Review Paper for Query Optimization for Declarative Crowdsourcing System

Nilesh. N. Thorat[1], A. B. Rajmane[2]
[1]PG student, Ashokrao Mane Group of institution, Vathar
[2]Associate Professor,Ashokrao Mane Group of institution, Vathar

***Abstract*— Main goal of declarative crowdsourcing system is to keep out of sight the complexities of the system. In crowdsourcing system data is placed at different web or data servers therefore optimization of query has major challenge. To address some queries we need to add more information to that queries that machines cannot solve. SQL query is being submitted by user which is being optimized on cost constraint depending on which execution plan is created, output is produced. The proposed system can be used for answering declarative queries posed over stored relational data together with data obtained on demand from the crowd. The proposed system is based on cost-based query optimization for crowdsourcing system, which also consider the latency that generates best possible and suitable query execution plan. It supports three types of query evaluation: select, join, complex-join, having three algorithm for each operation respectively.**

***Index Terms*- Crowdsourcing, query optimization, human intelligence tasks (HIT), Monetary cost.**

I. INTRODUCTION

Crowdsourcing is modern business which can be defined as the process of obtaining needed services, ideas, or content by collecting contributions from a variety of people, especially an online community rather than from employees or suppliers. Crowdsourcing is more efficient tool where some task cannot be performed solely with computers. Crowdsourcing takes an explicit and an implicit route which depend on situation. Explicit crowd-sourcing lets users to work together for evaluating, sharing and building different specific tasks, while implicit crowd-sourcing states that user has to solve a problem as a side effect of something else they are doing. With explicit crowdsourcing, users can evaluate particular items such as books,web-pages, or they may share their items. Users can also build artifacts with the help of providing information or editing other people work. Implicit crowd-sourcing has two types: standalone or piggyback. Standalone gives permission to people to solve problems as a side effect of the task they are actually doing, whereas piggyback collects users information through a third-party website to gather information.

Query optimization is the process in which we choose the most efficient way of execution for given SQL statement. The basic aim of optimizer is to generate the best execution plan for given SQL statement. The best execution plan has the lower cost between all possible query execution plans. The cost computation factors of query execution are depend on I/O overload, CPU and communication. The best method of execution depends on various conditions including how the query is written, the size of the data set on which query fetches its results, how the data is stored, and accessing structures it uses to fetch the data. The optimizer identify the best plan for a SQL statement by identifying various accessing methods, namely either full table scan or index scans and different join methods including nested loops and hash joins. Since the database has it's own internal statistics and tools at its disposal, the optimizer is usually in a good position as compared with the user to determine the best method of statement execution. Due to which, all SQL statements take advantage of the optimizer.

Recently some crowdsourcing system, are introduced such as CrowdDB, Qurk ,and Deco which provides declarative crowd interface with help of SQL query language. SQL is a non procedural language, so the optimizer is free to merge, reorganize, and process in any order. The interfaces of these systems hides the complexities of dealing with crowd source system which gives user friendly look. The above said systems mainly divided into three stages, in first stage compilation of the given query takes place ,in second stage it should generate an execution plan for a given query, and in third stage ,system should post human intelligence tasks (HITs) to the crowd. According to the plan, it should collect the answers, handle errors and resolve the inconsistencies in the answers.

Optimization mechanisms for traditional databases are mainly classified into rule based and cost based. A rule-based optimizer simply applies a set of rules such as rewriting rules for predicate push-down, join ordering. CrowdDB is an example system that uses a rule-based query optimizer. Rule based optimization is easy to implement, but it has limited optimization capabilities since complex queries cannot easily solved by it.Also it leads to ineffective execution plans.

Declarative queries allow the programmers to write data manipulation code without considering underlying data structure which include it. Increased level of abstraction above imperative code, they improves program readability and also creates opportunities for automatic query parallelization and query optimization.Declarative querying improves optimization capacity of system which provides best nearby solution to query which is fired at end of crowd system.There are so many ways to solve declarative query for crowdsourcing system, such as number of question included in query , the types or difficulties of the questions and cost of query execution. It is therefore important to design an efficient crowdsourcing query optimizer which should consider all good query plans and select the "best" plan based on the cost model and optimization objectives.

II.RELATED WORK

Davidson, Khanna, Milo,Roy[1] , state that Group-by and top-k are basic constructs in database queries. However, the base used to group and to order certain types of data –for example such as unlabeled photos clustered by the same person ordered by age – are not easy to evaluate by machines. While evaluating top-k and group-by queries with the help of crowd the answer may be either type or value questions. Suppose that two data elements are given, then answer to a type question is "yes" if the elements have same type, so they belong to same cluster or identical group ; i.e. two data elements are ordered based on answer to value question.Results from crowd source are fetched using predefined assumption but it may be incorrect. They introduced efficient algorithms for top-k and group-by for problems in crowd source systems, which gives results with high probability.

Ju Fan, Meiyu Lu, Beng Chin Ooi, Tan, Zhang[2], state that , the web is full featured data in terms of HTML tables .If these HTML tables are integrated ,it gives rise to a knowledge repository but semantic correspondences between web table columns need to be checked, it can be carried out with help of conventional schema matching but they won't produce good result, as sometime it may be incomplete. They proposed the system with two solutions for web table matching which solves semantic correspondences and schema matching. First, concept-based approach is designed which deals with mapping of each column of web table to best concept, which solves problems for columns which are disjoint ,due to incomplete values of columns. Second, hybrid machine crowdsourcing framework deals with incomplete column with concept matching tasks to the crowd under the constraint of budget and utilizing the crowdsourcing result to help the algorithm to produce the best matches for the rest of the columns.

Franklin,Tim Kraska, Ramesh, Reynold Xin [3] proposed CrowdDB system which performs a computationally difficult functions, such as matching, ranking, or aggregating results based on fuzzy criteria. CrowdDB takes input from human with the help of crowdsource system for providing information that is missing from the database which cannot easily get answered by database systems or search engines. CrowdDB resembles with traditional database system with some big change. Traditional database systems does not take human input for query processing. From an implementation point of view human-oriented query operators are needed to integrate as well as cleanse crowdsourced data. Performance as well as cost depends on a number of new factors including worker affinity, training fatigue, motivation and location.

Chien-Ju Ho, Jabbari and Vaughan[4], state that Crowdsourcing markets is a tool that collect data from very different workers. Worker uses label for classification of common tasks but it may be error prone, at a particular time it can be treated as spam

.The solution to this problem can be obtained by collecting labels for each instance from multiple workers. With the help of online primal-dual techniques, classification tasks of task assignment and label assisgnment for workers can be carried out in heterogeneous way. They show that adaptively assigning workers to tasks can lead to more accurate predictions at a lower cost when the available workers are diverse.

H. Park and J. Widom[5] proposed comprehensive system named “Deco” which deals with answering the query depending on stored relational data together with data obtained from the crowd . The basic objective is to fetch best query plan with the help of optimized query on the basis of user estimated monetary cost. Novel techniques are used in Deco’s query optimizer system which include cost model that can easily differentiate between “free” existing data versus paid new data. Cardinality estimation algorithm deals with changes to the database state during query execution. Plan enumeration algorithm uses common subplans repeatedly in a setting that makes its reuse challenging task.

A. D. Sharma, H. Garcia-Molina,A. Parameswaran, and A. Halevy[6],proposed a system named CrowdFind which deals with problem of searching some items which satisfy fixed properties within data set for people. Suppose that a person wants to identify total no of travelling photos from a travel website, since the data for this constraints may be very large, also monetary cost and latency would be large. They proposed optimal algorithm which has comparison capacity between statistic cost versus actual time to evaluate the query. They study the deterministic as well as error-prone human answers, along with multiplicative and additive approximations. Lastly, They study how to design the algorithms with specific expected cost and time measures.

III.PROPOSED WORK

The query optimization is used for following three types of queries

1.Selection query : The selection query is used to fetch data from database. SELECT is the most commonly used data manipulation language (DML) command. SQL SELECT statement has capacity to retrieve data from a table in the database. A query may retrieve information from specified columns or from all of the columns in the table. A selection query has one or more human-recognized selection conditions over the tuples within a single relation. Selection query has many applications in real crowdsourcing scenarios, such as filtering data and finding certain items.

2.Join query : The SQL join clause is used to combine records from two or more tables in a database. A JOIN is a nothing but merging fields from two tables by common value which is present to each table. Several operators can be used to join tables, including =, <, >, <>, <=, >=, !=, BETWEEN, LIKE, NOT; they be used to join tables. However, the most common operator is the equal symbol. Following are the types of Join query.

- ✓ INNER JOIN creates a new result table by combining column values of two tables (table A and table B) based upon the join-condition. The query compares every row of table A along with every row of table B to identify matching pairs of rows which satisfy the join-condition. When the join-condition is satisfied, column values for each matched pair of rows of table A and table B are merged into a result row.
- ✓ The SQL LEFT JOIN returns all rows from the left table, even if there are no matches in the right table. This can be interpreted that if the ON clause found 0 (zero) records in right table, the join will still return a row in the result , but including NULL value in each column from right table. That is a left join returns all the values from the left table(which satisfy join condition), plus matched values from the right table or NULL in case of no matching for join condition.

- ✓ The SQL RIGHT JOIN returns all rows from the right table, even if there are no matches in the left table. This can be interpreted as if the ON clause matches 0 (zero) records in left table, the join will still return a row in the result, but with NULL value in each column from left table.That is a right join returns all the values from the right table(which matches with join-condition), plus matched values from the left table or NULL in case of no matching for join-condition.

- ✓ The SQL FULL JOIN merges the results of both left and right outer joins into single record.The joined table will contain all records from both tables, and fills NULL value for missing matches on either side.

- ✓ The SQL SELF JOIN is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.

- ✓ The CARTESIAN JOIN or CROSS JOIN returns the Cartesian product of the bunch of records from the more than two joined tables or two joined tables. That is , it checks equivalence to an inner join where the join-condition always turns to be True or where the join-condition is not present from the statement.

  A join query controls human intelligence to combine tuples from two or more relations according to certain join conditions.

3.Complex (selection-join (SJ)) query : The proposed system supports more general queries containing both selection and join. These queries are used to help users to impose more complex crowdsourcing requirements.

A.System Architecture

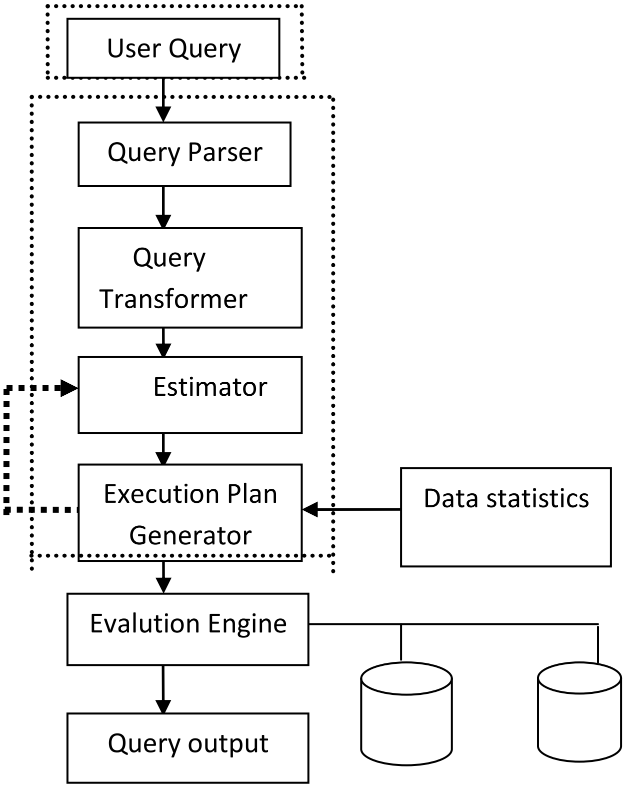

Fig 1: Architecture of Query Optimization for Declarative Crowdsourcing System

The proposed architecture has two model:

1.Data Model

It employs relational data model. The data is specified as a schema that consists of a set of relations R = { R_1,R_2, . .R_n}.These relations are designated by schema designers and can be queried by crowdsourcing users. Each relation R_i a set of attributes {A_{i1} , A_{i2} , . . ., A_{im} } describing properties of its tuples. Different from traditional databases, some properties of tuples are unknown before executing crowdsourcing.

2.Query Model

Query Q is an SQL query over the designated relations, and is semantics represents the results of evaluating Q over the relations using crowdsourcing.

IV.SCOPE OF THE WORK

The main goal of the proposed work is to find best query execution plan based on query optimization considering cost as well as latency constraint.

V.CONCLUSION

Different techniques for query optimization to support crowdsourcing have been discussed in detail. The best possible and best effective optimization algorithm is used for select, join, complex query. In the present time, simulated as well as real crowd experiments demonstrate the effectiveness of proposed system which produces best query plan which has a good balance between cost and latency.

REFERENCES

[1]. S. B. Davidson, S. Khanna, T. Milo, and S. Roy, “Using the crowd for top-k and group-by queries,” in Proc. 16th Int. Conf. Database Theory, 2013, pp. 225–236.

[2]. J. Fan, M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang, “A hybrid machine-crowdsourcing system for matching web tables,” in Proc. IEEE 30th Int. Conf. Data Eng., 2014, pp. 976–987.

[3] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, “CrowdDB: Answering queries with crowdsourcing,” in Proc.ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 61–72.

[4]. C.-J. Ho, S. Jabbari, and J. W. Vaughan, “Adaptive task assignment for crowdsourced classification,” in Proc. 30th Int. Conf. Mach. Language, 2013, vol. 1, pp. 534–542.

[5]. H. Park and J. Widom, “Query optimization over crowdsourced data,” Proc. VLDB Endowment, vol. 6, no. 10, pp. 781–792, 2013.

[6]. A. D. Sharma, A. Parameswaran, H. Garcia-Molina, and A. Halevy, “Crowd-powered find algorithms,” in Proc. IEEE 30th Int. Conf. Dta Eng., 2014, pp. 964–975.

[7].http://docs.oracle.com/database/121/TGSQL/tgsql_optcncpt.htm#TGSQL193