In the second one, the intermediate results are integrated into one, and further anonymized to achieve consistent k-anonymous data sets. We leverage Map Reduce to accomplish the concrete computation in both phases. A group of Map Reduce jobs is deliberately designed and coordinated to perform specializations on data sets collaboratively. We evaluate our approach by conducting experiments on real-world data sets. Experimental results demonstrate that with our approach, the scalability and efficiency of TDS can be improved significantly over existing approaches.

1.2 Problem Definition:

We analyze the scalability problem of existing TDS approaches when handling large-scale data sets on cloud. The centralized TDS approaches exploit the data structure TIPS to improve the scalability and efficiency by indexing anonymous data records and retaining statistical information in TIPS. The data structure speeds up the specialization process because indexing structure avoids frequently scanning entire data sets and storing statistical results circumvents recomputation overheads. On the other hand, the amount of metadata retained to maintain the statistical information and linkage information of record partitions is relatively large compared with data sets themselves, thereby consuming considerable memory. Moreover, the overheads incurred by maintaining the linkage structure and updating the statistic information will be huge when data sets become large. Hence, centralized approaches probably suffer from low efficiency and scalability when handling large-scale data sets. 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.
In cloud environments, computation is provisioned in the form of virtual machines (VMs). Usually, cloud compute services offer several flavors of VMs. As a result, 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. A distributed TDS approach is proposed to address the distributed anonymization problem which mainly concerns privacy protection against other parties, rather than scalability issues. Further, the approach only employs information gain, rather than its combination with privacy loss, as the search metric when determining the best specializations.

1.3. Context Diagram Of Project:

![Context Diagram](image)

TPTDS approach to conduct the computation required in TDS in a highly scalable and efficient fashion. The two phases of our approach are based on the two levels of parallelization provisioned by Map Reduce on cloud. Basically, Map Reduce on cloud has two levels of parallelization, i.e., job level and task level. Job level parallelization means that multiple Map Reduce jobs can be executed simultaneously to make full use of cloud infrastructure resources. Combined with cloud, Map Reduce becomes more powerful and elastic as cloud can offer infrastructure resources on demand, for example, Amazon Elastic Map Reduce service. Task level parallelization refers to that multiple Mapper/reducer tasks in a Map Reduce job are executed simultaneously over data splits.

To achieve high scalability, we 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. Then, we run a subroutine over each of the partitioned data sets in parallel to make full use of the job level parallelization of MapReduce. The subroutine is a MapReduce version of centralized TDS (MRTDS) which concretely conducts the computation required in TPTDS. MRTDS anonymizes data partitions to generate intermediate anonymization levels. An intermediate anonymization level means that further specialization can be performed without violating k-anonymity. MRTDS only leverages the task level Parallelization of MapReduce.

2. Algorithms And Flowcharts

Tptds Algorithm

**Algorithm 1. Sketch of Two-phase TDS (TPTDS).**

**Input:** Data set \( D \), anonymity parameters \( k, k' \) and the number of partitions \( p \).

**Output:** Anonymous data set \( D' \).

1. Partition \( D \) into \( D_i, 1 \leq i \leq p \).
2. Execute \( MRTDS(D_i, k', AL^i) \rightarrow AL_i', 1 \leq i \leq p \) in parallel as multiple MapReduce jobs.
3. Merge all intermediate anonymization levels into one, \( merge(AL_1', AL_2', \ldots, AL_p') \rightarrow AL^1 \).
4. Execute \( MRTDS(D, k, AL^1) \rightarrow AL^* \) to achieve \( k \)-anonymity.
5. Specialize \( D \) according to \( AL^* \), Output \( D' \).

Data Partition Algorithm

**Algorithm 2. Data Partition Map & Reduce.**

**Input:** Data record \((ID_r, r), r \in D\), partition parameter \( p \)

**Output:** \( D_i, 1 \leq i \leq p \).

**Map:** Generate a random number \( rand \), where \( 1 \leq rand \leq p \); emit \((\text{rand}, r)\).

**Reduce:** For each \( \text{rand} \), emit \((\text{null}, \text{list}(r))\).
Data Specialization Algorithm

**Algorithm 3. Data Specialization Map & Reduce.**

**Input:** Data record \((ID, r) , r \in D \); Anonymization level \(AL\).

**Output:** Anonymous record \((r^*, count)\).

**Map:** Construct anonymous record \(r^* = (p_1, (p_1, \ldots, p_m, sv)), p_i , 1 \leq i \leq m\), is the parent of a specialization in current \(AL\) and is also an ancestor of \(v_i\) in \(r\); emit \((r^*, count)\).

**Reduce:** For each \(r^*\), \(sum = \sum count\); emit \((r^*, sum)\).

3. Modules:

* Data Partition
* Anonymization
* Merging
* Specialization
* Obs

Modules Description:

**Data Partition:**

» In this module the data partition is performed on the cloud.
» Here we collect the large no of data sets.
» We are split the large into small data sets.
» Then we provides the random no for each data sets.

**Anonymization:**

» After geting the individual data sets we apply the anonymization.
» The anonymization means hide or remove the sensitive field in data sets.
» Then we get the intermediate result for the small data sets.
» The intermediate results are used for the specialization process.
» All intermediate anonymization levels are merged into one in the second phase. The merging of anonymization levels is completed by merging cuts. To ensure that the merged intermediate anonymization level AL never violates privacy requirements, the more general one is selected as the merged one.

Merging:

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

**Specialization:**

» After geting the intermediate result those results are merged into one.
» Then we again applies the anonymization on the merged data it called specialization.
» Here we are using the two kinds of jobs such as IGPL UPDATE AND IGPL INITIALIZATION.
» The jobs are organized by web using the driver.

**Obs:**

» The OBS called optimized balancing scheduling.
» Here we 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.
» The OBS approach is to provide the high ability on handles the large data sets.
4. Simulation Results:

SCALABLE TWO-PHASE TOP-DOWN SPECILIZATION APPROACH
FOR ANONYMIZATION

LOAD DATA SET ON CLOUD

METDB DRIVER

MERGING OPERATION

DETAILS OF DATA SETS

RELATION AND WEIGHT

ATTRIBUTE | ORIGINAL RELATION | WEIGHT

STATE | 8 | 1
DOB | 4 | 1
COLLEGE | 2 | 1
NAME | 2 | 1
PHONE | 6 | 1
DISTRICT | 8 | 1
ATTEND | 6 | 1
FAVORITE | 1 | 1

LOAD STUDENT DATA SETS

FILE 1
FILE 2
FILE 3
FILE 4

CONCLUSION & FUTURE ENHANCEMENT:

Here, we have investigated the scalability problem of large-scale data anonymization by TDS and proposed a highly scalable two-phase TDS approach using Map Reduce on cloud. Data sets 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-anonymized data sets in the second phase. We have creatively applied Map Reduce on cloud to data anonymization and deliberately designed a group of innovative Map Reduce jobs to concretely accomplish the specialization computation in a highly scalable way.
Experimental results on real-world data sets have demonstrated that with our approach, the scalability and efficiency of TDS are improved significantly over existing approaches. In cloud environment, the privacy preservation for data analysis, share and mining is a challenging research issue due to increasingly larger volumes of data sets, thereby requiring intensive investigation. We will investigate the adoption of our approach to the bottom-up generalization algorithms for data anonymization.

**Future Enhancement:**

Based on the contributions herein, we plan to further explore the next step on scalable privacy preservation aware analysis and scheduling on large-scale data sets. Optimized balanced scheduling strategies are expected to be developed towards overall scalable privacy preservation aware data set scheduling.

**REFERENCES:**


