A System for Query Processing and Optimization in SQL for Set Predicates using Compressed Bitmap Index

Jayant Rajurkar¹ T. Khan²

¹Department of Computer Science & Engineering ²Department of Information Technology

G. H. Raisoni College of Engineering, Nagpur (MS), India

Abstract— In data warehousing and OLAP application, within the database relation comparison between group of tuples with their attribute and values requires syntax and semantics that currently available Database Management System do not provide. Currently, complex SQL queries consist of scalar-level operations are often formed to obtain even very simple set-level semantics. Such queries are not only difficult to write but also challenging for a database engine to optimize, which results in costly evaluation. We studied aggregate function-based technique and Compressed bitmap index-based technique to processing set predicates. Bitmap index, which creates one bitmap vector for individual attribute value. It occupy less space than raw data and gives more opportunities for efficient query processing. We exploited the property of bitmap index and developed a very effective 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 (table) which results in speeds up the query processing significantly. Experiments verified our technique is much more efficient than existing algorithms in optimizing queries for set Predicates.

Key words: Data Warehousing, Bitmap index, Set predicates, OLAP, Querying processing and optimization, Word-Aligned Hybrid (WAH)

I. INTRODUCTION

Now a day’s, demand of querying the data in data warehouse and OLAP applications with the semantics of set-level comparison is very high. Suppose a company or institution seeking for candidates for the job with set of compulsory skills, company or institution may search their resume database. Skills of each candidate that is set of values are compared against the compulsory skills. Such sets are dynamically formed. Such process of set level comparisons can be performed using currently available SQL syntax and semantics without proposed system [1]. If the set level comparisons performed using currently available SQL syntax, resulting query may be more and more complex, with the result it may take too much time to process the query than necessary. Such complex query becomes a difficult for the user to formulate, which results in too much costly evaluation.

Aggregation query is type of Iceberg Query [3] which calculates and computes aggregate values above the particular threshold value. High aggregate values always carry out more necessary information. Aggregate functions are COUNT, MIN, MAX, SUM and AVERAGE etc. In this paper, main focus is on processing queries that have aggregation function with antimotone property [4] such as MIN, MAX, SUM and COUNT.

In this paper, our aim is to process and retrieve the data using compressed Bitmap indexes. Currently available GROUP BY clause can only and only do scalar value comparison by accompany HAVING clause. Aggregate functions COUNT, MIN, MAX, SUM and AVERAGE etc. produces single numeric value, which compared to another single aggregate value. We have presented Aggregate function based technique and compressed bitmap index based technique. Aggregate function based technique processes set predicates in the normal way as processing conventional aggregate function. Second technique is compressed bitmap index in which bitmap indices is created on each attributes. This 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 [1]. For the purpose of compression Word-Aligned Hybrid (WAH) [5] technique is used. This technique now a day’s can be applied on all types of attributes such as numeric attributes [6][7] high cardinality categorical attributes [6], text attributes [8] etc. This technique is efficient for data warehouse query processing and OLAP.

II. RELATED WORK

Now a day’s, Many database management systems provides definition of attributes consisting a set of values such as nested table in Oracle and SET data type in MYSQL. For the Set predicates, there is no need of data storage and representation, hence included in standard DBMS. In real world applications, according to need of query groups and corresponding set are usually dynamically formed. Users can dynamically formed set level comparisons without any limitation caused by database schema for set predicates. It also allows cross attribute set level comparison. In [10][11][12], grouping variables and associated set concepts was introduces as SQL extension in order to allow comparison of multiple aggregate functions over same grouping condition. This paper mainly focuses on processing of data using compressed bitmap index and predicting the sets.

Bin He et al.(2012) explained the properties of bitmap index and developed a very efficient and powerful bitmap index pruning strategy for processing queries. Bitmap Index pruning based technique removes the necessity of scanning and processing the entire data set (table) and thus results in processing of fast query processing. This technique is more efficient than existing algorithms generally used in recent databases. By checking these characteristics of bitmap indices, the opportunities of computing queries efficiently using compressed bitmap index. A naive way for computing query using bitmap indices is to do pairwise bitwise-AND operations among bitmap vectors of all necessary attributes. This technique is not very efficient because the product of the number of
bitmap vectors of all attributes is large and large portion of these operations are not necessary.

Elizabeth O’Neil et al. proposed FASTBIT and RIDBIT techniques. FastBit is a research tool developed for studying and analysing how compression methods affect bitmap indexes, and has been used in a number of scientific applications [12]. It organizes table data into rows and columns, where each table is vertically partitioned and each column stored in individual files, each partition typically consisting of many millions of rows. Bitmap indexes are applied continuously without partitioning into bit segments as in RIDBit technique. The index used in this study is the Word-aligned hybrid (WAH) compressed basic bitmap index. In FastBit tool bitmaps are generated for all values of entire index for one individual in memory before writing the index file.

III. PROPOSED METHOD

In proposed system we have presented Aggregate function based technique and bitmap index based technique with Word-Aligned Hybrid (WAH) compression technique. Performance of previously available algorithm suffers from empty bitwise AND result problem. Following example demonstrates the impact to processing query and retrieving the relevant results.

A. Example:
Let us consider, a table has 10,000,000 tuples with attribute P has 10,000 individual values, and attribute Q has 10,000 individual values. P and Q will have 100,000 bitmap vectors for individual attribute. In worst case, total number of pairwise bitwise-AND operations is 100,000×100,000 = 100,000,000,000, which is 100 times larger than number of tuples. Since number of distinct groups is bounded by the number of tuples n in the relation, we need at most n bitwise-AND operations to answer the query. Here, more than 99 percent of bitwise-AND operations are not useful.

This challenge of empty bitwise-AND result problem is eliminated by proposed algorithm. In aggregate function based technique, regular functions can be processed together with the grouping, in which memory is available for storing hash tables of all individual groups. This algorithm processes into 1-pass as well as 2-pass iterations. If number of groups become too large that hash table cannot be fit in the main memory, it adopted sorting based aggregation method or two pass hashing based technique in DBMSs. In 1-pass iteration, input table is sorted and partitioned by hash function. In second pass tuple in same group are loaded in main memory and operations are performed independently.

Following are steps of proposed algorithm:
1) Step 1: Bitmap index vector is created for individual attributes available in the tables of the database.
2) Step 2: Compression is provided to individual attributes using Word-Aligned Hybrid (WAH) technique.
3) Step 3: When the user fires the query to retrieve particular records, requested records will be accessed according to their compressed bitmap vectors of individual attributes.
4) Step 4: Results are computed using both the techniques and results are compared according to time to access that records from the tables.

(a) Table R (b) Bitmap indices for A, B
Fig. 1: An example of Bitmap index

Each value in the column corresponds to a bitmap vector of length r, in which the kth position of the vector is 1 if this value appears in the kth row and otherwise.

IV. RESULTS AND DISCUSSION

![Table R and Bitmap Indices](image)

Fig. 2: Records retrieved using Aggregate function Based Technique.[Total Execution Time =29611 ms]
The experiments were performed on the Intel i3 CPU @ 1.90 GHz processor with 4GB RAM memory. We conducted experiments on Aggregate function based technique, bitmap index based technique and compressed bitmap index based technique. The experiments are carried on road-accidents safety dataset taken from official web site of united Kingdom(UK). Dataset consist of road accidents data with casualties and vehicles from the year 1974 to 2012. Each tables in the Road-safety dataset consist of more than 5000000.

Fig. 3: Records retrieved using Bitmap Index Based Technique. [Total Execution Time = 12000 ms]

Fig. 4: Records retrieved using Compressed Bitmap Index Based Technique. [Total Execution Time = 5215 ms]

Fig. 5: Result Analysis

Records are retrieved using all presented techniques. In Fig. 2., time required to retrieve records using Aggregate function Based Technique is 29611 ms. In Fig. 3., time required to retrieve records using Bitmap index Based Technique is 29611 ms. In Fig. 4, Records retrieved using Compressed Bitmap Index Based Technique is 5215 ms. Result analysis is mentioned in Fig. 5. In proposed system time required to retrieve records is much less than existing system.

V. CONCLUSIONS

This paper presents an efficient algorithm for query processing using compressed bitmap indices and Aggregate function based technique. Our algorithm demonstrates better
performance over existing available schemes. Set predicates combined in a group, allow selection of dynamically formed groups and set values. We observed that bitmap index has benefits of saving disk access. Computation time can be reduced by conducting bitwise operations on Bitmap indexes. The problem of massive empty AND results, can be eliminated using algorithm with priority queues. We further developed Word-Aligned Hybrid (WAH) optimization strategy to further improve the performance of the system. The issue that we consider for future work, we will investigate in solutions when the data are of large size such that the bitmap of a single column does not fit in main memory.

REFERENCES
[1] Chengkai Li, Member, IEEE, Bin He, Ning Yan, Muhammad Assad Safiullah "Set Predicates in SQL: Enabling Set-Level Comparisons for Dynamically Formed Groups” IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 2, FEBRUARY 2014.