Journal of Basic and Applied Engineering Research
p-ISSN: 2350-0077; e-ISSN: 2350-0255; Volume 3, Issue 12; October-December, 2016, pp. 1038-1041

http://www.krishisanskriti.org/Publication.html

Extending the CrowdOp (Query Optimization) For Advanced SQL Operators

T. Latha[1] and A. Suresh Babu[2]

[1,2]Computer Science Engineering, Jawaharlal Nehru Technological University, Anantapur.
E-mail: [1]thalapanenilathachowdary@gmail.com, [2]sureshalladi.cse@jntua.ac.in

Abstract—*Query optimization has worked almost exclusively on reducing query execution time, while important qualities such as consistency as well as predictability have largely been ignored, even though most database users consider these individualities to be at least as significant as raw performance. The existing approaches are focused on cost-based query optimization. In traditional approach we optimized three kinds of queries those are 1) selection queries 2) join queries 3) complex selection-join queries. In this paper we proposed Extended Crowd Optimizer to support much advanced SQL operators like Sorting as well as Aggregation and also described how to incorporate correlation among SQL select as well as Join conditions into the optimizer for difficult queries.*

1. INTRODUCTION

Relational database systems possess the fashionable adoption and now its not simplest within the business environment for which they had been initially expected, but additionally for a lot of another category of structured data such as private, social, and also in scientific knowledge. Nonetheless, as information construction and use end up increasingly democratized via web, mobile and other applied sciences, the boundaries of the science are fitting extra obvious. RDBMSs makes a few key belief's about the exactness, integrity and accuracy of the information they retailer. When these acceptance declines to preserve, relational techniques will return improper or insufficient solutions to consumer queries, in the event that they return any solution in any respect. Crowd sourcing is among the setting up web 2.0 established marvels and has pulled in amazing consideration from each gurus and researchers in the course of the years. It may encourage the supply and organize effort of entities and associations. We believe that knowledge, techniques analysis is in certainly a kind function to make massive responsibilities to this increasing search zone and keep it in mind as an extra examination outskirt. In this manner, couple of experiences have defined what were entire and what have to be finished.

In this paper we are trying to present a selective checking the substrate of estimating with the intention to crowd search studies, including pretending establishments, study systems, and checking foci and determine a number of principal analyses for IS researchers from three features of view—the member, organization and framework. This analysis add to the IS writing and gives bits of advantage to experts, administrators, planners and association creators to raise distinct disorders in crowd sourcing plans. A querying increases the efficiency of the process and it depends upon the procedure that have efficiency to optimize and furnish a “near optimal” execution method for individual query, So that declarative crowd sourcing query may also be calculated in number of methods and the option of execution method has primary effect on total efficiency which entails the queries actually asked. The complications of the queries and the financial cost incurred and therefore it’s major to design an effective crowd sourcing query optimizer that's equipped to don't forget all just right query plans and pick the “pleasant” plan established on a price model and optimization pursuits. The objective of present method is to provide a declarative query interface by accessing the efficiency of query optimization plans for the crowdsourcing systems become a member of complicated queries in a crowdsourcing atmosphere making use of parallel procedure. Procedure should give in a position to separate the query and execute that query on parallel process.

2. RELATED WORK

A lot of analysis on optimization of query for crowdsourcing had done over the past few years. Some foremost strategies are mentioned here. Various strategies had been recommended and analysis has been achieved within the field of optimized query. Here many evolved approaches have been presented for optimized query and crowdsourcing. Chien-Ju et.al [8] states that Crowdsourcing fields have received beauty as an instrument for moderately gathering data from distinct employees and classify duties in which programmers offers labels for instances are among the many most usual duties posted because of man made errors and the fee of unsolicited mail, the labels gathered are as a rule noisy. They evaluate the hindrance of challenge undertaking and label assumptions for distinct classification duties. By applying on-line primal-dual methods that derive a provably close-premier adaptive project algorithm. This exhibits flexible assigning program to work

that can lead to extra correct forecast at minimize price when the available employees are diverse.

Hellerstein et al [7] proposed a few methods for utilizing employees on crowdsourcing phase such as Amazon's Mechanical Tur. This is foremost in optimized query for crowdsourcing to help predicate obtaining and in estimating query, when operating a bunch by a way of performing with a count or AVG mixture. There are many increasing methods to take away spammers and attackers looking to learn decision estimating and using depend estimation procedure and search that for pix calculating can be more amazing than sampled classifying, lowering the bulk of burden indispensable to reach at a guess that is inside 1% of proper fraction by as much as plan of consequence with decreased work latency. Liu et al [10] explained that target on how you can use manpower to compare objects for joining, sorting and aggregatin information by the normal operations in DBMS described general interface for query and the person interface for the works and submit to MTurk. They advise a quantity of optimization, together with venture batching, exchanging pairwise correlation with number rankings, and pre-filtering tables earlier than becoming a member of them, which scale back the total rate of strolling sort, joins and aggregate on the group.

3. PROPOSED WORK:

Frame work: Query Optimization

Query optimization operating many relational database administration methods.

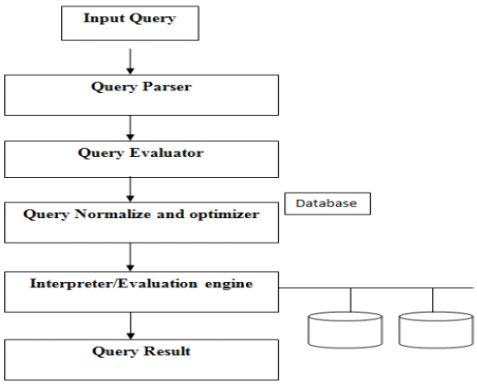


Fig. 1: Query Optimization Work Flow

The query optimizer makes an attempt to verify probably the most effective technique to execute a given query via considering that the viable query plans. In general, the query optimizer can't be accessed immediately by means of customers: If queries are passed to database server then they are parsed in the parser, after passing queries to optimizer for query optimization. A query generally means requesting information from database and Queries outcome are achieved by having access to consistent database data as well as controls it in the way that returns the requested information. On account that database formats are tricky, customarily, and above all for now not-very-easy queries that wanted information for asked query can also be accumulated from a database via getting access it in to unique methods through one-kind-of data structures and in exceptional orders. Each different manner requires distinctive processing time. Query optimization finds the great plan for query in terms of calculated monetary cost.

Query Optimizer

First the given query will be parsed and generates optimized plan. In query plan, preliminary query plan will probably be defining utilizing query as well as become a member of query. This member of query shall be split and Optimization methods are used for query joining and generate ultimate optimized plan. The optimization methods that optimize the query in sequential and parallel methods. First a query is passed through sequential combiner there itself we get the latency and cost for given query and then next it passed through parallel combiner in which it generates less latency and cost compared to sequential combiner.

Crowd Sourcing Executor:

The plan for query is then performed via Crowdsourcing Executor to provide human intelligence tasks nothing but HITs and post these tasks on crowdsourcing structures that founded on the Human intelligence tasks and solutions gathered from the group. Crowdsourcing Executor estimates that query and returns the gathered results to the consumer.

Quality Control:

It includes assigner and combiner. Assigner is to assign work for query in unique tables. Combiner works for join that effects all queries. By this we can get control on quality of query related to accuracy results and choose the best solution from the results.

Correlation between SQL Conditions

The most normally used SQL command is SELECT statement. SQL SELECT assertion is used to query or retrieve information from a table in the database. A query could retrieve by understanding targeted columns or from entire columns within the desk. To create a simple SQL SELECT for assertion, you have to specify the column(s) title as well as the desk name. The entire query is referred to as SQL Select statement. SQL Join is used to narrate knowledge in distinctive tables. A Join condition is a part of the SQL query that fetched rows from two or more tables. A SQL Join is used within the SQL WHERE Clause of select, replace, and delete statements. If a SQL Join is ignored and if it is invalid then Join operation will influence in a Cartesian product. The Cartesian product returns a number of rows equal to all rows in the tables being joined. After query passed through

sequential combiner as well as parallel combiner we can do join query optimization by using C-fill and C-join operators with additional attribute called quality. By this the result should be more accuracy with latency in milliseconds.

Complex Join Queries

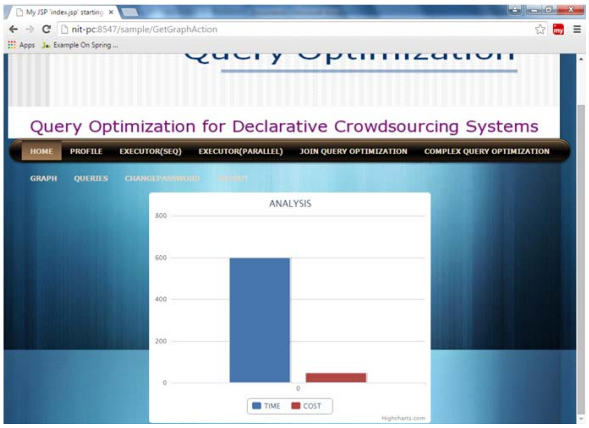
Another category of query optimization approach is complex query optimization. This may increasingly include each selection , join and c-fill operators. These queries can aid person's categorical trickier crowdsourcing requisites. For this the time constraint shouldn't be imposed, we optimize the plan for query similarly to natural databases. practicing some heuristic ideas is equivalent to pushing down selection and identifying the join order using C-fill and C-join operators to obtain latency in milliseconds and cost for that optimized complex query.

Sorting and Aggregation

In this sorting and aggregation techniques were used to obtain better result for the optimized queries. These operators can sort and aggregate the same optimized queries which will pass another time in the query optimizer and get reduced cost and latency while compared to the result before same optimized queries which can view in the table form. By this, we can generate better result for the same queries which will pass through sorting and aggregation process and obtain the reduced cost and latency query by overriding the before result.

4. RESULTS

In our experiment we developed efficient and effective optimization algorithm for select, join ,sorting, aggregation and complex selection-join queries.



From the above graph, we described optimized query execution time along with the cost. Finally, our experiment on both simulated and real crowd demonstrate the effectiveness of our query optimizer as well as validate our cost model along with latency model.

Result after sorting and aggregation

Query	Sequential Time

Query	Parallel Cost
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='BMW' and a.Model='XC60' and a.Style='Sedan'and r.make='BMW' and r.model='XC60' and i.color='White'	69
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Toyota' and a.Model='Avalon' and a.Style='SUV'and r.make='Toyata' and r.model='Avalon' and i.color='red'	57
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Velvo' and a.Model='Avalon' and a.Style='SUV'and r.make='Velvo' and r.model='Avalon' and i.color='White'	90
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Toyota' and a.Model='Avalon' and a.Style='SUV'and r.make='Toyata' and r.model='Avalon' and i.color='White'	123
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='BMW' and a.Model='X5' and a.Style='classic'and r.make='BMW' and r.model='S80' and i.color='red'	105

Query	Parallel Time
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='BMW' and a.Model='XC60' and a.Style='Sedan'and r.make='BMW' and r.model='XC60' and i.color='White'	19
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Toyota' and a.Model='Avalon' and a.Style='SUV'and r.make='Toyata' and r.model='Avalon' and i.color='red'	15
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Velvo' and a.Model='Avalon' and a.Style='SUV'and r.make='Velvo' and r.model='Avalon' and i.color='White'	26
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='Toyota' and a.Model='Avalon' and a.Style='SUV'and r.make='Toyata' and r.model='Avalon' and i.color='White'	37
Select a.id,a.Make,r.Review,i.color FROM AutoMobile a,Review r,Image i where a.Make=r.Make and a.Make=i.Make and r.Make=i.Make and a.Make='BMW' and a.Model='X5' and a.Style='classic'and r.make='BMW' and r.model='S80' and i.color='red'	31

5. CONCLUSIONS

In this paper we concluded efficient query optimization algorithm for sorting and aggregation techniques and described correlation between SQL complex Queries.We performed the Join query optimization and complex query optimization also with that we have done sorting and aggregation for the resulted optimized queries and in our experiment we can view the sequential queries time along with cost and also we can view parallel query time along with cost. Based on our experiment we proved that we can optimize the execution time and cost of the complex queries and overwrite the result by sorting and aggregation techniques for the same optimized queries.

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] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen, “An online cost sensitive decision-making method in crowdsourcing systems,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, pp. 217–228, 2013.

[5] Y. Gao and A. G. Parameswaran, “Finish them!: Pricing algorithms for human computation,” Proc. VLDB Endowment, vol. 7, no. 14, pp. 1965–1976, 2014.

[6] S. Guo, A. G. Parameswaran, and H. Garcia-Molina, “So who won?: Dynamic max discovery with the crowd,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2012, pp. 385–396.

[7] J. M. Hellerstein and M. Stonebraker, “Predicate migration: Optimizing queries with expensive predicates,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1993, pp. 267–276.

[8] 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.

[9] L. Hyafil and R. L. Rivest, “Constructing optimal binary decision trees is np-complete,” Inf. Process. Lett., vol. 5, no. 1, pp. 15–17, 1976.

[10] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang, “CDAS: A crowdsourcing data analytics system,” Proc. VLDB Endowment, vol. 5, no. 10, pp. 1040–1051, 2012.