Assortment of Materialized View: A Comparative Survey in Data Warehouse Environment

Deepika Kirti
University School of Information and Communication Technology
Guru Gobind Singh Indraprastha University
Dwarka, New Delhi

Jaspreeti Singh
University School of Information and Communication Technology
Guru Gobind Singh Indraprastha University
Dwarka, New Delhi

ABSTRACT
Data warehouse is a repository of large amount of data collected from multiple heterogeneous and distributed data sources. Data warehouse stores lots of data in the form of views, referred as materialized views which provide a base for decision support or OLAP queries. Materialized views store the result of queries which improves the query performance. One of the most important aspect in data warehousing is the selection of materialized views which minimizes the query response time and maintenance cost, given a limited storage space. In this paper, analysis of various approaches of view selection in data warehousing environment is done that have been proposed in the recent past and also provided a comprehensive study of these approaches based on various parameters such as issues addressed, query language supported, comparison to benchmark etc.

Keywords: Data warehouse; materialized view; view selection; benchmark.

1. INTRODUCTION
A data warehouse (DW) is a relational database that is designed for query processing and analysis rather than transaction processing. It usually contains historical data derived from transactional data, but can include data from other sources also. According to W.H. Inmon, “A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process”[1].

Data warehouse stores lots of data in the form of views, referred as materialized views (MV), these views are generated as per user requirement specified in queries [2]. A materialized view is a database object that stores the result of a query and makes them available for future use which is used to answer the query in a data warehouse [6]. Database retrieval of materialized view is just like a cache, which is copy of the data which can be retrieved quickly. However, the use of materialized views requires additional storage space and overhead of view maintenance when refreshing data warehouse [3]. The process of reflecting changes to a materialized view in response to the changes (inserts or update or delete) in the original database is known as View Maintenance that incurs view maintenance cost.

Generally, these are the following choices for materialized views [4]:

- Materializing all views in Data Warehouse: gives best performance at highest cost of maintenance.
- Leaving all the views virtual: gives poorest performance but lowest maintenance cost.

Because of view maintenance cost, it is impossible to make all views materialized under storage space constraints. There is need to select an appropriate set of views to materialize for answering queries, which is called materialized view selection (MVS) [3]. The materialized views are considered the best subset if it results in a minimum query cost as well as satisfies all the constraint. The materialized view selection problem is NP-hard [2].

The proposals about materialized view selection differ in several points [5] which are as follows: the way of determining candidate views; the framework used to capture relationships between candidate views; query optimization; selection of views in relational or multidimensional context; multiple or single query optimization; theoretical or technical solutions. For the purpose of a better understanding of materialized view selection based on the major driving factors of query processing and view maintenance cost, a comparison of the approaches that have been proposed so far has been done.

The paper organization is as follows:

In section 2, highlight and define the view selection problem and provide a brief description of the approaches that have been proposed in the recent past works. Section 3, presents a comparative study on various research works explored in previous section. Lastly, the conclusion is in section 4.
2. STATE-OF-ART

In this section, a brief description of various approaches designed for effective view selection in the recent past is given. From the beginning of data warehouse evolution, the main concern has been problem of view selection.

The most important issue while designing a data warehouse is to identify and store the most appropriate set of materialized view in data warehouse. Materialization of all views is not possible due to memory space and time constraints [11]. The main aim of view selection problem is to minimize either one of the constraint or a cost function (query processing cost and view maintenance cost)[7]. Hence, view selection problem is defined as a process of identifying and selecting a group of materialized views that are most closely associated to user defined requirements in the form of queries in order to minimize the query response time, maintenance cost and query processing time under certain resource constraints. This involves the optimization of two costs included in materialization of views: query processing cost and materialized view maintenance cost.

In [6] a framework is presented that automatically selects the materialized views and indexes for SQL databases. It has been implemented for performance tuning in SQL Server 2000.

Authors in [3] explore the use of an evolutionary algorithm for materialized views selection based on multiple global processing plans for queries. Also hybrid evolutionary algorithm is applied to solve three related problems i.e. query optimization, choose best global processing plan from multiple global processing plan and select materialized view from given global processing plan.

A VRDS algorithm [7] is developed for selecting views in data warehousing environment and also developed a framework based on view relevance.

In [2] AND /OR graph based approach is developed which handle the view selection problem in data cubes present in data warehouse by taking an example of TPC-D benchmark database. Authors have also proposed an optimization algorithm to select certain views.

Another graph based approach has been discussed in [8] in order to select a set of views for special cases under disk space and maintenance cost constraints. AND view graph have been discussed to evaluate the global plan for queries and OR view graphs focus on data cubes. Authors have proposed greedy heuristic based algorithm to handle the same.

In [36] authors developed a competitive A* algorithm for selection of materialized views under disk space constraint. In this paper, authors defined a new H-function, which guarantees to find the optimal solution, NIBS order is proposed to insert views during A* search, developed two powerful pruning techniques and two novel sliding techniques. And in [9] author revisit the problem of materialized view selection under disk space constraint. A new competitive A* algorithm is proposed and shows experimentally that new algorithm is powerful and efficient, and flexible approach.

The authors in [10] have used clustering techniques for computation of reduced tables and then materialized views are computed based on these reduced tables . Also presented a clustering based algorithm named as ASVMRT.

An approach to solve the issue of balancing the trade-off between performance and view maintenance [11]. The Authors have presented a two phase optimization technique (2PO) which is combination of simulated annealing (SA) and iterative improvement (II), using multiple view processing plan (MVPP).

The authors in [4] focus on reducing the cost of views refreshment on the basis of greedy algorithm and dynamic selection problem. In contrast to greedy algorithm, the application of views refreshment can be more suitable to queries. In contrast to dynamic selection algorithm, the strategy of materializing views based on cache updating can avoid the frequent substitution of views in the materialized view set, which can lower the efficiency.

A new approach for materialized view selection based on parallel simulated annealing (PSA) is presented in [12] that selects views from an input MVPP. The PSA algorithm approach generates solutions lesser than the heuristic algorithm. Authors experimentally shows that PSA provides significant improvement in the quality of the obtained set of materialized views as compared to heuristic method and sequential SA.

A MA based algorithm, based on memetic algorithm (MA) is presented in [13]. The memetic algorithm has been successfully applied to several NP-Hard combinatorial optimization problems and efficiency of the algorithm has been confirmed. Experiment result shows that the proposed algorithm performs better than heuristic algorithm and genetic algorithm.

Another approach for fast materialized view selection in distributed environment is node selection algorithm [14]. This algorithm shows that it performs better for query processing as compared to other materialized view selection strategies.

An approach based on metaheuristic cooperation is presented in [15]. This study shows that, the meta heuristic cooperation offers: very good performance ratio with an exact algorithm for reduced problem size; better result than the individual metaheuristics.; scalable approach.

In [16] authors have proposed a framework for candidate view selection using I-Mine algorithm, Index Support for item set mining to mine the frequent queries.

In EMVSIA algorithm [17], authors have implemented dynamic adjustment for static materialized views selection algorithm. The authors have conducted experiments to prove reduction of search space and time consumption.

In [18], author presents two algorithms to generate MVPP: one to generate a feasible solution expeditiously and the other provides an optimal solution by mapping the optimal MVPP generation problem as a 0-1

A method to select materialized view using top-k query algorithm is presented in [19]. The selection is based on the query frequency, view storage space and maintenance cost. The authors have experimentally shown that top-k query algorithm has a better performance than heuristic algorithm for lineage tracing query.
An algorithm of view selection to materialize in specialized data warehouse on the basis of data domain information is developed in [20]. The purpose of the algorithm is to minimize the materialization cost by selecting an optimal set of materialized view constrained by actual space from candidate view subsets.

A framework is proposed in [21] to provide an optimized version of view selection problem which intend to give the best combination of low query processing cost, low view maintenance cost and good query response.

3. COMPARITIVE STUDY

<table>
<thead>
<tr>
<th>Features</th>
<th>Authors</th>
<th>Issues Addressed</th>
<th>Proposed Work</th>
<th>Query Language Supported</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Tools Supported/ Implementation</th>
<th>Comparision to Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Jogekar &amp; ashish Mohd. (2013) [22]</td>
<td>Query response time+ query processing cost + view maintenance cost + storage constraint</td>
<td>MVS framework + algorithm preservation of existing Materialized view + materialized view maintenance</td>
<td>Not addressed</td>
<td>Finer query response time, reduced total cost associate with materialized view</td>
<td>Removes Low access frequency materialized views</td>
<td>Not addressed</td>
<td>Not done</td>
</tr>
<tr>
<td></td>
<td>Suchyukorn &amp; Auepanwiriyakul (2013) [23]</td>
<td>MVPP + Common sub expression for Materialized View Selection</td>
<td>Algorithm for Re-optimization improvement + cost model for MVS</td>
<td>SQL based</td>
<td>Total query processing cost of MVPP reduced, materialized view maintenance cost reduced</td>
<td>-</td>
<td>TPC-H databases of size 1GB</td>
<td>Done (TPC –H benchmark)</td>
</tr>
<tr>
<td></td>
<td>Nalini, Kumaravel Rangarajan (2011) [16]</td>
<td>Query response time + space constraint + query frequency consideration</td>
<td>Use of I-Mine algorithm + index Support for item set mining</td>
<td>Java, SQL</td>
<td>Can mine frequent queries in less computation time</td>
<td>I-Mine index needs to be rematerialize when transactional database is updated</td>
<td>SQL Server 8</td>
<td>Not done</td>
</tr>
<tr>
<td></td>
<td>Abdelmajid boukra Sadek Bourobi (2011) [15]</td>
<td>Optimization of response time</td>
<td>Approach based on meta-heuristic cooperation</td>
<td>Not addressed</td>
<td>Offer better result than individual metaheuristic c, scalable approach</td>
<td>Deals only with response time</td>
<td>Not addressed</td>
<td>Not done</td>
</tr>
</tbody>
</table>

The analysis of the various research works done in the area of materialized views of data warehouse has been done on the basis of several parameters such as issues addressed, proposed work, query language supported, advantages, disadvantages, tools supported or implementation and comparison to benchmark. The analysis and comparative study presented in this paper summarizes various aspects of materialized views of data warehouse that helps in understanding them in a convenient manner. This tabular comparison is presented in the reverse chronological order to understand the development in this area and shows the evolution for the same.
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Year</th>
<th>Methodology</th>
<th>Algorithm/Optimization</th>
<th>Not Addressed</th>
<th>ed</th>
<th>Depend Upon</th>
<th>Not Addressed</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>KV Badmaeva (2011) [20]</td>
<td></td>
<td>Optimal set of materialized view constrained by actual space from candidate view subset</td>
<td>Algorithm of view selection for materialization</td>
<td>Support the administrator by proactive method of materialized view selection in specialized data warehouse, designing cost reduced</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Karde &amp; Thakare (2010) [14]</td>
<td></td>
<td>Query cost + maintenance cost + storage space</td>
<td>Algorithm for creation and maintenance of views + algorithm for node selection</td>
<td>Query performance improved</td>
<td></td>
<td>d</td>
<td>Only distributed environment highlighted</td>
<td></td>
</tr>
<tr>
<td>Xin Li, Xu Qain, Junlin jiang, ziqiang wang (2010) [24]</td>
<td></td>
<td>Total maintenance cost + query response time</td>
<td>SFL algorithm for Materialized View Selection</td>
<td>Outperforms greedy heuristic algorithm &amp; genetic algorithm in terms of total maintenance cost, find global optimization solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xiangquan Song &amp; Lin Gao (2010) [25]</td>
<td></td>
<td>Sub optimal selection in global search area</td>
<td>Ant colony based algorithm for materialized view selection</td>
<td>Global optimal solution will be found by the feedback of ACA materialized view selection algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seyed Hamid Talebian &amp; Sameem Abdul Kareem (2010) [26]</td>
<td></td>
<td>Disk space constraint</td>
<td>Multi objective view selection problem using lexicographic Genetic algorithm</td>
<td>High degree of optimality as compared to optimal solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ashadevi, Balasubramanian &amp; Navneetham (2010) [21]</td>
<td></td>
<td>Maintenancce, storage of most cost effective views to be materialize</td>
<td>OSVP against OCEMS with aid of time</td>
<td>Only maintenance cost &amp; storage space constraint have been taken into account</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhou Zhang, Xia Sun, Ziqiang wang [13]</td>
<td></td>
<td>Cost effective view selection under storage</td>
<td>MA algorithm for selecting view</td>
<td>Faster computation time + comparison of GA space constraint</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Approach</td>
<td>Approach Details</td>
<td>Implemented Environment</td>
<td>Status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lijuan, Xuebin, Linshuang &amp; Qian (2009) [17]</td>
<td>Search space and time consumption consideration for view selection</td>
<td>EMVSDIA algorithm reduces search space and time consumption</td>
<td>Windows 2003 server, Database platform SQL server 2005</td>
<td>Not done</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dhote &amp; Ali (2009) [8]</td>
<td>Selection of views to minimize query response time</td>
<td>AND/OR DAG + Optimization algorithm SQL based</td>
<td>Does not work well for some cases, works only on lattice</td>
<td>Not addressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gui shang Yin &amp; Xiang Yu, Liandong Lin (2007) [4]</td>
<td>Reduced time of response time</td>
<td>Strategy on materialize view selection based on cache updating</td>
<td>Strategy of ad-hoc adjustment, improves the efficiency of the system, avoid repeating computations</td>
<td>Not addressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phuboobob and auepanwiri yakul (2007) [11]</td>
<td>Addresses the trade off between performanc e and view maintenanc e</td>
<td>Two phase optimization, a combination of PSA and MVPP SQL based</td>
<td>Only query response time factor has been made as the main focus</td>
<td>Done (TPC-H benchmark)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jing Li, Yao wang, qiang liu (2006) [28]</td>
<td>Query frequency + source data</td>
<td>Information content + density based Information content + density based</td>
<td>Efficiency transmission of data Focus only on selecting spatial tuples</td>
<td>Not addressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author(s)</td>
<td>Materialized View Selection</td>
<td>Algorithm</td>
<td>Solution Quality</td>
<td>Storage Space</td>
<td>Maintenance</td>
<td>Discussion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gang gou, Jeffrey Xu Yu &amp; Hongjun Lu (2006)</td>
<td>View selection under disk space constraint</td>
<td>A* algorithm</td>
<td>High solution quality, high running performance, high flexibility</td>
<td>Not addressed</td>
<td>Find optimal solutions only when S is small</td>
<td>Not addressed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yang &amp; Chung (2006)</td>
<td>Attribute value density + clustered tables + selection of views based on clustered/reduced tables</td>
<td>ASVMRT Algorithm for view selection</td>
<td>Faster computation time, Reduced storage space, 1.8 times performance better than conventional algorithms</td>
<td>Maintenance of reduced tables not addressed, updating reduced tables needs attention</td>
<td>In pubs database + ETRI</td>
<td>Not done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gupta &amp; Mumick (2005)</td>
<td>View selection under disk space constraint</td>
<td>AND/OR view graphs + greedy heuristic based algorithm</td>
<td>Optimal solution for special cases + polynomial time heuristics</td>
<td>Approximatio n concerns not addressed + problem in AND view graph not NP-hard</td>
<td>Not addressed</td>
<td>Done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valluri, Vadapalli, Karlapalem (2002)</td>
<td>Minimization of total query processing cost</td>
<td>Framework based on view relevance + VRDS algorithm to minimize processing cost</td>
<td>Better than greedy and MVPP, strikes a balance between query processing and view maintenance cost</td>
<td>Optimality could have been more</td>
<td>Not addressed</td>
<td>Not done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhang, Yao &amp; Jianyang (2001)</td>
<td>Query Optimization + Multiple query optimization + materialized view selection</td>
<td>Combination of heuristic and evolutionary algorithm</td>
<td>Gives better performance than both heuristic and evolutionary algorithm, performed alone</td>
<td>Trade-off between computation time and cost saving has not been answered</td>
<td>SUN OS 5.5 + GAlib + simulation software</td>
<td>Done</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aggarwal, Chaudhari &amp; Narasayya (2000)</td>
<td>Automated view and index selection</td>
<td>Framework for index and view selection + Candidate selection &amp; enumeration technique</td>
<td>Robust tool support + Both indexes &amp; view Selected</td>
<td>Only a part of physical design space addressed</td>
<td>SQL Server 2000</td>
<td>Done (TPC-H Benchmark)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4. CONCLUSION

In this paper, an analysis of different approaches proposed by the research community to deal with selection of materialized views in data warehouse environment has been done. Most of the techniques in materialized view selection mainly focus on reducing the total cost associated with the materialized view i.e. total query processing cost and maintenance cost which improves the query performance and select the views to materialize in a way that query response time also reduced under storage space constraint. These techniques have been examined on various parameters such as issues addressed, query language supported, comparison to benchmark etc. and provided a comparative study in a tabular manner. From the reverse chronology of study work of researchers in this area, it has been observed that as the recent proposals have tried to address the issues which were not resolved by earlier approaches, thereby semantically enriching this area. This
study work summarizes the research work done in the best possible manner.

5. FUTURE WORK
In this paper, comparison of various research work based on several parameters has been provided. As future work, direct this research towards various strategies of view adaptation and synchronization and batch oriented view maintenance strategies. A thorough investigation of methodologies to handle materialized view in highly distributed environments for query processing and analysis seems worth attention.

6. ACKNOWLEDGMENT
My sincere thanks to my honourable guide Ms. Jaspreeti Singh and others who have contributed towards the preparation of the paper.

7. REFERENCES


[31] Alka and Anjana Gosain, “A Comparative Study of Materialised View Selection in Data Warehouse Environment” 5th International Conference on Computational Intelligence and Communication Networks (CICN), 2013.


