ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

# Optimized Balanced Scheduling of Two Phase Top Down Specialization for Diabetes Patients using Map Reduce.

**KS Sendhil Kumar[1*] and , N Jaisankar[2].**

[1]Assistant Professor (Senior), [2]Professor, SCOPE, VIT University, Vellore, Tamil Nadu, India.

**ABSTRACT**

Cloud computing provides huge storage capacity to store all important and sensitive data as well as process the large data sets. Sensitive data means personal data about a person, which means there is something inside a person that is considered characteristically unique. For example individual people shares private data like bank account details, personal details, health records and financial data. All these private data which are stored in cloud environment needs high security and privacy. In this paper Optimized balanced scheduling is applied to perform anonymization on data sets. Here scheduling is based on the time and the size of the data sets. Anonymization approach provides privacy on individual people personal data. In large data sets it's very difficult to provide anonymizing approach; so two phase top down specialization approach is introduced to provide privacy as well as handling of large data sets in cloud. In first phase, the process of splitting large data sets into small dataset and applying anonymization on individual data sets takes place. In second phase intermediate results are merged into one and further anonymization process is applied to get the desired output. Here anonymization process is implemented using map reduce framework on cloud environment.

**Keywords:** Map Reduce, Anonymization, Balanced Scheduling, Two-Phase Top Down Specialization.

***Corresponding author***

ISSN: 0975-8585

## INTRODUCTION

Distributed systems allow for greater overall service performance than systems whose function is centralized in a single location. By spreading the computational load across different nodes, each location is under less stress. This technique allows each node to perform more efficiently, which increases the performance of overall service. One example of how this works is, In high demand messaging services, instead of dumping the load for every current user transaction onto a single server, transactions are spread across a number of different servers. In this way, the demand on each individual node is reduced, and the data each node receives percolate out to the other nodes in the background.

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. A MapReduce technique is a combination of Map() procedure that performs filtering and sorting (for example sorting of employees by first name into queues, each name is stored in one queue ) and Reduce() procedure that performs a summary operation (for example counting the number of employees in each queue, yielding name frequencies). The "MapReduce System" also called "infrastructure", "framework" is structured by marshalling the distributed servers, running various tasks in parallel, dealing with all interchanges and information exchanges between the different parts of the framework.

**"Map" step:** The master node takes the input data, partitions the problem into smaller sub-problems, and distributes them to worker nodes. A worker node does the same process again prompting 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 gathers the responses to all the sub-issues and consolidates them somehow to shape the output, the answer to the problem it was originally trying to solve.

Protection is the capacity of an individual or data about them and in this manner uncovers them specifically. The limits and substance of what is viewed as private contrast among societies and people, however share essential normal topics. Privacy is sometimes related to anonymity, the wish to stay unnoticed or unidentified in general society domain. When something is private to a person, it usually means there is something inside a person that is considered characteristically unique. The extent to which private data is uncovered accordingly relies on upon how the general population will get this data, which differs between places and over time.

TDS (Top Down Specialization) uses Taxonomy Indexed Partitions (TIPS) to improve the privacy on the data sets, but it follows the centralized approach so fails to handles the large data sets. Another one approach called distributed TDS approach, where it satisfies the distributed anonymization problem but not have the ability to handle large amount of data sets. The following problems are encountered when these approaches are used.

- The overall performance of the privacy providing is low.
- It only suitable for the small amount of data sets.
- The anonymization at each level is low.
- They are not following the scheduling strategies.

Two phase top down specialization approach has various outcomes to process the data efficiently and scalable with help of MapReduce [MR] jobs. Large amount of dataset can be handled using MR jobs. Map Reduce on cloud data has two levels of parallelization, i.e., job level parallelization and task level parallelization. Job level parallelization means numerous Map Reduce jobs are executed simultaneously to make full utilization of cloud framework. When MapReduce is incorporated with cloud, it turns to be more capable and versatile e.g. Amazon Elastic MapReduce service. Task level parallelization refers to numerous mapper/reducer tasks in a MapReduce jobs are executed simultaneously over data splits.

In this paper, we propose an exceedingly adaptable two-phase TDS approach for information anonymization in MapReduce on cloud. To make full utilization of the parallel capacity of MapReduce on cloud, specializations required in an anonymization procedure are split into two phases. In first phase, original datasets are partitioned into a group of smaller datasets, and these datasets are anonymized in parallel, producing intermediate results. In second phase, intermediate results are combined into one, and further anonymized to accomplish steady k-unknown information sets. We leverage MapReduce to accomplish the concrete computation in both phases. A

group of MapReduce jobs are designed with careful consideration and facilitated to perform specializations on information sets cooperatively.

Existing centralized top-down approach fails to handle extensive scale datasets in cloud. Two phase top-down specialization methodology defeats the above said drawback. This methodology gets input data and divides that into the small data sets. So that it consumes more time to anonymize the datasets. Optimized Balanced Scheduling (OBS) mechanism is proposed for performing anonymization process. In OBS individual dataset have separate sensitive field. Priority value is set for every data set sensitive field after thorough analysis, and then anonymization process is applied on this sensitive field depending upon the scheduling.

**Related Works**

The recent growth in cloud computing technology has significantly transformed everyone's observation on infrastructure architectures, software delivery and development models. This fast changeover in the direction of the cloud environment has fuelled worries on various critical issues in information systems, communication and information security [5]. From security viewpoint, more risks and challenges have been introduced from this migration to the clouds, crumbling a great part of adequacy of conventional protection mechanisms. As a result the objective 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 [6][7][8]. This paper proposes presenting a Trusted Third Party, tasked with assuring specific security characteristics within a cloud environment. The proposed solution calls upon cryptography, specifically Public Key Infrastructure working together with SSO and LDAP, to guarantee the authentication, integrity and confidentiality of involved data and communications. Here certificate based authorization method is adapted to provide security in cloud environment. The main benefit of this method is trusted third party is used to guarantee security. The drawback is overall performance of security issues are low compared with existing methodologies.

X. Xiao and Y. Tao et.al [1] proposed a novel technique, anatomy, for publishing sensitive data. Anatomy releases all the quasi-identifier and sensitive values in two separate tables. Integrated with a grouping mechanism, this methodology ensures privacy, and catches a lot of connection in the micro data. Linear-time algorithm is adapted to compute anatomized tables that obey the l-diversity privacy requirement and reduce the error of reconstruction of micro data. Broad investigations affirm that their strategy permits significantly more efficient data analysis than conventional publication method based on generalization. In particular, anatomy allows aggregate reasoning with average fault below 10%, which is lower than the fault obtained from a generalized table by orders of magnitude. Here rational anatomy method is adapted to provide privacy on cloud environment. Advantage of this method is, it provides high amount of accuracy and privacy. Drawback is, it failed to handle large amount of the data sets.

K. LeFevre et.al [2][3] presents a methodology for securing individual privacy, an vital issue in micro data distribution and publishing. Usually anonymization algorithms intend to fulfill certain privacy definitions with negligible effect on the quality of the resulting data. While a significant part of the previous literature has measured quality through straightforward one-size-fits-all measures, the quality of the data set is best judged regarding the workload for which the information will ultimately be utilized. The authors discusses a collection of anonymization algorithms that incorporate a target class of workloads, comprising of one or more data mining tasks as well as selection predicates. A broad experimental assessment indicates that this approach is often more efficient than previous techniques. In addition, the authors also consider the problem of scalability issues. This article describes two extensions that permit us to scale the anonymization algorithms to datasets much larger than main memory. The first extension is based on information from scalable decision trees, and the second is based on sampling, thorough performance evaluation indicates that these techniques are viable in practice. In this paper work load aware anonymization technique is used. Advantage of this technique is anonymization effect is high when compared to the existing system. Drawback is it also fails to handles the large amount of the data sets.

N. Mohammed et.al [4] discussed about the privacy concerns of sharing patient information between Hong Kong Red Cross Blood Transfusion Service (BTS) and public hospitals. Sharing healthcare data have turned into a crucial necessity in healthcare system management; conversely, inappropriate sharing and handling of healthcare data could make threats to patients' privacy. In this article, the authors take a broad view of patient personal information and privacy requirements to the issues of centralized anonymization and distributed anonymization, and recognized the real difficulties that make traditional data anonymization methods not relevant. Moreover, they

proposed a new privacy model called LKC-privacy to overcome the difficulties and present two anonymization algorithms to attain LKC-privacy in both the centralized and the distributed scenarios. Experimental analysis on real-life data reveals that their anonymization algorithms can effectively keep hold of the essential information in anonymous data for data analysis and is scalable for anonymizing large datasets. Here distribute anonymization and centralized anonymization techniques are utilized for providing privacy on cloud.

**Two-Phase Top-Down Specialization (TPTDS)**

Two-Phase Top-Down Specialization (TPTDS) approach is adapted to accomplish the computation required in TDS in a highly scalable and well-organized fashion. The two phases of TPTDS approach are based on two levels of parallelization provisioned by MapReduce on cloud. MapReduce processes the cloud dataset using two levels of parallelization. i.e., job level parallelization and task level parallelization. In job level parallelization multiple MapReduce jobs are executed at the same time to efficiently utilize the cloud infrastructure resources. Cloud computing offers infrastructure resources on-demand to the users. When MapReduce is integrated with cloud, it becomes more powerful and elastic. e.g. Amazon Elastic MapReduce service. In Task level parallelization multiple mapper/reducer tasks in a MapReduce job are executed at the same time over data splits. In first phase multiple jobs are executed parallel on data partitions, but the resultant anonymization levels are not identical. In order to obtain final consistent anonymous data set, in second phase, integration of intermediate results takes place and the entire data set are further anonymized.

In second phase all intermediate anonymization levels are merged together into one. The merging of anonymization levels is completed by merging cuts. The domain value is selected based on any one of the three conditions: (i) is identical to (ii) is more general than (iii) is more specific than. To guarantee 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, merging of **anonymization levels are performed in the same way iteratively.**

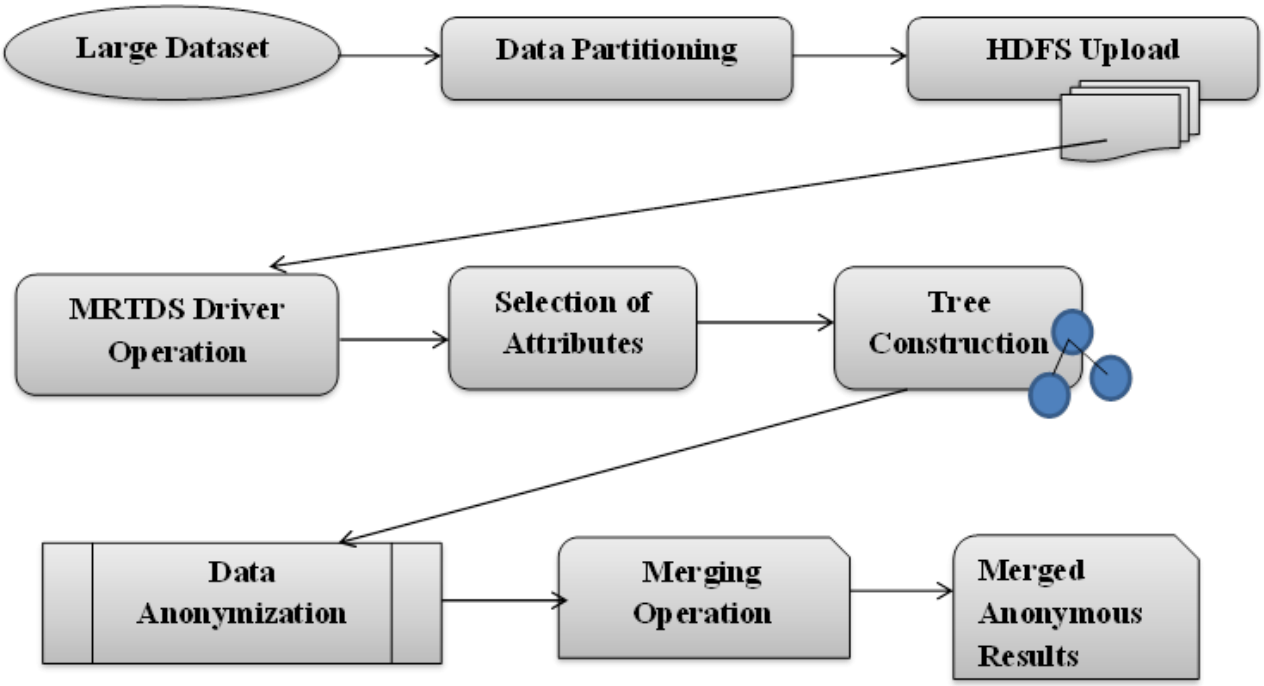


**Figure 1 System Architecture**

**Taxonomy Tree**

In this process, the root of the tree is diabetes. The children's of the root are symptoms of diabetes such as Body mass index value, Plasma glucose concentration, Blood pressure etc. Here the taxonomy tree is constructed based on the attributes of the dataset. The attributes are the patient diabetic symptoms. It shows the first level as the symptoms. There is no hierarchy than first level.

ISSN: 0975-8585

- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- 2-Hour serum insulin (mu U/ml)
- Body mass index (weight in kg/(height in m)^2)
- Diabetes pedigree function
- Age (years)
- Class variable (0 or 1)

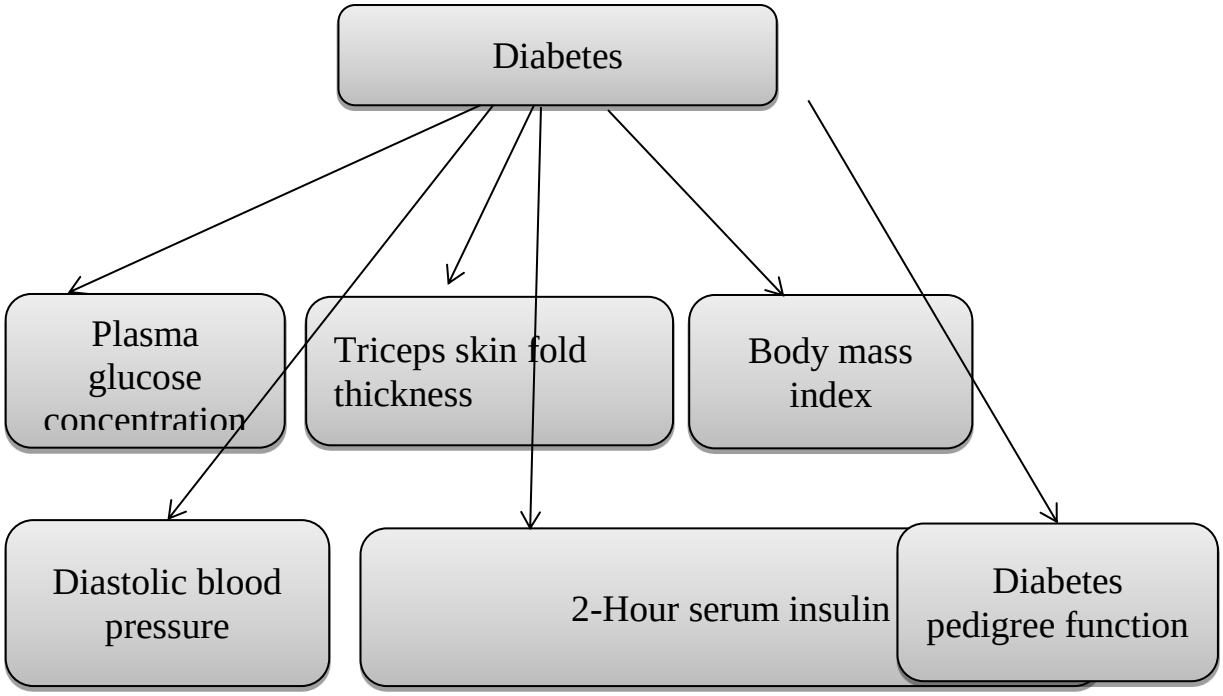


**Figure 2 Taxonomy Tree**

**IGPL Initialization- Information Gain per Privacy Loss**

IGPL is an exchange of metric that considers both privacy and information requirements, as the search metric. In this process the cut-off value is fixed as the attribute for the process. Here a label is fixed to each attribute to define score. After selecting the attribute it shows the relevant sibling in the form of 0 and 1.If it comes 1 means that attributes somewhat related to the selected attribute otherwise not.

**Quasi-Identifier Process**

Quasi-identifiers, representing groups of anonymous records, can lead to privacy breach if they are too specific that only a small group of people are linked to them. In this process Quasi-identifiers are attributes, which are fixed in the form of aligned one. After showing the aligned results, the weight for each attribute is evaluated dynamically. This is same for all mapped results.

**K-Anonymity**

The K-anonymity privacy model can combat such a privacy breach because it ensures that an individual will not be distinguished from other at least k-1 ones. The anonymity parameter is specified by users according to their privacy requirements. In this process sensitive parameters are identified and fixed as the key for anonymization. Here K is provided with the value 3 means three level anonymization is performed. So it will perform 3-Anonymity.

ISSN: 0975-8585

**Implementation of TPTDS Approach**

The implementation of Two-Phase Top-Down Specialization (TPTDS) approach is done with utmost care so that there is no slippage in satisfying the user's needs. Implementation is done using Jena a Java based API as front End. The Java Server Pages are used to create a user interactive page and Eclipse as the run time environment. The proposed system also use cygwin tool for MapReduce functions and WampServer for representing packages of independently-created programs installed on computers that use a Microsoft Windows operating system. Following are various modules of implementation process.

**Optimized Balanced Scheduling**

Optimized Balanced Scheduling (OBS) is a scheduling mechanism, which can be applied to anonymization process. OBS focuses on two kinds of scheduling (i) Size based scheduling (ii) Time based scheduling. In size based scheduling the data sets are partitioned into various partitions based on specified size. In time based scheduling anonymization process is applied on the data sets on a specified time. In OBS individual dataset have a separate sensitive field. A priority value is assigned to each sensitive field in the dataset. Then anonymization process is applied on this sensitive field depending on the type of scheduling. The OBS technique improves data locality in MapReduce thereby increasing the mapping efficiency. It is delay optimal in heavy traffic regime. It minimises the number of backlogged task as the arrival rate approaches the capacity region threshold.

**OBS Algorithm:**

Step 1: Let M be the size of the dataset D, $D_i$be the portioned dataset, NP be the number of partitions and PV be the partition value.

Step 2: Partition the dataset D into $D_i$of size M such that $D_i$, 1 <i< NP.

Step 3: Assign a priority value PV to each sensitive field in the dataset $D_i$.

Step 4: Apply anonymization on sensitive field in the dataset $D_i$ based on specific time interval.

**MapReduce Top Down Specialization –Two Phase (MRTDS)**

MRTDS plays acore role in the two-phase TDS approach, as it is invoked in both phases to concretely conduct computation. MRTDS apply anonymization process on partitioned data set to create intermediate anonymization levels. On the intermediate anonymization levels further specialization can be performed without violating the K-anonymity level. The algorithm for MRTDS approach is explained below.

**Algorithm:**

**Input:** Data set '$D_i$', partition size 'P', K- anonymity level

**Output:** Anonymous data set D*

Step 1:Split dataset D into $D_i$ such that $1 \leq i \leq P$

(i) Generate random number 'r',$1 \leq i \leq P$
(ii) Map (rand,r) to $D_i$

Step 2: Execute MRTDS (Di,Lp)→ Lip in parallel with K- anonymity level.

Step 3: Merge all anonymous levels of partitioned data into one set
$(L_{i0} + L_{i1}+L_{i2}+\ldots. +L_{ip})\rightarrow L_{ip}$

Step 4: Apply MRTDS (D,Lip) → $L^*$ to get K- anonymous dataset.

Step 5: Specialize D in same manner as $L^*$ to obtain $D^*$

**Data Partition**

In data partition large amount of data sets are collected and divide into small data sets. Then a random number is provided for each data set. In this process first the data is loaded, then the parameters present in the dataset and size of the dataset are analysed. After that based on the size of the data the dataset is partitioned and the data are upload in the cloud.Partitioning is the procedure of figuring out which reducer instance will receive which intermediate keys and values. Every mapper must decide for all of its output (key, value) pairs which reducer will receive them. It is essential that for any key, regardless of which mapper instance generated it, the destination partition is the same. 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.

**Anonymization**

Anonymization of data can ease privacy and security concerns and act in accordance with with legal requirements. Anonymization is not invulnerable counter measure that compromise current anonymization techniques which can expose protected information in released datasets.In anonymization process we are going to identify the sensitive parameters. After identification of sensitive parameters we apply anonymization to those parameters for viewing the parameters based on the anonymized results.After getting the individual data sets apply the anonymization technique. Anonymization means hide or remove the sensitive field in data sets. Data anonymization changes clear text data into a nonhuman clear and permanent form including but not restricted to pre image resistant hashes and encryption techniques in which the decryption key has been leftover.

**Merging**

In this step the intermediate results of the numerous small data sets are combined together. The MRTDS driver is used for categorizing small intermediate result for merging. The merged data sets are collected on cloud. The result of merging process is again applied in anonymization called specialization. In the merging process, we retrieve the results i.e. after retrieving the anonymized results of each and every splitted file. At last merge the entire resultant file into a single file. During this merging process the final result should be an anonymized one.

**Specialization**

In this step the intermediate results are combined into one. Then we again apply anonymization on the merged data it's called specialization. Here two kinds of jobs such as IGPL update and IGPL initialization are used. The jobs are structured by us using the driver. In the specialization process we evaluate two type of information, information gain and privacy loss for both initialization and updation phase.The driver arranges the execution of jobs. Note that we leverage anonymization level to manage the process of anonymization. It follows two steps: step 1 initializes the values of information gain and privacy loss for all specializations, which can be done by the job IGPL INITIALIZATION. Step 2 IGPL UPDATION is iterative. Firstly, the best specialization is selected from valid specializations in current anonymization level.

**Experiment Evaluation**

In order to evaluate the effectiveness and efficiency of our Optimized Balanced Scheduling (OBS) approach, we compared it with Two Phase Top Down Specialization (TPTDS) approach. Effectiveness of the algorithm is computed using scalability and utilization of data. Scalability of the algorithm is checked by testing both the algorithms on the large scale data sets. Data utilization is measured by computing the information loss (IL) caused by data anonymization process. The execution time of OBS and TPTDS are denoted as $T_{obs}$ and $T_{TP}$ respectively.

IL means information loss caused by data anonymization. For domain value $q$ in attribute $A_i$, $1 \le i \le m$, the IL of $q$ is defined as

$$IL(q)=(|Des(q)|-1)/|DOM_i|$$

Where $Des(q)$ is the set of descendant domain values of $q$. Further, the IL of an anonymous record $r^*$ is given by

$$IL(r^*)=\Sigma_{q \in r^*} w_i.IL(q)$$

ISSN: 0975-8585

Where $w_i$ specifies the penalty weight of attribute $A_i$. The value of $w_i$ is set to be 1⁄m i.e., all attributes are treated equally. The overall IL of an anonymous data set $D^*$ is calculated by

$$IL(D^*)=\Sigma_{r^* \in D^*} IL(r^*)$$

Basically, higher IL indicates less data utility.

The execution time and IL are affected by three factors, namely, data set size (S), number of data partitions (p), and the intermediate anonymity parameter ($k^l$ ). How these three factors influence the execution time and IL of OBS and TPTDS is observed in the following experiments.

We use the Diabetes dataset from UCI Machine Learning Repository, a publicly available dataset commonly used as a de facto benchmark for testing anonymization algorithms. The dataset contains health records of various diabetic patients. Each record contains 9 attributes such as Number of times pregnant, Plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/(height in m)^2), Diabetes pedigree function, Age (years), Class variable (0 or 1). Since we are evaluating the scalability with respect to data volume, the size of the original diabetes patient dataset is blown-up to generate a series of datasets.

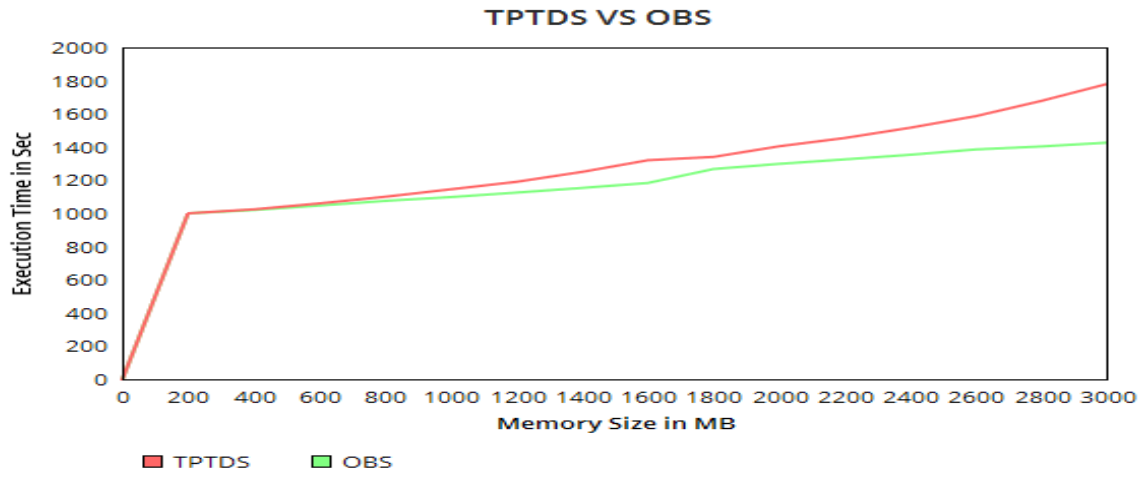


**Fig 3 Execution time w.r.t size of the dataset between TPTDS and OBS**

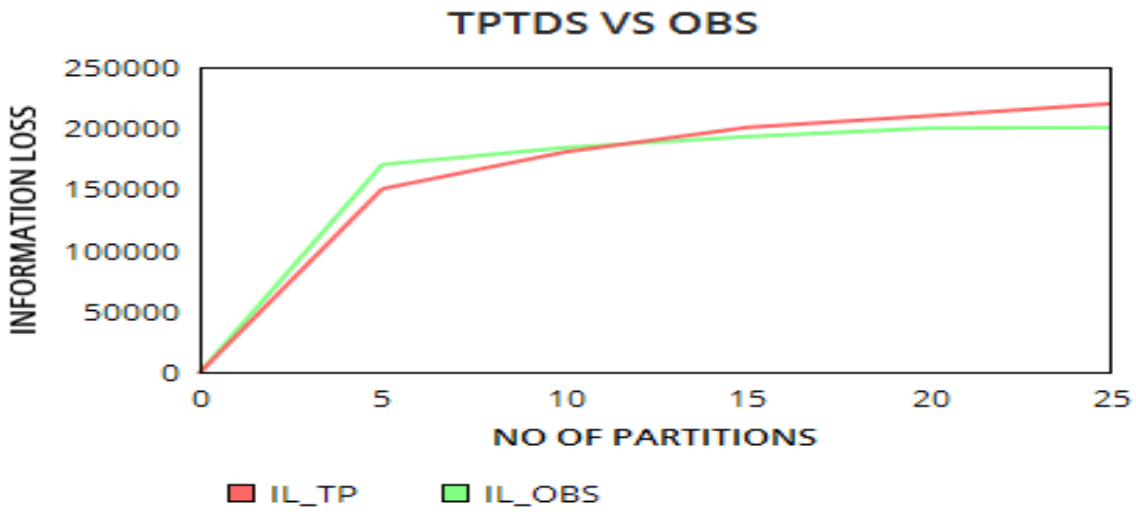


**Fig 4 Information Loss w.r.t Number of partitions between TPTDS and OBS**

ISSN: 0975-8585

**Table 1** Comparison of various Anonymization approaches.

| Paper Title | Idea | Approach | Strength | Weakness |
|---|---|---|---|---|
| k-anonymization as spatial indexing : Toward scalable and incremental anonymization. | R-tree index based approach yields a batch anonymization algorithm that has advantage of supporting incremental updates. | Building spatial index over data sets. | 1. Achieve high efficiency and quality.<br>2. High Accuracy | 1. More compaction is needed to achieve high quality<br>2. Different indexing algorithm provide different issues<br>3. High memory overload. |
| A secure distributed framework for achieving k-Anonymity. | Achieving k-anonymity using DkA-Datafly algorithm (Distributed k-anonymity) while satisfying the security definition of SMC (Secure Multiparty Computation). | 1. Uses bottom-up approach<br>2. Anonymize vertically partitioned data from various data sources without disclosing confidential information from one party to another. | Securely integrating and anonymizing multiple data sources. | Can't address scalability issues in TDS anonymization. |
| A scalable two phase Top-Down Specialization approach for Data Anonymization using MapReduce on Cloud. | Applies MapReduce on cloud to partition and anonymize data sets in parallel. | 1. Data Partition Map and Reduce.<br>2. Anonymization level merging.<br>3. Data Specialization. | Highly scalable and significant increase in efficiency of Top-Down approach. | 1. Privacy preservation in cloud.2.Sharing and mining of data is a challenging issue<br>3.Inadequacy in handling large datasets |
| Incognito: Efficient Full-Domain K-Anonymity | Implementation framework for full domain generalization using multidimensional data model together with suite of algorithms | Generalization of multidimensional data model. | To produce minimal full-domain generalization. perform up to an order of magnitude faster than previous algorithms on two real-life databases | Performance of incognito can be enhanced by materializing portions of the data cube, including count aggregates at various points in the dimension hierarchies. |

We measured the change of execution time of both $T_{OBS}$ and $T_{TP}$ with respect to the memory size when number of partition (P) is set to 1. The memory size (S) varies from 50 MB to 3 GB. The diabetes data set is big enough to evaluate the effectiveness of scalability of our approach. From fig 3 we can see that both $T_{OBS}$ and $T_{TP}$ increase slightly when the size of data set increases. There may be some slight fluctuations around the size 1600 MB – 1800MB which are mainly caused by the content of the dataset.

Both $T_{TP}$ and $T_{OBS}$ are suitable for smaller data sets. But after a particular point memory size S=800 we can see the difference between $T_{TP}$ and $T_{OBS}$ becomes larger and larger with the increase in the size of the dataset. $T_{OBS}$ grows steadily and linearly for large datasets. This shows that OBS approach becomes more efficient compared with TPTDS approach for larger data set.

**Conclusion and Future Work**

The conclusion of our proposed work is applying optimized balancing scheduling mechanism to two phase top down approach thereby increasing the mapping efficiency in large datasets. OBS provides the ability to handle the large amount of data sets. OBS minimizes the number of backlog tasks as the arrival rate approaches the capacity region threshold. OBS algorithm improves the data locality in MapReduce thereby increasing the mapping efficiency. It is delay optimal in heavy traffic regime. Here we provide privacy by effective anonymization approaches. In our future work is to reduce the handling effect of large amount of the data sets.

**REFERENCES**

1. X. Xiao and Y. Tao, (2006) 'Anatomy: Simple and Effective Privacy Preservation,'Proc. 32nd Int'l Conf. Very Large Data Bases(VLDB'06), pp. 139-150.

2. K. LeFevre, D.J. DeWitt., et al.(2005) 'Incognito: Efficient Full-Domain K-Anonymity,' Proc. 2005 ACM SIGMODInt'lConf' Management of Data (SIGMOD '05), pp. 49-60.

3. K. Lefevre, D.J. Dewitt., et al. (2008) 'Workload-Aware Anonymization Techniques For Large-Scale Data Sets' Acm Trans. Database Systems, Vol. 33, No. 3, Pp. 1-47.

4. Mohammed, Noman, et al. 'Anonymizing healthcare data: a case study on the blood transfusion service.' *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.

5. S. Chaudhuri, (2012) '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.

6. M. Armbrust, A. Fox., et al.(2010) , 'A View of Cloud Computing, 'Communication. ACM, vol. 53, no. 4, pp. 50-58.

7. L. Wang, J. Zhan, W. Shi and Y. Liang,(2012) 'In Cloud, Can Scientific Communities Benefit from the Economies of Scale?, 'IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 2, pp.296-303.

8. H. Takabi, J.B.D. Joshi and G. Ahn, (2010) 'Security and Privacy Challenges in Cloud Computing Environments, 'IEEE Securityand Privacy, vol. 8, no. 6, pp. 24-31.

9. D. Zissis and D. Lekkas, (2011) 'Addressing Cloud Computing Security Issues," Future Generation Computer Systems., vol. 28, no. 3, pp. 583-592.

10. X. Zhang, Chang Liu, S. Nepal., et al.(2012) '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.

11. N. Cao, C. Wang, M. Li, K. Ren., et al. (2011) 'Privacy- Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,' Proc. 31st Annual IEEE Int'l Conf. Computer Communications (INFOCOM'11), pp. 829-837.

12. P. Mohan, A. Thakurta, E. Shi., et al. (2012), 'Gupt: Privacy Preserving Data Analysis Made Easy,'Proc. 2012 ACMSIGMOD Int'l Conf. Management of Data (SIGMOD'12), pp. 349- 360.

13. J. Dean And S. Ghemawat, (2008), 'Mapreduce: Simplified Data Processing On Large Clusters,' Comm. Acm, Vol. 51, No. 1, Pp. 107-113.

14. B.C.M. Fung, K. Wang, R. Chen., et al. (2010) 'Privacy-Preserving Data Publishing: A Survey Of Recent Developments' Acm Computing Surveys, Vol. 42, No. 4, PP. 1-53.