



Implementation of Algorithms for Materialized View Selection and Preservation in Data Centric Environment

Gaurav Ambatkar*, Dr. Tripti Arjariya

Bhabha Engineering Research Institute,
Bhopal, India

Abstract— Now a day's online analytical processing for data centric environment represents some of the latest trends in computing environments for large scale processing and analysis of data. A data warehousing technology is nothing but essential key pieces of information which are used for making effective business strategy formulation and implementation. Any data warehouse success has depended on appropriate delivery process method. The delivery process should be quick and effective. Here the major focus is on query processing time when queries containing many joins and to resolve joins product is used with some desired condition and thus join queries are much more expensive in terms of execution speed is concern and thus to get the quick response of heavy I/O operations query or complex queries containing joins we required effective and accurate approach which decreases query processing time effectively. The solution to above said problem is creation of materialized view for frequently used queries and not all queries need to be materialized because it is not practically feasible due to materialized view storage space as well as its maintenance cost constraint. So to reduce query processing time proper queries can be selected on the basis of certain parameters like query frequency, query storage, query processing cost as well as maintenance cost. The implemented framework parameters are query frequency, query processing time and query storage to optimize the query processing time thereby resulting efficient data delivery system for data centric environment

Keywords— Data Centric Environment, Materialization, Threshold, Query processing cost, query frequency, query storage.

I. INTRODUCTION

A data centric environment is a warehouse of subjectively selected historical operational data which may answer complex, statistical or analytical queries. Data centric environment provides easy organization and maintenance of large data besides this it also provide fast retrieval and analysis of huge data in the desired manner as well as depth required now and then. As the operational data size always increases, the speed of accessing data for analysis of the data is playing an important role significantly. So to improve data processing speed the frequently used pre-computed intermediate results obtained from the query processing are stored in the data warehouse called as materialize views. In this way materialized view provides effective solution for the queries posted to the data warehouse, by preventing the users from accessing the base tables.

Materialized views acts just like a cache having physical storage space, so data can be retrieved quickly. But a materialized view requires additional storage space overhead as it required physical storage and thus that physical storage maintenance is also required when the data warehouse refresh. Data centric environment is capable of answering analytical queries and performing analysis in an efficient and quick manner, in the view of the fact the data warehouse research community give effective solutions for the problem of representing data in a form suitable for analytical queries, it has not totally addressed other performance issues like, query response time for a given complex queries, view maintenance time, etc.

The method of reflecting changes to a materialized view in response to the changes like inserts or update or delete operations on the base table is called as 'Materialized View Maintenance problem' It has associated a 'Materialized View Maintenance Cost'. Because of this view maintenance cost, it is very infeasible to make all views materialized under the specified storage space. This need to select an appropriate set of views to materialize for answering queries, this was denoted Materialized View Selection (MVS) and maintenance of the selected view denoted Maintenance of Materialized View (MMV).

Materialized views are really very important for improving performance in many business applications that's why recently database research community paying attention to the materialized view selection and maintenance. The primary intent of this research is to develop a framework for selecting views to materialize so as to achieve finer query response in low time by reducing the total cost associated with the materialized views. The proposed framework exploits all the cost metrics coupled with materialized views such as query execution frequency, query access cost, base-relation update frequency, view maintenance cost and the system's storage space constraints. The framework sustains existing materialized views periodically by removing views with low access frequency and high storage space. The queries with high access frequencies are selected for the view selection problem.

This paper is organized carefully to describe a related work of materialized view selection, preservation in section 2 as well as Materialized Views Selection ,preservation approach implementation framework details is explaining in section 3. Where as in section 4, we shown performed experimental result, and its discussion, in section 5, in the last we concluded the paper and provide the used references.

II. RELATED WORK

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

Dr. T.Nalini et al. [1] implemented an I Mining algorithm for materialized view selection they also focus on incremental maintenance of materialized view but preservation work was not addressed in this piece of work.

Ashadevi, B and Balasubramanian[2] also studied and proposed framework for selection of views which need to be materialize they considered views selection, including query processing frequencies, base relation, update frequencies, query access costs, view maintenance costs and the system's storage space constraints and then selects the most cost effective views to materialize and thus optimizes the maintenance storage, and query processing cost. They also addressed the preservation of existing materialized view framework.

Himanshu Gupta and Inderpal SinghMumick [3] proposed a useful greedy algorithm to minimize the maintenance cost associated with the materialized view selection framework. This paper gives view selection under disk storage space & maintenance cost constraints are addressed carefully.

Yang, J et al.[4] proposed a heuristics algorithm for multiple views processing plan (MVPP), which is used to present the materialized view problem formally.

Amit Shukla et al. [6] give a very simple but fast heuristic algorithm, PBS. PBS algorithm runs much faster than BPUS, and it is fast enough to make the exploration of the time-space trade -off .

P. P. Karde et.al. [8] work gives an overview of various techniques that are proposed as well as implemented in past and recent for selection of materialized view. The major issues related to maintaining the materialized view are also discussed work. Here some future point aspects are also stated that might be useful for many recent researchers.

Dr. Y.D.Choudhari et al.[11] had implement a novel CBFSMV algorithm for selection and preservation of materialized view using query clustering strategy that reduces the query execution time as compared base table access.

Our main intention is to materialize only effective queries or views by taking into consideration of complex query frequency, query processing time and storage space requirement.

III. IMPLEMENTED FRAMEWORK

To efficiently access the data centric environment analysis report data, materialized views are useful for the users to quickly get the desired search results of frequent with best storage and processing time queries. The main motive behind the implemented materialized view selection as well as preservation only most promising user queries or views by taking into consideration of query storage ,query frequency, query processing time etc. The implemented approach is applied on data centric model queries fired by the users. Technically it is not feasible to create materialized view for all user queries due to the storage space and maintenance cost constraints so the queries that are most frequently used by the users along with, the query processing cost and storage cost combination should be less are considered to be the efficient materialized view queries and only that queries materialized views were created. For experimentation point of view we have created a small database “ mvdb” containing schemas of five tables which shows the relationship between customer and supplier information. Before applying materialized view selection framework for selection of most prominent queries for customer and supplier relationship tables we need to generate large set of dummy records for the above specified database tables by building record generator. After generation of dummy record set next step is to create complex queries based on created records and then select most prominent queries from the created query set using implemented materialized view algorithms selection model.

3.1 Algorithm1: Record Generator Algorithm

Step1: Specify the number of record you want to insert inside mvdb database tables.

Step2: Generate specified number of random records using combination of table attributes values stored inside arrays with some auto generated primary values.

Step3: Store generated records inside specific tables with appropriate constraints.

3.2. Algorithm2: Prominent Materialized View Selection Algorithm

Assumptions:

$N \rightarrow$ Number of Query

begin:

Repeat for $I \leftarrow 1$ to N

Check query existing materialized View

if materialized view not exists

Find query storage space

Find query processing time

Find query access frequency

Else

```

Discard the query
end repeat
Repeat for I ← 1 to N
    Find query storage, processing and access cost
    Find query selection cost using weighted
    constants values
end repeat
Repeat for i ← 1 to N
    if query selection cost is greater then
    specific threshold value then
        Create materialized view for the selected query
    else
        Discard the query
    end repeat
end repeat
    
```

The above algorithms are used to create small database with dummy record set and find most outstanding queries on the basis of query frequency, processing time and storage space information then calculate each query selection cost and compare with specific threshold value using summation of selected queries. If the query selection cost is bigger than the materialized view selection threshold value then creating the materialized view for the selected query otherwise neglect that particular query.

The above algorithms for generation and selection of effective materialize views can be achieved the desired multi-objective result i.e. it provides the finest combination of query processing time, query storage and query frequency.

3.3. Algorithm3: Preservation of Existing Materialized Views

This below algorithm shows the preservation procedure for the materialized views which are frequently used (views which are currently active) as well as it will also shows the information of less frequently used materialized views (views which are not currently active) so that non active materialized views can be removed from memory.

```

for each Materialized View in database
    Find materialized view frequency
    Find materialized view storage
        Find materialized view cost based on
        materialized view frequency, storage
    If materialized View Cost is bigger than specific threshold value then
        preserve that materialized view
    else
        Remove current Materialized view;
end for
    
```

The materialized view preservation algorithm keep hold of the materialized views having frequent access frequency and less storage cost and eliminate the less prominent materialized view having very low access frequency and very high storage space for the materialization of active new most promising views.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section shows the running experiment results that are carried out using mvdb database schema by applying algorithm 1 and 2. The various sample user queries with its query frequency, storage space & processing time are shown in Table1 given below. Whereas each query, processing cost, frequency cost ,storage cost as well as selection cost and specific materialized view selection threshold which are calculated using algorithm2 are shown in Table2.

TABLE1: QUERY MATERIALIZED VIEW SELECTION PARAMETER VALUES

Sample Query	Query Frequency	Query Storage (bytes)	Query Processing Time(ms)
q1	5	181	16
q2	4	57	0
q3	3	3	0
q4	2	154	0
q5	2	119	0

TABLE 2: MATERIALIZED VIEW QUERY SELECTION COST INFORMATION

Sample Query	Query Frequency	Query Storage (bytes)	Query Processing Time(ms)	Selection cost
q1	1	1	1	0.976
q2	0.8	0.315	0	0.9
q3	0.6	0.017	0	1.205

q4	0.4	0.851	0	1.1
q5	0.4	0.657	0	1.039
Minimum Threshold Value =	0.536			

The sample queries having better selection cost is selected by comparing with threshold the results are shown in Table3.

TABLE 3: SELECTED QUERY INFORMATION

<i>Query</i>	<i>Selection cost (S_Q)</i>
<i>q2</i>	<i>0.74</i>
<i>q3</i>	<i>0.79</i>

Above table depicts only those queries which satisfy the multiple constraints criteria for query selection so here we are selecting only two most promising queries out of five sample queries.

4.1 Comparative Analysis

This section shows the comparative analysis of the implemented approach for the direct view access and materialized view access. Below given snapshot clearly represents that creation of materialized view based on proposed parameters which provides best results. So now we can be declared that: The-direct base table access approach required the highest query processing time with no view maintenance as well as storage cost. The materialized-views approach can provide the best query performance for complex query with some view maintenance and storage costs are incurred.

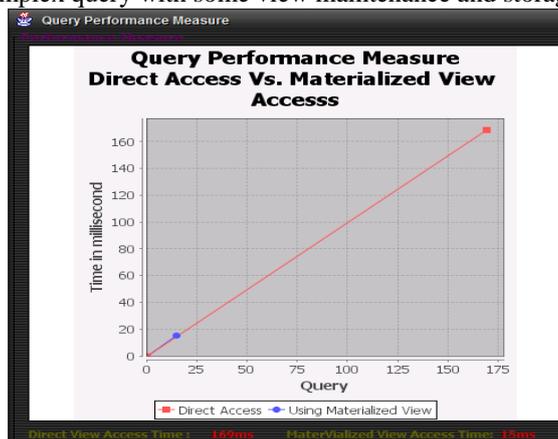


Figure 3: Shows comparison of execution time of the query using materialized view selection framework and execution time of the query if it is posted for original database (without framework).

V. CONCLUSIONS

This paper gives the exact idea regarding how to select most prominent queries out of bunch of complex time expensive queries. The selection process for materialization is based on query frequency, query processing time and query storage space where as preservation is based on materialized view storage and frequency.

For experimentation, the proposed framework is executed on the simulated data centric model using text file containing complex queries, to find the efficiency of the approach in selection of materialize view along with this we embedded preservation framework which shows effective results for preserving most useful materialized views and releasing undesired materialized views on the basis of preservation cost. For future prospective we try to apply this implemented model against some real-world data centric environment along with maintenance model for created materialized views.

REFERENCES

- [1] Dr.T.Nalini, Dr.A.Kumaravel , Dr.K.Rangarajan,"A Novel Algorithm with IM-LSI Index For Incremental Maintenance of Materialized View" JCS&T Vol. 12 No. 1 April 2012
- [2] B.Ashadevi, R.Balasubramanian," Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.10, October 2008
- [3] Gupta, H. & Mumick, I., Selection of Views to Materialize in a Data Warehouse. IEEE Transactions on Knowledge and Data Engineering, 17(1), 24-43, 2005.
- [4] Yang, J., Karlapalem. K., and Li. Q. (1997). A framework for designing materialized views in a data warehousing environment. Proceedings of the Seventieth IEEE International Conference on Distributed Computing systems, USA, pp:458.

- [5] V.Harinarayan, A. Rajaraman, and J. Ullman.“Implementing data cubes efficiently”. Proceedings of ACM SIGMOD 1996 International Conference on Management of Data, Montreal, Canada, pages 205--216, 1996.
- [6] Proc. 24th Int. Conf. Very Large Data Bases, 1998, pp. 488–499.
- [7] Wang, X., Gruenwalda. L., and Zhu.G. (2004). A performance analysis of view maintenance techniques for data warehouses. Data warehouse knowledge, pp:1-41.
- [8] Mr. P. P. Karde, Dr. V. M. Thakare. “Selection & Maintenance of Materialized View and It’s Application for Fast Query Processing: A Survey”. Proceedings of International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010
- [9] Abdulaziz S. Almazyad, Mohammad Khubeb Siddiqui. “Incremental View Maintenance: An Algorithmic Approach”. Proceedings of International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 03
- [10] Elena Baralis, Tania Cerquitelli, and Silvia Chiusano,” I-Mine: Index Support for Item Set Mining” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, april 2009
- [11] Y.D. Choudhari and Dr. S. K. Shrivastava, “Cluster Based Approach for Selection of Materialized Views”, International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 2, Issu