Volume 2, Issue 7, July 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Cluster Based Approach for Selection of Materialized Views

Yogeshree D. Choudhari[1]
Asst. Professor,
Department of Information Technology
K.D.K. College of Engineering, Nagpur, India

Dr. S. K. Shrivastava[2]
Director,
SBITM College of Engineering
Betul, India

ABSTRACT: ***A data warehouse (DW) is a database used for reporting and decision support services of an organization. A thousand of queries are fired in every second to actual database or data warehouse and replying each query in quick time with accuracy is a great concern. Queries to DW are critical regarding to their complexity and length. They often access millions of tuples, and involve joins between relations and aggregations. Materialized views are able to provide the better performance for DW queries. A novel algorithm is proposed for selection of materialized view using query clustering strategy that reduces the execution time as compared to response time for actual database.***

Keywords: Materialized view, clustering, Access Frequency, Threshold, %threshold

1. INTRODUCTION

Data warehouse can be considered as a repository of an organization's electronically and systematically stored data. Data warehouses are designed to facilitate reporting and analysis of data, focuses on data storage. The data warehouse is intended to provide decisions support services for large volumes of data. So how to rapidly respond to query request is a great challenge in data warehouse. Quick response time and accuracy are important factors in the success of any database. Many researches have been focused on the selection of MV for quick response time and low maintenance cost. The MV view creation and selection is based on the various parameters like access frequency, base update frequency etc.

1.1 MATERIALIZED VIEW:

A Materialized View (MV) is the pre-calculated (materialized) result of a query. Unlike a simple VIEW the result of a Materialized View is stored in a table. Materialized Views are used when immediate response is needed and the query where the Materialized View bases on would take too long to produce a result. Materialized Views have to be refreshed for updating it once in a while. It depends on the requirements how often a Materialized View is refreshed and how actual its content is. Basically a Materialized View can be refreshed immediately or deferred; it can be refreshed fully or to a certain point in time.
A MV is computed for SQL query since in a data warehouse, a materialized view relates to a SQL statement, that is materialized view corresponds to the result of SQL statement execution. [1]

If the same query is requested frequently with same rows and columns, the query has to be visited the database repeatedly for result increasing the response time. This paper proposes an approach of grouping in broader sense clustering the similar queries depending on certain parameters like access frequency to find the result from MV. The proposed work explores the area of query clustering for the selection of materialized view to decrease the response time and storage space.

2. RELATED WORK

A very few research work has been done about selection of materialized view using clustering approach. A significant work about dynamic clustering of Materialized view is done by [1]. It firstly clusters materialized views, and then dynamically adjusts materialized view set and eliminates the jitter which dynamic materialized view selection algorithm generally has. A heuristics algorithm is developed in [2] that can provide a feasible solution based on individual optimal query plans. It also maps the materialized view design problem as 0-1 integer programming problem. [3] in this paper, a framework for materialized view selection is proposed that exploits a data mining technique (clustering), in order to determine clusters of similar queries. It also proposes 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. An automatic strategy for the selection of XML materialized views that exploit a data mining technique, more precisely the clustering of the query workload is explained in [4]. To validate this strategy, it implemented an XML warehouse modeled along the XCube specifications.

Paper [5] proposes a greedy algorithm BPUS based on the lattice model. And paper [6] discusses the issue of materialized view selection with the B-tree index. Paper [7] proposes PBS algorithm which make the size of materialized view as selection criteria. Paper [8] proposes preprocessor of materialized view selection, which reduces the search space cost and time complexity of static materialized view selection

algorithm. These algorithms are based on the known distribution of query, or uniform distribution under the premise, which essentially are static algorithms. However, the query is random in actual OLAP system, so materialized view set which static algorithm generates cannot maximally enhance the query response performance in datawarehouse.In order to improve further query response performance in data warehouse, paper [9] proposes dynamic materialized view selection algorithm, FPUS algorithm, which is based on query frequency in unit space. It does not require knowing distribution of query, uniform distribution under the premise neither. However, it dynamically adjusts materialized view according to the collection of query. Paper [10] proposes DCO algorithm. The immediate adjustment strategy of these dynamic selection algorithms improves greatly query response performance.

3. COMPOSITION OF PAPER

The paper is divided into 3 parts: The first part consist of proposed methodology that will describe the problem definition and proposed algorithm, second part is the implementation of the proposed algorithm and third part consist of conclusions.

3.1 PROPOSED METHODOLOGY:

Queries to DW are critical regarding to their complexity and length. They often access millions of tuples, and involve joins between relations and aggregations. Materialized views are able to provide the better performance for DW queries. However, these views have maintenance cost, so materialization of all views is not possible. An important challenge of DW environment is materialized view selection because we have to realize the trade-off between performance and view maintenance.

To solve the problem, a clustering method is suggested in which similar queries will be clustered according to their query access frequency to select the materialized views that will reduce the execution time and storage space. When the query is posed, it will be compared with already clustered or existing query, and the precomputed MV will be returned as a result which will reduce the execution time of the query.

In this approach, a framework is created which will reduce the execution time of query when posed to this framework.

This execution time of the query with framework will be compared with execution time of the query if it is posed for original database (without framework).

3.2. PROPOSED CBFSMV ALGORITHM:

A novel framework is developed for the selection of MV using query clustering. The steps of the algorithm are as below.

I) Generation of random set of records for given tables in database by record generator.

II) Extraction or generation of all possible set of queries resolved by system on above created records.

III) Optimization of above set of queries according to their access frequency.

IV) Creation of MV according to query access frequency called as Threshold Value and according to Maximum Cluster Area Threshold %.

This **Maximum Cluster Area threshold %** is calculated using following formula.

Cluster Area:
Area of Table (At) = Ctot*Rtot
Area of Cluster (Ac) = Creq*Rreq

Maximum Cluster Area Threshold %(Ra) =Ac/At

Let Threshold_Area_Ratio = Tar

If Ra <= Tar Then
Create View
Else
Ignore

Where, Ctot: Total column of tables
Rtot: Total Records generated for table
Creq: Columns required by query
Rreq: Records required by query

V) According to above criterion of MV creation, 3 types of MV are created as follows.

- **Single query to Multi table MV.**
  In this response of single query is obtained from multiple MV table.
- **Single query to single table MV.**
  In this response of single query is obtained from single MV table.
- **Multiple queries to single table MV.**
  In this response of multiple similar queries will be obtained from single MV table.

VI) After creation of these 3 different types of MV, we will store these MV.

This creation of materialized view depends on the access frequency and threshold% as calculated above. This algorithm will not generate materialized view for the queries whose cluster area will be greater than threshold%. Thus instead of creating the framework it may directly fetch the data from database.

3.3 PROPOSED ARCHITECTURE OF FRAMEWORK FOR MV SELECTION:

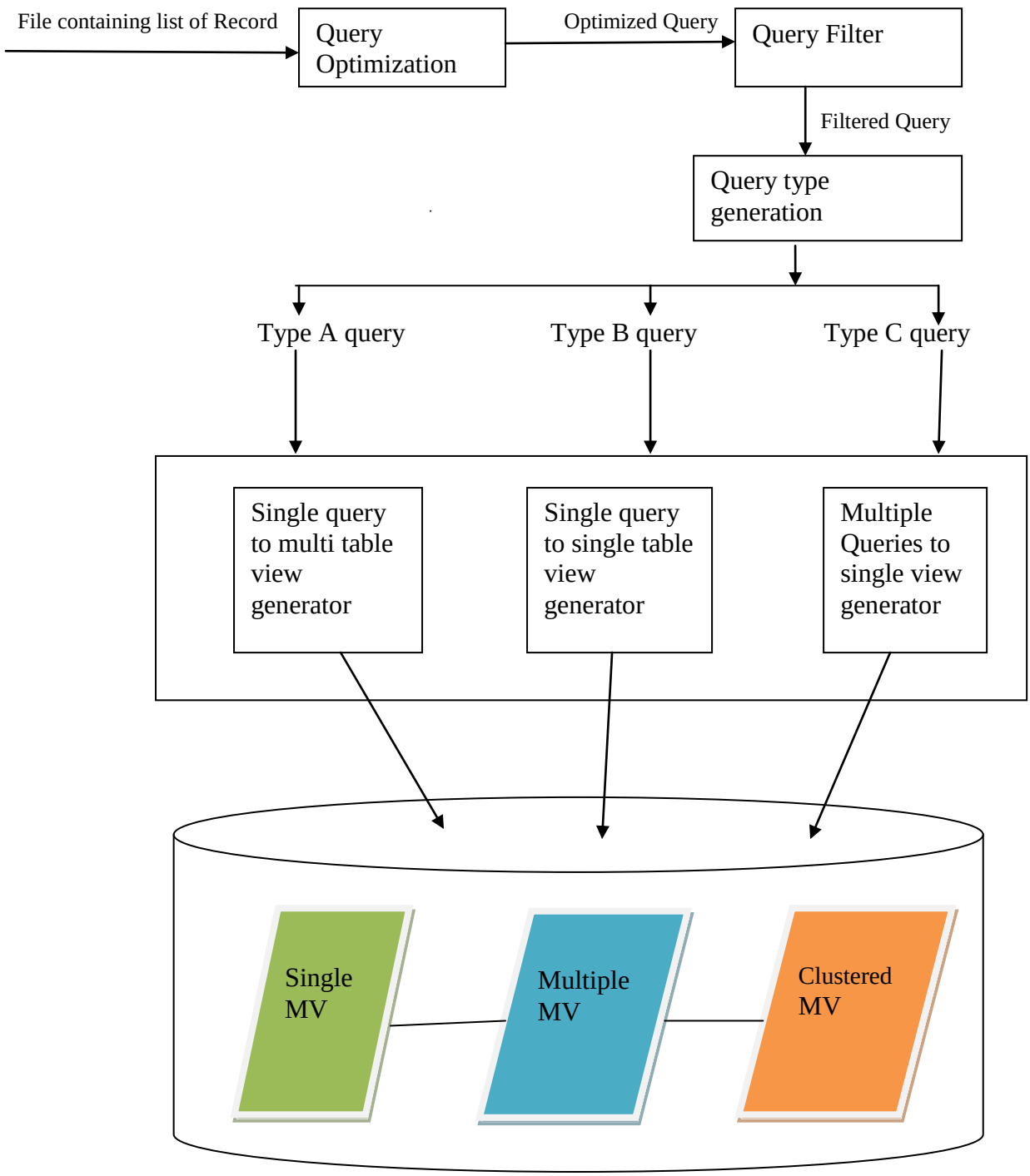

Fig 3.1:Proposed Architecture of Materialized View Selection Framework

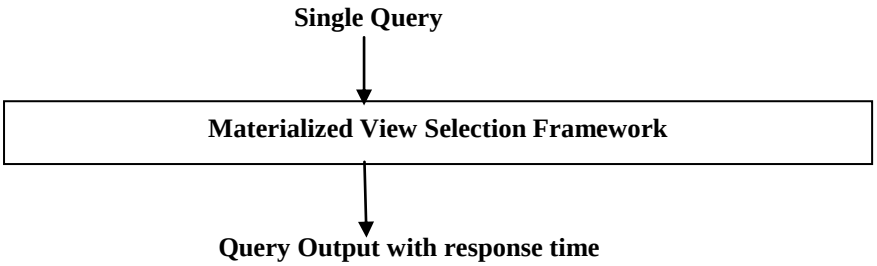

Fig3.2: Query output with Framework with Response Time

4. CONCLUSIONS

Thus the paper proposes algorithm for the materialized view design problem, e.g., how to select the set of views to be materialized so that the cost of processing a set of queries and storage space required storing the data for the materialized views is minimized. This approach realizes on analyzing the queries so as to derive common intermediate results which can be shared among the similar queries to reduce the response time and to eliminate the need for creation of same MV for the query. The proposed algorithm for determining a set of materialized views is based on the idea of reusing temporary results from the execution of the global queries. The cost model takes into consideration of both query access frequencies and % threshold.

The work presented here is the first stage research in selection of queries with high access frequencies, clustering them and creation of Materialized Views for the same. These high access frequency queries are further analyzed for required cluster area to create MV.

REFERENCES

[1] An Gong, Weijing Zhao, "Clustering-based Dynamic Materialized View Selection Algorithm**"** *Proceedings of Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, 2008, China,pp391-395.

2]Jian Yang,Kamlakar Karlapalem,Quing Li,''Algorithms for materialized view design in data warehousing environment."

3] K. Aouiche, P.Emmanuel Jouve, and J.Darmont ,"Clustering-Based Materialized View Selection in Data Warehouses"*TechnicalReport, University of Lyon 2, 2007.*

4] Hadj Mahboubi, Kamel Aouiche and Jérôme Darmont , "Materialized View Selection by Query Clustering in XML Data Warehouses" *Fourth International Conference on Computer Science and Information Technology-Jor dan.*

[5] Harinarayan V, Rajaraman A, Ullman J D, "Implementing Data Cubes Efficiently", *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1996, pp. 205-216.

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

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

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

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

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

[11] Mistry H, Roy P, Sudarshan S, et al, "Materialized view selection and maintenance using multi-query optimization", *Proceedings of SIGMOD'01*, 2001, pp. 307-318.

[12] Shukla A, Deshpande PM, Naughton JF, "Materialized view selection for multidimensional datasets", *Proceedings of the 24th International Conference on VLDB*, Morgan Kaufmann Publishers, San Francisco, 1996, pp. 51.

