A New Approach To Generate Optimized Automatic Query Using Crowdsourcing System

Adarsh H. More[1], Abhishek H. Naik[2], Pratik S. Maid[3], Sushant S. Lomte[4] , Kishor N. Shedge[5]

[1,2,3,4] *Students, Dept. of Computer Engineering,* [5]*Assistant Professor & H.O.D,Dept. of Computer Engineering, PRES's SVIT college of Engineering,Maharashtra,India*

Email ID

[1] idealaadarsh1@gmail.com
,[2]abhihnaik.an1709@gmail.com
,[3] pratikm607@gmail.com,
[4]sushlomte143@gmail.com,
[5]kishor.shedge2007@gmail.com

Abstract — Data is placed at different web or data servers in a crowdsourcing system. It is difficult to answer queries by machine. Main goal of declarative crowdsourcing system is to keep out of sight the complexities of the system. Optimization of the query is the biggest problem now days for crowdsourcing system. Declarative crowdsourcing is designed to hide the complexities and remove user the burden. The user has to submit an SQL query and the system takes the responsibility for compiling the query, generating the execution plan and evaluating in the crowdsourcing market. There are many alternative execution plans for given query and depending upon the cost constraints best and worst plans is considered. It supports three types of query evaluation "select, join & complex join" respectively.

Keywords—***SQL Query, Human intelligence task(HIT), Query optimization, Crowdsourcing.***

I. INTRODUCTION:

Most of the times user have basic knowledge about queries, however they do not know how to implement optimized queries. Generally there are many queries which generates the same output but user is unaware about the performance and cost minimization knowledge so they fail to fire exact queries. Ex: CrowdOp, Qurk, Deco.

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. A declarative system must first compile the query, generate an execution plan.

In Crowdsourcing total group of people create, discuss, and refine meaningful thoughts, jobs via the web. Crowdsourcing is used in translation, hand writing recognition and audio transcription.

Optimization function in query are applied in relational database systems. Query optimization is the process of Selecting the most relevant query plan. Declarative queries allow to write data manipulation code to the programmers. Increased abstraction level above imperative code, improves program readability and helps for automatic query parallelization and optimization..

In CrowdOp several important data management problems such as Quality control, Cost control, Latency control are identified and resolved. In quality control, relatively low-quality results or even noise may yield in crowdsourcing. For example, wrong answers may be intentionally given by malicious worker. Workers may have different levels of expertise, and an untrained worker may be incapable of accomplishing certain tasks. To achieve high quality, we need to tolerate crowd errors and infer high-quality results from noisy answers. In cost control, the crowd is not free, and if there are large numbers of tasks, crowdsourcing can be expensive.

II. LITERATURE SURVEY:

Crowdop is a cost-based optimization approach for declarative crowdsourcing systems. In this an efficient algorithm within the CrowdOp for optimizing 3 types of queries such as select query, join query and complex selection-join queries is used. It also considers both cost and latency in query optimization objectives and generates query plans that provide a good balance between the cost and latency.

A probabilistic approach is discussed which is used for supervised learning. In this gives an estimate of the actual hidden labels and also evaluate different experts. Output indicates that the proposed

method is superior to the commonly used majority voting baseline. Two key assumptions: (1) feature vector is not depended on performance of each annotator for a given instance and (2) conditional on the truth the experts are independent, that is, they make their errors independently.

Here items are compared for joining and sorting data which is the most common operations in DBMS. Qurk used MTurk platform to runs on top of Crowdsourcing.

Different difficult functions such as ranking, matching or aggregating results based on fuzzy criteria are computationally performed by CrowdDB system. CrowdDB takes input from human with the help of crowdsourcing 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 crowdsourced data. Cost as well as performance depends on a many new factors including training fatigue, worker affinity, location and motivation.

III.PROPOSED SYSTEM:

The proposed system uses both framework algorithm and selection for query optimization. There are systems that work on the query execution plans though datasets have some problematic values. The query optimization is used for following three types of queries:

1. Selection query : The selection query is used for the selection of conditions and to fetch data from datab/ase. SELECT is the most commonly used data manipulation language (DML) command. SQL SELECT statement is used to retrieve data from a table in the database.

2. Join query : The SQL join clause is used to combine records/tuples from two or more tables/relations in a database. A JOIN is a nothing but merging fields from two tables by common value which is present to each table.

3. Complex query : These support more queries containing both selection and join. These queries are used to help users to impose more complex crowdsourcing requirements.

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

IV.SYSTEM ARCHITECTURE:

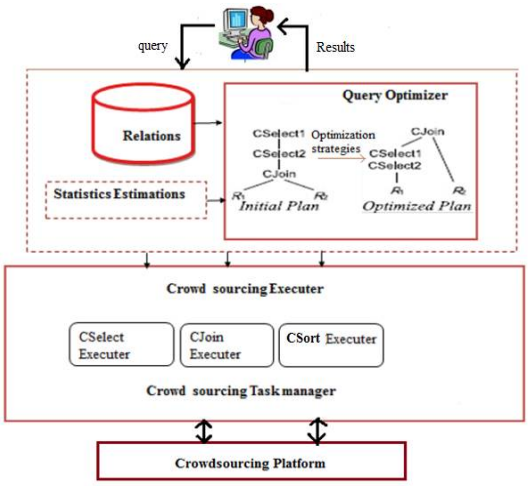

Fig 1: Architecture diagram

Fig. 1 illustrate the architecture of query processing in Crowdop. User first fill the query form for the required attributes and conditions. The query generator will automatically generate the SQL query and an initial plan of that query is generated. The initial plan of the SQL query is issued by a crowdsourcing environment for execution.

The executer will first call query optimizer. The query optimizer takes the logical query plan as input and produces an optimized query plan. This optimizer parses the query and produces a best cost and time efficient query plan. The parser is like a traditional relational database system, takes the query from the application as input. The initial query plan is then executed by crowdsourcing executor to generate human intelligence tasks(HIT).

This system overcome challenges such as: 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.

V. ALGORITHM AND MATHEMATICAL MODULE:

Algorithm:

Algorithm for Optimization framework(Q,C)

Input: Query Q, Cost C
Output: Query Q, Optimized plan
1: Initialize database and tables, load tables
4: Execute Query SELECT
5: Calculate Latency Max (Lmax)
6: Compute Query cost Lmax Lmin
7: Do Step 3 to 6 for JOIN and COMPLEX
8: Compare Latency

Mathematical Module:

The complete system S can be represented in terms of input, output and functions.

S = {I,O,F,U}
where ,
I :Input: {Q}
where ,
Q= No. of queries
O: Output: {Q, Qp, Qr, C}
where ,
Q= Query
Qp= Generated Query Plan
Qr= Query Result
C = Cost (Time Performance, Disk Space)
F: Functions: {Cq, Jq, Sq, U}
Cq= Perform complex query optimization and analyze cost using input query.
Jq= Perform join query optimization and analyze cost using input query.
Sq= Perform selection query optimization and analyze cost using input query.
U= Defines end user who wants to find Query result with best query plan.

VI. CONCLUSION

This is how we propose a cost-based query optimization that considers the cost-latency tradeoff and supports multiple crowdsourcing operators.We develop efficient and effective optimization algorithms for select, join and complex queries. Our experiments on both

2: Initialize C = nil
3: Calculate Latency Min (Lmin)

simulated and real crowd demonstrate the effectiveness of our query optimizer and validate our cost model.

REFERENCES:

[1] S. B. Davidson, S. Khanna, T. Milo, and S. Roy. Using the crowd for top-k and group-by queries. In ICDT, pages 225–236, 2013.

[2] J. Fan, M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang. A hybrid machinecrowdsourcing system for matching web tables. In ICDE Conference, 2014.

[3] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In SIGMOD Conference, pages 61–72, 2011.

[4] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen. An online cost sensitive decision-making method in crowdsourcing systems. In SIGMOD Conference, pages 217–228, 2013.

[5] Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. PVLDB, 7(14):1965–1976, 2014.

[6] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?
dynamic max discovery with the crowd. In SIGMOD Conference, pages 385–396, 2012.

[7] J. M. Hellerstein and M. Stonebraker. Predicate migration: Optimizing queries with expensive predicates. In SIGMOD Conference, pages 267–276, 1993.

[8] C.-J. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In ICML (1), pages 534–542, 2013.

[9] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. Inf. Process. Lett., 5(1):15–17, 1976.

[10] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: A crowdsourcing data analytics system. PVLDB, 5(10):1040–1051, 2012.

[11] A. Marcus, D. R. Karger, S. Madden, R. Miller, and S. Oh. Counting
with the crowd. PVLDB, 6(2):109–120, 2012.

[12] Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Humanpowered sorts and joins. PVLDB, 5(1):13–24, 2011.