Electronic Health Records Processing using MapReduce on Cloud

[1]Prafulla B.G.L, [2]Vernon Louis, [3]Chitra R
[1]Student, [2]Student, [3]Assistant Professor
[1]Department of Information Science Engineering,
[1]NIE Institute of Technology, Mysuru, India

***Abstract* - A cloud services require in big scale, for users to share a private data such as health records, transactional data for analysis of data or mining of that data which bringing privacy concerns. Recently, because of new social behavior, social transformation as well as vast spread of social system many cloud applications increase in accordance with the Big Data style, and make it a challenge for commonly used software tools to manage, capture and process the large-scale data within an elapsed time. In this paper, we are going to implement a scalable two-phase top-down specialization approach to anonymize large scale data sets of electronic health records using the MapReduce framework on cloud. In both phases of our project, we are going to design a group of inventive MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way.**

***Keywords* - Data anonymization, Top-Down Specialization, MapReduce, Privacy Preservation, Optimized Balanced Scheduling**

I.INTRODUCTION

Big Data refers to collection of data sets that is so large and complex that it becomes so difficult to process using traditional data processing applications. The challenges include data capture, curation, storage, search, sharing, transfer, analysis, and visualization. The trend to large data set is due to additional information derivable compared to separate smaller sets with same total of data, allowing correlations to be found to spot business trends, determine quality of research, prevent diseases, combat crime, and determine real-time road traffic conditions.

Cloud computing, a disruptive trend at present, poses a significant impact on IT industry and research communities [6]. Cloud computing provides massive communication power and storage capacity via utilizing a large number of commodity computers together, enabling users to deploy applications cost effectively without heavy infrastructure investment, so that the cloud users can concentrate on their core business. However, numerous potential customers are hesitant to take advantage of cloud due to privacy and security concerns.

Privacy is one of the most concerned issue in cloud computing. Personal data like electronic health records and financial transactional records are usually deemed extremely sensitive through these data can offer significant benefits if they are analyzed and mined by organizations such as disease research centers. For instance Microsoft Health Vault [3], an online health service, aggregates data from users and shares data with research institutes. To overcome the privacy issues data anonymization is widely adopted in non interactive data publishing and sharing scenarios. Data anonymization refers to hiding identity and/or sensitive data owners of data records.

Large-scale data processing framework like MapReduce have been integrated with cloud to provide powerful capability of application. We leverage MapReduce, a widely adopted parallel data processing framework, to address the scalability problem of the top-down specialization (TDS) approach. The TDS approach, offering a good tradeoff between data utility and data consistency is widely applied for data anonymization. To make full use of parallel capability of MapReduce on cloud, specializations required in an anonymization process are spilt into two phases. In the first one, original data sets are partitioned onto a group of smaller data set, and these data sets ate anonymized in parallel, producing intermediate results. In the second one, the intermediate results are integrated into one, and further anonymized to achieve consistent *k*-anonymous data sets. It leverages MapReduce to accomplish the concrete computation in both phases.

The major contributions of the project are threefold. First, we creatively apply MapReduce on cloud to TDS for TDS anonymization and deliberately design a group of innovative MapReduce jobs to concretely accomplish the specializations in a highly scalable manner. Second, we propose a two-phase TDS approach to gain high scalability via allowing specialization to be conducted on multiple data partitions in parallel during the first phase. Third, experimental results show that our approach will significantly improve the scalability and efficiency of TDS for data anonymization over existing approaches.

II.LITERATURE SURVEY

Big Data processing in cloud computing environments [1].With the rapid growth of emerging applications like social network analysis, semantic web analysis and bioinformatics network analysis, a variety of data to be processed continues to witness a quick increase. Effective management and analysis of large-scale data poses an interesting but critical challenge. Recently, big data has attracted a lot of attention from academia, industry as well as government.

Privacy preservation enriched MapReduce for Hadoop based Big Data applications [2]. In the proposed system four models to enrich overall anonymity of critical data sets has been developed. These models are privacy characterization model, anonymizer for data sets, data set update and privacy preserved data management. In the proposed system the data owner possesses authority and interface to introduce various security levels for its data to make it privacy preserved and anonymous. The proposed model facilitates data users to retrieve data sets in its anonymized form which ultimately provides user task without publishing critical detail information about original data. This system would not only facilitate anonymity of data sets in cloud infrastructure but also optimize data recomputation by means of its partial data retaining capacity. Thus, the proposed system would bring optimization not only in terms of privacy preservation but also with enhanced resource utilization in Big Data based applications.

Map Task Scheduling in MapReduce with data locality, throughput and heavy-traffic optimality [5]. Here the focus is to strike right balance between data-locality and load-balancing to simultaneously maximize throughput and minimize delay. We present new queuing architecture and propose a Map Task Scheduling algorithm constituted by joining the Shortest Queue policy together with the Max Weight policy. We identify an outer bound on the capacity region, and then prove that the proposed algorithm stabilizes any arrival rate vector strictly within its outer bound. It shows that the algorithm is throughput optimal and the outer bound coincides with the actual capacity region. The proposed algorithm is heavy-traffic optimal, i.e., it asymptotically minimizes the number of backlogged tasks as the arrival rate vector approaches the boundary of the capacity region.

Security and privacy challenges in cloud computing environments [4]. The cloud computing paradigm is still evolving, but has recently gained tremendous momentum. However, security and privacy issues pose as the key roadblock to its fast adoption. In this, security and privacy challenges are presented that are exacerbated by the unique aspects of clouds and show how they are related to various delivery and deployment models.

III. SYSTEM ANALYSIS

Existing System

The amount of data being exploded in accordance with the Big Data trend, existing centralized TDS approach is inefficient to handle the scalability issues and achieve privacy preservation on privacy-sensitive large-scale electronic health record data sets.

Proposed System

- The two-phase top-down specialization (TPTDS) approach using privacy preserving MapReduce on cloud provides ability to handle the high amount of large-scale electronic health record data sets.
- It provides the privacy by effective anonymization approaches such as *k*-anonymity.
- Introducing a Trusted Third Party, tasked with assuring specific security characteristics within a cloud environment.
- The proposed system calls upon cryptography, specifically Public Key Infrastructure to ensure the authentication, integrity and confidentiality of involved data and communications.

IV. SYSTEM DESIGN

Data Partition

The data partition is performed in the cloud. Here the original data set D is spilt into smaller number of data sets. Then provide random number of each data set. Partitioning is the process of determining which reducer instance will receive them.

Anonymization

Anonymization of data can mitigate privacy and security concerns and comply with legal requirement. After getting the individual data sets it applies anonymization. It means hide or remove the sensitive field in data sets. Then it gets the intermediate results for small data sets. The intermediate results are used for the specialization process. Data anonymization algorithm converts clear text data into nonhuman readable and irreversible form.

Merging

The intermediate results of several small data sets are merged here. The MapReduce edition of centralized TDS (MRTDS) driver [6] is used to organize the small intermediate results for merging. The merged data sets are collected on cloud. The merging of anonymization levels are completed by merging cuts. The merging results are again applied in anonymization called specialization.

Data Specialization

An original data set D is concretely specialized or anonymization in a one-iteration in MapReduce job. When we obtain the merged intermediate anonymization level AL*, we run MRTDS driver (D, *k*, AL*) on entire data set D, and get the final anonymization level AL*. Then Reduce function simply aggregate these anonymous records and counts the number of that particular records and its count represent a Quasi-identifier (QI) -group. The QI-groups constitute the final anonymous data sets.

Data Specialization

An original data set D is concretely specialized or anonymization in a one-iteration in MapReduce job. When we obtain the merged intermediate anonymization level AL*, we run MRTDS driver (D, *k*, AL*)on entire data set D, and get the final anonymization level AL*. Then Reduce function simply aggregate these anonymous records and counts the number of that particular records and its count represent a Quasi-identifier (QI) -group. The QI-groups constitute the final anonymous data sets.

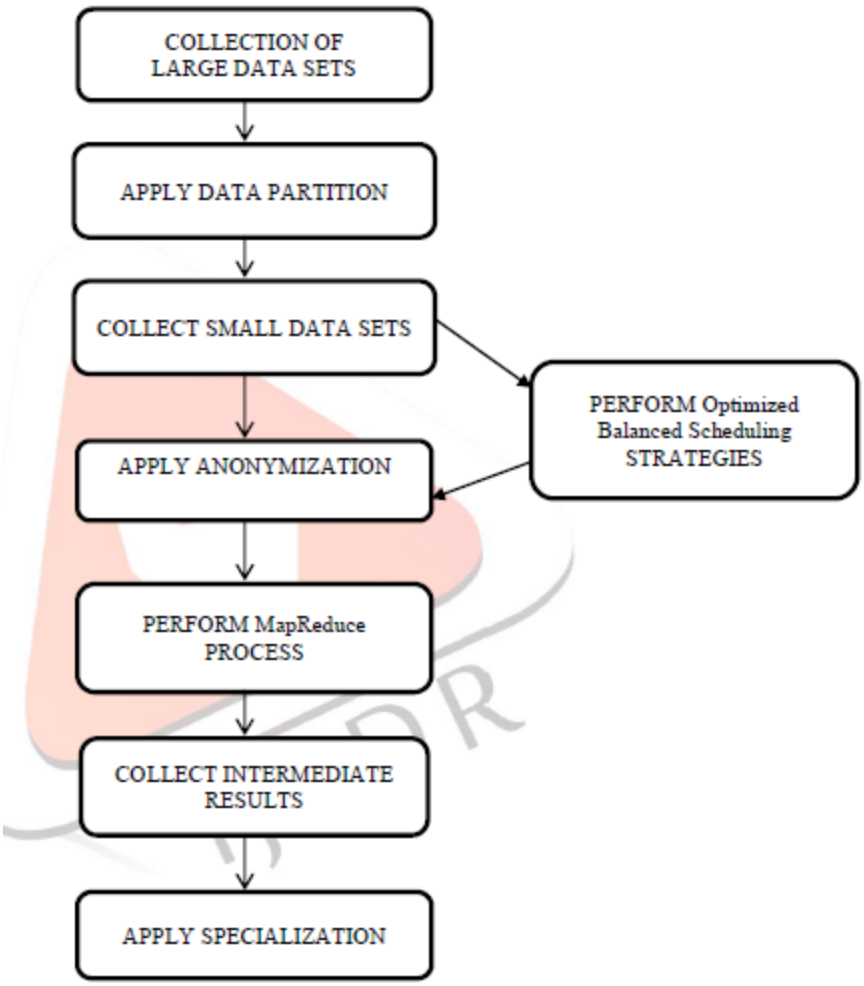

Fig 1 System Process Flow

MapReduce

Fig. 2 shows the MapReduce process flow. MapReduce [6] is a 5 step parallel and distributed computation.

STEP1: Prepare the Map() input – the "MapReduce system" designates Map processors, assigns the K1 input key value each processor would work on, and provides that processor with all the input data associated with that key value.

STEP 2: Run the user-provided Map() code – Map() is run exactly once for each K1 key value, generating output organized by key values K2.

STEP 3: "shuffle" the Map output to the Reduce processors – the MapReduce system designates Reduce processors, assigns the K2 key value each processor would work on, and provides that processor with all the Map-generated data associated with that key value.

STEP 4: Run the user-provided Reduce() code – Reduce() is run exactly once for each K2 key value proposed by the Map step.

STEP 5: Produce the final output – the MapReduce system collects all the Reduce output, and sorts it by K2 to produce the final outcome.

The MapReduce also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails the work can be rescheduled assuming the input data is still available.

Optimized Balanced Scheduling

The optimized balanced scheduling (OBS) mechanism is for scheduling map tasks to improve data locality which is crucial for the performance of MapReduce. The algorithm used is map task scheduling constituted by the Join the Shortest Queue policy together with the Max Weight policy. Here it concentrates on sensitive field in every data set and gives priority for this sensitive field. It focuses on the two kinds of scheduling called time and size. Here data sets are split in to the specified size and applied anonymization on specified time.

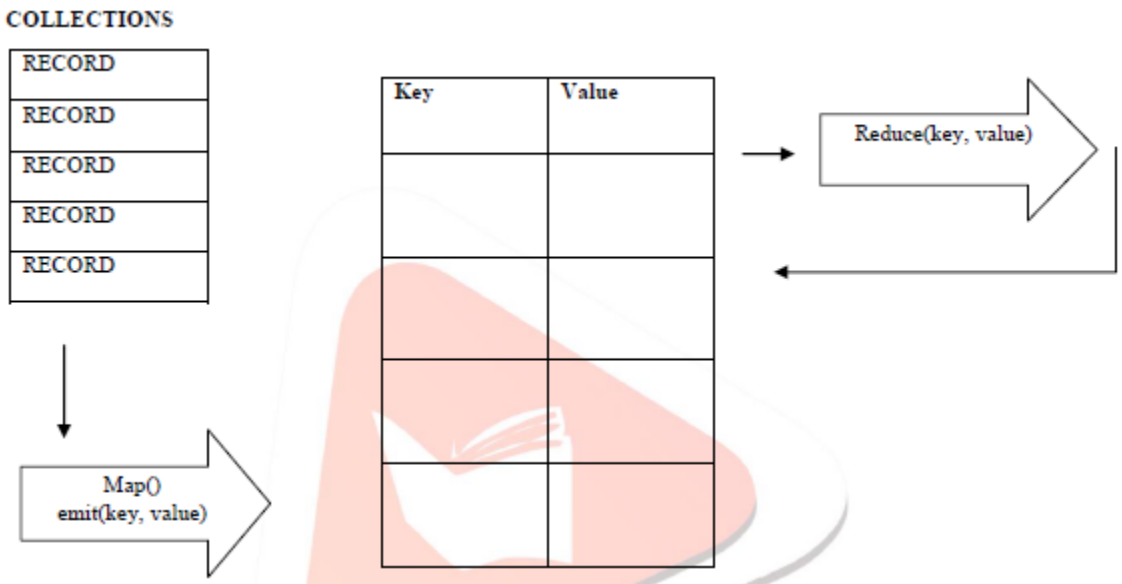

Fig 2 MapReduce Process Flow

V. APPLICATIONS AND CONCLUSION

This system is implemented to maintain an electronic health records for hospital management. This provides the overall detail of the patient's health issues. It provides the management to have abstract details of patients with particular disease in real quick time against the large database. It provides online uploading of patient's prescription details, referring made to other doctors, scanning reports so that the doctor can refer it online and provide the treatment. This system provides the government to gather information of number of patients with particular diseases which is helpful in take necessary measures.

It can be concluded that, the proposed system is efficient to handle large-scale data sets and preserving privacy by effective anonymization approaches so that the health records can be maintained electronically, which benefits the organizations in monitoring all the activities in an efficient manner.

REFERENCES

[1]. Changquing Ji, Yu Li, Wenming Qui, Uchechukwu Awada, Keqiu Li, "Big Data Processing in Cloud Computing Environments, 2012 International Symposium on Pervasive Systems, Algorithms and Networks.

[2]. Chayya S Dhule, Dr. Girijamma et al., "Privacy Preservation Enriched MapReduce for Hadoop Based Big Data Applications" March 2014, American International Journal of Research in Science, Technology, Engineering and Mathematics.

[3]. Microsoft Health Vault, http://www.healthvault.com

[4]. H. Takabi, J.B.D. Joshi and G. Ahnn, "Security and Privacy Challenges in Cloud Computing Environments", IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Nov. 2010.

[5]. W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Map Task Scheduling in MapReduce with Data Locality: Throughput and Heavy-Traffic Optimality" Arizona State Univ., AZ, Tech. Rep., Jul. 2012.

[6]. Xuyun Zhang, Lawrence T Yang, et al. "A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization using MapReduce on Cloud", IEEE Transactions on Parallel and Distributed Systems, Vol. 2, No. 2, February 2014.