Selection of Views for Materialization Using Size and Query Frequency

T.V. Vijay Kumar and Mohammad Haider

School of Computer and Systems Sciences,
Jawaharlal Nehru University,
New Delhi-110067, India

Abstract. View selection is concerned with selecting a set of views that improves the query response time while fitting within the available space for materialization. The most fundamental view selection algorithm HRUA uses the view size, and ignores the query answering ability of the view, while selecting views for materialization. As a consequence, the view selected may not account for large numbers of queries. This problem is addressed by the proposed algorithm, which aims to select views by considering query frequency along with the size of the view. The proposed algorithm, in each iteration, computes the profit of each view, using the query frequency and size of views, and then selects from amongst them, the most profitable view for materialization. The views so selected would be able to answer a greater number of queries resulting in improvement in the average query response time. Further, experimental based comparison of the proposed algorithm with HRUA showed that the proposed algorithm was able to select views capable of answering significantly greater number of queries at the cost of a slight increase in the total cost of evaluating all the views.

Keywords: Materialized Views, View Selection, Greedy Algorithm.

1 Introduction

Data warehouse stores subject oriented, integrated, time variant and non-volatile data to support processing of analytical queries [7]. These analytical queries, which are long and complex, consume a lot of time when processed against a large data warehouse. Further, the exploratory nature of these analytical queries contributes to high average query response time. This query response time can be reduced by materializing views over a data warehouse[9]. Materialized views contain pre-computed and summarized information, computed from the information stored in the data warehouse. They are significantly smaller in size, when compared with the data warehouse, and can significantly reduce the response time if they contain relevant and required information for answering analytical queries. The selection of such information and storing them as materialized view is referred to as the view selection problem[4]. View selection deals with selecting appropriate set of views that provide answers to most of the future queries. View selection is formally defined in [4] as "Given a database schema D, storage space B, Resource R and a workload of queries Q, choose a set of views V over D to

V.V. Das, G. Thomas, and F. Lumban Gaol (Eds.): AIM 2011, CCIS 147, pp. 150–155, 2011.

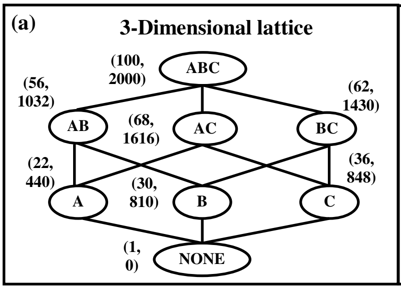
materialize, whose combined size is at most B and resource requirement is at most R". The number of possible views is exponential in the number of dimensions and for higher dimensions it would become infeasible to materialize all views due to space constraints[6]. Further, the space and resource constraint translates the views selection problem into an optimization problem that is NP-Complete[6]. Alternatively, views can be selected empirically, based on past query patterns[12], or heuristically using algorithms that are greedy, evolutionary etc. This paper focuses on greedy based view selection.

Greedy based view selection, in each iteration, select the most beneficial view for materialization. Among the several greedy based algorithms presented in literature [1, 2, 3, 5, 6, 8, 10, 11, 13, 14], the algorithm in [6] is considered the most fundamental one. This algorithm, which hereafter in this paper would be referred to as HRUA, selects the top-T beneficial views, from amongst all possible views, in a multidimensional lattice. HRUA computes the benefit of a view in terms of its cost, which is defined in terms of the size of the view. HRUA computes benefit as given below:

$$\text{BenefitV} = \sum\{(\text{Size}(\text{SMA}(W)) - \text{Size}(V)) \mid V \text{ is an ancestor of view } W \text{ in the lattice and } (\text{Size}(\text{SMA}(W)) - \text{Size}(V)) > 0\}$$

where Size(V) = Size of view V
Size(SMA(V)) = Size of Smallest Materialized Ancestor of view V

HRUA uses size of the view to compute the benefit, It does not consider the number of queries that can be answered by a view, referred to as its query frequency. As a consequence, the views selected using HRUA may not be beneficial with respect to answering most of the future queries. As an example, consider a three dimensional lattice shown in Fig. 1(a). The size of the view in million (M) rows, and the query frequency (QF) of each view, is given alongside the view. Selection of Top-3 views using HRUA is shown in Fig. 1(b).



(b) Selection of Top-3 views using HRUA

View	Size	QF	Benefit		
			1st Iteration	2nd Iteration	3rd Iteration
AB	56	1032	176	-	-
AC	68	1616	128	64	32
BC	62	1468	156	76	38
A	22	440	156	68	48
B	30	810	140	52	32
C	36	848	128	84	-

Fig. 1. Selection of Top-3 views using HRUA

HRUA selects AB, C and A as the Top-3 views. These selected views result in a Total View Evaluation Cost (TVEC) of 492. Considering the query frequency along with the size of each view, the Total Queries Answered (TQA) by the selected views AB, C and A is 3130, from among 6214 queries. An increase in this TQA value would result in more queries being answered by the selected views. The proposed algorithm aims to select Top-T profitable views for materialization that improves the TQA value by considering query frequency along with the size of each view. As a result, the views selected would be able to answer a greater number of queries thereby improving the average query response time. The paper is organized as follows: The proposed algorithm is given in section 2 followed by experimental results in section 3. Section 4 is the conclusion.

2 Proposed Algorithm

Unlike HRUA, the proposed algorithm aims to select views that are not only profitable with respect to size but are also capable of answering greater number of queries. The proposed algorithm, in each iteration, considers the query frequency, along with the size of each view, to select the most profitable view for materialization. The query frequency of each view reflects past trends in querying and is computed as the number of queries, posed in the past, that can be answered by the view. The proposed algorithm, as given in Fig. 2, takes the lattice of views along with the size and query frequency of each view as input and produces the Top-T views as output.

```
Input:   lattice of views L along with size and query frequency of each view
Output: Top-T views
Method:
  Let
    $V_R$ be the root view in the lattice, S(V) be the size of view V,  QF(V) be the query frequency of V in the lattice,
    SMA(V) be the smallest materialized ancestor of V, D(V) be the set of all descendent views of V, MV be the set
    of materialized views, P (V) = Profit of view V, $P_M$ = Maximum Profit, $V_P$ = View with maximum profit
    FOR $V \in L$
            SMA(V) = RootView
    END FOR
    REPEAT
            $P_M = 0$
            FOR each view $V \in (L - V_R \cup MV)$
                        $V_P = V$
                        $P(V) = 0$
                        FOR  each view $W \in D(V)$ and  $(S(SMA(W)) - S(V)) > 0$
                                    $P(V) = P(V) + \left|\frac{QF(SMA(W)) - QF(V)}{S(SMA(W)) - S(V)}\right|$
                        END FOR
                        IF  $P_M < P(V)$
                                    $P_M = P(V)$
                                    $V_P = V$
                        END IF
            END FOR
            $MV = MV \cup \{V_P\}$
            FOR $W \in D(V_P)$
                        IF $S(SMA(W)) > S(V_P)$
            $SMA(W) = V_P$
                        END IF
            END FOR
        Until |MV| < T
        Return MV
```

Fig. 2. Proposed Algorithm

The proposed algorithm, in each iteration, computes the profit of each view P(V) as given below:

$$P(V) = \sum \left\{ \left| \frac{QF(SMA(W)) - QF(V)}{S(SMA(W)) - S(V)} \right| \text{V is an ancestor of view W in the lattice and } (S(SMA(W)) - S(V)) > 0 \right\}$$

The profit of a view V is computed as the product of the number of dependents of V and the ratio of frequency difference between V and its smallest materialized ancestor and the size difference between V and its smallest materialized ancestor. The proposed algorithm (PA), in each iteration, computes profit of the as yet unselected views and selects, from amongst them, the most profitable view for materialization. The selection continues in this manner until T views are selected.

Let us consider the selection of the Top-3 views from the multidimensional lattice in Fig. 1(a) using PA. The selection of Top-3 views is given in Fig. 3.

View	Size	QF	Profit		
			1st Iteration	2nd Iteration	3rd Iteration
AB	56	1032	88	-	-
AC	68	1616	48	24	24
BC	62	1468	56	28	28
A	22	440	40	35	-
B	30	810	34	17	9
C	36	848	36	27	18

Fig. 3. Selection of Top-3 views using PA

PA selects AB, A and BC as the Top-3 views. The views selected using PA has a TVEC of 480, which is less than TVEC of 492 due to views selected using HRUA. Also, the views selected using PA have a comparatively higher value of TQA of 4598 against the TQA of 3130 due to views selected using HRUA. Thus, it can be said that PA, in comparison to HRUA, is capable of selecting views that account for a greater number of queries at a lower total cost of evaluating all the views.

In order to compare the performance of PA with respect to HRUA, both the algorithms were implemented and run on data sets with varying dimensions. The experimental based comparisons of PA and HRUA are given next.

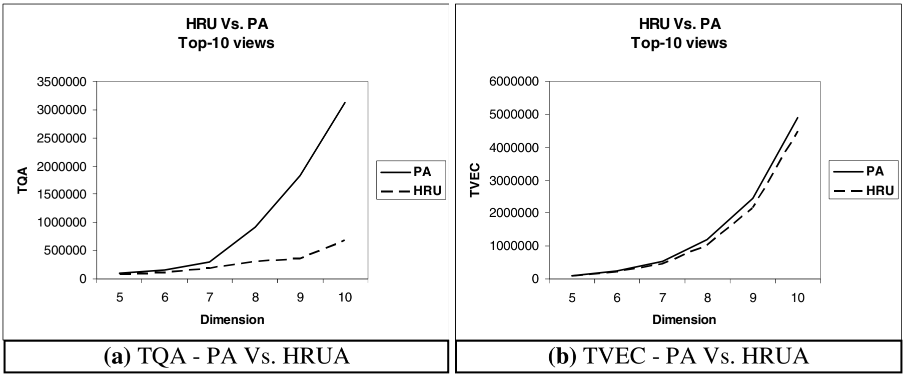
3 Experimental Results

The PA and HRUA algorithms were implemented using JDK 1.6 in Windows-XP environment. The two algorithms were experimentally compared on an Intel based 2 GHz PC having 1 GB RAM. The comparisons were carried out on parameters like TQA and TVEC for selecting the Top-10 views for materialization. The experiments were conducted by varying the number of dimensions of the data set from 5 to 10.

First, graphs were plotted to compare PA and HRUA algorithms on TQA versus number of dimensions. The graphs are shown in Fig. 4(a). It is observed from the graph that the increase in TQA, with respect to number of dimensions, is higher for PA vis-à-vis HRUA.

In order to ascertain the impact of higher TQA on TVEC due to views selected using PA, graphs for TVEC against number of dimensions were plotted and are shown in Fig. 4(b). It is evident from the graph that the TVEC of PA is slightly more than that of HRUA. This small difference shows that the PA selects views which are almost similar in quality to those selected by HRUA.

It can be reasonably inferred from the above that PA trades significant improvement in TQA with a slight increase in TVEC of views selected for materialization.



(a) TQA - PA Vs. HRUA

(b) TVEC - PA Vs. HRUA

Fig. 4. PA Vs. HRUA – (TQA, TVEC) Vs. Dimensions

4 Conclusion

In this paper, an algorithm is proposed that selects Top-T views from a multidimensional lattice using both the size and the query frequency of each view. The proposed algorithm, in each iteration, computes the profit of each view using the size and query frequency of the views and then selects, from amongst them, the most profitable view for materialization. Unlike HRUA, the proposed algorithm is able to select fairly good quality views that are able to account for large number of queries. This would result in improvement in the average query response time.

The experiment based comparison of PA with HRUA on parameters TQA and TVEC showed that PA was found to achieve higher TQA at the cost of a slight increase in the TVEC in respect of views selected for materialization. That is, PA is able to select views capable of answering significantly greater number of queries at the cost of a slight drop in the quality of views selected for materialization.

References

1. Agrawal, S., Chaudhuri, S., Narasayya, V.: Automated Selection of Materialized Views and Indexes in SQL Databases. In: Proceedings of VLDB 2000, pp. 496–505. Morgan Kaufmann Publishers, San Francisco (2000)
2. Aouiche, K., Darmont, J.: Data mining-based materialized view and index selection in data warehouse. Journal of Intelligent Information Systems, Pages, 65–93 (2009)
3. Baralis, E., Paraboschi, S., Teniente, E.: Materialized View Selection in a Multidimensional Database. In: Proceedings of VLDB 1997, pp. 156–165. Morgan Kaufmann Publishers, San Francisco (1997)
4. Chirkova, R., Halevy, A., Suciu, D.: A Formal Perspective on the View Selection Problem. The VLDB Journal 11(3), 216–237 (2002)
5. 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)

6. Harinarayan, V., Rajaraman, A., Ullman, J.: Implementing Data Cubes Efficiently. In: Proceedings of SIGMOD 1996, pp. 205–216. ACM Press, New York (1996)
7. Inmon, W.H.: Building the Data Warehouse, 3rd edn. Wiley Dreamtech, Chichester (2003)
8. Nadeau, T.P., Teorey, T.J.: Achieving scalability in OLAP materialized view selection. In: Proceedings of DOLAP 2002, pp. 28–34. ACM, New York (2002)
9. Roussopoulos, N.: Materialized Views and Data Warehouse. In: 4th Workshop KRDB 1997, Athens, Greece (August 1997)
10. Serna-Encinas, M.T., Hoya-Montano, J.A.: Algorithm for selection of materialized views: based on a costs model. In: Proceeding of Eighth International Conference on Current Trends in Computer Science, pp. 18–24 (2007)
11. Shah, A., Ramachandran, K., Raghavan, V.: A Hybrid Approach for Data Warehouse View Selection. Int. Journal of Data Warehousing and Mining 2(2), 1–37 (2006)
12. Teschke, M., Ulbrich, A.: Using Materialized Views to Speed Up Data Warehousing, Technical Report, IMMD 6, Universität Erlangen-Nümberg (1997)
13. Vijay Kumar, T.V., Ghoshal, A.: A Reduced Lattice Greedy Algorithm for Selecting Materialized Views. CCIS, vol. 31, pp. 6–18. Springer, Heidelberg
14. Vijay Kumar, T.V., Haider, M., Kumar, S.: Proposing Candidate Views for Materialization. CCIS, vol. 54, pp. 89–98. Springer, Heidelberg (2010)