

# Database Query Optimization in Crowdsourcing System

Pariyarth Jesnaraj<sup>1</sup> Dr. K. V. Metre<sup>2</sup>

<sup>1,2</sup>Department of Computer Engineering

<sup>1,2</sup>Savitribai Phule University, Nasik, India

**Abstract**— Data is placed at different web or data servers in a crowdsourcing system. It is difficult to answer queries by machine. Processing such queries requires human input for providing information that is missing from the database, for performing computationally difficult functions, and for matching, ranking, or aggregating. In crowdsourcing system optimization of query is the major challenge. In proposed system, the user submits SQL query which is optimized based on cost constraints. An execution plan is generated and output is produced to the user. This system can be used for answering queries posed over stored relational data together with data obtained on demand from the crowd. This system considers a cost-based query optimization which also considers the latency that generates best possible and suitable query execution plan. The proposed system uses efficient algorithms for optimizing three types of query evaluation i.e. select, join, and complex-join query.

**Key words:** Crowdsourcing System, SQL query

## 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 through online community rather than from employees or suppliers. Crowdsourcing is more efficient tool where some task cannot be performed solely with computers. Query optimization is a function of many relational database management systems. The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans. An SQL-like declarative interface is designed to encapsulate the complexities of dealing with the crowd and provide the crowdsourcing system an interface that is familiar to most database users. Many crowdsourcing systems, such as CrowdDB, Qurk and Dec, works on SQL like query language which provides declarative interface to the crowd. For a given query, the system must first compile the query, generate an execution plan, post human intelligence tasks (HITS) to the crowd according to the plan, collect the answers, handle errors and resolve the inconsistencies in the answers. Crowdsourcing has created a variety of opportunities for many challenging problems by leveraging human intelligence. For example, applications such as image tagging, natural language processing, and semantic-based information retrieval can exploit crowd-based human computation to supplement existing computational algorithms. Human workers in Crowdsourcing system solve problems based on their knowledge, experience and perception. It is therefore not clear which problems can be better solved by crowdsourcing than solving solely using traditional machine-based methods. Therefore, a cost sensitive quantitative analysis method is needed. The query optimization uses three types of queries:

**Selection Queries:** The selection query is used to select data from a database. The result is stored in a result table, called the result set. A selection query applies one or more human-recognized selection conditions over the tuples in a single relation.

**Join Queries:** An SQL JOIN query is used to combine rows from two or more tables, based on a common field between them.

**Complex Selection-Join Queries:** This query consists of both selections and joins. These queries can help user's express more complex crowdsourcing requirements.

## II. RELATED WORK

C. Raykar, L. H. Zhao [1] Discussed probabilistic approach. This approach is used for supervised learning. This used to evaluate different experts and also gives an estimate of the actual hidden labels. Output indicates that the proposed method is superior to the commonly used majority voting baseline. Two key assumptions: (1) performance of each annotator does not depend on the feature vector for a given in-instance and (2) conditional on the truth the experts are independent, that is, they make their errors independently.

Marcus, E. Wu [2], in this paper, authors compare items for sorting and joining data, two of the most common operations in DBMSs. MTurk platform is used and Qurk runs on top of crowdsourcing.

Franklin, Tim Kraska [3] proposed difficult functions, such as matching, ranking, or aggregating results based on fuzzy criteria are computationally performed by CrowdDB system. 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 do 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.

Marcus, Miller [4], proposed a several techniques for using workers on a crowdsourcing platform like Amazon's Mechanical Turk to estimate the fraction of items in a dataset that satisfy some property or predicate without explicitly iterating through every item in the dataset. This is important in crowdsourced query optimization to support predicate ordering and in query evaluation, when performing a GROUP BY operation with a COUNT or AVG aggregate. They compare sampling item labels, a traditional approach, to showing workers a collection of items and asking them to estimate how many satisfy some predicate.

H. Park, J. Widom [5], explained Deco’s cost-based query optimizer, building on Deco’s data model, query language, and query execution engine is proposed. Objective of Deco’s is to find the best query plan to answer a query. It also describes Deco’s cost-based query optimizer. The Primary goal Deco’s is to find the best query plan to answer a query.

Davidson, Milo, Roy [6], explained that to evaluating top-k and group-by queries using the crowd to answer either type or value questions. Given two data elements, the answer to a type question is “yes” if the elements have the same type and therefore belong to the same group or cluster; the answer to a value question orders the two data elements. The assumption here is that there is an underlying ground truth, but that the answers returned by the crowd may sometimes be erroneous. They formalize the problems of top-k and group-by in the crowd-sourced setting, and give efficient algorithms that are guaranteed to achieve good results with high probability.

Chien-Ju Ho, Jabbari and Vaughan [7], state that a Crowdsourcing market is a tool that collects 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 assignment 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.

Ju Fan, Meiyu Lu, Beng Chin [8], proposed a two-pronged approach for web table matching that effectively addresses the two difficulties. First, they propose a concept-based approach that maps each column of a web table to the best concept, in a well-developed knowledge base, that represents it. This approach overcomes the problem that sometimes values of two web table columns may be disjoint, even though the columns are related, due to incompleteness in the column values. Second, they develop a hybrid machine crowdsourcing framework that leverages human intelligence to discern the concepts for “difficult” columns. The overall framework assigns the most “beneficial” column to concept matching tasks to the crowd under a given budget and utilizes the crowdsourcing result to help our algorithm infer the best matches for the rest of the columns.

A. D. Sharma, H. Garcia-Molina [9], 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 costs 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. SYSTEM ARCHITECTURE

User will first fill the query form for the required attributes and conditions. The query generator will automatically generate the query and this SQL query is issued by a crowdsourcing environment for execution. The executor will first call query optimizer. This optimizer parses the query and produces a best cost and time efficient query plan. The query plan is then executed by crowdsourcing executor to generate human intelligence tasks. Based on the HIT answers collected from the crowd, crowdsourcing executor executes the query and returns the generated results to the user.

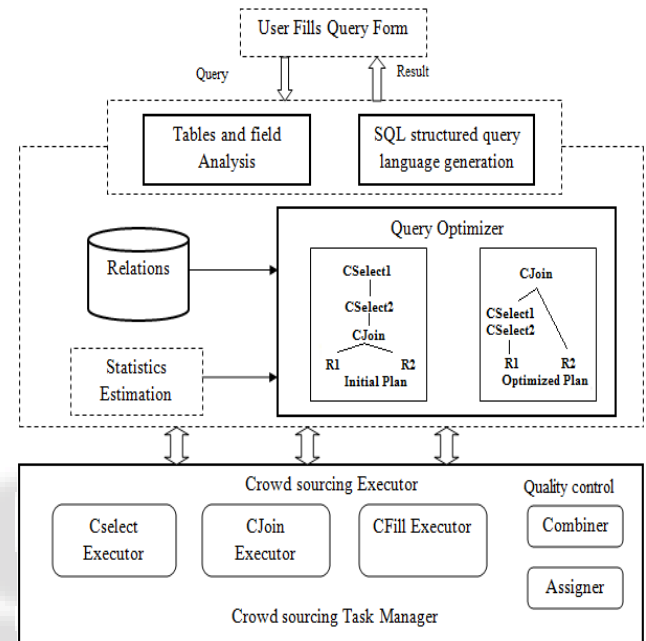


Fig. 1: System Architecture

#### Supporting cost-based based query optimization

In Cost Based Optimization cheapest execution plan is created for each SQL statement. This plan is the use the least amount of resources (CPU, Memory, I/O, etc.) to get the desired output. This can be a difficult task for DB engine as complex queries contains thousands of possible execution paths, and selecting the best one can be quite expensive.

#### Optimizing different crowdsourcing operator

In query plan, three types of crowd-powered operators, i.e., CSELECT, CJOIN and CFILL, are applied for evaluating query. FILL operator requests the group to fill in missing qualities in databases; SELECT operator abstracts the human operation of selecting objects satisfying certain conditions and JOIN operator is used for the group to match things as indicated by some criteria.

### IV. CONCLUSION

The proposed system can be used for answering queries posed over stored relational data together with data obtained on demand from the crowd. SQL query is being submitted by user which is being optimized on cost constraint depending on which execution plan is created, output is produced. A cost based query optimization that considers the cost-latency tradeoff and supports multiple crowdsourcing operators.

#### ACKNOWLEDGMENT

We are glad to express our sentiments of gratitude to all who rendered their valuable guidance to us. We would like to express our appreciation and thankful to Dr. M. U. Kharat, Head of Department, Computer Engineering., MET College of Engineering and Research Center, Nasik. We also thank the anonymous reviewers for their comments.

#### REFERENCES

- [1] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, 2010.
- [2] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller, "Human powered sorts and joins," *Proc. VLDB Endowment*, vol. 5, no. 1, pp. 13–24, 2011.
- [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] A. Marcus, D. R. Karger, S. Madden, R. Miller, and S. Oh, "Counting with the crowd," in *Proc. VLDB Endowment*, vol. 6, no. 2, pp. 109120, 2012.
- [5] H. Park and J. Widom, "Query optimization over crowdsourced data," *Proc. VLDB Endowment*, vol. 6, no. 10, pp. 781–792, 2013.
- [6] 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. 225236.
- [7] 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.
- [8] 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. 976987.
- [9] 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.