An Effective Cost Approach Technique Using Materialized View for Query Evaluation

P. P. Karde
Department of IT,
HVPM's College of Engg. & Tech, Amravati, India
p_karde@rediffmail.com

Dr. V. M. Thakare
Post Graduate Deptt. of Computer Science, SGB, Amravati University, Amravati, India
vilthakare@yahoo.co.in

Prof. S. P. Deshpande
Associate Professor,
MCA Department, HVPM Madals, Amravati
shrinivasdeshpande68@gmail.com

ABSTRACT

One of the important issues in data warehousing is the selection of a set of views to materialize in order to minimize the cost. Materialized view can provide the massive improvement in query processing and it is the crucial decision in a database for optimal efficiency. Materialized views have been found to be very effective at speeding-up queries and are increasingly being supported by commercial databases or data warehouse systems. However, one encounters the problem of view maintenance if all possible views are materialized in advance. Reducing query time by means of selecting a proper set of materialized view with a lower cost is crucial in databases. In this paper, we hereby propose a cost model for query execution and maintenance along with an efficient view selection algorithm. The main contribution of this paper is to speedup the selection process of materialized view by reducing the total cost of database query. The proposed methodology works well over other view selection methods.

Keywords

Database, Data warehouse, Materialize view selection, Materialize view maintenance, total cost, storage cost, net Benefit, Queries.

1 INTRODUCTION

Information based technologies are being heavily used in almost every industry including public and private sectors. However, the effort to provide effective data analysis for better decision support is quite costly. In an ever changing environment, timely & accurate information is as good as the time window allows. To make effective decisions one needs all related data available to provide timely and appropriate support, therefore it is a big challenge to built, operate and manage such an integrated data store in a cost effective way. Materialized view in data warehouse offers an excellent solution for it. [1,2]. To avoid accessing the original data sources and increase the efficiency of the queries posed to the data warehouse, some intermediate results stored in a data warehouse is called materialized view. [3,4]. A Materialized view provides indirect access to the table data by storing the results of query in a separate object. There are many research issues related to databases among which materialized view selection using an efficient methodology is one of the most challenging ones [4]. Therefore one simple criterion would be to select a set of materialized view which will minimize the total response time of queries. In this paper a methodology is described in which a proper set of views are selected for query processing & computation of cost. In next section previously related work is described in the field of MV selection.

Trade-off in Materialized View

The trade-off depends upon the appropriate selection of materialized view. It gives the best query performance when all views are materialized but it increases the cost to maintain these views.

Obviously, 100% materialization may be infeasible for large databases because it will require an excessive amount of disk space. Also, the time required to materialize a view is considerable. Therefore, 100% view materialization might take a long time to maintain it, which might not be affordable in today's environment.

When no views are materialized in data warehouse, it gives the poorest query performance and decrease the cost of view maintenance. In this case, one needs to access the raw data and answer each query. This approach will result in long retrieval time due to high CPU and disk load. But it does not need any extra storage space for the view materialization.

The third alternative is to materialize only a part of the database. But selecting the right set of views to materialize is the challenge. End users get the optimal solution when only some views are materialized & some are not, which depends on the frequency of query, frequency of updates & the cost of query that provides the optimal balance in data warehouse for query execution.[5,6,7,8,9]. This trade-off is considered in our research during the cost computation of query.

2 LITERATURE REVIEW

Since decade, researchers have been interested in materialized view selection. During the study, several attempts have been made to select the appropriate set of materialized view & maintain it which minimizes the cost of query. The problem of finding views for materialization to answer queries has traditionally been studied under the name of view selection. Its original motivation comes up in the context of data warehousing. It describes the advancements achieved in materialized view selection and also discussed the progress made in view selection problem with various approaches. In spite of the success of the various approaches, they often suffer with some problems like they couldn't acquire a good initial solution, not found a scalable solution and might not be an optimal for the best set of materialized views.

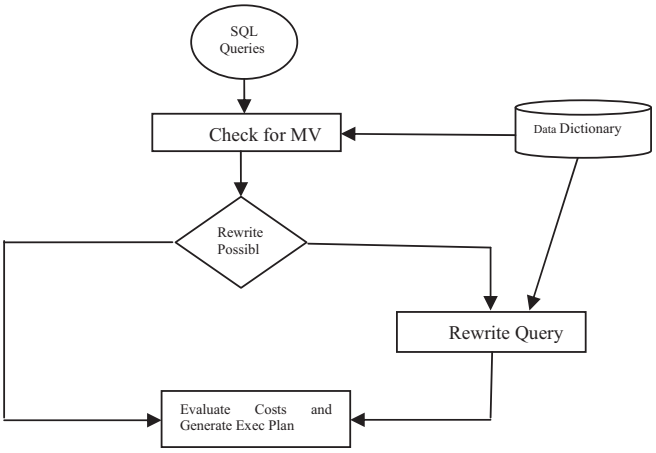
An evolutionary algorithm in [10] for materialized view selection is particularly suited for large and complex problems where little prior knowledge is available. This approach is based on multiple

global processing plans for queries. A hybrid evolutionary algorithm is applied to solve three related problems.

A theoretical framework for the general view selection problem and polynomial-time algorithms for some special cases presents in [11], which lower bound the benefit of the optimal solution. The view selection problem under the maintenance time constraint with two heuristics algorithms is explained in [12]. The key underlying these algorithms is to define good heuristic functions and to reduce the problem to some well-solved optimization problems. [53] again extends the work to address the problem of selecting views to materialize under the constraint of a given amount of total maintenance time. The proposed cost effective framework for materialized view selection explained in [14]. 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 framework sustains existing materialized views periodically by removing views with low access frequency and high storage space. VRDS algorithm presents in [15] based on view relevance to select views in. Due to the space constraint and maintenance cost constraint, the materialization of all views is not possible. Therefore, a subset of views needs to be selected to be materialized. The 0–1 integer programming technique in [16] used to obtain the optimal global processing plan and then a heuristic algorithm was employed to select the materialized views based on this global processing plan. It is worth noting that the optimal global processing plan found in such a way may not lead to the best set of materialized views. A heuristic algorithm proposed in [16], 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 &is used to present the problem formally. An entirely different strategy for implementing a materialized sample view is presented in [17]. The sampling algorithm is an online algorithm, which is used to produce a much larger sample by the structure as time progresses. This uses a new data structure called the ACE Tree to index the records in the sample view. At the highest level, the ACE Tree partitions a data set into a large number of different random samples such that each is a random sample without replacement from one particular range query. Harinarayan et al. [18] presented a greedy algorithm for selection of materialized view so that query evaluation costs can be optimized in the special case of "data cubes". However the cost of view maintenance and storage were not addressed in this piece of work. Himanshu Gupta & Mumick in [19] described the greedy algorithm to compute the maintenance cost and storage constraint in the selection of materialized view . AND-OR view graphs are introduced to represent all the possible ways to generate database views such that the best query path can be utilized to optimize query.

Query Rewrite Mechanism

The query rewrite facility is activated by including ENABLE QUERY REWRITE clause when creating the materialized view. However, query rewrite is only possible where the materialized view is stored is shown in fig. 1.1. The query optimizer automatically recognizes when an existing materialized view can be used to satisfy a request. Next, it transparently rewrites the request to use the materialized view [18, 24]. Queries are then directed to the materialized view and not to the underlying detail tables, resulting in a significant performance gain. There are the options to match the text in query: Full SQL text match, Partial text match & No match [18,25]



3 OVERVIEW OF OUR APPROACH

The materialized view design problem is the problem of selecting a set of views to materialize in the database or data warehouse which answer all queries of interest while minimizing the response time remaining within a given view maintenance cost window. In this research work, proposed approach will be used to select a right set of materialized view & maintained it in data warehouse by reflecting the changes in it. Generally the disk space constraint is reflected in most of the approaches to select the materialized view. But now a day's memory is available in cheap price i.e. becomes in-expensive. Many researchers have applied the heuristics to trim the search space, in order to get the results quickly. In order to avoid exhaustively searching the whole solution space and to obtain a better solution than that obtained by heuristic, therefore, during the selection of view & computation of query we may neglect this parameter to minimize the cost in data warehouse.

This approach appears the necessity for using optimization technique. One of the techniques used is the view materialization. Thus, when a query is introduced, we'll try to answer it by using these views rather than the entire warehouse. To implement this idea, the research work proposes ACE tree based materialized view selection method for query processing to minimize the response time. This algorithm is used for creating and maintaining materialized views using the tree based approach. Initially all records are arranged in ascending order of their key values. Then the middle record is selected as root element of tree. The records are then split till the threshold doesn't reach so that the leaf of tree should contain the number of records that will be available in

materialized view. Then the materialized view is created for each leaf node, indirectly each leaf represents materialized view that has to be created and maintained. The random walk algorithm is used as base for designing the node selection algorithm and gossip protocol is used to find the best set of the nodes.

1.1 Performance Evaluation

For the implementation of the system, Advanced Java and Microsoft SQL Server 2005 are used as development tools. In the initial step it shows the implementation of node selection approach and generates a sample view. The sample views are generated using ACE algorithm. After generating sample views user can fire query using sample views or without using sample views. The appropriate results are showed to user along with the processing time. The required processing time is displayed to user and comparison is carried out.

Initially Database (dataset) is sorted using any one of the field as mentioned in the phase of the tree generation. Then the materialized view is generated as per the number of leaf nodes created during the tree generation. The research work is carried on various available databases. These databases may present over different distributed database servers in the distributed network. For instance consider the BMC database which is considered here for elaboration of the implemented algorithms and methodology.

Algorithm A:

Threshold value- for number of records kept in materialized view: q

Inputs:

Total number of Records in database: N

Visit Nodes: S

Output:

Set of Materialized views : M

Begin

Arrange N in Ascending order of their key values

Select Middle record as a Root (node)

For all the records in databases available on S

If q> number of records in leaf

Split the number of records in equal set

Else create materialized view for the records which are present in leaf node.

Add the materialized view in view set

End

Algorithm B:

Total number of nodes in network : P

Number of nodes to visit : s

Jump size for randomly selecting nodes : k

Maximum tuples to be processed per node :r

Inputs:

Query with selection condition : Q

Node where query is initiated :Sink

Output: Query result to Sink (node where query is initiated)

Begin

Check number of active nodes

If number of nodes = 1

Execute query on that node

Else randomly select the nodes

Return result to Sink

Compute Processing time

Return this result to Sink

End

Cost Computation

The important issue is to select such a set of materialized views in order to minimize the total query processing time of data warehouse queries with a certain constraint. The constraint can be either disk-space constraint or maintenance-cost constraint. The disk-space constraint specifies the availability of the disk-space in a data warehouse, whereas the maintenance-cost constraint specifies how long all views must be updated, because changes to the source data result in recomputing the materialized views accordingly, which will be periodically done in a time window. All the cost metrics associated with the materialized views selection that comprise the query execution frequencies, base-relation update frequencies, query access costs, view maintenance costs can be considered during the implementation. These parameters optimizes the maintenance, storage and query processing cost as it selects the most cost effective views to materialize. Thus, an efficient data warehousing system for materialize view selection can be the best outcome. A cost model is presented to enable the evaluation of query cost, maintenance cost, storage cost and benefit associated with materializing each summary view in data warehouse.

Query Processing Cost

The proposed approach considers query processing cost, view maintenance cost, storage cost, net benefit and storage effectiveness for computing the total cost. The cost is calculated in terms of block size B. The cost of query processing is the frequency multiplied by the cost of query access from the materialized view. The total query cost 'C_{qr}' is given by

$$Total\,(C_{qr}) = \sum_{i=1}^{r} (f_{qi} * C_q(q_i))$$

Where,

f_{qi}' is the frequency of Query and

'$C_q(q_i)$' is cost of access for query 'q' using view 'q_i'.

Maintenance Cost

View maintenance is the process of updating precomputed views when the base fact table is updated. The maintenance cost for materialized view is the cost used for refreshing the view whenever the change is made to the base table. The maintenance cost is computed using the update frequency.

The re-computation of each view 'T_i' requires selection and aggregation from its ancestor view 'T_{ai}', and their joining with 'n' dimension tables T_{d1}, T_{d2},.....,. T_{dn}. Therefore the maintenance cost for the summary view 'T_i' and If there are j views in the data warehouse which are materialized, then the total maintenance cost *'Total (C_m)'* for these materialized views is given by

$$Total\ (C_m) = \sum_{i=1}^{j} (f_{ui} * C_m(T_i))$$

Where,

'f_{ui}' is the Update frequency of summary view 'v_i'.

'$C_m(T_i)$' is the cost of view maintenance in data warehouse.

The total cost of each view is calculated by summing the query processing cost and maintenance cost. Thus the total cost of materializing a view is

$$Total\ (C_{all}) = Total\ (C_{qr}) + Total\ (C_m)$$

The storage factor 'V' represents the estimated ratio of the storage capacity required by the data warehouse to the availability of hard disk space it is given by

$$V = (Total\ (C_{store}) + (1+Y) * T*S_b) / Total\ available\ Storage\ capacity$$

4 RESULTS & DISCUSSION

We experimented with the proposed approach to examine the quality of the solutions. We have implemented the algorithm in Advanced Java and MS-SQL Server 2005. All experiments are run on a P IV based PC's with 1 GB memory running Windows XP.

The experimental results are carried out on different databases like BMC, Northwind, Electricity, Web searches and all words databases are used to carry out the experiments using proposed method where query execution occurs by considering the query frequency, whether view is available or not and the number of records are available with summary view. The minimum time is computed by comparing the execution time and the available databases with & without using summary views.

The execution time is taken between the databases like, Northwind, Electricity, Web searches and all words using the proposed materialized view approach and without using the materialized view & it is identified that proposed methodology provides the flexible solution with minimum cost than without using materialized view, and reduced the total query response from the data warehouse. The research outcome is the creation of summary views alongwith the cost optimization design which will minimize the total cost of computation for query execution. In our approach we will find the total cost is based on the cost of query processing, cost of maintenance & storage cost by applying three different strategies: *All-virtual-views, All-materialized-views and Proposed materialized-views.* The user queries is shown in Table 4.1. This computes the availability of views in databases for the given query, query frequency, number of records.

Table 4.1: User Queries & Number of Record counts

User Queries	Query freq.	Views	Number of Records in view
SELECT SR, DO, AREA, CUSTOMER, EMTBRANCH, PRINCIPAL, MODEL, CNCCONTROL, MACHINESR, DELYON, STARTON, COMMON, COMMANBY, WARRENTYUPTO, REMARKS, TARGETDT FROM BMC ORDER BY DO;	2	BMC View	4387
SELECT DIVISIONSTATE, RESIDENTIAL, COMMERCIAL, INDUSTRIAL, TRANSPORTATION, ALLECTORS FROM ELEPRICEPERUSER ORDER BY ALLSECTORS;	1	ELEPRICE PER USER View	4660
SELECT URL, DATE FROM SEARCHES ORDER BY DATE;	1	SEARCHES View	3000
SELECT PRODUCTID, NAME, DEALER, PURCHASEDATE, QUANTITY, MANUFACTURINGDATE, SOLD, PRODUCTGRPID	1	PRODUCT DETAILS View	5564

FROM PRODUCTDETAILS GROUP BY PRODUCTID;			

The total cost is computed on the basis of query processing, maintenance and storage cost for the three materialized view strategies the *all-virtual-views* method, the *all-materialized-views* method and the *proposed materialized-views* method.

Table 4.2: Total Cost for Three Materialization Strategies

Strategy	Query Processing Cost	Maintenance Cost	Storage cost	Total Cost
All- virtual- views	16230	0	0	16230
All-materialized -views	1026	2689	1135	4850
Proposed-materialized-views	986	2380	380	3746

The total cost computation is also computed individually on each view as per the cost computation strategy described in proposed work & cost model. In the process of cost evaluation, actual cost of query processing (C_{all}), benefit (B_i), storage cost $C(V_i)$, maintenance cost $C_m(T_i)$ and net benefit is computed. The total cost is the actual query cost from the data warehouse. The net benefit and the storage effectiveness can be calculated to determine an optimal set of materialized views.

Analysis

Analysis between the other approaches and find out the execution time over these approaches. We compared our proposed algorithm with Memetic gorithm (HA) and Genetic Algorithm (GA). Table 4.3 reports the running time over 10, 20, 40, 60 and 80 respectively

Table 4.3 Running Time w.r.t Query (database)

Query	MA	GA	HA	Proposed Algorithm
10	1.5 Min	17.3 Min	1.2 Hr.	0.5 Min
20	7.4 Min	30.9 Min	5.3 Hr.	1.4 Min
40	16.8 in	52.4 Min	10.7 Hr.	2.3 Min
60	24.5 Min	1.6 Hour	21.4 Hr.	4.2 Min
80	36.3 Min	2.8 Hr.	35.6 Hr.	6.5 Min

5 SUMMARY & DISCUSSION

In this paper, we presented a new ACE algorithm for selection of proper set of materialized view based on the key element. We have also evaluated our methodology against other algorithms and from the above results, it is found that the proposed methodology works good & at reasonable level. We have computed the cost for three different materialization strategies and found that the proposed methodology can provide significantly better solution. The cost evaluation for various parameters in terms of number of blocks is also computed. Analysis of proposed approach with others is shown in results by comparing the running time & found less time over the others. An experiment is also taken to measure the different execution time for different database queries.

6 REFERENCES

[1] Yang D.L., Haung M.L., Hung M.C., "Efficient Utilization of Materialized Views in a Data warehouse". *International Journal of Information Technology, Vol 7, No.1, 2006.*

[2] Xu Y.J., Yao X., Choi C.H., "Materialized View Selection as Constrained Evolutionary Optimization". *IEEE Transactions on systems, Man & Cybernetics —Part C: Applications and Reviews, Vol. 33, No. 4, November 2003*.

[3] Zhang C., Yao X., and Yang J., "An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment", *IEEE Transaction on Systems, Man, and Cybernetics—Part C: Applications and reviews, vol. 31, No. 3, August 2001.*

[4] Widom J., "Research problems in data warehouse," in *Proc. 4th Int. Conf. Inform. Knowledge Manage. 1995*, pp. 25–30.

[5] Theodoratos D., Xu W., "Constructing Search Spaces for Materialized View Selection". *DOLAP'04, November 12–13, 2004, Washington, DC, USA*.

[6] Agrawal V.R., "Data Warehouse Operational Design: View Selection and Performance Simulation". *Ph.D. dissertation, Computer Science, the University of Toledo May 2005.*

[7] Chaudhuri S., and Narsayya V., "An Efficient, Cost-Driven Index Selection Tool for Microsoft SQL Server". *Proceedings of the 23rd VLDB Conference Athens, Greece, 1997.*

[8] Lee M., Hammer J., "Speeding up materialized view selection in Data Warehouses using a Randomized Algorithm". *International Journal of Cooperative Information System, Vol. 10 No.3, 2001*, pp.327-353.

[9] Oracle8*i* Tuning Release 8.1.5, "A Materialized View Concepts - Creating a Materialized View**".**

[10] Zhang C., Yao X., and Yang J., "An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment", *IEEE Transaction on Systems, Man, and Cybernetics—Part C: Applications and reviews, vol. 31, No. 3, August 2001*

[11] Gupta H,. "Selections of Views to Materialize in a Data Warehouse." *In Proc. of ICDT, pp.98-112, Delphi, January 1997.*

[12] Liang W., Wang H., and Orlowaska M,. "Materialized View Selection Under maintenance time Constraint", *Data & Knowledge Engineering 37(2001,)* pp.203-216

[13] Gupta A., & Mummick I. S. "Selection of Views to Materialize Under a Maintenance Cost Constraint", *In Proc. 7th Int. Conf. Database Theory, 1999*, pp. 453–470.

[14] Ashadevi B., Balasubramanian B. "Optimized Cost Effective Approach for Selection of Materialized Views in Data warehousing". *Journal of Computer Science & Technology Vol. 9 No. 1, April 2009*

[15] Valluri S.R, Vadapalli S., and Karlapalem K. , "View Relevance Driven Materialized View Selection in Data Warehousing environment*," Proceedings of the 13th Australian Database Conference (ADC2002), Melbourne, Australia, vol. 5*, pp. 187-196, *2002.*

[16] Yang J, Karlapalem K & Li Q., "Algorithms for materialized view design in data warehousing environment". *In the proceedings of the 23rd VLDB conference, 1997.*

[17] Joshi S., and Jermaine C., "Materialized Sample Views for Database Approximation". *IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 3, March 2008*

[18] Agrawal V.R., "Data Warehouse Operational Design: View Selection and Performance Simulation". *Ph.D. dissertation, Computer Science, the University of Toledo May 2005*

[19] Savagaonkar A., Limaye G., Nikam N., Kulkarni S., "Project Report on Materialized View Definition and Maintenance". *Indian Institute of Technology, Bombay November 13, 2007.*

[20] Boukra A., Ahmed M. and Bouroubi S., "Selection of views to materialize in data warehouse: A Hybrid solution*". International Journal of Computational Intelligence Research. ISSN 0973-1873 Vol.3, No.4 (2007),* pp. 327–334.

[21] Gorettiv K.Y. chan, Li Quing & Feng Ling, "Optimized Design of Materialized views in real life Data Warehouse Environment". *International journal of Information Technology vol.7 No.1 Sept.200.*

[22] Benjamin Arai, Gautam Das, Dimitrios Gunopulos, and Vana Kalogeraki, "Efficient Approximate Query Processing in Peer-to-Peer Networks," *IEEE Trans on Knowlwgde and Data Engg., Vol. 19, No. 7, Jul 2007.*

[23] Shantanu Joshi and Christopher Jermaine, "Materialized Sample Views for Database Approximation," *IEEE Trans on Knowledge and Data Engg., Vol. 20, No. 3, Mar 2008.*

[24] Olken F., and Rotem D., "Simple Random Sampling from Relational Databases," *Proc. 12th Int'l Conf. Very Large Data Bases (VLDB '86),* pp. 160-169, *1986.*

[25] Lizuan Z., Zhongxiao H., Liu C., "Selecting Materialized view Using Random Algorithm". *Data Network Security 2007, Proceeding vol.6570.*

Author Biographies

Mr. Pravin P.Karde received the Post Graduate Degree (M.E.) in Computer Science & Engineering from S.G.B. Amravati University, Amravati in the year 2006 & pursuing the Ph.D degree in Computer Science & Engineering. Currently he is working as an Assistant Professor & Head in Information Technology Department at H.V.P.M's College of Engineering & Technology, Amravati. His interest is in Selection & Maintenance of Materialized View.

Dr. V.M. Thakare is working as Professor & Head in Computer Science from last 9 years, Faculty of Engineering & Technology, Post Graduate Department of Computer Science, SGB Amravati University, Amravati. He has published 86 papers in various National & International Conferences & 20 papers in various International journals. He is working on various bodies of Universities as a chairman & members. He has guided around 300 more students at M.E / MTech, MCA M.S & M.Phil level. He is a research guide for Ph.D. at S.G.B. Amravati University, Amravati. His interest of research is in Computer Architecture, Artificial Intelligence and Robotics, Database and Data warehousing & mining.

Prof. S.P.Deshpande is currently working as Associate Professor at Post Graduate Department of Computer Science & Technology, MCA at Shree H.V.P.Mandal's Amravati since last 15 years. He has s published 25 papers in various national & International conferences & 5 papers in International journals. He has guided more than 100 students at Post Graduate level. His interest of research is Database management, Data Mining, Web based technologies, Artificial Intelligence.