IJRSET 2018 Volume 5, Special-Issue-2 Pages: 14-20

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

AN ADVANCED BOTTOM UP GENERALIZATION APPROACH FOR BIG DATA ON CLOUD

[1] M. Karthik Prabu
[1] Assistant Professor,
[1] Department of Computer Science,
[1] RVS Kumaran Arts and Science College,Dindigul

ABSTRACT: 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. In big data applications, 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. As a result is 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 Advanced Bottom up generalization 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. Original datasets are split up into a group of smaller datasets, and these datasets are anonymized in parallel, producing intermediate results. Then, the intermediate results are merged into one, and further anonymized to achieve consistent k-anonymous data sets. A group of MapReduce jobs are deliberately designed and coordinated to perform specializations on data sets collaboratively.

Keyword: [Cloud, Data Anonymization, MapReduce, Bottom-up Generalization]

1. INTRODUCTION

Big data is a popular term used to describe the exponential growth and availability of data, both structured and unstructured. 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 MapReduce. Typically, implementation of the MapReduce paradigm requires networked attached storage and parallel processing. The fact is that with so much data being generated by so many organisations 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. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud Computing is not a very new concept in IT, in fact Cloud Computing is a more advanced version of the Data Processing Service Bureaus that we had 40 years ago. The biggest and best known Cloud Computing providers include Amazon with EC2, Microsoft with Azure and Google with Google Apps (e.g. Gmail, Google Docs, and Google Calendar). However, as the shape of the cloud computing is emerging and developing rapidly both conceptually and in reality, the legal/contractual, economic, service quality, interoperability, security and privacy issues still pose significant challenge. A task of the utmost importance is to develop a secure way for data in a hostile environment so that the published data remain practically useful while individual privacy is preserved. Cloud computing offers the promise of big data implementation to small and medium sized businesses.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. An on-line cloud health service, aggregates data from users and shares the data with research institutes. Data privacy can be revealed with less effort by malicious cloud users or providers because of the failures of some traditional privacy protection measures on cloud. This can bring considerable economic loss or severe social reputation impairment to data owners. Hence, data privacy issues need to be addressed urgently before data sets are analyzed or shared on cloud. We creatively apply Map Reduce on cloud to BUG for data anonymization and deliberately design a group of innovative Map Reduce jobs to concretely accomplish the generalizations in a highly scalable way. Secondly, introduce a scalable Advanced BUG approach, which performs generalization on different partitioned data set and the resulting intermediate anonymizations are merged to find final anonymization which is used to anoymize the original data set.Results show that our approach can significantly improve the scalability and efficiency of BUG for data anonymization over existing approaches.

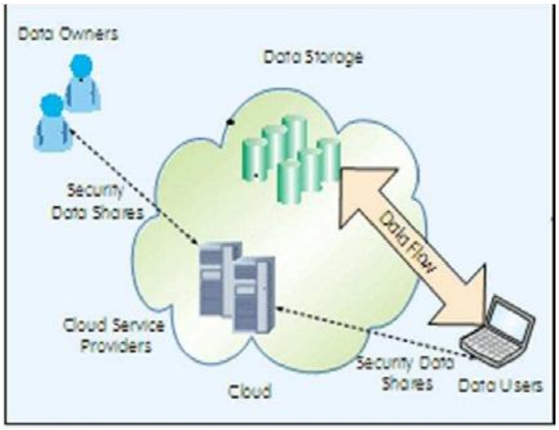

Figure 1: The Cloud System model

2. RELATED WORK AND PROBLEM ANALYSIS:

A wide verity of privacy models and anonymization approaches have been put forth to preserve the privacy sensitive information in 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. Most exiting algorithms exploit indexing data structure to assist the process of anonymization,specifically TEA (Taxonomy Encoded Anonymity) index for BUG.TEA is a tree of m levels. The ith level represents the current value for Dj. Each root to-leaf path represents a qid value in the current data table, with a (qid) stored at the leaf node. In addition, the TEA index links up the qids according to the generalizations that generalize them. The data structure TEA proposed in [239] can handle only a single QID. A new structure is required if the data holder wants to achieve LKC privacy using the Bottom-Up Generalization method because LKC privacy in effect is equivalent to breaking a single QID into multiple QIDs.

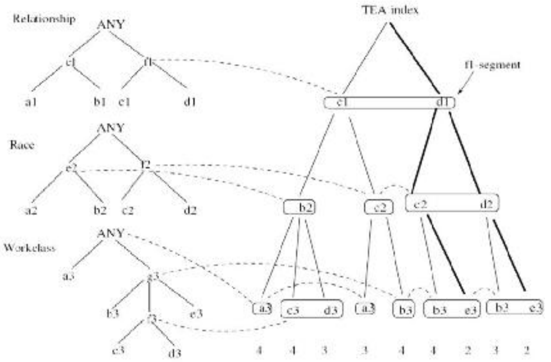

Figure 2: The TEA structure for QID

Although indexing data structures can speed up the process of data anonymization and the generalization process, because indexing structure avoids frequently scanning entire data sets and storing statistical results circumvents re-computation overheads, these approaches often fail to work in parallel or distributed environments like cloud systems because the indexing structures are centralized. There is an assumption that all data processed should fit in memory for the centralized approaches. Unfortunately, this assumption often fails to hold in most data-intensive cloud applications nowadays. Thus concluding that, 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. As in bottom up search strategy for finding optimization works well when the value of k is small. Centralized BUG lacks in high performance for certain value of k-anonymity parameter if they are used individually. In BUG, Calculating the ILPG and generalizing the data set involve accessing a large number of data records, thereby dominating the scalability and efficiency of bottom-up generalization. When generalize the information and privacy requirements to the problems of centralized anonymization and distributed anonymization, and identify the major challenges that make traditional data anonymization methods not applicable and suffer from scalable problem. MapReduce has been widely adopted in various data processing applications to push upward the scalability and efficiency. The MapReduce framework is scalable and fault-tolerant because each node in the cluster is expected to report back periodically with completed work and status updates. If a node remains silent for longer than the expected interval, a master node makes note and re-assigns the work to other nodes. As to BUG, the existing approach make full use of indexing data structure to promote efficiency, thereby falling short of high scalability and parallelization in cloud environments. Thus, it is valuable in investigating how to develop BUG algorithm with MapReduce in order to improve the scalability and efficiency. We also attach MapReduce to improve scalability and efficiency in our research on big data anonymization. A scalable advanced Bottom-Up Generalization (BUG) approach for data anonymization based on Map Reduce on cloud will make full use of the parallel capability of Map Reduce on cloud, specializations required in an anonymization process and the scalability and efficiency of centralized BUG are improved significantly over existing approaches.

3. METHODOLOGY

Bottom-Up Generalization is an efficient k-anonymization method. In a k-anonymous data set, each record is indistinguishable from at least k−1 other records with respect to QID. Basically, Bottom-Up Generalization (BUG) approach of anonymization is an iterative process starting from the lowest anonymization level.We leverage the information/privacy trade-off as the search metric for our approach, i.e., the Information Loss per Privacy Gain (ILPG). In this section we elaborate the Advanced BUG and MRBUG Driver. The Advanced BUG consists of following steps, data partition, run the MRBUG Driver on partitioned data set, combining the anonymization levels of the partitioned data set and applying generalization to original data set with integrated anonymization level without violating the k-anonmity.

3.1 Outline of Bi-Level BUG:

We propose a Advanced Bottom-Up Generalization (BUG) approach to improve the scalability and performance of BUG in advance fashion. The function of our approach is based on the two levels of parallelization provisioned by MapReduce on cloud. Basically, MapReduce on cloud has two levels of parallelization, i.e., job level and task level. Job level parallelization means that multiple MapReduce jobs can be executed simultaneously to make full use of cloud infrastructure resources. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data partitions. In advanced BUG, the dataset spit up is done by parallelized multi job mapreduce, and then the data set is anonymized by MapReduce Bottom up Generalization Driver without violating the k-anonymity. Then all the intermediate anonymization levels are integrated, ensures that the integrated intermediate anonymized level never violates K-anonmity, the more general one is selected as the integrated one. After obtaining the merged intermediate anonymization level, we excute the driver on the entire original data set, and get the resulting anonymization level. Then, the data set is anonymized by replacing original attribute values in it with the responding domain values obtained by the resulting anonymization level. The procedure for Advanced Bottom Up Generalization is as follows.

Input: Dataset DS, Anonymity k, k^1, partition p, record r, r DS, Anonymization Level AL_0

Procedure:

1. Scan DS and Generate a Random number ran, Emit (ran, r)
2. For each ran as parameters where 1 ≤ i ≤ p, results AL^1_i Emit (null, list(r))
3. Run MRBUG Driver with DSi , k1, ALo
4. Merge all AL^1_i where the resulting AL^2 should general or identical.
5. Run MRBUG Driver with DS, k, AL^2 as parameters, results AL*
6. For each attribute value v_i in r, find gen in AL^*
7. Generate r^*=(q_1, q_2, ..., q_m), q is quasi-identifier,m is number of attributes
8. If gen is INACTIVE then emit (r^*,count) For each r^* → ∑ **count** Emit (r^*,sum)

3.2 Data specification:

Here the original data set DS is partitioned into smaller ones. Let DSi, where 1 i p, denote≤≤ ≤ ≤ the data sets partitioned from DS, where p is the number of partitions. Specifically,a random number ran, where 1p, is generated for each data record. A record is assigned to the partition DSran.

Anonymization is not invulnerable counter Measures that compromise current anonymization techniques can expose protected information in released datasets. After getting the partitioned data set DSi, we run MRBUG (DSi , k1, ALo) on these data sets in parallelto derive intermediate ≤≤ anonymization levels AL1i where 1 i p.

3.3 Integrating the partitioned data:

Here the intermediate anonymization levels are merged into one (AL^2). The merging of anonymization levels is completed to ensure that the merged intermediate anonymization level (AL^2) never violates privacy requirements, the more general and identical one is selected as the merged one. If the intermediate anonymization levels AL^1_i satisfies privacy requirements, then the merged anonymization level AL^2 will not violate the privacy requirements. Then, MRBUG can further anonymize the entire data sets to produce final k-anonymous data sets.

3.4 MRBUG Driver:

MRBUG plays a main role in the Advanced BUG approach, as it is invoked by two times to concretely process generalization. Basically, a practical

MapReduce program consists of Map and Reduce functions, and a Driver that coordinates the macro execution of jobs. Each round of BUG iteration includes four major steps, namely, checking the current data set whether satisfies the anonymity requirement, calculating the ILPG, finding the best generalization and generalizing the data set according to the selected best generalization. An existing approach uses indexing data structure but MapReduce does not support indexing data structure. So for calculating ILPG we use mapreduce jobs. The procedure for MRBUG driver is as follows.

Input: Data Set DS, Anonymity k, Data record (ID_r, r), r DS, Anonymization Level AL_0

Procedure:

```
1.	Scan DS and initialize search metric ILPG
2.	For each(if Anonymity <k before generalization)
Find $gen_{best}$ and set it INACTIVE
3.	If set of gen and its siblings are INACTIVE then
Insert ← $gen_{new}$ to replace INACTIVE ones
4.	$AL_{i+1}AL_i$ ; Update search metric ILPG for all active gen
5.	Repeat the iteration
```

The MRBUG Driver starts by initiating the search metric values ILPG, next it checks for current anonymity satisfies the privacy requirements, the driver finds the best generalization of Highest ILPG and set genbest as INACTIVE that means genbest is not consider for next iteration. If set of all gen and its siblings are labelled as INACTIVE then insert new higher level generalization to replace all the INACTIVEs. Atlast reset new value for anonymization level and update the search metric ILPG. For initiating and updating the search metric ILPG involves accessing to the original data set and computing statistic information over the data set. The process of generalization also involves accessing to the original data set. The information loss of a generalization will not be affected when we perform other generalizations or insert a new generalization, while privacy gain will probably be impacted as the anonymity of the data set will change.The essential of computing anonymity of a data set is to find out the minimum QI-group size. The ILPG calculation results in Information gain, anonymity for generalizations and intermediate key value pair (key, count). Map and Reduce, defined over a data structure named key-value pair (key, value). Map takes a pair (k1, v1) as input and then outputs another intermediate key-value pair (k2, v2). These intermediate pair are consumed by the Reduce function as input. Reduce takes intermediate k2 and all its corresponding values list (v2) as input and outputs another pair (k3, v3).

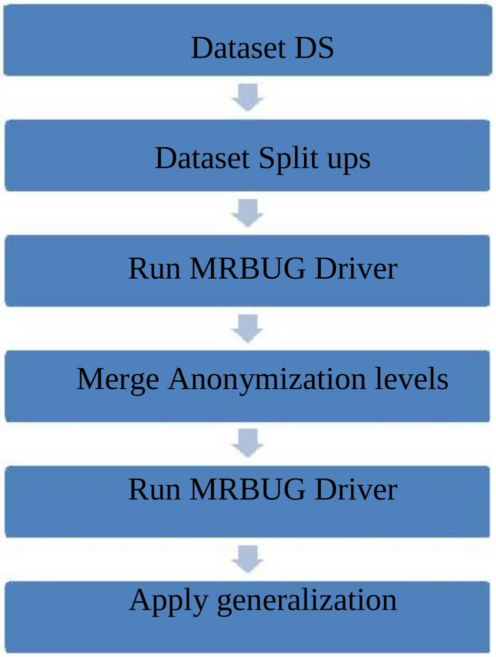

Figure 3: Flow Diagram

3.5 Data Generalization:

The original data set is concretely generalized for data anonymization by a MapReduce job. The Map function emits anonymous records and its count according to the current anonymization level. The Reduce function simply aggregates these anonymous records and counts their number. An anonymous record and its count represent a QI-group, and the QI-groups constitute the final anonymous data sets.

CONCLUSION AND FUTURE WORK

In this paper, we studied the scalability problem of data anonymization for big data applications on cloud using Bottom Up Generalization (BUG) and proposed scalable Advanced Bottom Up Generalization. The proposed BUG performed as Data partitioned, executing the driver producing intermediate results. Then, the intermediate results are merged and generalization is applied to produce anonymized data without violating k-anonymity. The MapReduce Framework is effectively applied on cloud for data anonymization and shows that scalability and efficiency of centralized BUG are improved significantly over existing approaches. In future optimized balanced scheduling strategies are expected to be developed towards overall scalable privacy preservation aware dataset scheduling. And also our method is designed for achieving k-anonymity; it can be modified to adopt the LKC-privacy model in order to accommodate the high-dimensional data.

SAMPLE EVALUVATION

To evaluate the effectiveness and efficiency of the Advanced BUG approach, we compare Advanced BUG with centralized BUG. We denote the execution time of the two approaches as Adv-BUG and cen-BUG, respectively. Our experiments are conducted in a cloud environment named Amazon Elastic MapReduce. Amazon Elastic MapReduce (EMR) is a web service that uses Hadoop, an open-source framework, to quickly & cost-effectively process vast amounts of data. Elastic MapReduce is a web service built on top of the Amazon cloud platform. All approaches are implemented in Java and standard Hadoop MapReduce API. We utilize the Health Care Sample data set from US Government Open Data Projects (Dataset).

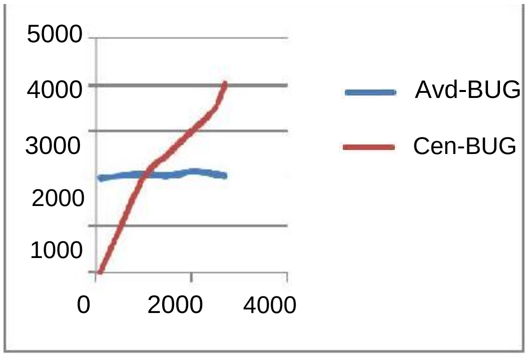

Figure 4: Change of execution time with respect to data size

X-axis: data size in MB
Y-axis: time in seconds

We measure the change of execution time Cen-BUG and Adv-BUG with respect to data size. We can see from Fig.4 that the execution time of the Advanced approach is kept under a certain level, while centralized BUG incur high execution time when the data increases in size. Hence, Cent-BUG suffers from scalability problem for large-scale data sets. The above experimental results demonstrate that our approach can significantly improve the scalability and efficiency compared with Centralized approaches.

REFERENCES

[1]. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M.Zaharia, “A View of Cloud Computing,” Commun. ACM, vol.53, no. 4, pp. 50-58, 2010.

[2]. 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.

[3]. H. Takabi, J.B.D. Joshi and G. Ahn, “Security and Privacy Callenges in Cloud Computing Environments,” IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, 2010.

[4]. D. Zissis and D. Lekkas, “Addressing Cloud Computing Security Issues,” Fut.

Gener. Comput. Syst., vol. 28, no. 3, pp.583-592, 2011.

[5]. 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.

[6]. Roy, S.T.V. Setty, A. Kilzer, V. Shmatikov and E. Witchel, “Airavat: Security and Privacy for Mapreduce,” Proc. 7th USENIX Conf. Networked Systems Design and Implementation (NSDI'10), pp. 297-312, 2010.

[7]. T. Iwuchukwu and J.F. Naughton, “K-Anonymization as Spatial Indexing: Toward Scalable and Incremental Anonymization,” Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB'07), pp. 746-757, 2007.

[8]. 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.

[9]. B.C.M. Fung, K. Wang and P.S. Yu, “Anonymizing Classification Data for Privacy Preservation,” IEEE Trans. Knowl..Data Eng., vol. 19, no. 5, pp. 711-725, 2007.

[10]. 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.

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

[12]. K. LeFevre, D.J. DeWitt and R. Ramakrishnan, “Mondrian multidimensional k-anonymity,” Proc. 22nd International Conference on Data Engineering (ICDE '06), article 25,2006.

[13]. T. Li, N. Li, J. Zhang and I. Molloy, “Slicing: A new approach for privacy preserving data publishing,” IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 3, pp. 561-574, 2012.

[14]. N.Li, T.Li, S.Venkatasubramiam “Closeness: A new privacy measure for datapublishing,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 7, pp. 943-956, 2010.

[15]. B.C.M. Fung, K. Wang, R. Chen and P.S. Yu, “Privacy preserving data publishing: A survey of recent developments,” ACM Computer Survey, vol. 42, no. 4, pp. 1-53, 2010.

[16]. M. Terrovitis, J. Liagouris, N. Mamoulis and S. Skiadopolous, “Privacy preservation by disassociation,” Proceedings of the VLDB Endowment, vol. 5, no. 10, pp.944-955, 2012.

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

[18]. D. E. Denning. Commutative filters for reducing inference threats in multilevel database systems. In Proc. of the IEEE Symposium on Security and Privacy (S&P), Oakland, CA, April 1985.

[19]. C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In Proc. of the 2nd Internationl Workshop on Privacy Enchancing Technologies (PET), pages 54–68, San Francisco, CA, April 2002.

[20]. J. Goguen and J. Meseguer. Unwinding and inference control. In Proc.of the IEEE Symposium on Security and Privacy (S&P), Oakland, CA,1984.