ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

An Approach for Efficient Query Processing For M.V. Selection and Maintenance

[1]Oshin Rangari, [2]Sapna Kose, [3]Swati Shirbhate, [4]Prof. Pravin O. Balbudhe

[1,2]Student B.E (8th sem), [3]Professor, Department of Computer Engineering,
Suryodaya College of Engineering, Nagpur, India

***Abstract*: A data warehouse as a storehouse is a repository of data collected from multiple data sources (often heterogeneous) and is intended to be used as a whole under the same unified schema. A data warehouse gives the option to analyze data from different sources under the same roof. If the executive of the company wants to access the data from all stores for strategic decision-making, future direction, marketing, etc., it would be more appropriate to store all the data in one site with a homogeneous structure that allows interactive analysis. In other words, data from the different stores would be loaded, cleaned, transformed and integrated together. To facilitate decision-making and multi-dimensional views, data warehouses are usually modeled by a multi-dimensional data structure. In this paper, we present a framework for selecting best materialized view so as to achieve the effective combination of good query response time, low query processing cost and low view maintenance cost in a specified storage space constraint. The framework implementation parameter includes query frequency cost, query storage cost and query processing cost. The framework select the best cost effective materialize views to optimize the query processing time thereby resulting efficient data warehousing system.**

***Keywords:* Data Warehouse Materialization, View-Maintenance, Access Frequency, Threshold, Query processing cost.**

I. INTRODUCTION

Data warehouse (DW) can be defined as subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decision [2]. It can bring together selected data from multiple database or other information sources into a single repository [3]. To avoid accessing from base table and increase the speed of queries posed to a DW, we can use some intermediate results from the query processing stored in the DW called materialized views. Therefore, materialized view selection involved query processing cost and materialized view maintenance cost. Materialized views are the derived relations, which are stored as relations in the database. When a base relation is update, all its dependent materialized views have to be updated in order to maintain the consistency and integrity of the database. The process of updating a materialized view in response to the changes in the base relation is called 'View Maintenance' that incurs a 'View Maintenance Cost'. Because of maintenance cost, it is impossible to make all views materialized under the limited space and time. This need to select an appropriate set of views to materialize for answering queries, this was denoted Materialized View Selection (MVS) and maintenance the selected view denoted Maintenance of Materialized View (MMV). [1-3]

Materialized views are 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.

II. RELATED WORK

The problem of finding appropriate views to materialize to answer frequent queries has been studied under the name of Materialized View Selection (MVS).

Dr. T.Nalini et al. [1] proposes an IM-LXI index for incremental maintenance of materialized view selection of materialized views so that query evaluation costs can be optimized as well as view maintenance and view storage was addressed in this piece of work.

Harinarayan et al. [21] presented a greedy algorithm for the selection of materialized views so that query evaluation costs can be optimized in the special case of "data cubes". However, the costs for view maintenance and storage were not addressed in this piece of work.

Yang et al. [5] proposed a heuristic algorithm which utilizes a Multiple View Processing Plan (MVPP) to obtain an optimal materialized view selection, such that the best combination of good performance and low maintenance cost can be achieved. However, this algorithm did not consider the system storage constraints.

Himanshu Gupta and Inderpal Singh Mumick [8] developed a greedy algorithm to incorporate the maintenance cost and storage constraint in the selection of data warehouse materialized views. We developed algorithms to select a set of views to materialize in a data warehouse in order to minimize the total query response time under the constraint of a given total view maintenance time. They have designed approximation algorithms for the specialcase of OR view graphs.

Chuan Zhang and Jian Yang [5] proposed a completely different approach, Genetic Algorithm, to choose materialized views and demonstrate that it is practical and effective compared with heuristic approaches.

Sanjay Agrawal et al. [6] proposed an end-to-end solution to the problem of selecting materialized views and indexes. Their solution was implemented as part of a tuning wizard that ships with Microsoft SQL Server 2000.

III. PROPOSED SYSTEM

Materialized View Selection and Maintenance Objectives are:

1] Decreased CPU consumption

2] Faster response times

It can be utilized by the users to obtain the quicker results once a set of views is materialized for the data warehouse.

IV. PROPOSED METHODOLOGY

I. MATERIALIZED VIEW SELECTION FRAMEWORK:

This section elaborates the created framework approach for the selection of materialized view. Materialized views are beneficial for the users to quickly get the search results for frequent queries. The ultimate aim behind the proposed materialized view selection framework is to materialize the user views by taking into consideration of query frequency, query processing cost and storage requirement of query. The developed framework is applied on data warehouse model, DW and a user's selected query file (UQF) that contains the list of queries used by the number of users. As it is practically impossible to create materialized view of all user queries due to the storage space constraints the queries that are frequently used by the users should be selected but, at the same time, the query processing cost and storage cost should be less. Accordingly, we have used the data ware house, DW that contains four tables. The schema of the data ware house used in the framework is represented with four various tables such as tblstudent (T1), tblmarks (T2), tblattendance (T3) and tblextraactivity (T4) The student table, which consists of following field records such as srno, studId, name, college branch and course where, studId is the primary key. The mark table contains one tuple for each subject marks, and its key is studId.

The attendance table contains details about the student attendance of each subject and its field records are srno, studId (foreign key), subject, class_date and attendance. The last extraactivity table contains each student extraactivity record using following fields srno, studId (foreign key),activity_name,student_post and extra_point.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

The first phase of materialized view selection is generation of huge random set of records for the above given database tables using random data insertion record generator. After that all possible set of complex queries are generated on above created records. The most prominent queries are selected from the given created query set

II. MATERIALIZED VIEW MAINTENANCE:

This section describes the detailed procedure of the designed approach to view maintenance. The principle behind the secondmodule is to handle the maintenance problem without recomputing the materialized views. For example, if the data warehouse gets updated (Addition and deletion of data source) after selecting materialized view, the corresponding updating data source should be reflected in the view. In order to deal with the updating and deletion of data source, the output of the query should be given by considering the updated data records without re-computing the whole process. Accordingly, we have designed an approach to view maintenance without accessing the data warehouse or view. The process of updation and deletion can be happened whenever the data sources are updating the records to the original data warehouse. The diagram given in figure 1 describes the datawarehouse updation from the data sources and figure 2 describes the overall procedure of the proposed approach.

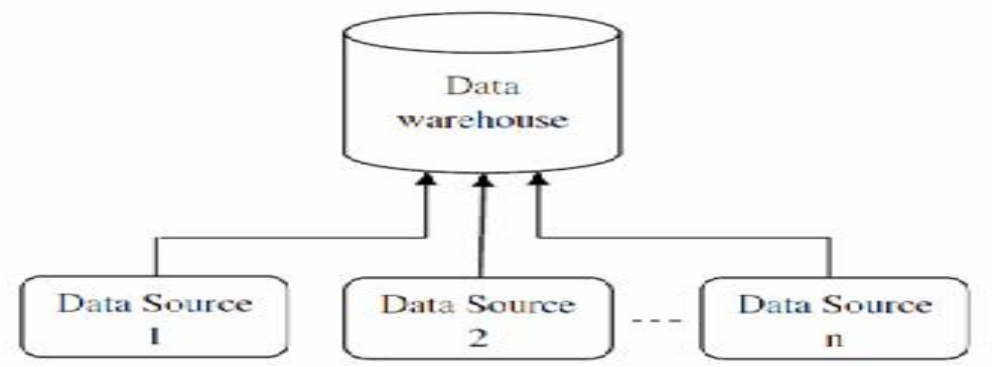

Fig1: Data warehouse updation from the multiple sources

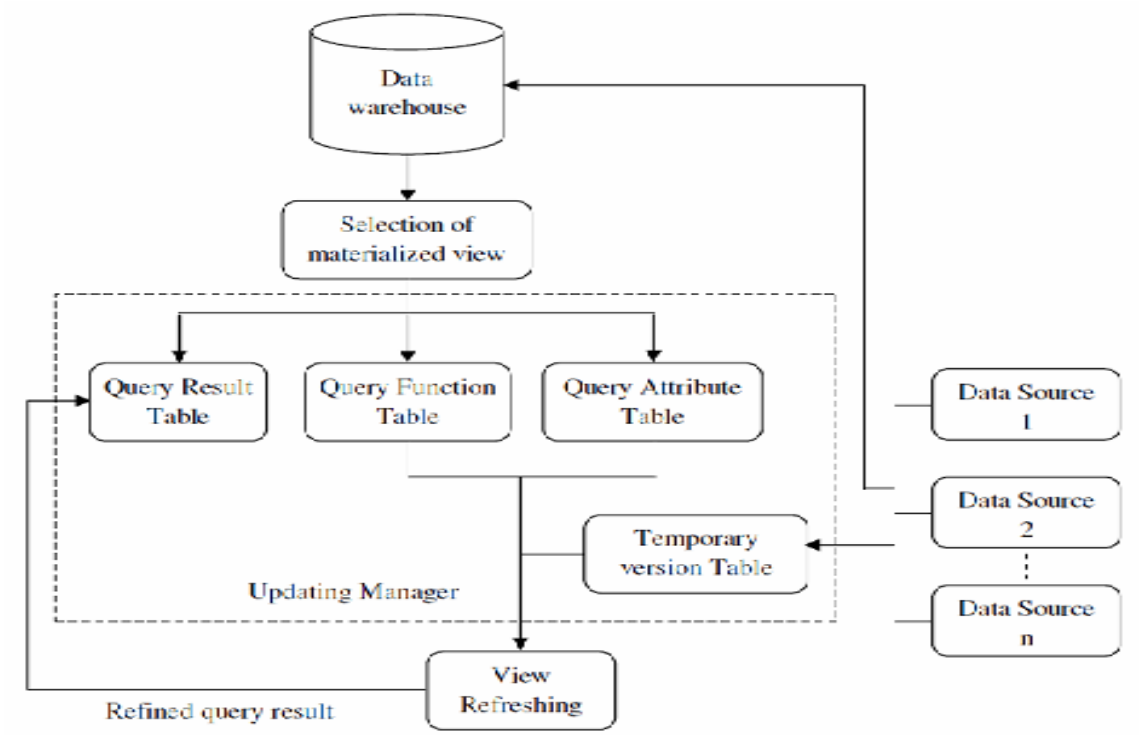

Fig 2: View Maintenance process

Before selecting new views for materialization, the existing materialized views are needed to be sustained based on their access frequency and storage space. The steps for the above process are given in Algorithm Steps of Algorithm:

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

Step1: Calculate

* Query frequency cost(Q_{FC}).

* Query storage cost (Q_{SC}).

Step2: Calculate Materialized views Maintenance cost on the basis of query frequency and storage cost as shown below.

$M_Q = \alpha * Q_{FC} + \beta (1 - Q_{SC})$;

Step3: Calculate Minimum M V Maintenance Threshold Value as shown here

$T_P = \sum_{i=1}^{N} M_Q / N$

Step4: Select queries having materialized view storage space is less than materialized view maintenance threshold value.

Step5: Build the materialized view for the selected query

V. RESULTS ANALYSIS

The section shows the running experiment results that are carried out using simulated student database schema by applying algorithm 1 and 2. The various typical user queries along with its query frequency, storage space & processing time are shown in Table1. Whereas query frequency cost, processing cost, storage cost, selection cost and minimum materialized view selection threshold is calculated using algorithm2 and shown in Table2.

Table: 1 Materialized View Query Selection Parameter Information

Query	Q AF	QSP (bytes)	QPT(ms)
Q1	5	16384	203
Q2	4	344064	0
Q3	3	65536	0
Q4	2	16384	31
Q5	1	16384	15

Table 2 Materialized View Query Selection Cost Information

Query	QFC	QSC	QPC	SQ
Q1	1	0.048	1	0.976
Q2	0.8	1	0	0.9
Q3	0.6	0.19	0	1.205
Q4	0.4	0.048	0.153	1.1
Q5	0.2	0.048	0.074	1.039
Minimum Threshold Value =		**1.0440**		

The queries having selection cost is greater than the minimum materialized view selection threshold value need to be materialized for quick query processing as shown in Table3.

Table3 Queries that satisfy selection criteria

Query	*Selection cost (SQ)*
Q3	*1.205*
Q4	*1.1*

Above table shows only those queries which satisfy the multiple constraints so here we are selecting only two queries having selection cost is greater than the minimum materialized view selection threshold value from the set of queries shown in Table1.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

VI. COMPARATIVE ANALYSIS

This section presents the comparative analysis of the proposed approach with the direct view access for analyzing the difference of query execution time.

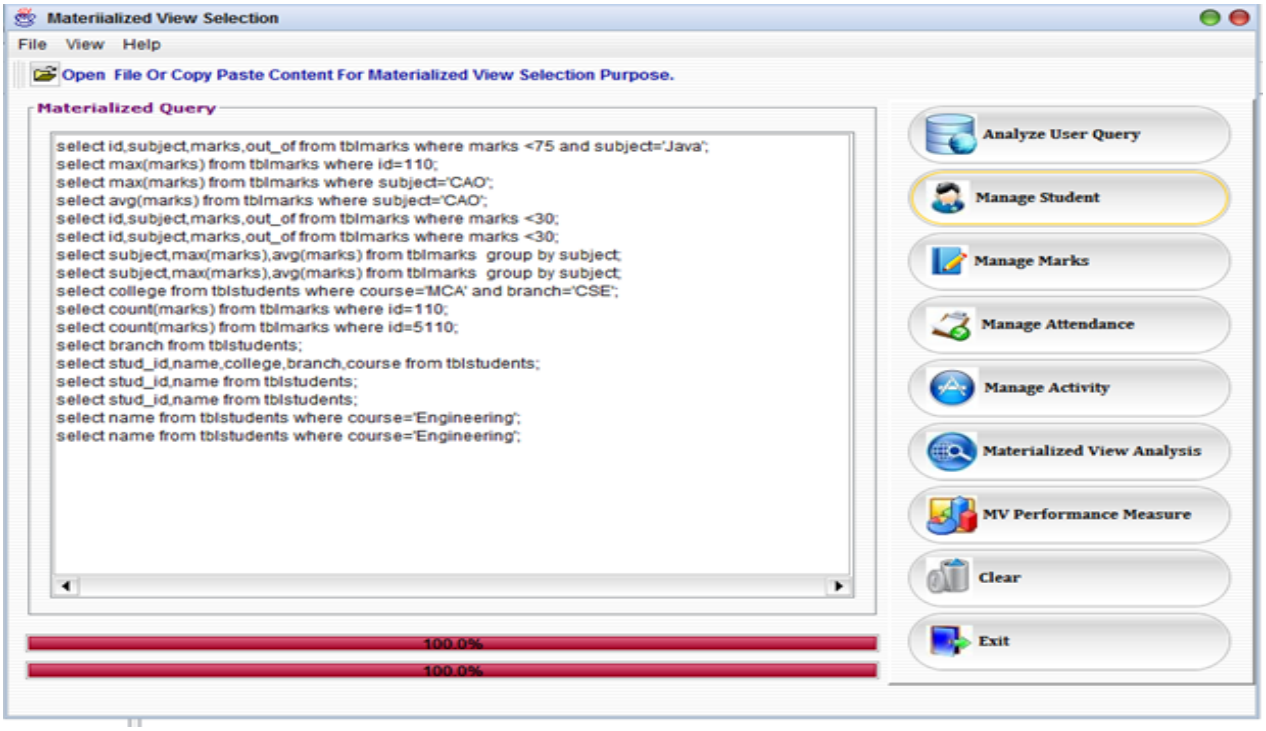

Figure 3: Main Window of Materialized View Selection Process.

Figure3:The above snapshot shows a scenario where a bunch of queries are given input to the materialized view selection main window using open file option. Here we can perform various operations like insertion, deletion, updation on dummy student database using manage student, manage activity, manage attendance and manage marks buttons. The analyze user query button is used to initialized the process of materialized view selection.

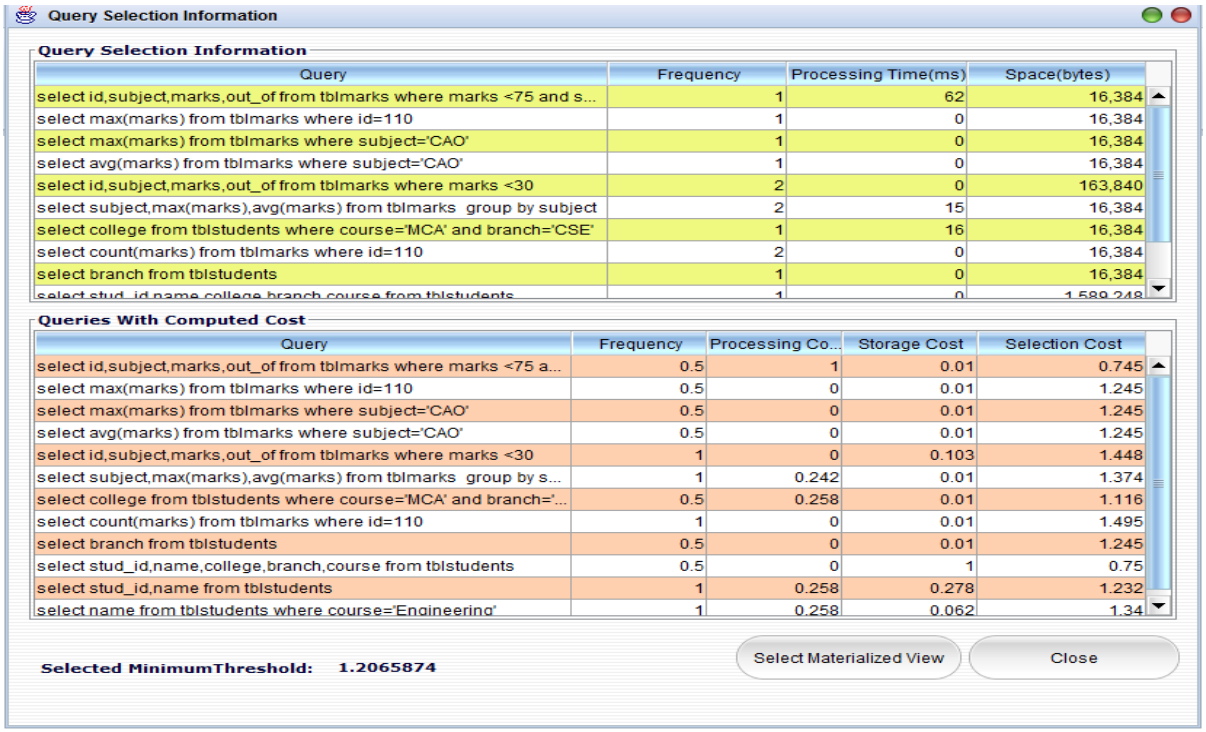

Figure: 4 Result Showing Query Information along with Query Frequency, Processing, Storage and Selection Cost

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

After pressing the analyze user query button all the user queries are analyze to calculate the query frequency, query processing time and query storage requirement and then with the help above information a query frequency cost, query processing cost and query storage cost is calculated using algorithm 2 and 3 after that selection threshold is calculated by summation of all the selection cost and then divide by the total number of queries to get the final selection threshold value which is shown in above figure 4.

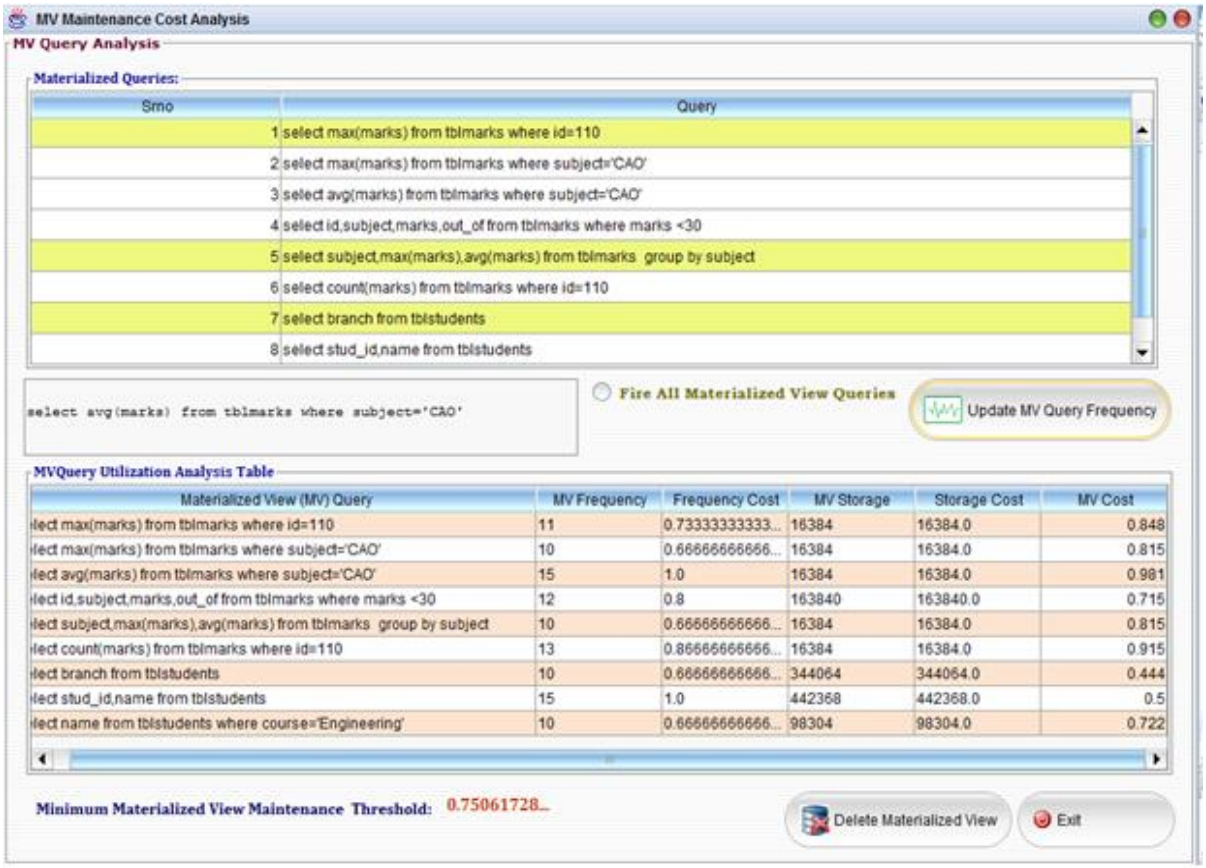

Figure: 5. Result of Materialized View Preservation Process

Figure 5.5 shows the preservation window of materialized view along with update materialized view query option. This window is used to sustain existing materialized views having high query frequency and low storage space and removing views with low access frequency and high storage space.

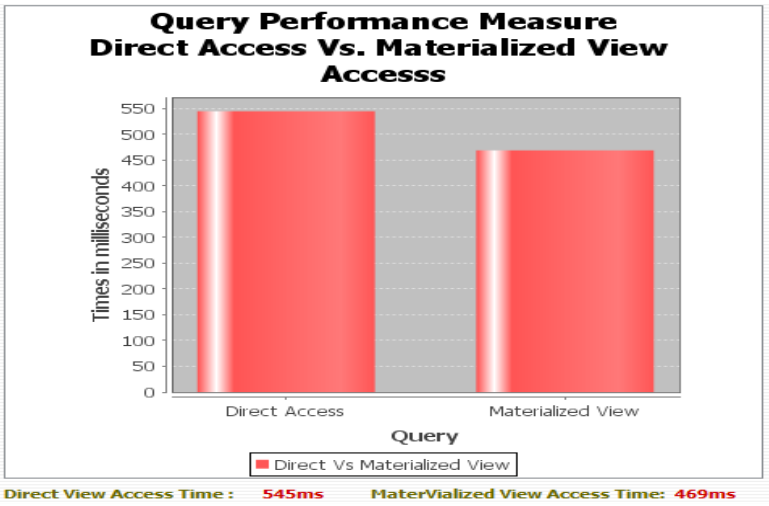

Figure 6. Shows comparison of execution time of the query using materialized view selection framework and execution time of the query if it is posted for original database (without framework)

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (171-178), Month: April - June 2015, Available at: www.researchpublish.com

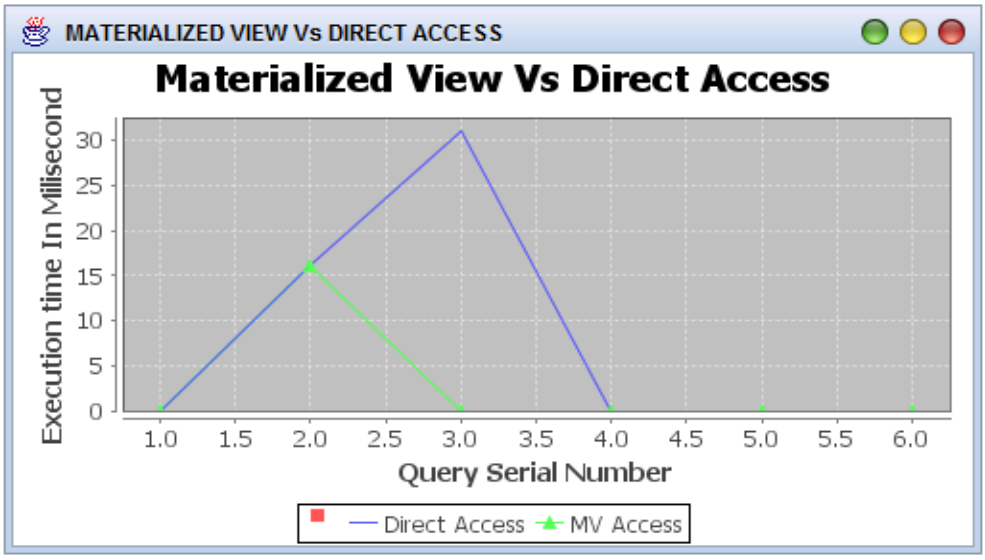

Figure 7. Above snapshot represents the calculation results, from which following observations can be stated: The all-direct base table access method requires the highest query processing cost with no view maintenance and storage costs are incurred. The all-materialized-views method can provide the best query performance with some view maintenance and storage costs are incurred.

VII. CONCLUSION

This paper gives the idea regarding best view selection for materialization and the effective incremental batch approach for materialized view maintenance.

The first approach is for materialized view selection, the essential constraints for materialize view selection are: query frequency, query processing time and storage space. The presented proposed methodology determines which queries are more beneficial using combination of query frequency, processing and storage cost for the creation of materialized view so as to achieve the quick query processing time.

The second approach is for materialized view maintenance(MVM) where after certain number of updates on the base table view maintenance process start in which selected materialized view queries attributes (queryattribute table records) are match with binary version table records which contains information of insert, update and delete queries fired on base table. If the query attribute table record are matched with binary version table records then materialized view maintenance is required otherwise there is no need to refresh the materialized view. This MVM approach avoids unnecessary refreshment of materialized view.

The third approach is for preserving best materialized views on the basis of high query frequency and low storage space and removes the materialized views with low access frequency and high storage space for the materialization of new views. For future research in this area could focus on validating this model against some real-world data warehouse.

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] . A. Shukla, P. Deshpande, and J. F. Naughton, "Materialized view selection for the multidimensional datasets," in 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 KhubebSiddiqui. "Incremental View Maintenance: An Algorithmic Approach". Proceedings of International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 03