Cong Liao

Critique - “Executing SQL over Encrypted Data in the Database-Service-Provider Model”

Summary

In this paper, the authors propose an effective approach to query the data stored in the database service provider and preserve data privacy at the same time. It encrypts the tuples in the table using common encryption techniques and table attributes using partition functions. The encrypted data are stored in the server side. When users request a query, it is translated into client side and server side queries based on the mapping functions. Most of the query computation given the basic operators implemented for the encrypted relations are executed at the server side and the optimized results are sent to the client side for decryption.

Pros

The paper has several merits. First, it provides an interesting approach addressing the privacy issue of data when using the database as a service. Instead of storing the original data in the service provider directly, it encrypt the original data by using partition/mapping functions and encryption techniques like RSA, AES, DES. etc., and store the encrypted data on the server side. Second, when executing the query, it rewrites the query into server side and client side queries, and performs most of the computation at the server side, which introduce minimum computation overhead to the client.

Cons

Encrypting the original data serves as an important basis for this approach to ensure the data privacy from the database service provider. The choice of the partition function can be risky when it comes to differential privacy issue. Besides, if the original data are constantly updated, the original partition/mapping functions would have to change frequently, which introduces additional difficulty in implementation and maintenance.

Questions

For numeric attributes, it is intuitive to create partition functions given the range of the attribute. However, for non-numeric attributes, for example, strings, what specific methods are used in the experiments?

Critique - “Human-powered Sorts and Joins”
Summary

In this paper, the authors propose the implementation of join and sort operations in a crowdsourced database Qurk. It runs over crowdsourcing platform Mechanical Turk and is able to translate query into HITs. Specifically, it presents three methods of join implementations, i.e., simple join, naive batching and smart batching. Results show that batching can greatly reduce the total number of HITs. For sorting, it evaluates three approaches, i.e., comparison-based, rating-based and hybrid of the two. The hybrid method produces more accurate results and less costs.

Pros

The paper successfully leverages the advantages of human-powered computation using crowdsourcing platform. It provides novel approaches of executing sort and join operations on the crowdsourced database, which produce accurate and reliable results in a reasonably low cost. This gives more insights into how to utilize the power of crowdsourcing systems to perform the tasks that normally machines alone are difficult to achieve a better performance.

Cons

It would be better if different types of dataset are used in the experiments rather than simply images. Besides, the use of the system seem to be limited in terms of the sensitivity level of the data. Data privacy usually considered in the traditional database systems would be an issue in such an application.

Questions

Is the Qurk system able to support more basic operations as those in the traditional database?