ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Big-Data Processing With Privacy Preserving Map-Reduce Cloud

R.Sreedhar[1], D.Umamaheshwari[2]

PG Scholar, Computer and Communication Engineering, EBET Group of Institutions, EBET Knowledge Park, Tirupur-638108, Tamilnadu, India[1]

Assistant Professor, EBET Group of Institutions, EBET Knowledge Park, Tirupur-638108, Tamilnadu, India[2]

Abstract- A large number of cloud services requires users to share private data like electronic health records for data analysis or mining, bringing privacy concerns. Anonymizing data sets via generalization to satisfy certain privacy requirements such as k-anonymity is a widely used category of privacy preserving techniques. 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. 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. An introduce the scalable two-phase top-down specialization approach to anonymize large-scale data sets using the MapReduce framework on cloud. In both phases of approach is deliberately design a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. Experimental evaluation results demonstrate that with this approach. The scalability and efficiency of top-down specialization can be improved significantly over existing approaches. An introduce the scheduling mechanism called Optimized Balanced Scheduling to apply the Anonymization. Here the OBS means individual dataset have the separate sensitive field. Every data set sensitive field and give priority for this sensitive field. Then apply Anonymization on this sensitive field only depending upon the scheduling.

Key Words: Data Anonymization, Top-Down specialization, Map-Reduce, Cloud, Privacy Preservation, OBS, Data Partition, Data Merging

I. INTRODUCTION

Big data is the term for a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, transfer, analysis, and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, determine quality of research, prevent diseases, combat crime, and determine real-time roadway traffic conditions.

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

Distributed systems are groups of networked computers, which have the same goal for their work. The terms parallel computing, and distributed computing have a lot of overlap and no clear distinction exists between them. The same system may be characterized both as parallel and distributed the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as parallel or distributed using the following criteria. In parallel computing, all processors may have access to a shared memory to be exchange information between processors. In

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processor.

A highly scalable two-phase TDS approach for data anonymization based on MapReduce on cloud. To make full use of the parallel capability of MapReduce on cloud, specializations required in an anonymization process are split into two phases. In the first one, original datasets are partitioned into a group of smaller datasets, and these datasets are 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 leverage MapReduce 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 demonstrate 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 apply MapReduce 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.

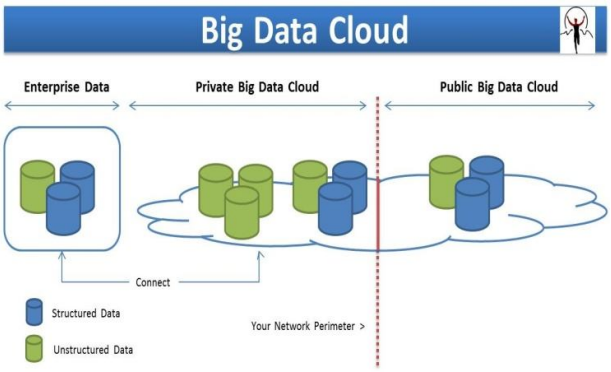

Fig1.BigData on cloud

Thirdly, experimental results show that the approach can significantly improve the scalability and efficiency of TDS for data anonymization over existing approaches.

II. RELATED WORKS AND PROBLEM ANALYSIS

In the work [1] Author said a new technique, called "certificate base authorization to provides the security" in cloud environment. The recent emergence of cloud computing has drastically altered everyone's perception of infrastructure architectures, software delivery and development models. Projecting as an evolutionary step, following the transition from mainframe computers to client/server deployment models, cloud computing encompasses elements from grid computing, utility computing and autonomic computing, into an innovative deployment architecture. This rapid transition towards the clouds, has fuelled concerns on a critical issue for the success of information systems, communication and information security. From a security perspective, a number of unchartered risks and challenges have been introduced from this relocation to the clouds, deteriorating much of the effectiveness of traditional protection mechanisms. As a result the aim of this paper is twofold; firstly to evaluate cloud security by identifying unique security requirements and secondly to attempt to present a viable solution that eliminates these potential threats. This paper proposes introducing a Trusted Third Party, tasked with assuring specific security characteristics within a cloud environment. The proposed solution calls upon cryptography, specifically Public Key Infrastructure operating in concert with SSO and LDAP, to ensure the authentication, integrity and confidentiality of involved data and communications. The solution, presents a horizontal level of service, available to all implicated entities, that realizes a security mesh, within which essential trust is maintained. Here it using the certificate base authorization to provides the security in cloud environment. The over all performance of the security issues are low compare with existing approaches.

In the work [2] Author said a new technique, called "work load aware anonymization techniques and classification and regression" Protecting individual privacy is an important problem in microdata distribution and publishing. Anonymization algorithms typically aim to satisfy certain privacy definitions with minimal impact on the quality of the resulting data. While much of the previous literature has measured quality through simple one-size-fits-all measures and

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

argue that quality is best judged with respect to the workload for which the data will ultimately be used. This article provides a suite of anonymization algorithms that incorporate a target class of workloads, consisting of one or more data mining tasks as well as selection predicates. An extensive empirical evaluation indicates that this approach is often more effective than previous techniques. In addition consider the problem of scalability. The article describes two extensions that allow to scale the anonymization algorithms to datasets much larger than main memory. The first extension is based on ideas from scalable decision trees, and the second is based on sampling. A thorough performance evaluation indicates that these techniques are viable in practice. Here it using the certificate base authorization to provides the security in cloud environment. The overall performance of the security issues are low compare with existing approaches. Here are using the work load aware anonymization techniques and classification and regression. It also fails to handles the large amount of the data sets.

In the work [3] Author said a new technique, called "distribute anonymization and centralized anonymization" Sharing healthcare data has become a vital requirement in healthcare system management; however, inappropriate sharing and usage of healthcare data could threaten patients' privacy. In this article, it study the privacy concerns of sharing patient information between the Hong Kong Red Cross Blood Transfusion Service (BTS) and the public hospitals. It generalize their 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. Furthermore propose a new privacy model called LKC-privacy to overcome the challenges and present two anonymization algorithms to achieve LKC-privacy in both the centralized and the distributed scenarios. Experiments on real-life data demonstrate that the anonymization algorithms can effectively retain the essential information in anonymous data for data analysis and is scalable for anonymizing large datasets. Handling of the large scale data sets are very difficult. Here it using the distribute anonymization and centralized anonymization to provides the privacy on cloud. Handling of the large scale data sets are very difficult.

III. PROPOSED SYSTEM

In this section, the Authors[1] introduces a two phase top-down specialization approach and it introduce the scheduling mechanism called Optimized Balanced Scheduling(OBS) to apply the anonymization. Here the OBS means individual dataset have the separate sensitive field.

A. Privacy Preserving Map-Reduce Cloud

The main drawback of the approach is centralized top-down approach. It's does not have the ability for handle the large scale datasets in cloud. Its overcome by it invent the two phase top-down specialization approach. This approach get input data's and split into the small data sets. Then it apply the anonymization on small data sets to get intermediate result. Then small data sets are merge and again apply the anonymization. Here the draw backs of proposed system is there is no priority for applying the anonymization on datasets. So that its take more time to anonymize the datasets. So it introduce the scheduling mechanism called Optimized Balanced Scheduling(OBS) to apply the anonymization. Here the OBS means individual dataset have the separate sensitive field. It analyze the each and every data set sensitive field and give priority for this sensitive field. Then apply anonymization on this sensitive field only depending upon the scheduling.

B. Two Phase Top Down Specialization

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. 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. Combined 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 identical. To obtain finally consistent anonymous data sets, the second phase is necessary 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 completed by merging cuts. Specifically, let in and in be two cuts of an attribute. There exist domain values and that satisfy

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

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 violates privacy requirements, the more general one is selected 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. The following lemma ensures that still complies privacy requirements.

IV. MAPREDUCE

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map() procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() procedure that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The MapReduce System is orchestrates by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, providing for redundancy and fault tolerance, and overall management of the whole process.

The model is inspired by the map and reduces functions commonly used in functional programming, although their purpose in the MapReduce framework is not the same as their original forms. Furthermore, the key contribution of the MapReduce framework are not the actual map and reduce functions, but the scalability and fault-tolerance achieved for a variety of applications by optimizing the execution engine once. MapReduce libraries have been written in many programming languages, with different levels of optimization. A popular open-source implementation is Apache Hadoop. The name MapReduce originally referred to the proprietary Google technology and has since been generalize MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware). Computational processing can occur on data stored either in a file system (unstructured) or in a database (structured). MapReduce can take advantage of locality of data, processing data on or near the storage assets to decrease transmission of data.

"Map" step: The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

"Reduce" step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel – though in practice it is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase - provided all outputs of the map operation that share the same key are presented to the same reducer at the same time, or if the reduction function is associative. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than commodity servers can handle a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism 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. Another way to look at MapReduce is as a 5-step parallel and distributed computation.

1. 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.

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

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.

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

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

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

Logically these 5 steps can be thought of as running in sequence – each step starts only after the previous step is completed – though in practice, of course, they can be intertwined, as long as the final result is not affected. Following Algorithm Used To MapReduce System.

1: class MAPPER

2: method MAP (string t, integer r)

3: EMIT (string t, pair(r,1))

1: Class COMBINER

2: method COMBINE (string t, pairs [(s_1,c_1),(s_2,c_2)..])

3: sum ←0

4: cnt←0

5: for all pair (s,c) € pairs [(s_1,c_1),(s_2,c_2)..] do

6: sum←sum+s

7: cnt←cnt+c

8: EMIT(string t, pair (sum,cnt))

1:Class REDUCER

2: method REDUCER (string t, pairs [(s_1,c_1),(s_2,c_2)..])

3: sum ←0

4: cnt←0

5: for all pair (s,c) € pairs [(s_1,c_1),(s_2,c_2)..] do

6: sum←sum+s

7: cnt←cnt+c

8: r_{avg}←sum/cnt

9: EMIT (string t, pair (r_{avg}, cnt))

The above following algorithm is used to map reduce operation. In many situations the input data might already be distributed among many different servers, in which case step 1 could sometimes be greatly simplified by assigning Map servers that would process the locally present input data. Similarly, step 3 could sometimes be sped up by assigning Reduce processors that are as much as possible local to the Map-generated data they need to process. MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel through in practice it is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase provided all outputs of the map operation that share the same key are presented to the same reducer at the same time, or if the reduction function is associative. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than commodity servers can handle a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism 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.

A. Data Partition

The data partition is performed on the cloud. Here it collects the large no of data sets. It are split the large into small data sets. Then provides the random no for each data sets. Partitioning is the process of determining which reducer instance will receive which intermediate keys and values. Each mapper must determine for all of its output (key, value) pairs which reducer will receive them. It is necessary that for any key, regardless of which mapper instance generated it, the destination partition is the same. If the key cat is generated in two separate (key, value) pairs, they must both be reduced together. It is also important for performance reasons that the mappers be able to partition data independently they should never need to exchange information with one another to determine the partition for a particular key. It is necessary that for any key, regardless of which mapper instance generated it, the destination partition is the same. If the key cat is generated in two separate (key, value) pairs, they must both be reduced together. It is also important for performance reasons that the mappers be able to partition data independently they should never need to exchange information with one another to determine the partition for a particular key.

B. Big Data Analytics Specialization

Big Data Analytics specialization will prepare students to address real-life problems along each of those dimensions. For instance, it is not uncommon for digital archives to store terabytes and even petabyte of data in hundreds of data repositories supporting thousands of applications. Maintaining such data repositories requires knowledge in ultra-large scale distributed systems, virtualization technologies, cloud computing, unstructured and semi-structured data

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

management, optimization methods based on data replication and data migration, as well as in advanced data protection techniques. The exponential growth of the amount of data calls for competence in advanced dynamic data processing techniques, including scalable data processing methods and technologies; data stream management; and large-scale process monitoring, modeling and mining. In order to comprehensively analyze such volumes of information from disparate and various disciplines, information professionals will need to master advanced data integration techniques and business intelligence tools, crowd sourcing technologies, large-scale information fusion, data-intensive computation and semantic data management. After gets the intermediate result those results are merged into one. Again applies the anonymization on the merged data its called specialization. Here using the two kinds of jobs such as IGPL Update and IGPL Initialization. The jobs are organized by using the driver.

C. Anonymization

Anonymization of data can mitigate privacy and security concerns and comply with legal requirements. Anonymization is not invulnerable countermeasures that compromise current anonymization techniques can expose protected information in released datasets. After gets the individual data sets it applies the anonymization. The anonymization means hide or remove the sensitive field in data sets. Then it gets the intermediate result for the small data sets. The intermediate results are used for the specialization process. Data anonymization algorithm that converts clear text data into a nonhuman readable and irreversible form including but not limited to preimage resistant hashes and encryption techniques in which the decryption key has been discarded.

Direct Anonymization Algorithm

DA(D,I,k,m)

1.scan D and create count-tree

2.initialize Cout

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.Return Cout

12. for i :=1 to Cout do

13. initialize count=0

14. scan each transactions in Cout

15. Seperate 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. if ka 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

D. Merging

The intermediate results of the several small data sets are merged here. The MRTDS driver is used to organize the small intermediate result for merging. The merged data sets are collected on cloud. The merging result is again applied in anonymization called specialization.

E. Optimized Balancing Scheduling

The OBS called optimized balancing scheduling. Scheduling map tasks to improve data locality is crucial to the performance of MapReduce. Many works have been devoted to increasing data locality for better efficiency. However, to the best of our knowledge, fundamental limits of MapReduce computing clusters with data locality, including the capacity region and theoretical bounds on the delay performance have not been studied. In this paper it addresses these problems from a stochastic network perspective.

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

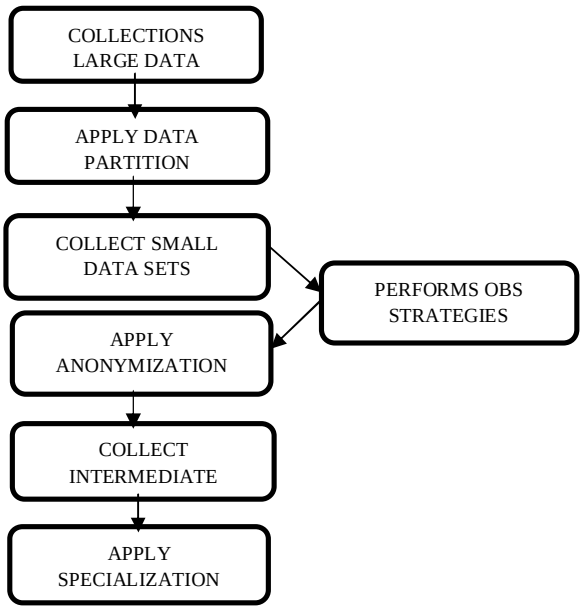

Fig3. Dataflow diagram

Our focus is to strike the right balance between data-locality and load-balancing to simultaneously maximize throughput and minimize delay. It presents a new queuing architecture and proposes a map task scheduling algorithm constituted by the Join the Shortest Queue policy together with the MaxWeight policy. It identifies an outer bound on the capacity region, and then prove that the proposed algorithm stabilizes any arrival rate vector strictly within this outer bound. It shows that the algorithm is throughput optimal and the outer bound coincides with the actual capacity region. Further it studies the number of backlogged tasks under the proposed algorithm, which is directly related to the delay performance based on Little's law. It proves that 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. Therefore the proposed algorithm is also delay optimal in the heavy-traffic regime. Here it focus on the two kinds of the scheduling called time and size. Here data sets are split in to the specified size and applied anonymization on specified time. To provide the high ability on handles the large data sets.

V. CONCLUSION

The conclusion of the proposed work is it using the two phase top down approach to provide ability to handles the high amount of the large data sets. And here it provide the privacy by effective anonymization approaches. In the future work is reduce the handling effect of large amount of the data sets. Its implements the optimized balancing scheduling. Where its based on the time and size of the data sets. In this paper it have investigated the scalability problem of large-scale data anonymization by Top-Down Specialization 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. It 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.

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.

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

[11]. M. Almorsy, J. Grundy, I. Mueller, "An analysis of the cloud computing security problem," In the proc. of the 2010 Asia Pacific Cloud Workshop, Colocated with APSEC2010, Australia, 2010.

[12].NIST, "Standards for Security Categorization of Federal Information and Information Systems. FIPS-199", <csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199.pdf>, Accessed Dec 2010.

[15]. R. Sherman, Distributed systems security, Computers & Security 11 (1) (1992).

[16]. D. Polemi, Trusted third party services for health care in Europe, Future Generation Computer Systems 14 (1998) 51–59.

[17]. S. T. Maguluri, R. Srikant, and L. Ying, "Heavy traffic optimal resource allocation algorithms for cloud computing clusters," in Int. Teletraffic Congr. (ITC), Krakow, Poland, 2012.

[18]. 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., Tempe, AZ, Tech. Rep., Jul. 2012.