Efficient Algorithms for Materialized View Selection in Data Warehousing Environment

[1] Ashish Mohod, [2] Manoj Chaudhari

[1] MTech. 4th SEM, Computer Science & Engineering
Department of Computer Science & Engineering,
Nagpur, India

[2] Department of Computer Science & Engineering,
Bhagwati Chaturvedi College of Engineering,
Nagpur, India

Abstract

The notion of data warehouse can be defined as subject-oriented, integrated, nonvolatile and time-variant collection of data in support of making management's decision effectively, for the success of data warehouse accurate and timely consolidated information as well as quick query response times is the fundamental requirement. To avoid accessing from base table and increase the speed of queries posted to a Data warehouse, we can use some pre-computed intermediate results from the query processing stored in the data warehouse called materialized views. The result of effective Materialized view selection provides an efficient data warehousing system. However, the materialized view needs to be effectively maintained to keep its contents integrated and consistent with the contents of its data sources. The materialized views have maintenance cost, that's why materialization of all views is not possible.
.

Keywords: Data Warehouse Materialized View, View-Maintenance, Access Frequency, Threshold.

1. Introduction

One of the most important decisions in designing data warehouse is selecting only those views to materialize which eliminates the overhead associated with expensive joins and aggregations for a large set of important class of queries. A materialized view is pre-computed data stored in a table that transparently allows users to query huge amounts of data much more quickly than they could access from the base table. Database retrieval of the materialized view is just like a cache, which is copy of the data that can be retrieved quickly. To select an appropriate set of view is the important target that reduces the entire query response time, however to maintain the selected views is critical but very important aspect of building effective data warehouse. The process of reflecting changes to a materialized view in response to the changes (inserts or update or delete)

in the base relation is called as 'View Maintenance' that incurs a 'View Maintenance Cost'. Because of view maintenance cost, it is impossible to make all views materialized under the limited space constraints. 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). [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.

This paper is organized as follows. We describe a related work of materialized view selection and materialized view maintenance in section 2, Materialized Views Selection framework implementation details is explaining in section 3. In section 4, we shown experimental result, and its discussion, in section 5, we concluded the paper and section 6 is used to provide the references.

2. 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 I-Mine algorithm for the selection of materialized views so that query evaluation costs can be optimized as well as maintenance and storage was addressed in this piece of work.

Ashadevi, B and Balasubramanian[2] proposed framework for selecting views to materialize(i.e., View selection problem), which takes in to account all the cost

IJCSN International Journal of Computer Science and Network, Volume 2, Issue 3, June 2013
ISSN (Online) : 2277-5420 www.ijcsn.org

72

metrics associated with the materialized 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.

Himanshu Gupta and Inderpal SinghMumick [3] developed a greedy algorithm to incorporate the maintenance cost and storage constraint in the selection of materialized views for data warehouse.

Yang, J et al.[4] proposed a heuristics algorithm based on individual optimum query plans. Framework is based on specification of multiple views processing plan (MVPP), which is used to present the problem formally.

Harinarayan et al. [5] proposed a greedy algorithm for the materialized views selection so that query evaluation costs can be optimized in the special case of "data cubes". This paper provides good trade-offs between the space used and the average time to answer query. Here, the costs for view maintenance and storage were not addressed in this piece of work.

Amit Shukla et al.[6] proposed a very simple and fast heuristic algorithm, PBS, to select aggregates for pre computation. PBS algorithm runs faster than BPUS, and is fast enough to make the exploration of the time-space trade -off feasible during system configuration.

Wang, X et al.[7] View maintenance techniques are classified into four major categories : self maintainable recomputation, not self-maintainable recomputation, self maintainable incremental maintenance and not self maintainable incremental maintenance. Self-maintainable Incremental maintenance performs the best in terms of both space usage and number of rows accessed.

Main objective of this paper is is to materialize the effective candidate views by taking into consideration of query frequency, query processing cost and space requirement along with view maintenance cost.

3. Implementation Details of Materialized Views 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.

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 generation of all possible set of complex queries are generated on above created records. The queries are selected from the given created query set using Algorithm 1.

Assumptions:

Q_S	→	Given set of queries
Q_{AF}	→	Queries access frequency
T	→	Threshold value
AL_{SQ}	→	Array List of selected queries

3.1. Algorithm 1

1: begin:
2: for each query in Q_S
3: find the frequency of each query Q_{AF}
4: if (Q_{AF} >= T) then
5: Add query to Array List AL_{SQ};
6: end if
7: end for

The candidate queries having access frequency greater than the threshold value T are selected for materialized

IJCSN International Journal of Computer Science and Network, Volume 2, Issue 3, June 2013
ISSN   (Online) : 2277-5420   www.ijcsn.org

73

view selection problem but, at the same time, the query processing and storage cost should be less thus queries processing time and storage cost may be calculated using Algorithm 2.

Assumptions:

Q_{Tot} →   Total no of queries having Q_{AF} >= T

Q_{MFreq} →   Maximum query frequency

Q_{PT} → Query processing time

Q_S → Query storage

Q_{MPT} → Maximum Query processing time

Q_{MS} → Maximum Query storage

Q_{PC} → Query processing cost

Q_{SC} → Query storage cost

Q_{FC} → Query frequency cost

Q_{RR} → Query result record storage value
        i.e Query data length and index length

Q_{TBE} → Query time before execution

Q_{TAE} → Query time after execution

Q_{CT} → Query cost table

S_Q → Query selection cost

M_T → Minimum threshold

α, β & γ → Weighted constant values in between 0 to 1

3.2 Algorithm 2

1: begin:

3: Repeat for I ← 1 to Q_{Tot}

4: Q_{FC} ← Q_{AF} /Q_{MFreq} ;

5: Q_{PC} ← Q_{TAE} - Q_{TBE} / Q_{MPT} ;

6: Q_{SC} ← Q_{RR}/ Q_{MS} ;

7: Q_{CT} ← Q_{FC} ;

8: Q_{CT} ← Q_{PC} ;

9: Q_{CT} ← Q_{SC} ;

10: end repeat

14: [Find selection cost]
    Repeat for I ←1 to Q_{Tot}
    S_Q = γ* Q_{FC} +β *Q_{PC}+ α(1- Q_{ST}) ;
    Q_{CT} ← S_Q ;
    end repeat

15: [Select MV Selection Threshold]
      M_T =$\sum_{i=K}^{K} S_Q$ / Q_{Tot}

16: [Select materialized view having good query response, low processing and storage cost]

Repeat for i ←1 to Q_{Tot}

S_Q ← Q_{CT} [i]
       if (S_Q >= M_T) then
     Build the materialized view for the selected query

17:   else
     Discard the query
    end repeat

Fig1

Thus, the above algorithm for selection of materialize views can be achieved the desired multi-objective i.e. it provides the best combination of better query response, low query processing cost and low storage space cost.

4. Experimental Results and Discussion

The section shows the running experiment results that are carried out using above simulated student database schema by applying algotithm1 and 2. The various typical user queries are shown below along with its query frequency, to be calculated using algorithm 1. The processing time, query result size along with query frequency cost, processing cost, storage cost, selection cost and minimum materialized view selection threshold is calculated using algorithm2.
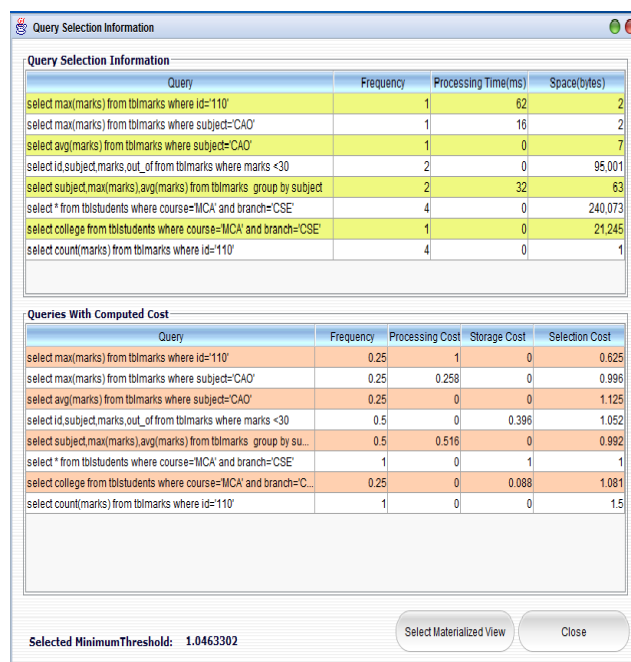

Fig1: Materialized View Query Selection Parameter Information Window

Above fig 1: showing query selection information frame window containing query frequency, query processing time, query result storage in bytes with query frequency, processing, storage and selection cost.

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 fig2.

IJCSN International Journal of Computer Science and Network, Volume 2, Issue 3, June 2013
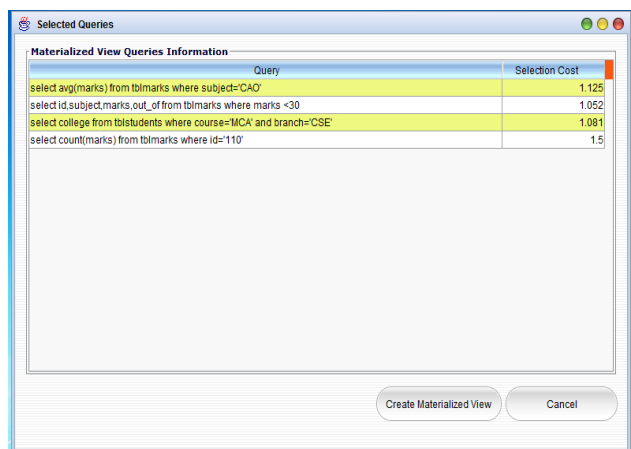ISSN   (Online) : 2277-5420   www.ijcsn.org

74

Fig 2: Selected Materialized View Queries Information Window

Fig 2 shows only those queries which satisfy the multiple constraints so here we are selecting only four queries having selection cost is greater than the minimum materialized view Selection threshold value from the set of queries
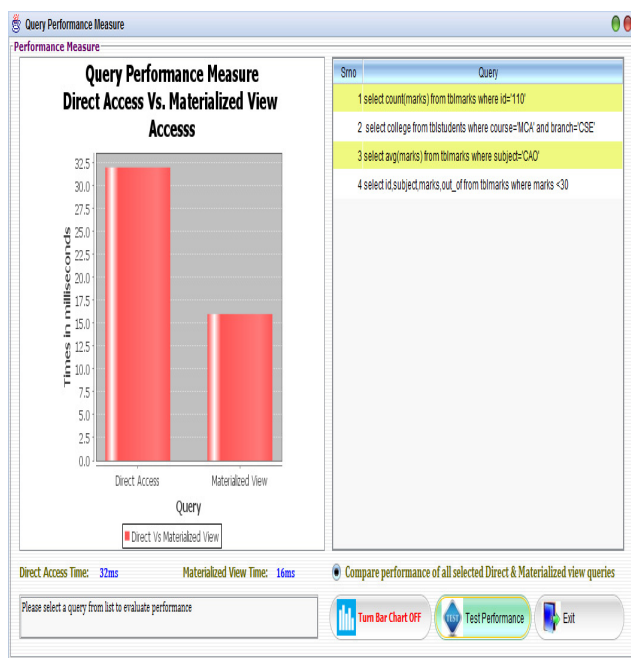

Figure 3 Shows comparison of execution time of the query using materialized view selection framework and execution time of the query if it is posted for original database (without framework).

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.

5. Conclusion

As materialized view store the precomputed data it is used to improve query performance, by minimizing query processing time. But due to view maintenance cost it is impossible to create materialized view of all the queries. Thus how to select the set of queries to be materialized so that query performance increases significantly and storage cost for storing materialized view minimized.

This paper gives the idea regarding how to select a set of materialized view with the help of various parameters like: frequency of query cost of query processing and storage space. We have presented proposed methodology that determines which queries are more beneficial for the creation of materialized view so as to achieve the high query performance.

For experimentation, the proposed framework is executed on the simulated student data warehouse model using list of query, to find the efficiency of the proposed approach in selection of materialized view. For future research in this area could focus on validating this model against some real-world data warehouse systems and also concentrate on incremental materialized view maintenance framework.

Acknowledgments

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 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, Issue 7, July 2012.

Authors

Ashish Mohod received the B.E. degree from U.C.O.E, Umred, and State-Maharashtra, India. He is pursuing MTech. in Computer Science and Engineering from Bhagwati Chaturvedi College of Engineering, Nagpur. Maharashtra, India. He was a Software developer between 2005 to 2009. He has presented more than 5 papers, in national conferences and publishes many international journal papers. He has organized workshops, state/national project events, seminars in various organizations. His interest of research is in Effective Materialized View Selection and Maintenance process.

Manoj S. Chaudhari was born in Nagpur, Maharashtra in 1982.Currently he is working as an Assistant Professor & Head of Computer Science and Information Technology department of Bhagwati Chaturvedi College of Engineering from last 3 years. His interests of research is in, Artificial Intelligence, Database and Data warehousing & mining