International Journal of Database Theory and Application
Vol.9, No.4 (2016), pp.247-258
http://dx.doi.org/10.14257/ijdta.2016.9.4.23

Sharing Attribute Names Based LSH Across Cloud Relational Database

Tan Gexu[1], Jing Xu[1 *], Liu Zhengnan[1] and Qiao Bin[2]

1. College of Information Engineering, Northwest A &F University, Yangling, Shaanxi 712100, China
2. Department of Generic Technology Promotion, Coordinating Centre of Scientific and Technological Resources, Shaanxi, 710077, China.
**Corresponding Author: Jing Xu, E-mail: jingxu@nwsuaf.edu.cn, gexutan@gmail.com*

Abstract

The problem of sharing private information in cloud relational database (CRDB) can be solved with existing techniques such as information integration with minimal sharing. However, there is no appropriate solution for the share of attribute names between two tenants. We proposed a scheme for sharing attribute names based on locality-sensitive hashing (AS-LSH). Then we proposed AS-sim protocol based simhash, which could be used in sharing attributes across CRDB. By the simulation tests, the precision and recall of AS-sim protocol are both over 90% when the length of the Chinese-attribute is longer than ten and the threshold of similarity ranges from 60% to 70%. Due to high efficiency, the scheme is suitable for the mess data, especially in the cloud.

***Keywords**: cloud relational database; attribute sharing; locality-sensitive hash; simhash*

1. Introduction

A database deployed and virtualized in a cloud computing environment is called cloud database [1]. With the properties of high scalability, high availability, multi-tenant, supporting efficient resource distribution and others, cloud database is the future direction for database technology [2]. Cloud database is classified as relational and non-relational based on data model [1]. According to reference [3], relational database still has advantage and occupies more than 80% although non-relational database has developed rapidly in recent years. Transaction is a crucial role in database operations [4], which are supported by relational database perfectly and by non-relational database hardly. For above reasons, we study in the cloud relational database (CRDB).

With the rapid development of technology, cloud computing has a number of unresolved key issues. Security is one of the vital factors which restrict the popularization and application of cloud computing. For instance, in 2009, Amazon S3 encountered twice interrupts and led a series of services to crash in February and July respectively [5]. In the same year, Microsoft Azure suffered an outage on March [5]. Because security incidents would cause serious losses and impact the confidence of users, it is an important prerequisite to protect the security and privacy of data in cloud application.

While the ownership of database is separated from its usufruct, the ownership of data is separated from the maintenance of the data. The cloud service provider (SP) may leak private information that belongs to tenant in manipulating data. It is necessary for sensitive data to be stored in cloud database after being encrypted by tenants. In the CRDB, sharing data safely among tenants is crucial. The tenants, especially those have

ISSN: 2005-4270 IJDTA

relative businesses, may have the requirement of sharing some data across database safely to obtain valuable information.

By literature retrieval, the related research is scanty. Based on secure multi-party computation, Rakesh Agrawal *et al*. [6] put forward a new paradigm of minimal necessary information sharing across private databases for intersection, equijoin, intersection size, and equijoin size without the third party.But this method had a potential security liability because the raw data was not encrypted. Carlo Curino *et al*. [7] introduced relational cloud, a scalable relational database-as-a-service (DBaaS) for cloud computing environments that could enable SQL queries to be processed over encrypted data based on adjustable privacy. CryptDB [7], a set of techniques with an acceptable impact on performance (22.5% reduction), was used in relational cloud to protect privacy which employed different encryption levels for different data, based on the types of queries. Inspired by the identity-based encryption, Li Ling [8] proposed the approach where a plurality of attributes from the users were combined to describe the data accurately, then the selection of identity were abstracted into the relativity analytics of attributes, and the threshold parameter was chose flexibly for different requirements at last. Based on secure multi-party computation, Jing Xu *et al*. [9-10] proposed two protocols about encrypted data equijoin and its size sharing across relational database, in which the two tenants could implement the equijoin across database and get the size of the equijoin for some specific attributes. The attributes must be appointed before executing those queries because only the values were shared in the solutions from Rakesh Agrawal [6] and Jing Xu *et al*. [9-10] Considering the attribute names from users were alike and different than each other, above solutions belonging to exactly match had complex progress and low efficiency.

To solve the question of sharing attributes in CRDB, in the theory of approximate nearest neighbor (ANN), we propose a scheme for sharing attribute name based on locality-sensitive hashing (AS-LSH). At first, both table name and attributes are put as a whole (in hereinafter, using attribute to represent the combination of table name and attribute name). Secondly the similarities of attributes from two tenants are computed in the methods of text relevance. Finally, the attributes whose similarity is higher than a certain threshold are shared. A protocol of attributes sharing based simhash (AS-sim) is presented to share attributes across CRDB. For users, the solution has an important significance in protecting privacy to achieve approximate matching for attributes across databases.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we develop a scheme of sharing attribute names across CRDB based LSH. And we describe the AS-sim protocol in Section 4. In Section 5, we implement the AS-sim protocol and test its performance. We end with a summary and directions for future works in Section 6.

2. Related Works

2.1. Approximate Nearest Neighbor

The nearest-neighbor (NN) problem, also be called best match and post office problem, is widely used in computer science including pattern recognition, multimedia data retrieval, vector compression, computational statistics and data mining. There are exact nearest neighbor, approximate near neighbor (ANN) and randomized nearest neighbor.

The definition of ANN, or (*r*,*c*)-*NN* [11] is: Given a set P of points in a d-dimensional space R^d, construct a data structure which given any query point q, reports any points within at most c times the distance from q to p, where p is the point in P closest to q. The c in this notion is closely associated with the threshold of similarity δ in this paper.

Suppose there are two tenant-users R and S who want to share some attributes in CRDB. The essence of finding familiar attributes by NN is comparing the relative similarity of texts. If the similarity of some attributes is larger than the certain threshold, it meets user's requirement. R and S can share the data above the attribute.

The approximation of ANN reduces the dependence on dimension from exponential size to polynomial size. Kd-tree, balltrees and LSH are the data structures to solve ANN problem [11]. For the hash algorithm, the raw inputs are definitely different when their hash values are different; the probability is negligible to obtain the raw inputs from the hash value. LSH is a kind of special hash function to protect the privacy for user.

2.2. Locality-Sensitive Hashing

In 1998, P. Indyk and R. Motwani [12] proposed locality-sensitive hashing (LSH), which is widely used in text processing, image retrieval, fingerprint identification and so on. LSH maps the vectors in high dimensions to low dimension space based on random projection, which can solve the question of ANN and NN. The idea of LSH is to construct a family of functions that hash objects into buckets, then objects that are similar will be hashed to the same bucket with high probability.

The definition of LSH [11]: A family H is called (r, cr, P_1, P_2) –sensitive where H of hash functions mapping R^d to some universe U if for any q , $p \in R^d$.

if $\| p-q \| \leq R$, then $P_H[h(q)=h(p)] \geq P_1$;

if $\| p-q \| \geq cR$,then $P_H[h(q)=h(p)] \leq P_2$;

In order for a LSH family to be useful, it hasto satisfy $P_1 > P_2$.

The neighbor points in the original data space still have the high probability to be neighbor after same mapping or projections, the nonadjacent points have the low probability to be mapped into same bucket.

LSH has many algorithm implements according various distances or similarities [13], such as simhash for angle-based distance [14], min-hash for Jaccard coefficient [15], p-stable distribution LSH for ℓp distance [16]. The accuracy of the algorithm depends on the LSH function.

Many LSH solutions suffer from "curse of dimensionality" in mass data, in which the efficiency of LSH solutions will be too low to use in high dimension. But simhash can be used in large-scale data to detect near-duplicates for web crawling by Google.

2.3. Simhash

Simhash, one of the solutions for LSH, is proposed by Moses Charikar [14]. The input of simhash is a vector and the output is an *f*-bit signature. In brief, assume the input is a set of features that have a certain weight.

For instance, a word may be used as the feature in the document and its frequency may be taken as the weight. The main steps to generate an *f*-bit fingerprint are as follows [17]:

(1) We maintain an *f*-dimensional vector *V*, each of whose dimensions is initialized to zero;

(2) A feature is hashed into an *f*-bit hash value;

(3) These *f* bits (unique to the feature) increment/decrement the *f* components of the vector by the weight of feature as follows: if the *i*-th bit of hash value is 1, the *i*-th component of *V* is incremented by the weight of feature; if the *i*-th bit of hash value is 0, the *i*-th component of *V* is decremented by the weight of feature;

(4) When all features have been processed, some components of *V* are positive while others are negative;

(5) The signs of components determine the corresponding bits of the final fingerprint.

3. AS-LSH Scheme

In CRDB, a database is shared among tenants in physical while it must be separated in logical by renting the service. When business counterparts share the same database, most tenants have demand of sharing some values for certain attributes. The attribute names must be specified before sharing the values in the existing schemes. Tenants may give the different names to attributes that have the same meaning. Since the attribute names may imply some meaning and leak the privacy of tenant, it is necessary to share the attribute fairly between tenants under privacy protection.

A scheme of AS-LSH is proposed, whose entire flow is shown in Figure 1.

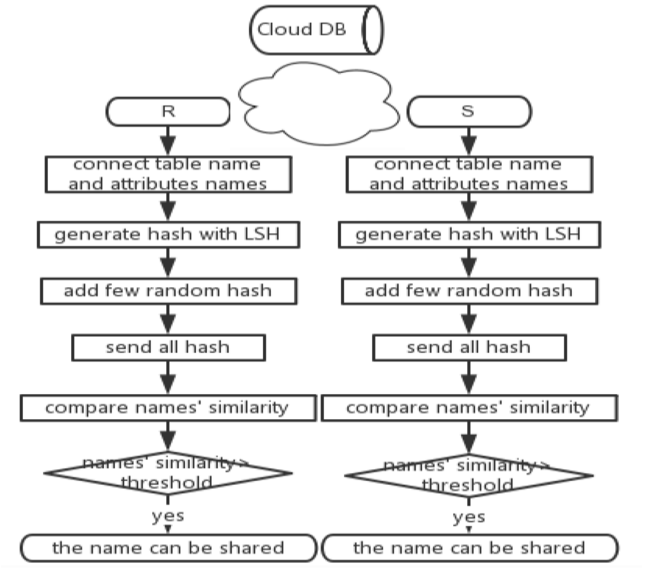
Cloud DB
R
S
connect table name and attributes names
connect table name and attributes names
generate hash with LSH
generate hash with LSH
add few random hash
add few random hash
send all hash
send all hash
compare names' similarity
compare names' similarity
names' similarity> threshold
names' similarity> threshold
yes
yes
the name can be shared
the name can be shared

Figure 1. AS-LSH Entire Flow

In the scheme, table name and attribute are linked as the whole attribute information; the similarity between attributes is analyzed by text relevance; the attributes is extracted, whose similarity is higher than the threshold. According to the user's requirement, the threshold is set in the interval between 0 and 1. This approach could not meet the condition of "need to know" where the collision of hash is unavoidable, but satisfy the rule of minimum necessary information sharing [6]. In order to obtain the mutual attribute, two tenants R and S need to exchange some additional information that may leak their privacy. Under the premise of availability of data, by obscuring a specific set [18], the privacy of the attribute number can be protected by adding uncertain data to the hash values.

According to the attribute names sharing scheme in Figure 1, it is assumed that the two tenants R and S are in the same CRDB physically with the same database license. In order to make tenant to obtain an exclusive user experience, the SP takes various methods to ensure that the tenants belong to different logical databases. The attributes cannot be operated until two tenants determine which ones can be shared. Just considering the attribute names and ignoring the table names have slight significance; according to first test, the match precision was poor when the attribute length was too short. Therefore the attribute name and its table name are connected as the whole attribute, which can lengthen the name and enhance the precision. To get the raw hash, R and S use LSH to process the attribute (table names and attribute names) respectively; then, in order to avoid leaking the set of attribute names, R and S add some pseudo-hash-values to the hash set of attributes individually; in the next, R and S send its hash set to each other, and calculate the similarity of those sets which depends on LSH algorithm implements; the attributes whose similarity is higher than a certain threshold can be shared eventually.

The selection of threshold has significant influence on the precision of sharing. The same attributes are going to be shared if the attribute names matched exactly, in other words, the similarity is 100%; the precision of sharing attributes is reducing when the similarity of the attributes decreased. Because the recall may rise up when detected the attributes whose meaning are same but names are unlike, the appropriate threshold of similarity is chosen to obtain the best performance which means the high precision and recall. Although adding some pseudo hash values can affect the match precision in sharing attributes, the impact will be too small to affect the performance of protocol in term of the result of second test.

4. A Protocol of Sharing Attributes Across CRDB Based Simhash

Because of different methods in measuring distance and similarity, LSH has various algorithm implementations. Based on the simhash algorithm which can measure the vectors distance with angle and resolve the problem of "curse of dimensionality", the protocol of sharing attributes across CRDB (AS-sim) is proposed as follows.

There are two tenants R and S in logical databases D_R and D_S respectively. The table names are T_{R1} ,…, T_{Rm} in D_R and T_{S1} ,…, T_{Sm} in D_S , and the attribute names are $A_{R11}, \ldots, A_{R1i}$ in T_{R1} and $A_{S11}, ..., A_{S1i}$ in T_{S1} . The set W is composed with table names and attribute names, such as $W_{R1} = T_{R1} \| A_{R11} \in W_R$, $W_{S1} = T_{S1} \| A_{S11} \in W_S$. C_R is generated by simhash to W_R , and C_S corresponds to W_S . F_R , F_S are the sets of pseudo-C_R and pseudo-C_S , which are generated randomly. $V_R = C_{R1}$,..., C_{Rj} , F_{R1} ,…, F_{Rp} , $V_S = C_{S1}, ..., C_{Sk}$, $F_{S1}, \ldots, F_{Sq}$. $C_R^{'}$ and $C_S^{'}$ have similar items in the set $V_R \cup V_S$ in R, $C_R^{''}$ and $C_S^{''}$ also have similar items in the set $V_R \cup V_S$ in S. $A_R^{'}$ is the sharable attributes that found by R, $A_S^{'}$ is the sharable attributes in S. R has the requested threshold δ_1 for similarity, and S has δ_2 . Concrete steps are described as follows:

1) R and S want to share some mutual attributes from D_R and D_S , they choose δ_1 , δ_2 individually;

2) R and S generate W_R and W_S ;

3) R and S generate C_R and C_S using simhash;

4) R and S choose F_R and F_S to compose $V_R = C_{R1}, ..., C_{Ri}, F_{R1}, \ldots, F_{Rm}$, $V_S = C_{S1}, ..., C_{Sl}$, $F_{S1}, \ldots, F_{Sn}$;

5) R and S send V_R and V_S each other;

6) R compare the similarity of V_R and V_S , R choose $C_R^{'}$ whose similarity is higher than δ_1 ;

7) S compare the similarity of V_R and V_S , S choose $C_S^{'}$ whose similarity is higher than δ_2 ;

8) To get $A_R^{'} \in A$, R finds $W_A^{'}$ in $W_R^{'}$ corresponding to $C_R^{'}$;

9) To get $A_S^{'} \in A$, S finds $W_A^{'}$ in $W_S^{'}$ corresponding to $C_S^{'}$.

5. Tests

5.1. Test Environment

The tests were not in complete cloud computing environment, only be testing in stimulation environment. The topology of networks is shown as Figure 1, in which cloud

DB is stimulated by MySQL on the server of CPU Intel(R) Core i7-2600 3.4GHz; memory DDR3 8192Mbytes 802.7MHz; OS Windows 7 64bits, tenants R and S is on the PC of 2.6GHz Dual-core Intel i5. The language in the tests is Java. The analyzer is Lucene and IKAnalyzer.

The most literatures use various datasets with different scales [19]. The SIFT features [20] are extracted from Photo-tourism [21] and Caltech 101 [22]; the GIST features [23] are extracted from LabelMe [24] and Peekaboom [25]. However, above datasets belong to image processing fields, are not suitable for the structured text-data. The relational cloud from MIT uses TPC-C benchmark that simulates a complete computing environment where a population of users executes transactions against a database [7]. The TPC-C benchmark was not suitable for the extracted features. So the datasets were self-designed in our research.

There are two tenants R, S in CRDB. R is a supermarket management system that stores information about the supermarket including product name, quantity, price and others; S is a new distribution center and supplied for several supermarkets. R and S do not want leak total information, such as the product types and item number. They only want to share certain information such as product names which they all have. R, S can exchange the value of these attributes if they have found the mutual attributes such as food name. Then they can cooperate for the certain goods in the future.

T_{R1} is the table of "food department product", whose attributes contain "name", "category", "sales", "inventory", "price", " profit" and so on. T_{R2} is the table of "logistics staffs information" storing some staff information, whose attributes include "staff number", "name", "age", "gender", and so forth. T_{S2} is the table of "food product information", whose attributes cover "name", "class", "manufacturer", "quantity", "price", and others; T_{S4} is the table of "supplier", which contains "manufacturer name", "product name", "shipment", "warehouse" and so on.

The ultimate aim in above example is to obtain the attribute of "product name" of food that they both have, then compute queries for this attribute across T_{R1} and T_{S2}. R and S want to protect privacy information such as the number of all attributes.

5.2. Test Scheme

The tests [17] validate that it is reasonable for a repository of 8B web-pages, 64-bit simhash fingerprints and hamming distance 3. Nevertheless, the conclusion was for English and can hardly be used in this question. On the premise of Chinese attributes, the first test started with the measurement of the precision for the similar attribute with different length; the next testing was the effect of pseudo-attributes on similarity; the third test was the precision-recall trade-offs for various thresholds; the time efficiency for the AS-sim protocol was the last one. The length of hash was 256 bits based on the trade-offs between safety and efficiency. The holistic steps for the tests are described as follows:

1) Generate 5 pairs similar attributes whose lengths vary from 2 to 20, measure the average similarity and determine the proper length of the attributes used in the following tests.

2) Generate the pseudo-attributes sets whose sizes range from 50 to 10000, then calculate the number and proportion whose similarity are higher than the assumed value, analyze the impact of those pseudo-attributes.

3) The similar attributes and pseudo-attributes are mingled, measure the precision and recall for various thresholds.

4) Calculate the time costs for different sizes of the attribute sets in the AS-sim protocol.

The evaluations of AS-sim protocol are in three aspects: space consumption, time efficiency and query quality. The space consumption depends on the length of the hash. Time costs in comparing the similarity of hash values are the most. The query quality is decided by the precision and recall of AS-sim protocol. The precision, also called positive predictive value, is the measure of exactness (or quality) that is the retrieved fraction of relevant instances. Recall, known as sensitivity, is a measure of completeness (quantity) that is fraction of relevant instances that are retrieved.

1) Precision = $\frac{|\{relevant\ texts\} \cap \{retrieved\ texts\}|}{|\{retrieved\ texts\}|}$

2) Recall = $\frac{|\{relevant\ texts\} \cap \{retrieved\ texts\}|}{|\{relevant\ texts\}|}$

The perfect precision score of 1.0 means that every retrieved result by a query was relevant and a perfect recall score of 1.0 means that all relevant documents were retrieved by the query.

5.3. Procedure and Analyze

(1)The measure of the precision for different length of similar attribute

As the length of the similar attributes is closely related with their similarities, we need to determine the length of attributes for the stable result. We choose 5 pairs similar attributes whose lengths vary from 2 to 20, measure the average similarity. The result is showed in Figure 2.

It can be seen from the Figure 2, the similarity is rising from 30 percent to 80 percent dramatically when the length is lower than six; the similarity has a sluggish increase from 80% to 87% while the length is up to 9; when the length of attribute is larger than 9, the similarity is stable between 87% and 90%. Since the low similarity can cause low quality for this protocol, it is meaningless when the length is below 6; it can work but cause some adverse effect when the length is between 6 and 9; the similarity is tending towards stable when the length is longer than 10. The recommended parameter, as well as the length of attribute, is 10.

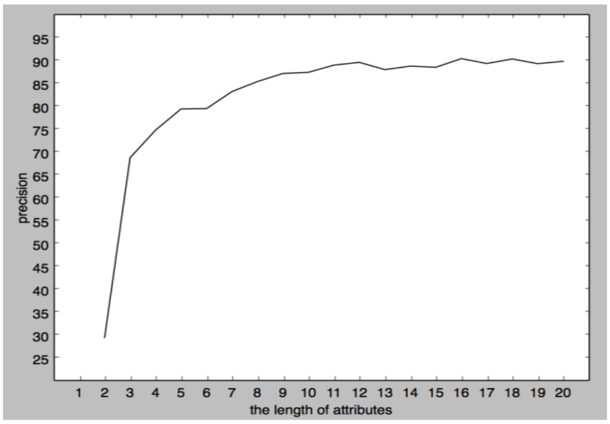

Figure 2. Similarities of the Attributes with Different Lengths

(2) The impact of pseudo-attributes on similarity

According to the results of the first test, the length of attribute is set to 10 (for Chinese Character). This part is testing the similarity of pseudo-attributes sets whose sizes range from 50 to 10,000; the next step is calculating the number and proportion of the attributes whose similarity is above certain thresholds. The results of the numbers in shown in Figure 3 and the proportion data is in Figure 4.

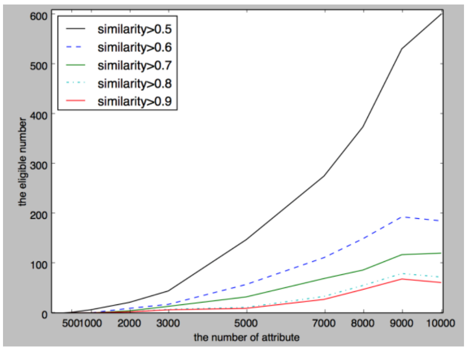

Figure 3. Amount of Eligible Attributes in Different Attribute Sets

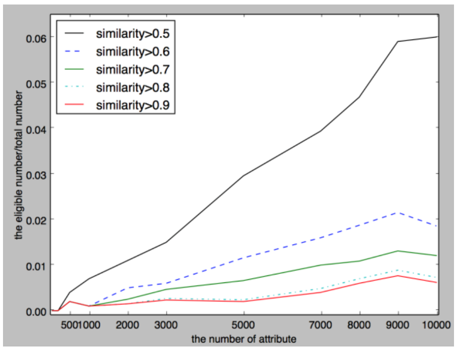

Figure 4. Proportion of Eligible Attributes in Different Attribute Sets

It can be indicated from the Figure 3 that the numbers of eligible attributes decrease from 601 to 62 when the similarity increases from 0.5 to 0.9 in the 10,000 attributes. At the same time, it can be seen from the Figure 4 that proportion of eligible attributes decrease from 6.01% to 0.62%. With the increased number of items, the eligible attributes is rising as well as the proportion, but this tendency trends towards smooth and steady after 9,000.

When the count of the random attributes is lower than 500, the test results show that there are no eligible attributes whose similarity surpasses 0.5; the added pseudo-attributes will not affect the performance of protocol when the threshold is over 0.5; the number of eligible attributes is growing slowly with the increasing of attribute counts that have little effect on the protocol.

(3) The precision-recall trade-offs for thresholds

This test is to obtain the precision and recall for the attributes with the different thresholds. The ten attributes in R are identical with those in S, and mingled with other 90 random attributes; R and S use the AS-sim protocol and measure the precision with different thresholds. The concrete result is shown in Figure 5. After the replacement of the same attributes with attributes whose similarity is 89%, the result is in Figure 6. The length of the attribute is 10 and the length of hash value is 256.

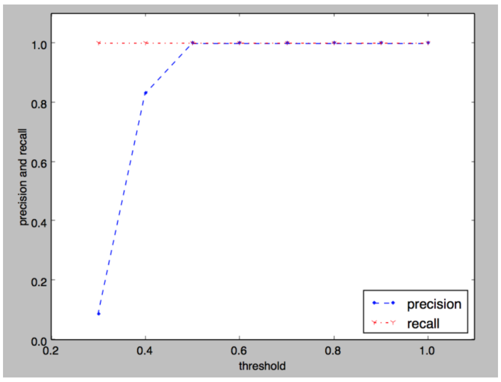

Figure 5. Precision-Recall Trade-Offs for Thresholds for Same Attributes

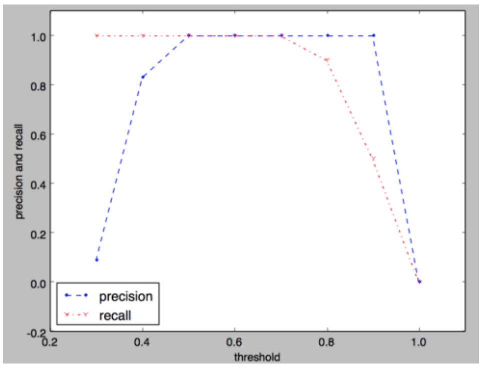

Figure 6. Precision-Recall Trade-Offs for the Attributes Whose Similarity is 89%

It appears from Figure 5 for the same attributes, the precision and recall of the attributes is 100% and suitable to exchange when the threshold exceeds 0.5. In the Figure 6, the precision is raising form 8 percent to 100 percent quickly when the threshold from 0.3 to 0.5; then the precision touches to zero when the increase of the threshold is over the average similarity; the recall reduced to zero after the threshold passes 0.7 and it is 100% in the other time. In conclusion, when the similarity of attributes is 89%, the precision and recall both are in the highest when the threshold between 0.5 and 0.7.

For the attributes with different similarity, the users should choose the threshold flexibly. To get the high precision and recall, the recommended threshold is between 0.6 and 0.7.

(4) The time efficiency for the AS-sim protocol

Two tenants apply the AS-sim protocol to exchange attribute sets with different sizes. The running time is in Figure 7.

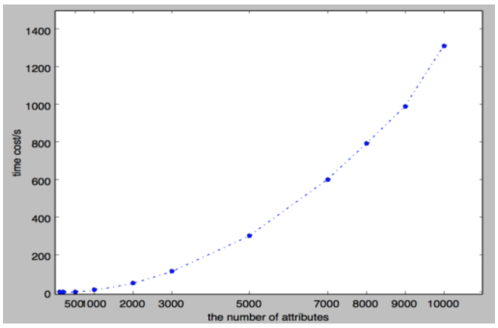

Figure 7. Time-Consumption of the AS-Sim Protocol in Attribute Sets of Varying Sizes

As is reflected in the Figure 7, the time expense of this protocol is growing with the increase of the number of attributes. The protocol can be completed in 0.309 seconds when the items of attributes is 100; the time is 0.783 seconds as the attributes is 200 and so on; the cost reaches to 1,311.904 seconds, namely 20 minutes when the number of items is 10,000. Compared with other steps, the step of searching the eligible attributes whose similarities satisfy the specify threshold cost most time. Compared with the values of attributes, the number of attribute names is very smaller, so the time consumption in AS-sim protocol will meet the needs of most tenants.

6. Conclusions

Toward the question that two tenants could not share their attribute names in CRDB with current techniques such as information integration with minimal sharing, AS-LSH scheme is proposed, and AS-sim protocol is implemented by Java in this paper.

The exchange quality (precision and recall rate), time consumption and some required parameters are tested, which include the tests of attribute similarities with different lengths, impact of pseudo-attributes on similarity, the precision-recall trade-offs for various thresholds and the time efficiency for AS-sim protocol. The space consumption was determined based the balance between security and efficiency. The relative parameter, the length of the hash values, is 256 bits. From the tests, it can be shown that the protocol has the best performance when attribute length is longer than ten characters and the threshold ranges from 60% to 70%; high performance means the high precision and recall that both are over 90% there. The protocol has high efficiency and is appropriate for the mess data such as cloud computing.

This paper use the approach in information retrieval preliminary solved the problem of sharing the names for the attributes. Some interesting directions for future research include:

(1) The protocol only targeted at the Chinese character, not including the English vocabulary and Figures;

(2) This article only study the sharing between two tenants but there still are requirements for multi-tenants in actual world.

(3) According to the tests and theoretical analysis, the security of this scheme is tolerable. However, all over the theory of LSH, it lacks of deep research on security of this hash and the existing research is also somewhat thin.

Acknowledgement

This work was financially supported by the National High Technology Research and Development Program of China (2013BAD15B02), and Chinese Universities Scientific Fund (2452015195, 2452015199, 2014YB067, QN2012033).

References

[1] J. P. Yoon, “Access control and trustiness for resource management in cloud databases”, In Grid and Cloud Database Management. Springer-Verlag, **(2011)**, pp. 109-131.

[2] L. Z. Yu, L. Y. Xuan, L. Chen, X. Yi and Z. Quan, “Research on cloud databases”, Journal of Software, vol. 23, no. 5, **(2012)**, pp. 1148–1166.

[3] TOPDB. Topdb top database index, http://pypl.github.io/DB.html. Dec. 12. 2015.

[4] R. Elmasri, “Fundamentals of database systems”, Number 553. Pearson Education India, **(2008)**.

[5] Y. Yang, C. Zhao and T. Gao, “Cloud computing: Security issues overview and solving techniques investigation”, In Intelligent Cloud Computing, Springer-Verlag, **(2014)**, pp. 152–167.

[6] R. Agrawal, A. Evfimievski and R. Srikant, “Information sharing across private databases”, In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, SIGMOD ’03, New York, NY, USA, **(2003)**, pp. 86-97.

[7] C. Curino, E. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan and N. Zeldovich, “Relational Cloud: A Database Service for the Cloud”, In 5th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, **(2011)**.

[8] L. Ling, “Research on some issues of data security in cloud computing services”, Master’s thesis, University of Science and Technology of China, **(2013)**.

[9] J. Xu, L. Bingbing and H. Dongjian, “A protocol of encrypted data equijoin sharing across private database”, Journal of Xi’an Jiaotong University, vol. 46, no. 8, **(2012)**, pp. 37-42.

[10] J. Xu, L. Shuqin and T. Gexu, “A protocol of equijoin size sharing across encrypted relational database”, Journal of Ssichuan University (Engineering Science Edition), vol. 46, no. 3, **(2014)**, pp. 95-101.

[11] G. Shakhnarovich, T. Darrell and P. Indyk, “Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)”, the MIT Press, **(2006)**.

[12] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality”, In Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM, **(1998)**, pp. 604-613.

[13] J. Wang, H. T. Shen, J. Song and J. Ji, “Hashing for similarity search: A survey”, arXiv preprint arXi: vol. 1408, no. 2927, **(2014)**, pp. 1-29.

[14] M. S. Charikar, “Similarity estimation techniques from rounding algorithms”, In Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing, STOC ’02, New York, NY, USA, **(2002)**, pp. 380-388.

[15] A. Z. Broder, “On the resemblance and containment of documents”, In Compression and Complexity of Sequences, IEEE, vol. 1997, **(1997)**, pp. 21-29.

[16] M. Datar, N. Immorlica, P. Indyk and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions”, In Proceedings of the twentieth annual symposium on Computational geometry, **(2004)**, pp. 253–262.

[17] G. S. Manku, A. Jain and A. D. Sarma, “Detecting near-duplicates for web crawling”, In Proceedings of the 16th international conference on World Wide Web, **(2007)**, pp. 141–150.

[18] G. Y. Hong, T. Y. Hai, T. S. Wei and Y. D. Qing, “Knowledge hiding in database”, Journal of Software, vol. 11, no. 18, **(2007)**, pp. 2782–2797.

[19] W. J. Dong, H. T. Shen and T. Zhang, “A Survey on Learning to Hash”, Journal of Latex Class Files, vol. 13, no. 9, **(2014)**, pp. 1-22.

[20] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International journal of computer vision, vol. 60, no. 2, **(2004)**, pp. 91-110.

[21] N. Snavely, S. M. Seitz and R. Szeliski, “Photo tourism: exploring photo collections in 3d”, In ACM transactions on graphics (TOG), vol. 25, **(2006)**, pp. 835-846.

[22] L. F. Fei, R. Fergus and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories”, Computer Vision and Image Understanding, vol. 106, no. 1, **(2007)**, pp. 59–70.

[23] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope”, International journal of computer vision, vol. 42, no. 3, **(2001)**, pp. 145–175.

[24] B. Russell, A. Torralba, K. Murphy and W. Freeman, “Labelme: a database and web-based tool for image annotation”, (tech. rep.), **(2005)**.

[25] L. V. Ahn, R. Liu and M. Blum, “Peekaboom: a game for locating objects in images”, In Proceedings of the SIGCHI conference on Human Factors in computing systems, **(2006)**, pp. 55–64.