Acceleration Of Query Response In Distributed System Using Materialized View Approach

Pooja G.Dhawane  
Department CSE, R.T.M.N.U.  
poojadhawane@gmail.com

Prof. Garima Singh makhija  
Department CSE, R.T.M.N.U.  
garima11makhija21@gmail.com

Abstract—A data warehousing technology is nothing but essential key pieces of information which are used by industries for the business strategy formulation and implementation. The success of data warehouse depends on appropriate data delivery method. The delivery process should be quick and accurate. This can be achieve by enhancing the query execution i.e. by minimizing the query response time. Performance of query can be improved by different approaches like hybrid integration system where most recent data is stored at mediator i.e. common access point to distributed system, optimization of query, use of proper data structure, reduce complexity of query etc. leaving all alternatives, we are planning to use materialized view approach. Materialized view is used to store the result of queries. We are going to store the result of queries based on the frequency of queries fired by the user. The queries having high frequencies i.e., based on the user demandable queries materialized view is designed. This will save the time of revisiting the base table whenever any query is fired frequently and also saves the time of cost of operations like join, aggregations etc.

Keywords—Materialization, query frequency, optimization, data cluster, data warehouse.

1. INTRODUCTION
Data warehouse is an approach to the integration of data from multiple, possibly very large, distributed, heterogeneous databases and other information sources. [1]

In existing system [2] the attributes which are most probably to extract where placed at the common access point called mediator to the distributed system. This has reduced the access time for the getting the data from original source of data. But if the attributes placed at the mediator are not proper then performance of the query will not be up to the mark. The primary intent of this research is to develop a system which will reduce the execution time for query i.e., getting the response from database as early as possible while in turn improves the performance of queries. We are going to take the help of materialized view. To avoid accessing the original data sources and increase the efficiency of the queries fired on data warehouse, some results in the query processing are stored in data warehouse.

2. RELATED WORK
Effective query processing using materialized view approach has been studied and implemented by many researchers by their own ways to achieve query optimization so as to decrease processing time.

In Hybrid integration system [2] some part of data were placed on mediator i.e. common access point to data warehouse. The attributes of interest are extracted from various data sources then possible values are assign to them and this information is placed on mediator. But if the attributes chosen are in appropriate then query performance will not be at the mark. Thus, we are trying to eradicate this problem using materialized view which directly stores the result of queries executing frequently.


Harinarayan proposed greedy algorithm [4] using data cubes to select the materialized view but this work does not consider the storage space required.

Sanket Patel used Tree based algorithm [5] for the selection & maintenance of materialized view. The decision of which data set should be materialized must be made on the basis of the system workload, which is a sequence of queries and updates that reflects the typical load on the system.

Y.D.Choudhary [6] proposed a novel CBFSMV algorithm for selection of materialized view using query clustering strategy that reduces the time as compared to response time for actual database.

In [7], authors have implemented dynamic adjustment for static materialized view selection algorithm that is EMVSDIA (Efficient Materialized view selection Dynamic Improvement algorithm). Experiments demonstrates EMVSDIA has excellent implementing efficiency and reach to expected efforts.
B. Ashadevi, Dr. R. Balasubramanian [8] implemented an algorithm that is projected for choosing the views to materialize on basis of their weight acquired in the query set. A greedy algorithm is used to incorporate the maintenance cost and storage constraint in the selection of data warehouse Materialized View. It reduces complexity of algorithm compare to previous algorithm [9].

Hybrid algorithm is combination of Evolutionary algorithm and Heuristic algorithm. The result of algorithm show that applying an evolutionary algorithm to either global processing plan optimization or materialized view selection for a given global processing plan can reduce total query and maintenance cost. Excessive computation time of pure Evolutionary algorithm and unsatisfactory quality of solution of Heuristic algorithm can be overcome by using Hybrid algorithm[10].

3. IMPLEMENTATION FRAMEWORK

After studying different research paper and analysing different algorithm to select MV. By keeping in mind the goal of our work to enhance performance of query in data warehouse using MV. We have divided the work into four parts, as shown below.

3.1 Creation of data warehouse
3.2 Data search by attribute
3.3 Creation of Materialized view
3.4 Search using MV

3.1 Creation of data warehouse
In this phase we collect raw data i.e, dat file is used. This raw data is loaded in temp.table. After cleaning the data, data warehouse is created. Now data is available by executing query on them. once the data warehouse is created it is used to create cluster data warehouse. These clusters are distributed on different systems. Clusters are formed on attributes.

3.2 Data search by attribute
We fire the queries on data warehouse and record the time needed to search the record based on attribute.

3.3 Creation Of Materialized View
In this phase MV is created by firing the queries. Each of search query is added in .txt file as per that MV is created. The size of MV depends on what type of database is used. It works like cache memory.

3.4 Search using MV
User fires the query. If data is available in MV user does not visit the base table if miss occurs then record is search in original data source. The time required to search the data using MV get reduced as visiting to base table is avoided.

4. EXPERIMENTAL RESULTS

This section shows the running experimental results that are carried out using materialized view schema by applying steps A, B, C, D. The sample queries are fired on database and following outcomes are obtained.

Fig.1. selection of Raw Data
Fig. 2. Cleaning of Raw Data

Fig. 3. Data Clustering on Attribute Country, Time zone, Continent
Fig. 4. Query posted on data warehouse without MV and time is recorded.

Fig. 5. Graphical representation of search time to retrieve data from base source.
CONCLUSION

From the above experimental result, we can see that with the help of materialized view the time to visit the base table is saved. Hence, the performance of query is improved to 15 to 16 times. Because MV is having the result of frequently fired queries. Thus, by implementing the materialized views in distributed system, we are trying to get the combination of good query response time from which query processing cost should be minimized.
6. REFERENCES


[8] B. Ashadevi and Dr. R. Balasubramanian, "Optimized Cost Effective Approach for Selection of materialized Views in Data Warehousing", JCS&T Vol. 9 No. 1, April 2009.
