ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Hadoop Environment

Rupali Sudam Sukre, Prof.Nilesh Sable

Dept. of Computer Engineering, Imperial College of Engineering, Wagholi, Pune, India

ABSTRACT*:* Order is a central issue in information examination. Preparing a classifier requires to get a substantial gathering of information. Discharging individual particular information in its most particular state represents a risk to individual security. This paper displays a commonsense and gainful calculation for deciding a conceptual rendition of information that covers delicate data and stays valuable for institutionalizing organizing. The examination of information is actualized by practicing or specifying the level of data in a top-down and base up way until a base security necessity is traded off. This top-down and base up specialization is useful and productive for taking care of both complete and consistent properties. Our technique abuses the situation that information as a rule contains excess structures for arrangement. While speculation may evacuate few structures, different structures start to offer assistance. Our outcomes demonstrate that standard of characterization can be protected notwithstanding for exceedingly restrictive security prerequisites. This work has huge applications to both open and private areas that share data for common point of preference and profitability. Probes genuine information demonstrate that the nature of order can be protected notwithstanding for exceptionally prohibitive obscurity necessities.

KEYWORDS: Information anonymization; top-down specialization; base up speculation; MapReduce; protection safeguarding; hadoop.

I.INTRODUCTION

Data are becoming the new raw material of business. Big Data is defined as data whose extent, rangeand complexity require new construction, techniques,algorithms, and analytics to manage and extract value andhidden knowledge from it. Lots of data is being collected and warehoused through Web data, e-commerce, purchases at department, Bank/Credit Card transactions,Social network etc.The progress and innovation is no longer hindered by the capability to collect data. But, by the capacity to manage, analyze, summarize, visualize, and discover knowledge from the collected data in a timely manner and in a scalable fashion is important to leverage utilization. Many applications that employ data mining techniques involve mining data that include private and sensitive information about the subjects. One way to enable powerful data mining while preserving privacy is to anonymize the data set that includes private information about subjects before being released for data mining. One way to anonymize data set is to handling its content so that the records adhere to k-anonymity. Two common handling techniques used to achieve k-anonymity of a data set are generalization and suppression. Generalization refers to substituting a value with a less specific but semantically persistent value, while suppression refers to not releasing a value at all. Generalization is more commonly enforced in this domain since suppression may dramatically reduce the quality of the data mining results if not properly used.

MapReduce is framework or a programming model that allows carrying out tasks in parallel across a large cluster of computers. It mainly consists of two functions namely Map and Reduce. it is a challenge for existing anonymization approaches to achieve privacy preservation on privacy sensitive large-scale data sets due to their insufficiency of scalability. Data anonymization refers to hiding identity and/or sensitive data for owners of data records. Then, the privacy of an specific can be effectively preserved while certain aggregate information is exposed to data users for diverse analysis

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

and mining.Dataanonymility adopting TDS approach and uses Mapreduce which solves scalability issue by using job level and task level parallelization. Job level parallelization means that multiple MapReduce jobs can be executed simultaneously to make full use of hadoop infrastructure assests. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data splits. To make full use of the parallel capability of MapReduce on hadoop can be used in such environments.

II.RELATED WORK

Wanchun Dou et al [1] proposed the cross mists are framed with the private cloud information assets and open cloud administration segments. Cross cloud administration sythesis gives a substantial weaving machine proficient to huge scale enormous information handling. Private cloud declines to reveal all points of interest of their administration exchange records. History record based administration streamlining strategy (Hiresome-II) is a security mindful cross cloud administration piece system. QoS history records are utilized to figure the cross cloud administration sythesis arrangement. Kmeans calculation is utilized as an information separating device to pick agent history records. It decreases the time many-sided quality of cross cloud administration structure arrangement for enormous information. Kaitai Liang et al [2] portrayed the framework imparts the huge information to protection and security shield in the middle of senders and beneficiaries. Protection safeguarding figure content multi sharing hardware is utilized to achieve unsigned information imparting to administration customers. Information imparting is accomplished to figure content assault control instrument. Cross cloud environment based information sharing is not bolstered. This exploration work intends to tackle the above issues. To save secrecy, some understood encryption systems are foreseen in the fiction, for example, mysterious. By utilizing these primitives, the establishment and the objective of information can be cosseted furtively. Xuyun Zhang et al [3] composed the framework deals with the enormous information imparting to nearness mindful neighborhood recording anonymization instrument. Adaptable two-stage grouping methodology and vicinity mindful agglomerative bunching calculation are utilized to impart enormous information to protection. Protection safeguarded enormous information mining operations are not bolstered. A pragmatic and generally embraced strategy for information protection safeguarding is to anonymize information by means of clearing articulation to fulfill a given security model. JoonsangBaek et al [4] depicted enormous information gathering from power administrations is imparted to security and protection in mists. Character based encryption, mark and intermediary re-encryption systems are fused in secure distributed computing based structure for huge information process. Cross cloud based administration organization is not bolstered. The fundamental thought of our system is to construct a various leveled constitution of distributed computing focuses to give different sorts of registering administrations for data administration and huge information investigation. Xiaoyong Li et al [5] recommended that the framework deals with the trust check between the client, merchant and administration assets in numerous cloud environment. Administration administrator mindful trust scheme(SOTS) assesses the trust utilizing multi trait based model. Asset security level evaluation is not improved. This versatile weaver can defeat the edges of customary trust plans, whereby the trusted administrators are inclined physically or naturally.

Rajiv Ranjan et al [6] depicted the cloud asset provisioning routines are utilized to assemble enormous information applications. Iterative ordinal enhancement (IOO) strategy is utilized to amass huge information applications with high adaptability in mists. Protection and security elements are not bolstered. Virtualized mists present show variability in assets. Versatility has how turned into the simple component of distributed computing as it empowers the capacity to enthusiastically include or evacuate virtual machine cases when workload changes.

Yanfeng Zhang et al [7] proposed the aftereffect of information mining get to be fusty and bygone over the long run. Incremental handling is a promising weaving machine to inspiring mining results. Iterative guide decrease models are utilized to execute mining on huge information air. Incremental preparing augmentation to Guide Reduce structure is utilized for mining enormous information. Private cloud based information offering is not customized to the mining model. Fine-grain incremental preparing utilizing MRBGstore, General-reason iterative calculation with unassuming expansion to Map Reduce API, Incremental handling for iterative calculation these are the three novel components. The fundamental thought for incremental calculation for Map is direct. We basically speak to the Map capacity for the embedded or erased. K means is a regularly utilized grouping calculation that segments focuses into k bunches. Luis M.Vaquero et al [8] dercribed the virtual machines shares and excitedly stacks the huge information qualities taking into account the client wishes. Enormous

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

information provisioning administration consolidates various leveled and distributed information sharing strategies to accelerate information stacking into the VMs utilized for information handling. Information exchange planning is not upheld. The grouping of undertaking expected to set up a major information for parallel examination on an arrangement of recently conveyed VMs is as per the following: apportioning, Data dispersion, Application setup, Load information in memory. There is four methodology are executed in information conveyance brought together approach, semi-unified methodology, various leveled approach, P2P approach.

Weikuan Yu et al [9] proposed the virtual rearranging technique is utilized to empower capable information development and lessen I/O for guide decrease rearranging. Guide Reduce uses rearranging period to universally trade the go-between information produced by the mapping stage. Multi cloud information development is not upheld. Virtual rearranging is acknowledged through a blend of three strategies including a three-level fragment table, close request consolidating, and dynamic and adjusted combining sub trees. MapReduce is advanced by Google as an extremely straightforward yet persuasive project demonstrate that offers parallelized calculation, adaptation to internal failure and conveying information handling. Qi Zhang et al [10] composed errand level booking plans are utilized to distribute assets for guide decrease. Stage and Resources data mindful scheduler for Mapreduce bunches (PRISM) structure is utilized to assign assets. Security and protection elements are not considered. Guide Reduce can essentially diminish the running time of information serious occupations.

III.**BACKGROUND**

A MapReduce program is consists of a Map() procedure that performs filtering and sorting (such as sorting students by email into queues, one queue for each email) and a Reduce() procedure that performs a summary operation (such as counting the number of students in every queue, resulting name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, executing the various tasks in parallel, keeping all communications and data transfers between the various parts of the system, and giving for redundancy and fault tolerance.

The model is inspired by the map and reduces functions commonly used in programming, even though their purpose in the MapReduce framework is not the same as in their original forms. The main contributions of the MapReduce framework are not the actual map and reduce functions, but the extensibility and fault-tolerance gained for a variety of applications by optimizing the execution engine once. A single-threaded implementation of MapReduce will usually not be faster than a traditional implementation. When the optimized distributed shuffle operation (which reduces network communication cost) and fault tolerance features of the MapReduce framework come into play, is the use of this model beneficial. MapReduce libraries have been written in multiple programming languages, with separate levels of optimization. A famous open-source implementation is Apache Hadoop. The name MapReduce originally referred to the proprietary Google technology but has since been generalized.

The Hadoop distributed file system (HDFS) is a scalable, distributed and portable file-system written in Java for the Hadoop framework. A Hadoop cluster has typically a single name node plus a cluster of data nodes, redundancy options are available for the name node due to its importance. Each data node serves blocks of data over the network using a block protocol specific to HDFS. The file system makes use of TCP/IP sockets for communication. Clients use RPC(remote procedure call) to communicate between each other. HDFS stores large files (typically in the range of gigabytes to terabytes) across no of machines. It achieves reliability by replicating the data across hosts, and hence theoretically does not need RAID storage on hosts (but to improve I/O performance some RAID configurations are still useful). With default replication value, 3, data is stored on three nodes: two on the same rack, and one on a separate rack. Data nodes can communicate with each other to adjust data, to move copies around, and to keep the replication of data. HDFS is not POSIX-compliant, since the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The advantage of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

IV.PROBLEM ANALYSIS

4.1 Local-Recoding Anonymization Scheme

To facilitate subsequent discussion, we briefly introduce the concept of local-recoding anonymization as background

TABLE 1 Basic Symbols and Notions

Symbol	Notations
D	A data set containing n data records.
R	A data original record, r 2 D and r ¼hv$_1$; ...;v$_m$QI ; sv$_1$;...;sv$_m$Si, where v$_i$, 1 i m^{QI}, is a quasiidentifier attribute value, and sv$_j$, 1 j m^S, is a sensitive attribute value, m^{QI}, m^{S}are the number of the two types of attribute, respectively.
TT$_i$	The taxonomy tree of categorical attribute att$_i$.
DOM$_i$	The set of all domain values in TT$_i$for categorical attribute att$_i$, or all domain intervals for numerical attributeatt$_i$.
V$_i$	The set of attribute values of att$^{QI}_i$.
SV$_i$	The set of sensitive values of att^S_i.
Qid	A quasi-identifier, qid ¼hq$_1$; ...;q$_m$QI i, q$_i$ 2 DOM$_i$.
QID	The set of quasi-identifiers, QID ¼hDOM$_1$;...;
QIG	DOM$_m$QI i.
	The quasi-identifier group containing all records with the same quasi-identifier.

knowledge. Local recoding, also known as cell generalization, is one of the schemes outlined in [5]. Other schemes include full-domain, sub-tree and multidimensional anonymization. Local recoding generalizes a data set at the cell level, while global recoding generalizes them at the domain level. The last three schemes mentioned above are global recoding. Generally, local recoding minimizes the data distortion incurred by anonymization, and therefore produces better data utility than global recoding.

Table 1 lists some basic symbols and notations. Each record in D consists of both quasi-identifier attributes and sensitive attributes. Quasi-identifier attributes are the ones that can be potentially linked to individuals uniquely if combined with external data sets, e.g., age and sex. If a sensitive value is associated to an identified individual, economic loss or reputation damage to the person probably occur. Thus, quasi-identifiers are usually anonymized to preserve privacy, while sensitive values are often kept in the original form for the sake of data mining or data analytics. The consequence of anonymization is that data are partitioned into a set of groups, and each is represented by an anonymous quasi-identifier. Such a group is named as QIgroup, denoted by QIG in Table 1. In this way, individual privacy is preserved while aggregate information is still available for data mining or analytics.

We consider both numerical and categorical attributes for local recoding herein, and assume that a taxonomy tree is given for a categorical attribute. To facilitate the discussion, it is assumed that the attributes are arranged in order, i.e., for quasi-identifier attributes, the scheme is ATT$_{QI}$ ¼ hatt$^{QI}_1$;..;att^{QI}l$_{QI}$;att^{QI}lQIþ1;...;att$^{QI}_m$QI i, where the first l^{QI}attributes are

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

numerical while the rest are categorical, and for sensi-tive attributes, the scheme is^{S}m i, where the first l ATTS ¼ hattS1 ;...;attSlS ;attSlSþ1;...;att$_S$ S. attributes are numerical while the rest are categorical.

Based on the notions above, the local recoding scheme is formally described as follows. Data records in D can be regarded as data points in a high dimensional space. Local-recoding scheme defines a set of functions on mostly overlapping multidimensional regions which cover V_1 ,V_m, where region overlapping means that multiple regions probably contain records with identical quasi identifiers. Specifically, a function '$_l$:R_l! QID, is defined for a region R_l, where l is an arbitrary number indexing the region. In fact, a region corresponds to a QIgroup. Therefore, the core sub-problems of local recoding are how to construct multidimensional regions and how to choose functions f'$_l$g. In our approach, we leverage the clustering technique to build multidimensional regions with keeping proximity privacy of sensitive attributes in mind. For each region, the categorical quasi-identifier attribute values are generalized to their lowest common domain value in the taxonomy tree, and the numerical ones are replaced by an interval that covers them minimally. Unlike local recoding, the sub-tree and full-domain schemes have one function over each attribute, i.e., f_i : V_i ! DOM_i, 1 i m, and the global multidimensional scheme has a single function over all the attributes, i.e., f : V_1 V_m! QID.

Preserving privacy is one side of anonymization. The other one is retaining aggregate information for data mining or analytics over the anonymous data. Several data metrics have been proposed to capture this [5], e.g., minimal distortion (MD) [6], ILoss[7] and discernibility metric (DM) [7]. With a data metric, the problem of optimal local recoding is to find the local-recoding solution that makes the metric optimal. However, theoretical analyses demonstrate that the problem under most not-trivial data utility metrics is NP-hard [5]. As a result, most existing approaches [9], [11], [18], [19] just try to find the minimal local recoding instead to achieve practical efficiency and a near optimal solution, where the minimal local recoding means that no more partitioning operations are allowed when building multidimensional regions under a certain privacy model. Our proximity-aware two-phase clustering approach herein also follows this line.

So far, only the k-anonymity privacy model has been employed to preserve privacy against record linkage attacks in existing clustering based anonymization approaches. The k-anonymity privacy model requires that for any qid 2 QID, the size of QIG qidð Þ must be zero or at least k, so that a quasiidentifier will not be distinguished from other at least k 1 ones in the same QI-group [6]. Usually, it is assumed that an adversary already has the knowledge that an individual is definitely in a data set, which occurs in many real-life data like tax data sets. After local recoding, the upper-bound size of a QI-group is 2k 1under the k-anonymity privacy model. If there were a QI-group of size at least 2k, it should be split into two groups of size at least k to maximize data utility.

4.2MapReduce Basics

MapReduce [8], a parallel and distributed large-scale data processing paradigm, has been extensively researched and widely adopted for big data applications recently [9]. Integrated with infrastructure resources provisioned by cloud systems, MapReduce becomes much more powerful, elastic and cost-effective due to the salient characteristics of cloud computing. A typical example is the Amazon Elastic MapReduce service.

Basically, a MapReduce job consists of two primitive functions, Map and Reduce, defined over a data structure named key-value pair (key, value). Specifically, the Map function can be formalized as Map: ðk$_1$;v$_1$Þ ! ðk$_2$;v$_2$Þ, i.e., Map takes a pair (k_1, v_1) as input and then outputs another intermediate key-value pair (k_2, v_2). These intermediate pairs are consumed by the Reduce function as input. Formally, the Reduce function can be represented as:Reduce:ðk$_2$; listðv$_2$ÞÞ ! ðk$_3$;v$_3$Þ, i.e., Reduce takes intermediate k_2 and all its corresponding values listðv$_2$Þ) as input and outputs another pair (k_3;v_3). Usually, (k_3, v_3) list is the results which MapReduce users attempt to obtain. Both Map and Reduce functions are specified by users according to their specific applications. An instance running a Map function is called Mapper, and that running a Reduce function is called Reducer, respectively.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

4.3 Motivation and Problem Analysis

In this section, we analyze the problems of existing approaches for local-recoding anonymization from the perspectives of proximity privacy and scalability. Further, challenges of designing scalable MapReduce algorithms for proximity-aware local recoding are also identified.

Little attention has been paid to the local-recoding anonymization scheme under proximity-aware privacy modes. As mentioned in Section 2, most existing local-recoding approaches concentrate on combating record linkage attacks by employing k-anonymity privacy model. As demonstrated in existing work [7], [8], [12], [20], however, kanonymity fails to combat attribute linkage attacks like homogeneity attacks, skewness attacks and proximity attacks. For instance, if the sensitive values of the records in a QI-group of size k are identical or quite similar, adversaries can still link an individual with certain sensitive values with high confidence although the QI-group satisfies kanonymity, resulting in privacy violation. Accordingly, a plethora of privacy models have been proposed to thwart such attacks as shown in Section 2. But these models have been rarely exploited into the local-recoding scheme except the work in [13]. This phenomenon mainly results from two reasons analyzed as follows.

The first one is that, unlike global-recoding schemes, k-anonymity based approaches for record linkage attacks cannot be simply extended for attribute linkage attacks. Since global-recoding schemes partition data sets according to domains, they can be fulfilled effectively in a top-down fashion. This property of global-recoding schemes ensures that k-anonymity based approaches can be extended to combat attribute linkage attacks though checking extra privacy satisfiability during each round of the top-down anonymization process [5]. However, the local recoding scheme fails to share the same merits because it partitions data sets in a clustering fashion where the top-down anonymization property is inapplicable. Although Wong et al. [13] proposed a topdown approach for local recoding, the approach can only achieve partially local recoding because global recoding is exploited to partition data sets as the first step and local recoding is only conducted inside each partition. Consequently, their approach will incur more data distortion compared with the full potential of the local-recoding scheme.

The second reason is that most proximity aware privacy models have the property of non-monotonicity [20], which makes such models hard to achieve in a top-down way, even for global-recoding schemes. Formally, monotonicity refers to that if two disjoint data subsets G_1 and G_2 of a data set satisfy a privacy model, their union G_1 [G_2 satisfies the model as well. Monotonicity is a prerequisite for top-down anonymization approaches because it ensures to find minimally anonymized data sets. Specifically, if a data set does not satisfy a privacy model, we can infer that any of its subsets will fail to satisfy the model. Thus, when anonymizing data sets in a top-down fashion, we can terminate the process if further partitioning a subset violates the privacy model. However, most proximity-aware privacy models such as ð";mÞ-anonymity and ð;dÞ^k-dissimilarity fail to possess the property of monotonicity. As a consequence, most existing anonymization approaches become inapplicable with such privacy models [20]. A two-step approach based on the Mondrian algorithm [14] is presented in [11] to obtain ð";mÞ-anonymous data sets, via k-anonymizing a data set first, and adjusting partitions to achieve the privacy requirements then. However, this approach targets the multidimensional scheme, rather than local recoding investigated herein. Furthermore, proximity is not integrated into the search metric that guides data partitioning in the twostep approach, potentially incurring high data distortion. Wang and Liu [20] proposed an anonymization model XColor under the ð;dÞ^k- dissimilarity model, yet there is still a gap between its theoretical methodology and a practical algorithm as they acknowledged.

In terms of the analyses above, achieving the local-recoding scheme under proximity-aware privacy models is still a challenging problem. To our best knowledge, no previous work focuses on this problem. Motivated by this challenge, we propose a proximity-aware clustering approach for local-recoding anonymization.

Existing clustering approaches for anonymization are inherently sequential and assume that the data sets processed can fit into memory [9], [18], [19]. Unfortunately, the assumption often fails to hold in most big data applications in cloud nowadays. As a result, the approaches often suffer from the scalability problem when encountering big data applications. Even if a single machine with huge memory could be offered, the I/O cost of reading/writing very large data sets in a serial manner will be quite high. Thus, parallelism is not an option but by far the best choice for big data applications. Utilizing a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

bunch of small and cheap computation nodes rather a large expensive one is more costeffective, which also coheres to the spirits of cloud computing where computation is provisioned in the form of various virtual machines.

We attempt to leverage MapReduce in cloud to address the scalability problem of clustering approaches for anonymization. However, designing proper MapReduce jobs for complex applications is still a challenge as MapReduce is a constrained programming paradigm. Usually, it is necessary to consider the problems like which part of an application can be parallelized by MapReduce, how to design Map and Reduce functions to make them scalable, and how to reduce network traffics among worker nodes. The answers to these questions often vary for different applications. Hence, extensive research is still required to design MapReduce jobs for a specific application.

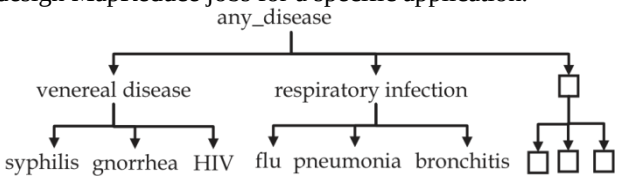


Fig. 1. A taxonomy tree of attribute Disease.

5 Proximity-Aware Clustering Problem of Local-Recoding Anonymization

Due to the non-monotonicity property of proximity-aware privacy models and characteristics of local recoding, clustering is a natural and promising way to group both quasiidentifier attributes and sensitive attributes. Hence, we propose to model the problem of local-recoding anonymization under proximity-aware privacy models as a clustering problem in this section. Specifically, a proximity-aware privacy model is formulated in Section 5.1 and the clustering problem is formalized in Section 5.2.

5.1Proximity-Aware Privacy Model

In big data scenarios, multiple sensitive attributes are often contained in data sets, while existing proximity-aware privacy models assume only one single sensitive attribute, either categorical or numerical. Hence, we assume multiple sensitive attributes in our privacy model, including both categorical and numerical attributes. As the discussion of proximity privacy attacks stems from numerical attributes, existing proximity-aware privacy models assume that categorical attribute values have no sense of semantic proximity [12], [20]. That is, categorical values are only examined whether they are exactly identical or different. Also, privacy models for categorical attributes only aims at avoiding exact reconstruction of sensitive values via limiting the number or distribution of sensitive values without considering semantic proximity [7], [8]. However, sensitive categorical values often have the sense of semantic proximity in real-life applications because the values are usually organized in a taxonomy tree in terms of domain specific knowledge. For instance, a taxonomy tree of diseases is presented in [17]. A similar one is depicted in Fig. 1 to facilitate our discussion.

Privacy breaches can still take place even if an anonymous data set already satisfies existing privacy models like l-diversity or t-closeness. For instance, an individual identified in a three-diverse QI-group with sensitive values {syphilis, gnorrhea, HIV}, will be associated with "venereal disease" with 100 percent confidence based on the taxonomy tree in Fig. 1. This inference can lead to severe privacy breach. We call such an attack as "categorical proximity breach". The core of the breach is the semantic proximity among categorical values defined by the domain specific knowledge, which is ignored in previous privacy models.

With the notion of proximity of categorical sensitive values, we extend the proximity privacy model ð;dÞ^{k}-dissimilarity in [12] to multiple sensitive attributes including both categorical and numerical ones. Our privacy model is named as ð^{þ};dÞ^{k}-dissimilarity, where "þ" implies proximity of categorical values is taken into account.

To capture the dissimilarity between sensitive values of two records, the distance metric should be defined first. Let where the meaning ofsv;sv0 2 SVidenote two sensitive values from two records,SViis already described in Table 1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

5.2 Proximity-Aware Clustering Problem of Local Recoding Anonymization

Due to the non-monotonicity property, the satisfiability problem of ð^{þ};dÞ^{k}-dissimilarity is hard. The satisfiability problem is whether there exists a partition that makes the anonymous data set satisfy dissimilarity. Based on the results in [12], we have the following theorem.Theorem 2. In general, the dissimilarity satisfiability problem is NP-hard.Proof. Since the dissimilarity satisfiability problem is proved to be NP-hard in [12], the conclusion holds as well for the dissimilarity satisfiability problem. The reason is that dissimilarity can be regarded as ð^{þ};dÞ-dissimilarity in a specific setting where categorical proximity is ignored, and a single sensitive attribute rather than multiple ones is considered.

In terms of the complexity result in Theorem 2, it is interesting and practical to find an efficient and near-optimal solution that can make the proximity among records in a QI-group as low as possible. Due to the non-monotonicity of ð"^{þ};dÞ^{k}-dissimilarity, top- down partitioning at domain levels fails to work for this privacy model. But clustering records with low proximity of sensitive values is still a promising way to make the proximity as low as possible. As clustering is also a natural and effective way for the local recoding scheme, we propose a novel clustering approach for local recoding by integrating proximity among sensitive values. Specifically, our approach attempts to minimize both data distortion and proximity among sensitive values in a QI-group when conducting clustering, unlike all the existing k-member clustering approaches (kMCs) that consider the former only. The clustering problem we attempt to address is referred to as Proximity-Aware Clustering problem, which is essentially a two-objective clustering problem.

VI.TWO-PHASE PROXIMITY-AWARE CLUSTERING USING MAPREDUCE

Except where otherwise noted, the proximity-aware clustering problem refers to the single-objective proximity-aware clustering problem hereafter. To address the SPAC problem in big data scenarios, we propose a two-phase clustering approach where agglomerative clustering method and point-assignment clustering method are employed in the two phases, respectively. We outline the sketch of the twophaseclustering approach in Section 5.1. Then, the algorithmic details of the two phases are elaborated in Sections 6.2 and 6.3, respectively. We illustrate the execution process of our approach and analyze the performance in Section 6.4.

6.1 Sketch of Two-Phase Clustering In order to choose proper clustering methods for the SPAC problem, some observations of clustering problems for data anonymization should be taken into account. First, the parameter k in the k-anonymity privacy model is relatively small compared with the scale of a data set in big data scenarios. Since the upper-bound of the size of a cluster for local-recoding anonymization is 2k 1, the size of clusters is also relatively small. Accordingly, the number of clusters will be quite large. Second, under the condition that the size of any cluster is not less than k, the smaller a cluster is, the more it is preferred. The reason is that this tends to incur less data distortion. Ideally, the size of all clusters is exactly k. Third, the intrinsic clustering architecture in a data set is helpful for local-recoding anonymization, but building such an architecture is not the final purpose.

Given the observations above, the agglomerative clustering method is suitable for local-recoding anonymization, as the stopping criterion can be set as whether the size of a cluster reaches to k. Moreover, the agglomerative clustering method can achieve minimum data distortion in the sense of the defined distance measure. Most existing approaches for k-anonymity mentioned in Section 2 employ greedy agglomerative clustering approaches. But they construct clusters in a greedy manner rather than combine the two clusters that have the minimum distance in each round, which results in more data distortion. But the optimal agglomerative clustering method suffers the scalability problem when handling large-scale data sets. Its time complexity is Oðn^{2} log nÞwith utilizing a priority queue. Worse still, the agglomerative method is serial, which makes it difficult to be adapted to parallel environments like MapReduce.

From the perspective of scalability, the point-assignment method seems to be ideal for local-recoding anonymization in MapReduce. The point-assignment process is to initialize a set of data records to represent the clusters, one for each, and assign the rest records into these clusters. The process is repeated until certain conditions are satisfied. Point assignment can be conducted in a scalable and parallel fashion in MapReduce. However, the set of representative records will become quite large according the observations above. Approximately, its size will be 1=k of an original data set. This fact makes it is a challenge to distribute such representative records to MapReduce workers who conduct point assignment according to the representative records independently. Another problem is that the size of each cluster is

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

uncontrollable in the point-assignment process. Thus, the size of a cluster can exceed the upper-bound 2k 1or be less than k, especially when a data set has high skewness. Extra effort is often required to adjust clusters to proper size. Given the pros and cons of the two clustering methods for local-recoding anonymization, we propose a two-phase approach that combines both methods based on MapReduce. In the first phase, we leverage point-assignment clustering method to partition an original data set into t clusters, where t is not necessarily relevant to k. For convenience, a cluster produced in the first phase is named as b-cluster. In the second phase, the agglomerative clustering method is run on each b-cluster simultaneously as 'plug-ins', which is similar to [31]. In this way, our approach shares the merits of both methods but avoids the drawbacks. Specifically, the twophase approach can produce quality anonymous data sets with the agglomerative clustering method and gain high scalability with the point-assignment method. In addition, no extra adjustment is required.

Algorithm 1 describes the main steps in the two-phase clustering approach. Similar to the spirit of t-means family [12], [13], we propose the t-ancestor clustering algorithm for point-assignment method. To avoid confusion, we employ the term 't-means' rather than 'k-means' which is commonly used in literature. The details of the t-ancestor algorithm and the agglomerative algorithm will be presented in Sections 6.2 and 6.3, respectively.

Algorithm 1. Sketch of Two-Phase Clustering

Input: Data set D, anonymity parameter k.

Output: Anonymous data set D.

1: Run the t-ancestor clustering algorithm on D, get a set of b-clusters: C^b¼f$C_1{}^b$;...;$C_t{}^b$g;

2: For each b-cluster $C_i{}^b$2C^b, 1 i t: run the agglomerative clustering algorithm on $C_i{}^b$, get a set of clusters C_i¼fC_{i1}; ...;C_{imi}g;

3: For each clusterby replacing each attribute value with a general one;C_j2C, where C¼[$^t_{i¼}$1C_i, generalize $C_j$to C_j

4: Generate D ¼[mj¼j1Cj, where mj ¼ Pti¼1 mi.

As t is usually required in advance, we roughly estimate it and demonstrate that the two-phase clustering algorithm is scalable in big data scenarios. Let N be the capacity of a MapReduce task worker, i.e., either a Mapper or a Reducer. Concretely, the capacity of N here means that the worker can accomplish the agglomerative clustering on a data set of size N within an acceptable time. The value of t is estimated as t jDj=N. Then, the expected maximum size of b-clusters jDj=t can be less than the worker capacity N. In fact, the skewness in a data set will affect the maximum size of b- clusters. Thus, the higher the degree of skewness is, the larger t should be. As k N according to the aforementioned observations, t will be much smaller than jDj=k which is the case if the point-assignment clustering method is exploited directly on anonymization. Hence, the set of t representative records is relative small and can be distributed to each MapReduce worker efficiently. As such, our approach can handle large-scale data sets in a linear manner with respect to the number of MapReduce workers, which can be accomplished with ease in cloud environments due to their scalability.

6.2 t-Ancestor Clustering for Data Partitioning One core problem in the point-assignment method is how to represent a cluster. Similar to t-medians [32], We propose to leverage the 'ancestor' of the records in a cluster to represent the cluster. More precisely, an ancestor of a cluster refers to a data record whose attribute value of each categorical quasi identifier is the lowest common ancestor of the original values in the cluster. Each numerical quasi-identifier of an ancestor record is the median of original values in the cluster. The notion of ancestor record also attempt to capture the logical centre of a cluster like t-means/medians, but t-ancestors clustering is more suitable for anonymization due to categorical attributes in the clustering problem herein.

To facilitate t-ancestors clustering, we take quasi-identifier attributes but sensitive ones into consideration. This will rarely affect the proximity of sensitive values in a final cluster, because the clustering granularity in the first phase is rather coarse. Accordingly, we leverage the distance measure (8) to calculate the distance between a data record and an ancestor. Usually, an ancestor is not a real data record in the data set, but the (8) can still be employed to calculate the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

distance between two vectors of attribute values. Except where otherwise noted, a record r in this section refers to the quasi-identifier part.

Initially, the t ancestors in the first round of point assignment are t records which are dedicatedly selected as seeds. In general, the selection of such t records influences the quality of clustering to a certain extent. To obtain a good set of seeds, we pick data records that are as far away from each other as possible. Concretely, we accomplish seed selection via a MapReduce job SeedSelection which outputs a set of seeds: S ¼ fr$_1$;...;r$_t$g. The Map and Reduce functions of SeedSelection are described in Algorithm 2. In this job, only one Reducer is utilized for seed selection due to the serial nature of the algorithm. To make it scalable to big data, we sample an original data set by emitting a record to the Reducer with probability N=jDjin the Map function, so that only about N records in total go to the Reducer. In the Reduce function, the first seed is picked at random, and then we repeatedly pick the record whose minimum distance to the existing seeds is the largest until the number of seeds reaches t.

Algorithm 2. SeedSelection Map and Reduce

Input: Data record r, r 2 D.

Output: A set of seeds S ¼fr$_1$;...;r$_t$g.

Map: Generate a random value rand, where 0 rand 1; if rand N=jDj, emit (1, r).

Reduce: 1: Select a random record r from listðrÞ, S r;

2: WhileFindjSrj2<listt: ðrÞthat maximizes min$_{r02S}$ dðr;r$_0$Þ;

S r;

3: Emit (null, S).

The t-ancestors clustering algorithm exploits Lloyd-style iteration refinement technique [22]. Each round of iteration consists of two steps, namely, expectation (E) and maximization (M). In the E step, data records are assigned to their nearest ancestor and constitute a b-cluster. In the M step, the ancestor of a b-cluster is recomputed according to the records in the cluster. The new set of ancestors is used in the E step of the next round. Ideally, it is expected that the iteration converges, i.e., the assignments no longer change after a finite number of rounds. However, a Lloyd-style clustering algorithm using a different distance measure other than euclidean distance fails probably to converge, or is very slow to converge. In practice, two widely-adopted stopping criteria are employed together: 1) the difference of ancestors between two continuous rounds is smaller than a predefined threshold; 2) the rounds of iteration arrive at predefined number. Formally, let S^i and S^{ðiþ1Þ} be the two sets of seeds in round i anddenotedbyði þ 1Þ,dðrespectively. TheSi;Sðiþ1ÞÞ, is defined as the average distancedifference between them, between their records:

$$dðS^i;Sð^iþ^1ÞÞ ¼Xj¼1 dr_j^i;r^ð_j^{iþ1Þ}t!: (10)$$ t

The first stopping criterion is quantified by d Sð^i;

Sðiþ1ÞÞ< t, where t is a predefined threshold. Let u denote the predefined maximum number of iteration rounds. The t-ancestors clustering algorithm stops if either of the criteria above is satisfied. Ultimately, the algorithm is described in Algorithm 3.

Algorithm 3. t-Ancestors Clustering

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Input: Data set D; parameter t; thresholds t, u.
Output: t b-clusters C^b¼f$C_1{}^b$;...;$C_t{}^b$g.

1: Run job SeedSelection, get initial seeds S^0; i 0;

2: Run job AncestorUpdate, get ancestors $S^{ðiþ1Þ}$;iWhile dðS^i;$S^{ðiþ1Þ}$Þt and i u, repeat Step 2; 3: Return b-clusters with ancestors $S^{ðiþ1Þ}$. i þ 1;

In each round of the while-loop in Algorithm 3, a MapReduce job named as AncestorUpdate is designed to fulfill the E and M steps. Specifically, the Map function of the job is responsible for point assignments in the E step, while the Reduce function accomplishes the re-computation of ancestors in the M step. The Map and Reduce functions are described in Algorithm 4. Two subroutines, MedianðÞand AncestorðÞ, are utilized in the Reduce function to calculate the medians of numerical attributes and ancestors of categorical attributes, respectively. Note that the Reduce function is scalable with setting t appropriately, and one Reducer can process more than one b-clusters in sequence if t is large enough.

Algorithm 4. AncestorUpdate Map and Reduce

Output:Input: Data recordSeeds of roundminr, r 2i,DS; seeds of round$_{ði}$i$_{þ}$1$_{Þ}$ ¼f$_r$1$_{ði}$i$_{þ}$1$_{Þ;...}$i,$_{;r}$S$_{ðt}{}^i$i$_{þ}$¼1$_{Þ}{}^f$gr. 1;...;rtg.

Map: 1: d þ1;

2: For j: 1 to t

If dðr;rminjÞ<d^{min}, then d^{min}dðr;r_jÞand j^{min} j; 3: Emit (j , r).

Reduce: 1: For l: 1 to m^{QI}

If att^{QI}_l is numerical, then v_lMedianðlistrðÞ;lÞ;

2: EmitElse(jv,lr$_{ði}$i$_{þ}$1Ancestor$_{Þ}$ ¼h$_v$1$_{;...}$ðlistr$_{;v}$m^{ðÞ}QI;li$_{).}$Þ;j

6.3Proximity-Aware Agglomerative

Clustering Unlike Section 6.2, we leverage the proximity-aware distance measure (9) for the agglomerative clustering in this section. In the agglomerative clustering method, each data record is regarded as a cluster initially, and then two clusters are picked to be merged in each round of iteration until some stopping criteria are satisfied. Usually, two clusters with the shortest distance are merged. Thus, one core problem of the agglomerative clustering method is how to define the distance between two clusters. To coincide with the objective in the SPAC problem, we leverage the complete-linkage distance in our agglomerative clustering algorithm, i.e., the distance between two clusters equals to the weighted distance between those two records (one in each cluster) that are farthest away from each other. In fact, after merging such two clusters, the distance between them is the diameter of the new cluster. Formally, the distance between clusters $C_x$and C_y denoted as dðC_x;C_yÞ, is calculated by: d C x;Cy ¼ jCxj þ Cy x maxx y dist r x;ry : (11)

r 2C ;r 2C_y

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

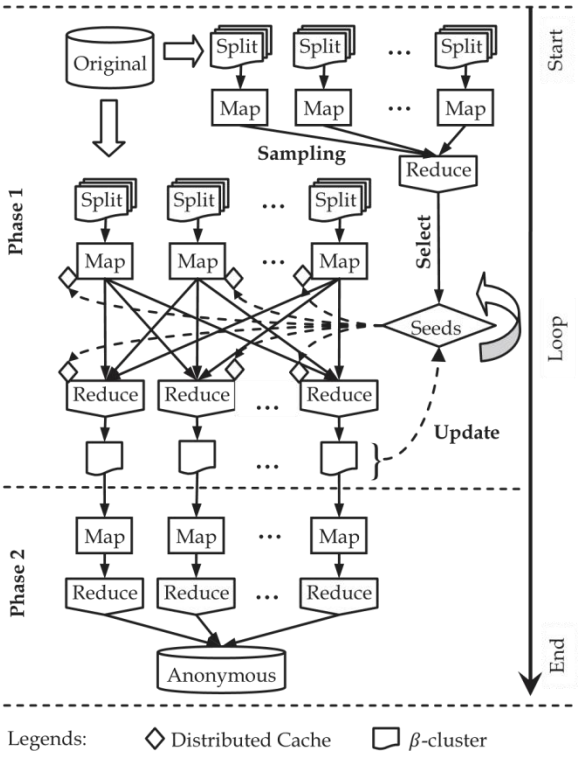
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Dissimilar to traditional agglomerative clustering algorithms, a cluster in our algorithm will not be considered for further merging if its size is equal to or larger than k. If no two clusters are of size less than k, the merging process stops. Accordingly, the maximum size of a cluster after merging is 2k 2. It is possible that a single cluster of size less than k remains after merging. We assign each data record of the left cluster to a cluster of size less than 2k 1 who is the nearest to the record. In an extreme case that all clusters are already 2k 1, we randomly pick a cluster and assign certain records from it to the left cluster to make the size of the latter be k. Note that there are only at most k 1 clusters if the extreme case takes place. Finally, every cluster resulting from the proximity-aware agglomerative clustering algorithm has at least k records, but no more than 2k 1records.

Based on the analyses above, Algorithm 5 presents the proximity-aware agglomerative clustering algorithm formally. We leverage a priority queue PQueueto retain distance between any two clusters, which aims at improving the performance of the agglomerative method. In the whileloop, the two clusters with the shortest distance are merged in step 3.1, and then the PQueueas well as the set of clusters $C^{ði þ1Þ}$ are adjusted in step 3.2 and 3.3. In step 4, we cope with the remaining cluster mentioned above without considering the extreme case.



Algorithm 5. Proximity-Aware Agglomerative Clustering Input: Data set C^b; k-anonymity parameter k.

1:Output:Initialize each record inClustersC¼fC_1;...C^b;Cas a cluster,^{n}g. C^0 ¼f$C_1{}^0$;...;$C_n{}^0$g;

C ;0 ;0i 00; 0 0 0 0

2: 8C_x;C_y2C , x 6¼ y, PQueuehdC_x;C_y;dðC_x;C_yÞi;

3: While PQueueis not empty

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

3.1:3.2: Delete entries involvinghCCðix0þ;C1Þ y0 ;dCðCinfx0 ;CCx0y0;CÞi y0 g;PQueueC$_x$0 or;CCy0z0 in PQueueCx0 [Cy;0 ;

4: 3.3:If ElseIfCC81zC0C^{ð}0iþ21k;Þð^{i}, thenþ1ÞC;PQueueðiþC1Þ [fCChði[z0þgf1;ÞCC,

---

thenz00g;C; z0 ;d8ðrC20;CCz000Þi, find a cluster; ðiþ Þ ¼ 1 and C00 2C

^{C}2C ^{and} jCj2k 2, minimizing dðfrg;CÞ, and C C [frg.

A MapReduce job named as AgglomertiveClustering is designed to wrap Algorithm 5. Specifically, Algorithm 5 is plugged in the Reduce function of the job. After a Reducer collects all data records of a b-cluster, Algorithm 5 is executed to generate final clusters (QI-groups). The Map function is relatively simple, which just emits a record and its corresponding cluster.

VII.CONCLUSIONS AND FUTURE WORK

Protection saving information examination and information distributed are getting to be not kidding issues in today's continuous world. That is the reason diverse methodologies of information anonymization systems are proposed. There are different anonymization systems present and they basically centered on k-obscurity which involves both speculation and concealment. The speculation calculations and its usage for ensuring the security of information utilized essentially for information examination.

Specifically, the paper displayed a base up speculation for changing particular information to less particular yet semantically reliable information for security assurance. TDS methodology utilizing MapReduce are connected on hadoop to information anonymization and intentionally outlined a gathering of inventive MapReduce employments to solidly perform the specialization calculation in a profoundly adaptable manner.

REFERENCES

[1] Wanchun Dou, Xuyun Zhang, Jianxun Liu and Jinjun Chen, "HireSome-II: Towards Privacy-Aware CrossCloud Service Composition for Big Data Applications", IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 2, February 2015.
[2] Kaitai Liang, Willy Susilo and Joseph K. Liu,"Privacy- Preserving Ciphertext Multi-Sharing Control for Big Data Storage", IEEE Transactions On Information Forensics And Security, Vol. 10, No. 8, August 2015
[3] Xuyun Zhang, Wanchun Dou, Jian Pei, Surya Nepal, Chi Yang, Chang Liu and Jinjun Chen, "Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud", IEEE Transactions On Computers, Vol. 64, No. 8, August 2015.
[4] JoonsangBaek, QuangHieu Vu, Joseph K. Liu, Xinyi Huang and Yang Xiang, "A Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid", IEEE Transactions On Cloud Computing, Vol. 3, No. 2, April/June 2015.
[5] Xiaoyong Li, Huadong Ma, Feng Zhou and XiaolinGui , "Service Operator-Aware Trust Scheme for Resource Matchmaking across Multiple Clouds", IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 5, May 2015.
[6] Rajiv Ranjan, Lizhe Wang, Albert Y. Zomaya, DimitriosGeorgakopoulos, Xian-He Sun and Guojun Wang, "Recent Advances in Autonomic Provisioning of Big Data Applications on Clouds", IEEE Transactions On Cloud Computing, Vol. 3, No. 2,April/June 2015
[7] Yanfeng Zhang, Shimin Chen, Qiang Wang and GeYu,"i2MapReduce: Incremental MapReduce for Mining Evolving Big Data", IEEE Transactions On Knowledge And Data Engineering, Vol. 27, No. 7, July 2015
[8] Luis M. Vaquero,AntonioCelorio, Felix Cuadrado and Ruben Cuevas, "Deploying Large-Scale Datasets on- Demand in the Cloud: Treats and Tricks on Data Distribution", IEEE Transactions On Cloud Computing, Vol. 3, No. 2, April/June 2015.
[9] Weikuan Yu, Yandong Wang, XinyuQue and Cong Xu, "Virtual Shuffling for Efficient Data Movement in MapReduce",IEEE Transactions On Computers, Vol. 64, No. 2, February 2015.
[10] Qi Zhang,MohamedFatenZhani, Yuke Yang, RaoufBoutaba and Bernard Wong "PRISM: Fine Grained Resource-Aware Scheduling for MapReduce", IEEE Transactions On Cloud Computing, Vol. 3, No. 2, April/June 2015.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

[11] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. SIGOPS Oper. Syst. Rev., 37(5):29–43, 2003.
[12]C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1099–1110, New York, NY, USA, 2008. ACM.
[13] K. Palla. A comparative analysis of join algorithms using the hadoop map/re-duce framework. Master's thesis, School of Informatics, University of Edinburgh, 2009.
[14] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In SIGMOD '09: Proceedings of the 35th SIGMOD international conference on Management of data, pages 165–178, New York, NY, USA, 2009. ACM.
[15] M. Stonebraker, D. Abadi, D. J. DeWitt, S. Madden, E. Paulson, A. Pavlo, and A. Rasin. Mapreduce and parallel dbmss: friends or foes? Commun. ACM, 53(1):64–71, 2010.
[16]A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyck-off, and R. Murthy. Hive: a warehousing solution over a map-reduce framework. Proc. VLDB Endow., 2(2):1626–1629, 2009.
[17] J. Venner. Pro Hadoop. Apress, 1 edition, June 2009.
[18] S. Viglas. Advanced databases. Taught Lecture, 2010. http://www.inf.ed. ac.uk/teaching/courses/adbs.
[19] T. White. Hadoop: The Definitive Guide. O'Reilly Media, 1 edition, June 2009.
[20] H.-C. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker. Map-reduce-merge: simplified relational data processing on large clusters. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 1029–1040, New York, NY, USA, 2007. ACM.