IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Special Issue-8, Apr 2019 E-ISSN: 2347-2693

A Comparative Study of Query Processing and Optimization Techniques

A. Regita Thangam[1*], S.John Peter[2]

[1]Dept. of Computer Science, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-12, Tamilnadu, India.
[2]Dept. of Computer Science, St.Xavier's College,Palayamkottai, India

Corresponding Author: regitaraja@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7si8.15 | Available online at: www.ijcseonline.org

Abstract— The importance for optimization arises from the flexibleness provided by modern user interfaces to databases. With the widespread applications of Database Management Systems, users have to deal with an enormous amount of data. Therefore, it is necessary to store this data in such a way that it is retrieved from the information within the quickest possible manner to satisfy the request from a user. Databases are most helpful in representing information in an organized manner. It provides the user with the flexibility to acquire correct, reliable and timely data for effective decision making process. Thus, the importance of database systems is increasing day by day. At the same time, the complexity in queries is also increasing day by day which makes the problem of determining the best query optimization technique. Query optimization in databases continues to be an important issue in various fields for a long period of time. To reduce the execution cost, we need to reformulate the complex query with computationally equivalent and more efficient one. Now in this paper, we analyse and compare the performance of various query processing techniques like Aggregate function based approach, Reduced function based approach, Bitmap Index based approach, Filtered Bitmap Index based approach to process queries with set predicate.

Keywords— query optimization, Aggregate function based approach, Reduced function based approach, Bitmap Index based approach, Filtered Bitmap Index based approach, set predicates.

I. Introduction

Query processing is mainly concerned with the execution of query. It also refers to the activities involved in extracting data from a database. In query process, one of the most critical and important step is query optimization.

Query optimization technique is used to access the database in an efficient manner. It refers to the process of producing an optimal execution plan for a given query. It is an art of obtaining the desired information in a reliable and timely manner. It is also a process of transforming a query into an equivalent form which can be evaluated more efficiently. The objective of query optimization is to provide minimum execution time and maximum throughput. Query optimization plays a significant role in tuning overall performance of the database systems.

Currently, Query optimization has become a popular topic in database research. Initially, databases were primarily used for transaction processing of the data. But now, databases are also used to facilitate reporting and analysis on the consolidated, historic data. The great success of information systems is due to the development of sophisticated query optimization technology, where users pose queries in a declarative way using SQL and the optimizer of the database system finds an efficient way to execute these queries.

In recent days, the demand for querying the data in larger databases with the semantics of set-level comparison is very high. Suppose we want to find the clients who watched the match on the set of particular days on the given world cup match database. Dates of each candidate that is set of values are compared against the dates in the query condition. Such sets are dynamically formed. If the set level comparisons performed using currently available SQL syntax, resulting query may be more and more complex. Such complex query becomes a difficult for the user to formulate, which results in too much costly evaluation.

The SQL query to find the students with extra-curricular activities "sports" and "dancing", as follows:

```
SELECT sname FROM Studprofile
GROUP BY sname HAVING SET(extraactivity) CONTAIN
{'sports', 'dancing'}
```

Given the above query, after grouping, a dynamic set of values on the attribute extraactivity is formed for each unique sname, and the groups whose corresponding SET (extraactivity) contain both “singing” and “dancing” are returned as query answers.

The SQL query to find the books with the authors Kala and Maria only. For this query, the EQUAL operator can be used as below:
SELECT bookname FROM bookdetails
GROUP BY bookname HAVING SET(author) EQUAL {‘Kala’, ‘Maria’}

For the decision making example, suppose we have a table Ratings (dept, avg_rating, month, year). The following exemplary query finds the departments whose monthly average ratings in 2018 have always been poor (assuming the rating is from 1 to 5):
SELECT dept FROM Ratings WHERE year = 2018 GROUP BY dept HAVING SET(avg_rating) CONTAINED BY {1, 2}

In this query, CONTAINED BY is used to capture the set-level condition. The CONTAINED BY operator is a logical operator that allows you to compare a value against a set of values. This operator returns true if the value is within the set of values. Otherwise, it returns false or unknown.

Without the explicit notion of set predicates, the query semantics may be captured by using sub-queries connected by SQL set operations (UNION, INTERSECT, EXCEPT), in coordination with join and GROUP BY. Such queries may be quite complex for users to formulate which results in too much costly evaluation. On the contrary, the set predicate construct according to embodiments of the present invention explicitly enables set-level comparisons. The concise syntax makes query formulation simple and also facilitates the efficient native support of such queries in a query engine [22].

The query syntax also allows to compare the sets defined on multiple attributes. A query with multiple set predicates can be supported for Boolean Operators such as AND, OR and NOT and the aggregate functions that are defined by the database server, such as AVG, SUM and COUNT.

This paper is organized as follows. Section I contains the introduction about query processing, the related work laying the stage for various query processing approaches is discussed in section II. The Comparative study is specified in section III. Eventually, section IV concludes this paper.

I. RELATED WORK

Query optimization tries to reduce the response time of a given query. It also refers to the process by which the best execution strategy for a given query is found from a set of alternatives. There is a high demand of querying data with set-level comparisons. Users can dynamically form set level comparisons without any limitation caused by database schema for set predicates. In many applications, there is a need to integrate data and operations that are external to the database.

Tejy Johnson et al. [19] proposed an efficient multi-level relational mapping algorithm for dependency rule generation to improve the query optimization. The method starts with the preprocessing of the input query which identifies the relational objects and splits the input query into number of small queries. Based on the dependency measure and also the rule being generated the strategy sorts the query part and based on the sorted order the query parts are executed to provide efficient results and reduce the time complexity.

Nowadays, a lot of database management schemes offer many features to handle the collection of values like nested table found in Oracle and also SET data type as in MYSQL. Data storage and depiction is not required for Set predicates as standard database management system includes them. In many applications, in accordance with requirement of query collections, matching sets are normally created dynamically. It is possible for users to create set level contrasts dynamically without having any limitation due to schema of database fir set predicates [20].

In [10], collection of variables and related set concepts were proposed as extension of SQL for allowing correlation of the multiple aggregate activities upon the same assembling condition. This study focuses totally on data processing with the utilization of condensed bitmap index in addition as prediction of the sets. Query processing on set-valued attributes and set containment joins have been extensively studied in [4].

Panos Kalnis et al. [6] proposed an efficient way for concurrent execution of multiple queries to increase the throughput. The importance of Multi-Query Optimization in the context of relational database query processing is explained by J.Chen et al. [1].

An efficient algorithm Filtered bitmap index based approach for processing queries with set predicates was proposed in [21]. This algorithm has the benefits of saving disk access and the computation time was reduced by reducing the number of iterations. In this algorithm the groups and the corresponding sets are formed according to the query needs which results in speeds up the query processing.

Swathi Kurunji et al. [13] presented an algorithm for processing ad-hoc multi-join query in cloud environment and reduce communication overhead. An extension of traditional query rewrite techniques was proposed by Albrecht et al. [3]. Derivability of multidimensional aggregates is the condition that has to be fulfilled to compute the result of an aggregate query based on the values of one or more aggregate views. They presented the conditions for derivability in a large number of relevant cases which go beyond previous approaches.

Rank join operators combine two or more relations and produce the k-combinations with the highest score. The rank join problem [7] has been dealt in the literature by extending rank aggregation algorithms [5] to the case of join in the setting of relational databases. The Cost-Aware with Random and Sorted access (CARS) pulling strategy was proposed by Davide Martinenghi et al. [14] for retrieving the k-combinations with the highest aggregate score that can be formed by joining the results of heterogeneous search engines. They optimized such a strategy with respect to an additive cost model that considers both sorted access and random access.

The distributed query optimization is one of the hardest problems in the database area [8]. For a given query, there is the possibility of more than one algebraic query. Some of these algebraic queries are better than others. The quality of the query is defined in terms of expected performance.

Effective decision making is significant in a very global competitive environment wherever business intelligence systems are getting an important part of each organization. The core of such systems is a data warehouse, which stores historical and consolidated data from the transactional databases, supporting complicated ad hoc queries that reveal interesting information [6].

Modern database systems use a query optimizer to identify the most efficient plan to execute declarative SQL queries. The role of query optimizers is critical for the decision-support queries featured in data warehousing and data mining applications. Pawan Meena et al. [11] proposed an abstraction of the architecture of a query optimizer and the technical constraints of advanced issues in query optimization.

A global index based optimization strategy for range query and analysis was proposed by Hui Zhao et al. [12] and they do some tests to evaluate the correctness and efficiency at the end. The strategy was first checking whether user requests can be optimized by using the global index knowledge.

Aggregate function based technique and Bitmap index based technique was proposed by Chengkai Li et al. [16] to process query with set predicates. Aggregate function based technique processes set predicates in the normal way as processing conventional aggregate function. Second technique is more efficient because it focuses on only those tuples which satisfies query condition and bitmaps of appropriate columns. Such index structure is applicable on many different types of attributes. This technique processes queries such as selections, joins, multi-attribute grouping etc.

Jayant Rajurkar et al. [17] developed a bitmap pruning strategy by using Word Aligned Hybrid (WAH) compression for processing queries which eliminates the necessity of scanning and processing the entire data set. This technique is used for optimizing queries with set predicates. The set predicates have several advantages than the set-valued attributes together with set containment joins which can support set-level comparisons.

II. COMPARATIVE STUDY

The syntax of SQL has been extended for supporting set predicates. Set predicate gets well with GROUP BY and also HAVING clauses, as it compares a set of tuples to a group of values. It can also be linked with logical operators such as AND, Or and NOT. Here we discuss various approaches for processing queries with set predicates containing the three kinds of set operators CONTAIN (⊇), CONTAINED BY (⊆) and EQUAL (=).

A. Aggregate Function based approach

Using the semantics of set predicates, a set predicate-aware query plan could potentially be much more efficient by just scanning a table and processing its tuples sequentially. The key to such an approach is to perform grouping and set-level comparison using one-pass iteration of tuples. The idea resembles however regular aggregate functions are often processed along with grouping. Hence a method that handles set predicates as aggregate functions was designed.

B. Reduced Function based approach

It is used to evaluate the queries with set predicates containing the three kinds of set operators {⊇, ⊆, =}. In the above Function based approach, the process of matching technique is applied for all the records and it is a very time consuming process. But in this approach, the matching technique is applied group wise and exit from the loop if some condition arises. The set predicate value for each tuple is checked with the condition values specified in the query. If any one of the condition field doesn't match, then further checking will be skipped for the current group for the operators EQUAL and CONTAIN and the time complexity will be reduced. In detail, a tuple is skipped if the corresponding group is already disqualified and the operator is ⊇ or =. Thus it will do the process very efficiently with reduced time complexity. This approach has the benefits of saving disk access and the computation time was reduced by reducing the number of iterations.

C. Bitmap Index based approach

The Bitmap index technique is used to process query with set predicates. The set level comparison is performed by one-pass iteration of tuples. The bitmap index-based approach uses bitmap indices on individual attributes. Based on single-attribute indices the simple data format and bitmap operations make it convenient to integrate various operations in a query, including dynamic grouping of tuples and set-level comparisons. This methodology brings several benefits by investing the distinguishing characteristics of bitmap index.

D. Filtered Bitmap Index based approach

In this approach, the groups and corresponding sets are dynamically formed according to query needs. It supports the set predicate operators CONTAIN, CONTAINED BY and EQUAL. This approach is also based on bitmap index based technique. There exists a bitmap for each unique attribute value. The vector length equals the number of tuples within the indexed relation. In this algorithm the groups and the corresponding sets are formed according to the query needs which results in speeds up the query processing. During query processing some filtered conditions are applied for EQUAL and CONTAIN operators to skip the unnecessary checking which helps us to reduce the iterations. Thus this approach has the benefits of saving disk access and the computation time was reduced by reducing the number of iterations.

III. Results and Discussion

The experiments are performed on the Intel I3 processor with 4GB RAM memory. This paper presents various approaches for processing query with set predicates. The efficiency of these approaches is proved by using benchmark dataset worldcup-98 which is collected from the website *http://ita.ee.lbl.gov/html/contrib/WorldCup.html*. The WorldCup98 data set contains 1,352,804,107 tuples, which correspond to all the access requests made to the 1998 World Cup website between April 30, 1998 and July 26, 1998. Each tuple has information such as the time of the request, the type of the requested file, the file size, the server that handled the request, the client identifier and so on. These algorithms are implemented in Matlab.

Query1: We designed a sample query to analyse the processing time of various approaches by using benchmark dataset worldcup-98.

To find the total traffics for clients who had visited in two consecutive days- July 18th, July 19th.

```
SELECT clientID, SUM(Bytes) FROM clients
GROUP BY clientId
HAVING SET(date) CONTAIN {0718,0719}
```

It identifies the clients who visited in both days July 18th, July 19th. The keyword CONTAIN represents a superset relationship, i.e., the set variable SET(date) is a superset of {0718,0719}

Results: The results of executing the above query using Aggregate function based approach, Reduced function based approach, Bitmap Index based approach, and Filtered Bitmap Index based approach on the Worldcup-98 data set is shown in TABLE I.

The Fig.1 shows comparison of query processing time among the above existing techniques. From Fig.1, it shows that Filtered Bitmap Index-Based approach is more efficient than other existing approaches.

TABLE I. EXECUTION TIME WITH DIFFERENT APPROACHES FOR QUERY1

Approach	Execution Time
Aggregate function based approach	165 secs
Reduced function based approach	152 secs
Bitmap Index based approach	143 secs
Filtered Bitmap Index based approach	95 secs

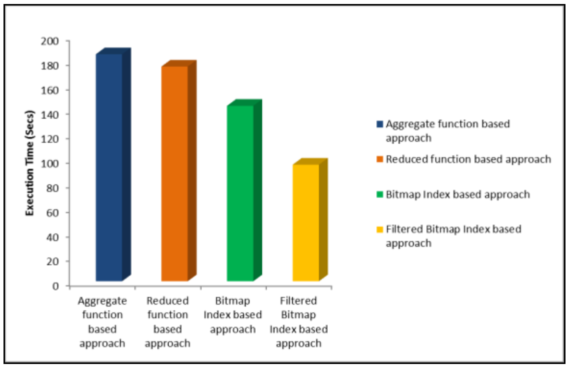

Fig. 1. Execution time analysis with different approaches for quey1.

IV. Conclusions

This paper presents an extensive analysis about Set Predicates for supporting group-level comparisons. The four evaluation methods are presented to process set predicates. It shows that Filtered Bitmap Index-Based approach is more efficient to process query with set predicates. This algorithm has the benefits of saving disk access and the computation time was reduced by reducing the number of iterations. The estimation governs the better evaluation of set predicates and producing efficient query plans.

References

[1] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang. NiagaraCQ, "A scalable continuous query system for internet databases", Published in Proc. SIGMOD, pages 379–390, 2000.

[2] Martin Arlitt and Tai Jin, "A Workload Characterization Study of the 1998 World Cup Web Site", IEEE Network, vol. 14, no. 3, pp. 30-37, May/June 2000.

[3] J. Albrecht, W. Hümmer, W. Lehner, L. Schlesinger, "Query Optimization By Using Derivability In a Data Warehouse Environment", Published in the Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP, DOLAP - 2000, pages 49-56.

[4] Y. Ioannidis, "The History of Histograms(abridged)", Published in theProceedings of the 29th VLDB Conference, 2003.

[5] R. Fagin, A. Lotem, and M. Naor, "Optimal Aggregation Algorithms for Middleware", Published in Computer and System Sciences, vol. 66, no. 4, pp. 614-656, 2003.

[6] Panos Kalnis, Dimitris Papadias, "Multi-query optimization for on-line analytical processing", Published in Information Systems, Volume-27,Issue 5, July 2003.

[7] I.F. Ilyas, W.G. Aref, and A.K. Elmagarmid, "Supporting Top-k Join Queries in Relational Databases", Published in VLDB J., vol. 13, no. 3, pp. 207-221, 2004.

[8] Alaa Aljanaby, Emad Abuelrub, Jordan and Mohammed Odeh, "A Survey of Distributed Query Optimization", published in The International Arab Journal of Information Technology, Vol. 2, No. 1, January 2005.

[9] C. Olston, B. Reed, U. Srivastava, R. Kumar and A. Tomkins, "Pig Latin: A Not-so-Foreign Language for Data Processing", Proc. ACM SIGMOD International Conference Management of Data, pp. 1099-1110, 2008.

[10] Chatziantoniou, D. and E. Tzortzakakis, "Asset Queries: A Declarative Alternative to Mapreduce", Published in ACM SIGMOD Record, 38(2): 35-41, June 2009.

[11] Pawan Meena, Arun Jhapate & Parmalik Kumar, "Framework for Query Optimization", published in the International Journal of Computer Science and Information Security, Vol. 9, No. 10, October 2011.

[12] Hui Zhao, Shuqiang Yang, Zhikun Chen, Songcang Jin, Hong Yin and Long Li, "MapReduce model-based optimization of range queries", Published in 2012, 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012).

[13] Swathi Kurunji, Tingjian Ge, Benyuan Liu, Cindy X. Chen, "Communication Cost Optimization for Cloud Data Warehouse Queries", Published in the Proceedings of the IEEE 4th International Conference on Cloud Computing Technology and Science 2012.

[14] Davide Martinenghi and Marco Tagliasacchi, " Cost-Aware Rank Join with Random and Sorted Access", Published in the IEEE Transactions On Knowledge And Data Engineering, VOL. 24, NO. 12, DECEMBER 2012.

[15] P. Arpitha, "Query Optimization In Data Warehouse", Published in the International Journal of Engineering Research & Technology, Volume 2, Issue 8, 2013.

[16] Chengkai Li, Bin He, Ning Yan, M. Safiullah "Set Predicates in SQL: Enabling Set-Level Comparisons for Dynamically Formed Groups", IEEE Transactions on Knowledge and Data Engineering , Vol. 26, No. 2, FEBRYARY 2014.

[17] J. Rajurkar, T. Khan, "A System for Query Processing and Optimization in SQL for Set Predicates using Compressed Bitmap Index", International Journal for Scientific Research & Development Vol. 3, Issue 02, 2015.

[18] A.Regita Thangam and S.John Peter, "An Extensive Survey on Various Query Optimization Techniques" Published in the International Journal of Computer Science and Mobile Computing, Volume-5, Issue- 8, August 2016.

[19] Tejy Johnson and S.K. Srivatsa, "Multi Level Relational Mapping Algorithm Based Dependency Rule Generation for Query Optimization", Published in the American-Eurasian Journal of Scientific Research, vol. 11, no. 2, pp. 72-78, 2016.

[20] Rhia Mariam George and A. Ronalad Doni, "Query Processing and Optimization Using Set Predicates", Published in the American-Eurasian Journal of Scientific Research, vol. 11, no. 5, pp. 390-397, 2016.

[21] A.Regita Thangam and S.John Peter, "Efficient Processing and Optimization of Queries with Set Predicates using Filtered Bitmap Index" Published in the International Journal of Computer Sciences and Engineering, Volume-5, Issue-11, Nov 2017.

[22] A.Regita Thangam and S.John Peter, "Efficient Processing of Queries with Set Predicates using Reduced Function based Approach", published in the Proceedings of the International Conference on Recent Trends in Multi-Disciplinary Research, pp.11, Dec 2018.

Authors Profile

Mrs. A.Regita Thangam is working as an Assistant Professor in St.Xavier's College, Palayamkottai, She earned his M.C.A. degree from M.S. University, Tirunelveli. She also earned his M.Phil from Alagappa University, Karaikudi. Now She is doing Ph.D. (Reg.No: 12203) in Computer Applications at Department of Computer Science and Research Centre, St.Xavier's College, Palayamkottai, Affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli, Tamil Nadu, India. She has published research papers in International and National journals.

Dr. S.John Peter earned his M.Sc. and M.Phil. from Bhradhidasan University, Trichirappli. The M.S University, Tirunelveli awarded his Ph.D. degree in Computer Science for his research in Data Mining. He is the Head of the department of computer science, and the Director of the computer science research center, St. Xavier's College (Autonomous), Palayamkottai, Tirunelveli. The Manonmaniam Sundaranar University, Tirunelveli has recognized him as a research guide. He has published research papers in International, National journals and conference proceedings. He has organized Conferences and Seminars at the National level.