ABHIYANTRIKI
An International Journal of Engineering & Technology
(A Peer Reviewed & Indexed Journal)
Vol. 3, No. 1 (January, 2016) http://www.aijet.in/ eISSN: 2394-627X

Study of Various Methodologies for Selection of Materialized View

Priti Badar*
Assistant Professor
Department of Information Technology
NHCE College of Engineering, Bangalore, India

Abstract

Fast reply time and correctness are significant factors in the success of any database. In huge databases mostly in distributed database, query reply time plays a vital role as timely access to information and it is the essential condition of successful business application. A data warehouse utilizes multiple materialized views to proficiently process a given set of queries. The main objective of data warehousing is to choose an appropriate set of views that minimizes the total cost associated with the materialized views. The materialization of all views is not achievable because of the space constraint and maintenance cost constraint. Materialized views selection is one of the vital decisions in designing a data warehouse for optimal efficiency. Choosing a suitable set of views that minimizes the total cost related with the materialized views is the main aim of data warehousing. This paper discusses various methodologies for choosing views to materialize so as to get the best grouping of good query reply, low query processing cost and low view maintenance cost in a given storage space constraints. These methodologies take into account all the cost metrics related with the materialized views selection. They pick the most cost effective views to materialize and thus optimize the maintenance, storage and query processing cost, thereby resulting in a proficient data warehousing system.

Keywords*: Clusters, DMVSC, MVPP, OLAP, FPUS.*

****Author for correspondence*** priti_badar@yahoo.co.in

1. Introduction

The growth of data has led to the recent availability of outsized archives of data in business and association. The choice making process is faced by serious problems due to the employment of this volume of data. These problems can be controlled by the mounting new data models and decision support systems. Warehousing is a promising technique that regains the data from distributed autonomous probably heterogeneous data sources and integrates the retrieved data. The data warehousing technologies is the source for the efficient embarking of many industries, for instance, transportation, manufacturing financial services, telecommunications, utilities and healthcare. In order to gather information from many data sources, a data warehouse uses an update-driven method that converses through networks both locally and internationally. In order to offer efficient answer for the queries posted to the data warehouse, the midway results obtained in the query processing are collected in the data warehouse. Data warehouse is projected to offer decision support services for huge volume data. So, how to rapidly reply to query request is certainly a great challenge in data warehouse. A view is a resultant relation defined in terms of stored relations. A data warehouse holds multiple views and it is referred the materialized views as the views accumulated in the data warehouse. The materialization of views is the most significant ordeal in data warehousing. It is impractical to materialize all possible views as large computation and space is necessitated. Consequently, the key concern in data warehousing is the "view selection problem" that contracts with the selection of appropriate set of views to materialize that hits a stability among computational cost and increased query performance. The selection of the materialized views is exaggerated by numerous factors. Thus, the procedure of selecting the suitable views to materialize in warehouse implementation is a serious subject. Materialized view performs query through pre-calculation view and stores in the form of table. When OLAP inquiries arrive, the search results can be directly obtained from the materialized views, thus avoiding the difficult and integrated operation from the underlying data, thereby effectively raising query response speed. So, materialized view is an efficient way to develop the performance of multi-dimension analysis. The aim for materialized view selection is to select a group of suitable materialized view in the space constraint, and finally data warehouse has better query reply performance.

The paper discusses on an outline for selecting views to materialize in order to get improved query response in low time by the decline of the total cost involved with the materialized views. All the cost metrics related with materialized views like the query execution frequency, query access cost, base-relation update frequency, view maintenance cost and the system's storage space constraints are employed by the discussed outline. An algorithm is discussed for selecting the views to materialize on basis of their weightage in the provided query set. The paper also discusses clustering-based dynamic materialized view selection algorithm. It initially clusters materialized views, and then dynamically regulates materialized view set.

2. Literature Review

The difficulty of finding views to materialize to reply queries has been discussed under the name of view selection. Harinarayan et al. [5] suggested a greedy algorithm for the selection of materialized views so that query evaluation costs can be optimized in the special case of "data cubes". Yang et al. [6] proposed a heuristic algorithm which utilizes a Multiple View Processing Plan (MVPP) to get an optimal materialized view selection, such that the best combination of fine performance and low maintenance cost can be obtained. Gupta and Mumick [7] developed a greedy algorithm to include the maintenance cost and storage constraint in the selection of data warehouse materialized views. Shukla et al. [8] presented a simple and fast heuristic algorithm, PBS, to choose aggregates for pre-computation. PBS runs numerous orders of magnitude faster than BPUS, and is fast enough to make the exploration of the time-space tradeoff feasible during system configuration. Gupta and Mumick [4] developed algorithms to pick a set of views to materialize in a data warehouse in order to reduce the total query response time under the constraint of a given total view maintenance time. Zhang and Yang [12] proposed a completely different approach, Genetic Algorithm, to select materialized views and exhibit that it is practical and effective compared with heuristic approaches. Zhang et al. [2] explored the utility of an evolutionary algorithm for materialized view selection based on multiple global processing plans for queries. Lee and Hammer [11] proposed a proficient solution to the maintenance cost view selection problem using a genetic algorithm for computing a near optimal set of views used to explore for a near optimal solution. Yu et al. [1] proposed a new constrained evolutionary algorithm for the maintenance-cost view-selection problem. Aouiche et al. [3] presented a framework for materialized view selection that utilizes a data mining technique (clustering), in order to determine clusters of similar queries. They also proposed a view merging algorithm that constructs a set of candidate views, as well as a greedy process for selecting a set of views to materialize.

In order to enhance query response performance in data warehouse, Tan Hong Xing presented dynamic materialized view selection algorithm, FPUS algorithm, which is based on query frequency in unit space. It does not need knowing distribution of query, uniform distribution under the premise. However, it dynamically regulates materialized view according to the collection of query. Zhang BL proposes DCO algorithm. The immediate adjustment strategy of these dynamic selection algorithms enhances greatly query response performance. However, these dynamic materialized view selection algorithms usually replace low gain materialized view with a new high gain alternative view according to certain evaluation criteria.

3. Outline for Selection of Materialized Views

This section discusses the cost effective outline for materialized view selection. The discussed outline uses all the cost metrics related 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 explained as follows:

Given a set of queries R and a quantity Q (existing storage space) and maintenance time TM and available materialized views MV, the view selection problem is to choose a set of views V to be materialized, that diminish total cost coupled with materialized views under storage space and maintenance cost constraints. The storage space constraint is the space which should not be surpassed by materializing the views. The maintenance cost constraint is the total time which should not be exceeded while maintaining the materialized views. The outline maintains existing materialized views periodically by replacing views with low access frequency and high storage space. The queries with high access frequencies are chosen for the view selection problem. Then the query access cost and maintenance cost of selected views are computed. 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 *Selected Set of View is* calculated. The total cost of each view is computed and views with optimum cost under the maintenance and space constraints are chosen for materialization. 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. Then the views with minimal cost whose maintenance time and storage space covers within the given constraints are chosen for materialization. Exploration of the time-space trade off feasible during system configuration.

4. Dynamic Materialized View Selection Algorithm based on Clustering (DMVSC)

a) Definition

In a data warehouse, a materialized view relates to a SQL statement in fact, in other way materialized view corresponds to the result of SQL statement execution. So, the concern that materialized view can be categorized as a class can be transformed into a judgment that the corresponding SQL statement similarity is higher than the defined threshold. Hence, clustering materialized view is transformed to clustering corresponding SQL statement. In order to make the concern simple and not miss the generality, this section resolves that query sentence belongs to the SPJ structure, namely select, project and join, without child query sentence. In order to evaluate the similarity between two SQL statements, five criteria are given in following:

(1) Decide whether there is the same or contained base table set.
(2) Decide whether there is the same complete equivalence connectivity condition.
(3) Decide whether there is equal or contained scope equivalence condition.
(4) Decide whether there is equal other kind of equivalence condition.
(5) Decide whether there is the same or contained output column.

The above five criteria are not equally important, whose weightage are different, (1) maximal, (5) minimal, and (2), (3), (4) the same weightage which is between (1) and (5). In the calculation of the similarity of the SQL statement, the contribution of behind condition is less than front condition. This can be well understood, if the base table sets of the two statements are not the similar or contained each other, the results of the two statements will be not too same definitely. So, the weightage of front condition is bigger.

The Query Processing Cost of Query (qQC(q,M))

Let M be a set of materialized view in data warehouse, therefore the query processing cost of query q will be the minimal cost value of receiving result from M.

So, $QC(q,M)=Min\{c(q,v),v \in M\}$.

The Maintenance Cost of Materialized View vMC(v,M)

There are two maintenance strategies, which are incremental updating and recalculation. Therefore if M is a set of materialized view in data warehouse, the maintenance cost is the minimal value of two strategies. *IMC*(*v*,*M*) denotes the incremental updating maintenance cost, and *RMC*(*v*,*M*) denotes the recalculation maintenance cost.

So, $MC(v,M) =min\{IMC(v,M),RMC(v,M)\}$

The Total Cost of Query q SC(q,M)

The total cost of query q is the weighted sum between the query q processing cost and the maintenance cost of view set S which is relevant to the result of query.

b) Discussion on DMVSC

On the very beginning, as to the query qx, DMVSC algorithm decides whether to search the corresponding direct view from materialized view set. If it gets a direct view, it will immediately return the result, otherwise it will recommence. Secondly, the algorithm will compute the similarity between qx and every class of materialized view set. And if the similarity is higher than or equal to the threshold, then qx will be classified into this class; otherwise qx will be considered as a new class. Concerning the above two kinds of situation, it requires to process separately. Under the first situation, when replacing original view with a new view, corresponding category in materialized view set is only needed to update. Thus, it decreases the view replacement scope greatly, which not only reduces the time complexity of algorithm, but also retained the diversity of materialized view set. Under the second situation, it replaces materialized view which has the lowest gain in the whole materialized view set according to traditional view replacement principle. Finally, it enhances overall query response performance of data warehouse.

5. Conclusion

The selection of views to materialize is one of the most significant issues in designing a data warehouse. The view-selection problem has been discussed in this paper by means of taking into account the necessary constraints: maintenance cost and storage space. Paper presents a outline for selecting views to materialize so as to get the best combination of good query response, low query processing cost and low view maintenance cost in a given storage space constraints. The presented outline considered all the cost metrics coupled 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. With the base of proposing materialized view similarity function, the paper also discusses DMVSC algorithm. It firstly clusters materialized views, and then dynamically adjusts materialized view set. So, it makes materialized view set have relatively higher query response performance to a variety of types of query, instead the adjustment of materialized view set tends to make a high response performance to one category query, and a poor response performance to other types query.

References

[1] 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, vol: 33, no: 4, pp: 458–467, 2003.

[2] 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.

[3] 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.

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

[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] 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.

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

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

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

[10] Gupta, H. & Mumick, I., Selection of Views to Materialize in a Data Warehouse. IEEE Transactions on Knowledge and Data Engineering, vol: 17, no: 1, pp: 24-43, 2005.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] B.Ashadevi, R.Balasubramanian, "Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", proc. of the International Journal of Computer Science and Network security Vol. 8, No. 10, pp. 236-242, 2008.

[16] Gupta H, Harinarayan V, Rajaraman A, et al, “Index Selection for OLAP”, *Proceeding of International Conference on Data Engineering*, 1997, pp. 208-219.

[17] Shukla A, Deshpande P M, Naughton J F, “Materialized View Selection for Multidimensional Datasets”, *Proceedings of the 24th VLDB Conference*, 1998, pp. 488-499.

[18] Zhang Bai Li, Sun Zhi Hui, and Sun Xiang, “Preprocessor of Materialized Views Selection”, *Journal of Computer Research and development*, 2004, pp. 1645-1651.

[19] Tan Hong xing, Zhou Long xiang, “Dynamic Selection of Materialized Views of Multi-Dimensional Data”, *Journal of Software*, 2002, pp. 1090-1096.

[20] Zhang BL, Sun ZH, Zhou XY, et al, “A Dynamic Cache Optimized Algorithm of Static Materialized Views”, *Journal of Software*, 2006, pp. 1213-1221.