Impact of delete and update queries on DBEst++ approximate query processing engine

Shyam Sundar Murali Krishnan
shyamkrishnan@ou.edu

Ramya Sai Vuyyuru
ramya.vuyyuru@ou.edu

Chapter 1

Project Proposal

1.1 Category of the project

The general sub-domain which the project is based on is the **AI- Enabled data Management**. The category that this project comes under is **The extended system with additional components (category-3)**

The reason that our project falls under this category is because our aim is to add the functionalities of delete queries and update queries and analyze how the DBEst++ engine behaves to these kind of queries because these queries has not been analyzed on the previous work. Based on the analysis we will try to modify the algorithm within the engine if it does not work well for these new set of functionalities.

The major objectives this project is as follows:

1. Include the functionalities of the delete and update queries by feeding the engine with series of delete and update queries.

2. Analyze the engine once these new functionalities have been embedded and analyze the behaviour of this engine by calculating the relative error and response time.

3. If the existing approach does not work well for the new set of functionalities we try to suggest the possible change in the algorithm within the engine.

1.2 Significance of the Project

The tuning of queries and other parameters in a database management system has been a recent topics of research. It has been proved that Machine Learning has been in help in providing effective results in tuning the parameter for the database (Van Aken et al., 2017). One of the functionality that the database looks for is the approximate query processing. The approximate query processing is the way of predicting the results of the query even before the query is executed in the database. In recent years many such query processing engines have been developed. Some of the popular engines are DBEst (Ma and Triantafillou, 2019) and DeepDB (Hilprecht et al., 2019). Both these engines uses Machine Learning algorithms in order to predict the answer to the queries given. In the case of DBEst the machine learning models like kernel density estimator and regression models have been used. In the case of DeepDB the Relational sum product networks has been used. Despite the success of these models in terms of accuracy and efficiency, there was scope of more improvements that can be done in terms of accuracy, response time and memory overheads.

These improvements were made by extending the work of DBEst model called the DBEst++ (Ma et al., 2021). This engine used Mixture Density Network (MDN) for predicting the answer to the queries. It was proved from this work that DBEst++ was much better model in terms of accuracy, response time and memory overheads. But all these analysis is done only with insertion and selection queries. Many other query structures has to be analyzed especially the delete queries in this engine. So, our aim through this project is to analyze the DBEst++ engine performance for delete and update queries and investigate the behaviour of this engine towards these queries and if doesn't work considerably well then suggest some changes to these algorithm so that it can accommodate these queries as well.

1.3 Research Methodology and Time Table

The generalized methodology for this project is as follows

1. To collect the data from the database which has delete queries and the update queries.
2. Do the data pre-processing by using the word embedding model which converts the words into useful set of numeric vectors.
3. Understand the way for calculating the response time because it a parameter for understanding the performance of the engine.
4. Analyze the engine once these new functionalities have been embedded and analyze the behaviour of this engine by calculating the relative error and response time.
5. If the existing approach does not work well for the new set of functionalities try to suggest the possible change in the algorithm within the engine.

The time table for the whole project is as follows

Tasks	Starting date	Ending date	Deliverables	Person-in-charge
Proposal Preparation	02/07/2022	02/14/2022	Proposal submitted	Shyam
Response time calculate	02/14/2022	02/21/2022	Knowing Response time	Ramya
Run DBEst++	02/21/2022	02/28/2022	Knowing DBEst++	Shyam
Collect data	02/21/2022	02/28/2022	Dataset ready	Ramya
Report Preparation	02/28/2022	03/07/2022	Submit report	Shyam
Vectorize data	03/07/2022	03/14/2022	Data to train	Ramya
Run DBEst++	03/14/2022	03/21/2022	DBEst++ new results	Shyam
Report Preparation	03/21/2022	03/28/2022	Submit report	Ramya
Analyze results	03/28/2022	04/06/2022	Results compared	Shyam
Improving algorithm	04/06/2022	04/13/2022	Achieving objective	Shyam, Ramya
Final Report	04/13/2022	05/02/2022	Submit report	Shyam
Final Presentation	04/13/2022	05/02/2022	Presentation	Ramya

Bibliography

Benjamin Hilprecht, Andreas Schmidt, Moritz Kulessa, Alejandro Molina, Kristian Kersting, and Carsten Binnig. Deepdb: Learn from data, not from queries! *arXiv preprint arXiv:1909.00607*, 2019.

Qingzhi Ma and Peter Triantafillou. Dbest: Revisiting approximate query processing engines with machine learning models. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1553–1570, 2019.

Qingzhi Ma, Ali Mohammadi Shanghooshabad, Mehrdad Almasi, Meghdad Kurmanji, and Peter Triantafillou. Learned approximate query processing: Make it light, accurate and fast. In *CIDR*, 2021.

Dana Van Aken, Andrew Pavlo, Geoffrey J Gordon, and Bohan Zhang. Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM international conference on management of data*, pages 1009–1024, 2017.