An Approach for Fast Execution of Ad-Hoc Queries in Massive Data using Indexing and Off-line Analysis

Vaibhav Jain, Mangesh Wanjari
1,2Dept. of CSE, Autonomous, SRCOEM Nagpur, Maharashtra, India

Abstract

Nowadays there is growing demand for massive data [1] processing and analysis applications. Accessing this big data may take large time depending upon the query fired. It is also observed that about 90% of the queries are fired on about 10% of the data. So there should be a mechanism for handling this 10% of data and the respective queries. Semantic analysis is used to match the queries. Processing this big data also consumes time. Every time a query is fired, the information is retrieved from the big data. If again the same query is fired, the same procedure gets repeated. One approach is to store these results temporarily into some other data base, and later access this data base to fetch the results just prior to main query execution if some similar query is fired.

We will provide an approach for fast execution of such queries to retrieve the useful data with the use of Semantic processing and Indexing.

Keywords

Big Data, Semantic Processing, Indexing.

I. Introduction

Semantic processing [4] determines the possible meanings of a sentence by focusing on interactions among word-level meanings in the sentence. Semantic query optimization provides local optimization to sub queries and also reduces unnecessary data transmission. The main idea behind semantic query optimization is to use semantic rules about data to transform a query into a more efficient but semantically equivalent query. Two queries are semantically equivalent if they return identical answers from a database state that is consistent with the semantic knowledge [4].

Semantic query optimization is not widely used in practice largely because it is difficult to encode useful semantic knowledge.

In the earlier stage, most of researches on query optimization were aimed at the traditional database such as MySQL, Oracle, and DB2 etc. Later, more and more data was stored into the distributed environment [1]. And so, many research theories and methods started focusing on it. The query optimization of massive data is becoming more and more popular in the recent research with the development of cloud computing and cloud storage technologies. The processing of such big data includes structured data processing, semi-structured data processing and non-structured data processing, along with the uncertain optimization processing and data query. The massive non-structured data processing mainly uses MapReduce [5] computing model. In order to improve the query and processing efficiency to the big data, lots of others internet massive data computing framework such as Hadoop, Hadoop++, Spark, CrowdDB and Yale university’s HadoopDB are proposed based on the MapReduce [5][1], in these recent years. The objective of all these new computing frameworks is to improve the query efficiency to big data. Query optimization methods will improve the efficiency a lot to the massive data processing.

While retrieving the information from the big data, some of the queries repeat more often than the others. The results of such queries can be stored into some temporary buffer and later can be fetched directly when the similar query is fired so as to reduce the time. The queries can be exactly or partially similar. Semantic analysis is used to find the matching between these queries. This paper discusses about the technique to retrieve the data faster by fast execution of ad-hoc queries using offline analysis [2], semantic matching and indexing. A threshold is decided to calculate the similarity measure of the queries. Ad-hoc queries are given as input, semantic matching is performed, threshold is checked, and the result is retrieved based on the threshold.

II. Related Work

A lot of work has been done in the field of semantic analysis and data mining. This paper is influenced from the SemanQuery Architecture [1], the work of Guigang Zhang et al and the S4 System [2] by Xiao Yu et al. First paper introduces the semantic analysis of the queries from the big query network. If the big query network has the query plans similar to that of the users’ query, SemanQuery will get the query plans’ query paths in the big query network and execute these query plans respective to the query paths. If users’ query plans are not found by the SemanQuery in the big query network, it will add the query plans to big query network and create a new big query network.

The S4 System takes a sub graph query as input, and retrieves from an information network a list of sub graphs that satisfy the query criteria. The S4 System produces off-line data mining results as indices, utilizes semi-structure information encoded in information networks and answers semantic similarity based sub-structure queries efficiently. It uses Similarity Indexing and Structured Indexing.

III. Proposed Approach

• We assume a network of all the previously executed queries, there is a small network of semantically matched queries with comprehensive information.

• The results of these queries are stored in some temporary buffer so as to fetch them directly later when required.

• The other part takes user queries as input.

• The runtime analysis of this query is done, in which semantic analysis is performed.

• Based on this Query evaluation is done.

• Thus similarity measure is calculated considering the threshold decided.

• If the similarity measure is greater than the threshold, the result that is stored is fetched directly with executing the original query. Otherwise, the query is executed normally.

• This saves the time required to fetch the similar data again and again from the big data, resulting in fast execution of the user query.

We will focus our work on Indexing, Similarity Measure and Validation part. The assumptions are made related to previous work completed in the area. These include a big query network of all the fired queries, big data in structured form, any -a simple one for a start- input query, already generated runtime query plans. Similarity measures can be calculated using various semantic
techniques and threshold. The temporary results can be stored into views which can easily and efficiently be updated and at runtime everytime a new result is being added to it. This will help fetch the intermediate results from the view and if the range of the result is greater than that in the present view, it will be updated to store this new range and so on. Thus saving a big time of information retrieval from the big data as the view is relatively much smaller than original database.

Indexing will help check the query into the query table and fetch it faster. Also the results matching to the query will be fetched using the indexing.

Two separate databases, one for the queries and the other for temporary results, would be required to maintain and update continuously with every new query being executed.

Fig. 1: Proposed Approach

**IV. Conclusion and Future Work**

Semantic analysis of the online queries makes it useful for their fast execution. The buffer contains the results of the queries which are fetched as the threshold value of the similarity measure crosses. Indexing makes it faster to search the result from the buffer. Various techniques of semantic matching of the queries can be used to make it more efficient and faster. Also buffer management techniques can be used for easy data retrieval and storage. The semantically matched queries database and the buffer must be updated regularly according to the changes.

**References**


[2] Xiao Yu, Yizhou Sun, Peixiang Zhao, Jiawei Han, "Query-Driven Discovery of Semantically Similar Substructures in Heterogeneous Networks", Department of Computer Science, University of Illinois at Urbana-Champaign.


Vaibhav Jain has received his B.E. degree in Computer Engineering from MAEER’s Maharashtra Academy of Engineering, Alandi, Pune, Pune University in 2011. He is pursuing Mtech in Computer Science and Engineering from Shri Ramdeobaba College of Engineering and Management (Autonomous), Nagpur. His research interests include query optimization and semantic analysis.

Mangesh Wanjari has received his B.E. in Computer Technology from Nagpur University in 2002. He has received his Master of Technology (MTech) in Computer Science and Engineering from VNIT, Nagpur in 2009. After having some industrial experience he has joined the teaching field. He is an associate professor in Ramdeobaba College of Engineering and Management, Nagpur. His research interests include Database Technologies, Query Optimization and Semantic Analysis.