Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment

B.Ashadevi
Assistant Professor, Department of MCA,
Velalar College of Engg. and Technology,
Erode, Tamil Nadu, India.

Dr.R.Balasubramanian
Dean of Basic Sciences,
Velammal Engineering College,
Chennai, Tamil Nadu, India.

Summary
A data warehouse uses multiple materialized views to efficiently process a given set of queries. The materialization of all views is not possible because of the space constraint and maintenance cost constraint. Materialized views selection is one of the crucial decisions in designing a data warehouse for optimal efficiency. Selecting a suitable set of views that minimizes the total cost associated with the materialized views is the key objective of data warehousing. In this paper, we present a framework for selecting views to materialize so as to achieve the best combination of good query response, low query processing cost and low view maintenance cost in a given storage space constraints. The framework takes into account all the cost metrics associated with the materialized views selection, including query execution frequencies, base-relation update frequencies, query access costs, view maintenance costs and the system's storage space constraints. The framework selects the most cost effective views to materialize and thus optimizes the maintenance, storage and query processing cost, thereby resulting in an efficient data warehousing system.

Keywords:
Data Warehousing, Views, Materialization, View Selection, View-Maintenance, Query processing cost, Storage space.

1. Introduction

In recent years, outsized archives of data are available in industry and organization, which results from the accumulation of data. Utilizing these bulk data for decision-making may lead to decisive problems. The rise of new data models and decision support systems provide a way to manage these problems. Warehousing is an emerging technique for retrieval and integration of data from distributed autonomous possibly heterogeneous information sources [1]. A data warehouse is an information base that stores a large volume of extracted and summarized data for On-Line Analytical Processing and Decision Support Systems [2]. Many Industries: manufacturing financial services, transportation, telecommunications, utilities and healthcare, has been effectively set out based on data warehousing technologies.

A data warehouse system utilizes an update-driven approach to collect data from many data sources by means of communicating locally and internationally through networks. A data warehouse system provides a solid platform of consolidated historical data for analysis, and distributes such analysis to users locally and remotely [3]. The intermediate results obtained in the query processing are stored in the data warehouse, to provide effective solution for the queries posted to the data warehouse, which in turn can prevent the users from accessing the original data sources [4]. A data warehouse contains multiple views where a view is a derived relation defined in terms of base (stored) relations. The views stored in the data warehouse are referred to as the materialized views.

Materialized views are physical structures that improve data access time by precomputing intermediary results. At the same time, the use of materialized views requires additional storage space and entails maintenance overhead when refreshing the data warehouse [5]. Data warehouse is capable of answering queries and performing analysis in an efficient and quick manner, in the view of the fact that integrated information is directly available at the warehouse with differences already resolved [1]. Though the data warehouse research community provides effective solutions for the problem of representing data in a form suitable for analytical queries, it has not completely addressed other performance issues such as, query response time for a given aggregated query, view maintenance time, etc.

The most important tribulation in data warehousing is the materialization of views. Materializing all possible views is impractical, which entails large computation and space. Therefore, the key issue in data warehousing is to select appropriate set of views to materialize that hit a steadiness between computational cost and increased query performance, commonly referred as "view selection problem"[6]. In a dynamic environment choosing suitable set of views to materialize is not such an easy task, this includes more factors to be taken into consideration. Several factors affect the choice of materialized views. Therefore selecting the suitable views to materialize has become an imperative issue in warehouse implementation.

The materialized view selection problem is NP-hard [6]. Several methods have been proposed in the literature to

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

address the materialized view selection problem such as [5], [6 -13]. In order to obtain better solutions with respect to view maintenance and query processing costs, evolutionary approaches (genetic algorithms) have recently been proposed [3], [4], [14], [15]. However, the possibility of infeasible solutions creates some problems. The constraint of storage space was dealt within [10]. The constraint of maintenance cost was dealt within [6], [16] and [3]. In [4] a genetic algorithm is proposed with no constraint, in [13] a genetic algorithm is proposed for this problem under the constraint maintenance cost by introducing a function of penalty and in [3] a new genetic algorithm without using function of penalty is presented. In [17], the Genetic Algorithm is further refined by modifying genetic operators and the repairing scheme of infeasible solutions. Most of these approaches make use of AND – DAG for the query representation. The AND-DAG approach which uses tree based structure resulting in computational complexity and more traversal time.

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. The intermediate views in the queries are represented in a simple format rather using AND-DAG, which uses tree based structure resulting in computational complexity and more traversal time. An algorithm is proposed for the selection of views to materialize based on their weightage in the given query set and storage space. Then the query access cost and maintenance cost of selected views are calculated. The total cost of each view is calculated and views with minimum cost under the maintenance and space constraints are selected for materialization.

The rest of the paper is organized as follows; Section 2 presents a brief review of related works in materialized views selection. Section 3 presents the proposed framework for materialized views selection. The experimental results are given in Section 4 and conclusions are summed up in Section 5.

2. Related Works

The problem of finding views to materialize to answer queries has traditionally been studied under the name of view selection. Its original motivation comes up in the context of data warehousing.

Harinarayan et al. [7] 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. [8] 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 [9] developed a greedy algorithm to incorporate the maintenance cost and storage constraint in the selection of data warehouse materialized views. "AND-OR" view graphs were introduced to represent all the possible ways to generate warehouse views such that the best query path can be utilized to optimize query response time.

Amit Shukla et al. [10] proposed a simple and fast heuristic algorithm, PBS, to select aggregates for precomputation. PBS runs several orders of magnitude faster than BPUS, and is fast enough to make the exploration of the time-space tradeoff feasible during system configuration. Himanshu Gupta and Inderpal Singh Mumick [6] 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 special case of OR view graphs.

Chuan Zhang and Jian Yang [15] 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. [11] 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.

Chuan Zhang et al. [4] explored the use of an evolutionary algorithm for materialized view selection based on multiple global processing plans for queries. They have applied a hybrid evolutionary algorithm to solve problems. Minsoo Lee and Joachim Hammer [14] proposed an efficient solution to the maintenance-cost view selection problem using a genetic algorithm for computing a near-optimal set of views used to search for a near optimal solution.

Panos Kalnis et al. [12] proposed the application of randomized search heuristics, namely Iterative Improvement and Simulated Annealing, which select fast a

sub-optimal set of views. The proposed method provided near-optimal solutions in limited time, being robust to data and query skew. Jeffrey Xu Yu et al. [3] proposed a new constrained evolutionary algorithm for the maintenance-cost view-selection problem. Constraints were incorporated into the algorithm through a stochastic ranking procedure. No penalty functions were used.

Ziqiang Wang and Dexian Zhang [17] proposed a modified genetic algorithm for the selection of a set of views for materialization. The proposed algorithm is superior to heuristic algorithm and conventional genetic algorithm in finding optimal solutions. Kamel Aouiche et al. [5] proposed a framework for materialized view selection that exploits a data mining technique (clustering), in order to determine clusters of similar queries. They also proposed a view merging algorithm that builds a set of candidate views, as well as a greedy process for selecting a set of views to materialize.

3. Materialized Views Selection Framework

This section explains the proposed cost effective framework for materialized view selection. The proposed framework exploits all the cost metrics associated with materialized views such as query frequency, query access cost, base-relation update frequency, view maintenance cost and the system's storage space constraints. The materialized view selection problem can be described as follows: Given a set of queries Q and a quantity S (available storage space) and maintenance time MT and existing materialized views Mv, the view selection problem is to select a set of views M to be materialized, that minimize total cost associated with materialized views under storage space and maintenance cost constraints. The storage space constraint is the space, which should not be exceeded by materializing the views. The maintenance cost constraint is the total time, which should not be exceeded while maintaining the materialized views.

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. The intermediate views in the queries are represented in a simple form rather using AND-DAG, which uses tree based structure resulting in computational complexity and more traversal time. An algorithm is proposed for the selection of views to materialize based on their weightage in the given query set and storage space. Then the query access cost and maintenance cost of selected views are calculated. The total cost of each view is calculated and views with optimum cost under the maintenance and space constraints are selected for materialization. The proposed framework is discussed detailed in the following sub-sections.

3.1 Preservation of Existing Materialized Views

This sub-section details the preservation of the existing materialized views. Before selecting new views for materialization, the existing materialized views are sustained based on their access frequency and storage space. The steps for the above process are given in Algorithm1.

Assumptions:

M_V → Vector of Materialized views
N → Total no of materialized views
MS → Memory size of materialized views
$Thres$ → Threshold value
AF → Access frequency of Materialized views

Algorithm 1:

for each Materialized View in M_V
    $W = 2\log(\mathrm{AF}) - \log(\mathrm{MS})$
    if (W < Thres) then
        Remove current Materialized view;
    end if
end for

The above algorithm removes the materialized views with low access frequency and high storage space for the materialization of new views.

3.2 Weightage Based View Selection

This sub-section details the initial selection of views based on their weightage in the given query set and storage space. Instead of selecting all the queries, the queries which have high access frequency are selected for the view selection problem. The queries are selected from the given query set using Algorithm 2.

Assumptions:

Q → Given Set of Queries
Q_{AF} → Access Frequency of Queries
Φ → Threshold value
SQ → Vector of selected queries

Algorithm 2:

for each query in Q
    if $(Q_{AF} > \Phi)$ *then*
        Add query to vector SQ ;
    end if
end for

The queries having access frequency greater than the threshold value Φ are selected for materialized view selection problem. After that the conditional clauses in each query are represented in a simple using Algorithm 3.

Assumptions:

SQ → Selected set of Queries

Q_C → 2D Array of conditional clauses

Q_{SV} → 2D Array of integer values of Q_C

Algorithm 3:

```
for each query in SQ
        if the query has conditional clauses then
                Q_C[i] = conditional Clause(CQ)
        end if
end for
```

Each distinct conditional clause in Q_C is mapped to an integer value and the count of each distinct clause is calculated using Algorithm 4.

Assumptions:

DCC → Distinct conditional clauses

$DCCI$ → Mapped Integer values of distinct conditional clauses

$DCCI_C$ → Count of Distinct Integer values

Algorithm 4:

```
Set Q_SV [0][0] = 1;
Set DCC[0] = Q_C [0];
Set DCCI[0] = Q_C[0];
Set Status = false;
Set count = 0;

for each(i) row in Q_C
  for each (j) conditional clause C_C in row
    for each (k) conditional clause C_C1 in DCC
      if (C_C = C_C1)
        Q_SV[i][j] = DCCI[k];
        Status = true;
      end if
    end for
      if (status = false)
        Q_SV[i][i] = DCCI[SizeofDCCI - 1] + 1;
        DCC[size of DCC] = C_C;
        DCCI [size of DCCI] = DCCI [size of DCCI -1] + 1;
      end if
        status = false;
    end for
end for

for each (i) integer value a in DCCI
      for each row in Q_SV
            for integer value b in row
                  if (a = b)
                  count++;
                  end if
            end for
      end for
      DCCI_C[i] = count;
      count = 0;
end for
```

Then the views are selected based on their weightage in the given query set and storage space using Algorithm 5. Then views with weightage greater than a threshold value α are selected for further process.

Assumptions:

M_U → Vector of Storage space needed to store result of conditional clause

M_{Tot} → Total storage space needed

CC → Count of conditional clause

CC_{Tot} → Total Count

SV → Selected set of views

Algorithm 5:

```
for each conditional clause in DCC
      F1 = CC / CC_Tot;
      F2 = (1 - (M_U / M_Tot);
      W = 2 log(F1) + log(F2);
      If (W > α);
            Add current conditional clause based view
            to SV for further process;
      end if
end for
```

3.3 Query Processing Cost

The cost of query processing is query frequency multiplied by the cost of query access from the materialized views. The query processing cost of each view from SV is calculated using the following formula.

$$QP_{COST} = 1/\sum_{i=1}^{N} Freq * Ca(V)$$

Where N is the total no of queries, Freq is the frequency of query and Ca (V) is the cost of access for query q using view V.

3.4 View Maintenance Cost

View maintenance is the process of updating pre-computed views when the base fact table is updated. The maintenance cost for materialized view is the cost used for refreshing this view whenever a change is made to the base table. The maintenance cost is calculated using update frequency and the priority value of the base table. A priority value in the range 1 – 10 is assigned for each base table based on its importance. The maintenance cost is calculated using Algorithm 6.

Assumptions:

P → Priority of Base tables

UF → Update frequency of Base tables

Algorithm 6:

for each view in SV

for each base table

$$VM_{COST}[i] = 1/(P[i] * (1/UF[i]);$$

end for

end for

3.5 Materialized View Selection

The total cost of each view is calculated by summing the query processing cost and maintenance cost. Then the views are sorted in ascending order based on their total cost.

$$TotCost = QP_{COST} + VM_{COST};$$

Then the views with minimal cost whose maintenance time and storage space falls within the given constraints are selected for materialization.

4. Experimental Results

In this section, we present the results of our experimental analysis. We have implemented all the algorithms of the proposed framework in Java. The Algorithm 1 has successfully removed the existing materialized views with low access frequency and high storage space and thus freed the space for the materialization of new views. The Algorithm 2 has successfully selected the queries with high access frequencies for the view-selection problem. The conditional clauses from each selected query were extracted by Algorithm 3. Then these conditional clauses were represented using a single different representation instead of using AND –DAG graph. The Fig. 1 depicts the sample representation for seven queries.

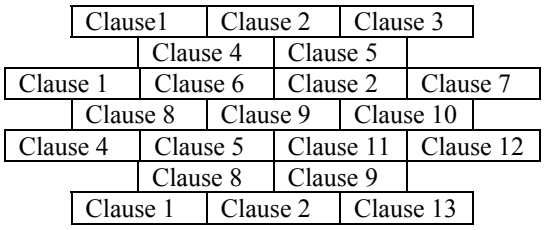

Fig. 1 Simple representation format

From the available views, some views were initially selected based on Algorithm 5. Table 1 depicts the available views and their weightage calculated using Algorithm 5.

Table 1: Available Views and their Weightage

Available Views	Frequency	Storage Space	Weightage
Clause 1	3	100	3.52
Clause 2	3	5	3.87
Clause 3	1	30	5.99
Clause 4	2	25	4.62
Clause 5	2	30	4.60
Clause 8	1	5	6.07
Clause 7	1	15	6.04
Clause 8	2	40	4.57
Clause 9	2	15	4.65
Clause 10	1	20	6.02
Clause 1	1	10	6.05
Clause 12	1	15	6.04
Clause 13	1	15	6.04

We have set the threshold value of α as α = 4.85 for the above case to select views for further process. The Fig. 2 shows the total cost of materializing selected views in different stages of our framework. From Fig. 2 we can conclude that, our framework has significantly reduced the cot from 600 to lesser than 200. Our framework has concluded the list of views with minimum cost for materialization under the storage space constraints and maintenance cost constraints by considering all the cost metrics associated with the materialized views.

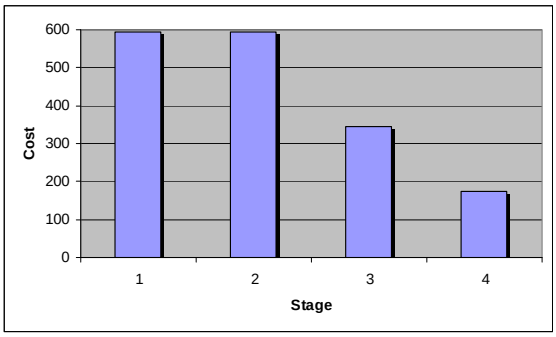

Fig. 2 Total cost of selected views in different stages of framework.

5. Conclusion

The selection of views to materialize is one of the most important issues in designing a data warehouse. The view-selection problem has been addressed in this paper by means of taking into account the essential constraints: maintenance cost and storage space. We have presented a framework for selecting views to materialize so as to achieve the best combination of good query response, low query processing cost and low view maintenance cost in a given storage space constraints. The presented framework considered all the cost metrics associated with materialized views such as query execution frequencies, base-relation update frequencies, query access costs, view maintenance costs and the system's storage space constraints. The most cost effective views have been selected for materialization by the framework and the maintenance, storage and query processing cost of the views have been optimized.

References

[1] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom, "View Maintenance in a Warehousing Environment." In Proceedings of the ACM SIGMOD Conference, San Jose, California, May 1995.

[2] S. Chaudhuri and U. Dayal. "An Overview of Data Warehousing and OLAP Technology". SIGMOD Record, 26(1): 65-74, 1997.

[3] J. X. Yu, X. Yao, C. Choi and G. Gou. Materialized view selection as constrained evolutionary optimization. IEEE Transactions on Systems, Man and Cybernetics, Part C, 33(4): 458–467, 2003.

[4] C. Zhang, X. Yao, and J. Yang. An evolutionary Approach to Materialized View Selection in a Data Warehouse Environment. IEEE Transactions on Systems, Man and Cybernetics, vol. 31, no.3, pp. 282–293, 2001.

[5] K. Aouiche, P. Jouve, and J. Darmont. Clustering-based materialized view selection in data warehouses. In ADBIS'06, volume 4152 of LNCS, pages 81–95, 2006.

[6] H. Gupta, I.S. Mumick, Selection of views to materialize under a maintenance cost constraint. In Proc. 7th International Conference on Database Theory (ICDT'99), Jerusalem, Israel, pp. 453–470, 1999.

[7] 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.

[8] J.Yang, K. Karlapalem, and Q. Li. "A framework for designing materialized views in data warehousing environment". Proceedings of 17th IEEE International conference on Distributed Computing Systems, Maryland, U.S.A., May 1997.

[9] H. Gupta. "Selection of Views to Materialize in a Data Warehouse". Proceedings of International Conference on Database Theory, Athens, Greece 1997.

[10] A. Shukla, P. Deshpande, and J. F. Naughton, "Materialized view selection for multidimensional datasets," in Proc. 24th Int. Conf. Very Large Data Bases, 1998, pp. 488–499.

[11] S. Agrawal, S. Chaudhuri, and V. Narasayya, "Automated Selection of Materialized Views and Indexes in SQL Databases," Proceedings of International Conference on Very Large Database Systems, 2000.

[12] P. Kalnis, N. Mamoulis, and D. Papadias, "View Selection Using Randomized Search," Data and Knowledge Eng., vol. 42, no. 1, 2002.

[13] 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.

[14] M. Lee and J. Hammer, Speeding up materialized view selection in data warehouses using a randomized algorithm, International Journal of Cooperative Information Systems, 10(3): 327–353, 2001.

[15] C. Zhang and J. Yang, "Genetic algorithm for materialized view selection in data warehouse environments," Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, LNCS, vol. 1676, pp. 116–125, 1999.

[16] C. -H. Choi, J. X. Yu, and G. Gou, "What difference heuristics make: Maintenance-cost view-selection revisited," in Proc. Third Int. Conf. Web-Age Information Management, 2002.

[17] Ziqiang Wang and Dexian Zhang, Optimal Genetic View Selection Algorithm Under Space Constraint, International Journal of Information Technology, vol. 11, no. 5, pp. 44 - 51,2005.

[18] Gang Gou; Yu, J.X.; Hongjun Lu., "A* search: an efficient and flexible approach to materialized view selection Systems," IEEE Transactions on Man, and Cybernetics, Part C: Applications and Reviews, Vol. 36, no. 3, May 2006 pp: 411 - 425.

Mrs. B.Ashadevi received B.Sc., MCA from the university of Kamaraj university Madurai, M.Phil from the University of Mother Teresa Womens University in 1997, 2000 and 2004 respectively, where she is currently pursuing the PhD. She was a Lecturer between 2000 and 2006. Currently, she is an Assistant professor in the Department of MCA. Her current research interests include Knowledge and Database Engineering. She has

authored a book on Database Management systems and published paper in international journals.

Dr.R.Balasubramanian received B.Sc (Mathematics) – 1967 at Govt.Arts College, Coimbatore, M.Sc (Mathematics) – 1969 at PSG Arts College, Coimbatore and Ph.D – 1990 at PSG College of Tech. On completion of his M. Sc program he served CIT, Coimbatore for two years as Associate Lecturer. In June 1971 he joined PSG Tech as Associate Lecturer served PSG Tech till November 2000. During his stay at PSG Tech he saw few phases of professional caderes like, Lecturer, Special Temporary Assistant Professor and Assistant Professor. During November, he opted for voluntarily retirement and joined Sri Krishna College of Engineering and Technology as Professor and Head of the Department of Mathematics and Computer Applications. He has published more than 15 research papers in national and international journals. Attended a number of short-term courses and conferences to enrich his knowledge. He has organized a National level conferences and chaired many technical sessions of the mathematical conferences organized elsewhere in the country. He has authored a series of books on Engineering Mathematics and Computer Science. He has supervised one PhD thesis and several M. Sc (App. Math), MCA project works. His interest includes, applied mathematics, partial differential equations, Data structures. He was the principal investigator of UGC sponsored research projects. He is a member of board of studies of Avinashilingam deemed University and chairman board of examinations of several Universities. He is a life member of many professional bodies like ISTE, ISTAM, CSI. His mission is to impart highest quality of mathematics and mould younger generation.