Materialized View Selection Algorithm: A Survey

Hema S. Botre  
Smt.Bhagwati Chaturvedi  
College of Engg.Nagpur

Prof. M. S. Chaudhari  
Smt.Bhagwati Chaturvedi  
College of Engg.Nagpur

Abstract: Quick response time and accuracy are important factors in the data warehouse. In distributed database, query response time plays an important role to enhance the query processing. A data warehouse is the collection of multiple materialized views to efficiently process a given set of queries. Materialized views are found useful for fast query processing. Because of the space constraint and maintenance cost constraint, materialization of all views are not possible. Therefore Materialized view selection is one of the important decisions in designing a data warehouse for greater efficiency. Selecting the suitable set of view in order to minimize the total cost associated with the materialized view is the important task in the data warehouse. This work will implement an algorithm which will give the fast query processing and can reduce the space on to the disk.

Introduction:

In distributed environment, a large amount of information exchange into database systems of various organizations. Therefore there should be some provision for organizations to manage such tremendous volume of data. Online analytic processing (OLAP) system provides some ways to take decision from such data. OLAP systems helps in making decisions by firing group-by SQL queries on to the database. Traditional databases are used for online transaction processing (OLTP) applications. Because of OLAP queries are involved in summarizing historical data that have been collected from different sources, it can’t process complex OLAP queries. OLTP applications can access a small number of data from a single local operational database.

To overcome the weakness of traditional databases, data warehouse have been developed. A data warehouse is a huge database system that collects, summarizes, and stores data from multiple remote and heterogeneous information sources. A data warehouse acquired the collection of materialized views, which are pre-computed as well as summarized data from multiple operational sources. To avoid accessing the original data sources, some intermediate results are stored in a Data warehouse. These intermediate results are called materialized views. Materialized views speed up query processing but on the other hand they have to be refreshed when changes occur to the data sources. Hence, there are two costs involved in the selection of materialized view i.e. query processing cost and the materialized view maintenance cost. Focus of this work is what views should be materialized in order to get fast query processing and view maintenance cost should be minimum?

Related Work:

Harinarayan [1] 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. This work proposed an algorithm that pick a good set of queries to materialize which may be able to answer to the other queries. Initially this work has proposed TPC-D benchmark database
that showing why it is important to materialize some part of
data cube but not all of the cube. Second lattice framework
proposed the greedy algorithm which pick the right view to
materialize with respect to various constraint. Finally focused
on hypercube lattice and investigated the time-space trade-off
in detail. The result has shown that the views in some cases
form a memory hierarchy with different access times usually
assigned to different memory space like cache or main
memory. This work has not mentioned view maintenance
and storage cost.

T.Nalini, Dr. A.Kumaravel, Dr.K.Rangarajan [2]
have been proposed I-mine mining techniques from which the
frequently user accessible queries will be generated. Then,
respective set of views can be selected to materialize by
minimizing the total query response time and/or the storage
space along with maximizing the query frequency. This work
has proposed I-Mine index which is a structure provides tight
integration of item set extraction in a relational DBMS. Since
no constraint is matching during the index creation phase, I-
Mine provides a complete form of the original database.
Implementation of this work is based on the FP-tree data
structure, which is very effective in providing a compact
representation of relation R. FP-growth implemented the
construction of a memory structure called FP-tree which
discovers all frequent item sets in a depth-first manner. In this
piece of work first sort the queries in a descending order based
on frequency and at the same time, the queries are sorted in an
ascending order according to the storage cost and query
processing cost. Then selected the top which are satisfying
multiple objectives can be possibly selected. By using I-Mine
indexing technique, frequently accessing queries can easily
find out and can materialized that view only. The
incremental updates of the index is the further extension of
this work which is yet to be satisfied.

MR. P. P. Karde, DR. V. M. Thakare [3] proposed
tree based approach is used for creating and maintaining
materialized views. In the first part of the work all records are
arranged in ascending order with respect to their key values.

Then pick up the middle record as root element of tree. The
records are then split up to the threshold value therefore the
leaf of tree will contain the number of records that will be
available in materialized view. Now Each leaf represents
materialized view that has to be created and maintain. Second
part of the work decides the nodes in the distributed
environment for which materialized view should be created,
and maintained. The random walk algorithm implemented to
design the node selection algorithm and gossip protocol find
out the best set of the nodes. The total cost is calculated on the
basis of query processing, maintenance and storage cost for
three materialized view strategies the all-virtual-views method,
all-materialized-views method and the proposed materialized-
views method. The experimental result shows that all-
virtualviews method requires the highest cost of query
processing as compared to other and view maintenance and
storage costs are not required. The all-materialized-views
method can provide the best query performance but contain
more cost of view maintenance with that this method requires
the minimum query processing cost. Perhaps its maintenance
and storage cost are the highest. The proposed-materialized-
views method contain a minimum query processing cost and
its total cost is also minimized.

Himanshu Gupta and Inderpal Singh Mumick [4]
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 an algorithms for the special case of
AND-OR view graphs. In this work view selection problem
has been done by using AND view graphs. The work has
proposed greedy heuristic for selecting views which contain
maximum benefit per unit space. With this algorithm, this
work has proposed a greedy-Interchange Algorithm then
Inner-Level Greedy Algorithm, Multi-level Greedy Algorithm.
All algorithm has proved that it provide a solution within a
constant factor of optimal.

Ziqiang Wang, Dexian Zhang [5] has proposed
modified genetic algorithm under the space constraint. Each
chromosome is encoded with binary bit string (0 or 1) therefore AND-OR view graph is encode as a binary string. Instead of graphs binary string can directly simplify the implementation of genetic algorithm. In this work AND view graph implemented in binary string, then use breadth-first graph traversal to traverse all nodes and produced an ordered list of all nodes. In this order 0 denotes that corresponding node is not materialized and 1 indicate that node is materialized. The experimental results shows that this algorithm is more faster and accurate that the heuristic and conventional genetic algorithm.

B.Ashadevi, Dr. R. Balasubramanian [6] implemented an algorithm that is projected for choosing the views to materialize on basis of their weight acquired in the query set. This work deals with the preservation of the existing materialized views. If the current view require low access frequency and more storage space then can remove that view and can free up the space for another view to get materialized. It required a computation for calculating an access frequency and storage space.

For every materialized view

\[ W = 2 \log \text{(Access frequency)} - \log \text{(Materialized space)} \]

If \( W < \text{threshold value} \) then

Remove Current MV

End if

End for

In this way can remove current MV and can materialize another view. They have compared optimized CEMS (Cost Effective approach for Materialized view Selection) against CEMS in terms of time. The optimized CEMS consumes less time than the CEMS algorithm.

Sanket Patel and Deepak Dembla [7] have proposed tree based materialized view selection, in which views are selected at the time of query processing and then selects nodes in the distributed environment for the execution of the query.

Conclusion:

The selection of views to materialize is one of the most important issues in designing a data warehouse. So as to achieve the best combination of good query response where query processing cost should be minimized in a given storage space constraints. The space constraint is the most important factor while selecting the views to be materialized. The Proposed approach will implement such an algorithm that will select the materialized views to enhance the query processing and minimize the space & storage constraint.

References:


