Exploring Change – A New Dimension of Data Analytics

The paper presents new methods for data change analysis. Most of the real world databases update themselves frequently. This introduced new method explores data and its changes to understand the change dynamics of the data. This information can be used for predicting next changes location and even values. Same information can be used for changing the data back in situations like vandalism. Paper also shows how these methods can be used on different data sources like Wikipedia, DBLP and IMDB.

Introduction section is very informative about data exploration. It explains different change types in the data and also gives example scenarios for each of them to reinforce the subject. The change cube uses 4 different attributes for storing every change but 3 of them (timestamp, id, property) are enough to identify changes. Both data changes and schema changes use these 4 attributes but schema changes have a namespace in the attribute values. Using the same format for both of them good for efficient computation, storing space and readability. In addition, database trigger, log table and other constraint changes can be stored in the same format. The authors show every change in the table format and using color heat map for number of changes increases users' understanding and benefits readability. Lastly, for improving this system, authors collected feedback from the users and they implemented shortcuts into the system for faster and better usability.

I think one of the weak point of this algorithm is storing each change operation in the database. That means each update/delete operations are going to be stored like log files. It also means that each update or delete operations has to complete 2 tasks which are updating the value and writing the change values into the change table. These extra operations can suffer database performance. Also, paper doesn't mention the performance aspect of the algorithm but I think in heavy load or extreme situations, algorithm may be miss some of the update operations to store its changed values. Since it doesn't look at the performance of the algorithm, we don't know anything about scalability of the overall system. Lastly, I don't think it was necessary to explain some of the basic database operations like sorting and splitting (with mathematic background) in this paper.

Transaction store in the separate table and each tuple has id for primary key value. It isn't necessary but they could use more precise timestamps for primary key value. In short term this would decrease the performance. However, in long term instead of storing unique id values, storing same length timestamps could be more space efficient. Since, they are precise, they would be unique because I don't think 2 different operations can occur in the same milliseconds. Besides, the authors mention some operations for transactions like sorting, slicing and splitting. There are some UI pictures in the paper but I am not sure these operations have easy to use UI elements. Finally, in the future they may be implement machine learning techniques into the system for deeper data exploration or change classification.

2019-03-02 Baran Kaya, kayab@mcmaster.ca, 400284996