, instead of applying specialization for entire data. So that data Anonymization is also achieved. The Map Reduce Framework is effectively applied on cloud for data Anonymization and shows that scalability and efficiency of centralized BUG are improved significantly over existing approaches. We will investigate the adoption of our approach to the bottom-up generalization algorithms for data Anonymization. Based on the hand-outs Herein, we plan to further discover the next step on scalable privacy preservation aware analysis and scheduling on big-scale data sets.

REFERENCES:

[1].B.C.M. Fung, K. Wang, and P.S. Yu, "Anonym zing Classification Data for Privacy Preservation," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 5, pp. 711-725, May 2007.
[2].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.

[3].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.
[4].W. Jiang and C. Clifton, "A Secure Distributed Framework forAchieving k-Anonymity," VLDB J., vol. 15, no. 4, pp. 316-33,2006.
[5].J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processingon Large Clusters," Comm. ACM, vol. 51, no. 1, pp. 107-113,2008. [6].X. Xiao and Y. Tao, "Anatomy: Simple and Effective PrivacyPreservation," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB'06), pp. 139-150, 2006.
[7].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.
[8].N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," Proc.IEEE INFOCOM, pp. 829-837, 2011.
[9].Y.Chen, X.Wang and Y.Ruan,"Sedic:Privacy-Aware Data Intensive Computing on Hybrid Clouds,"Proc. 18th ACM Conf. Computer and Comm. Security (CCS '11),pp. 515-526, 2011.
[10].N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," Proc.IEEE INFOCOM, pp. 829-837, 2011.
[11]. H. Takabi, J.B.D. Joshi, and G. Ahn, "Security and Privacy challenges in cloud computing environments".Volume no. 8, no. 6, pp. 24-31, Nov. 2010.
[12] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," ACM Computing Surveys, vol. 42, no. 4, pp. 1-53, 2010.