© September 2018 | IJIRT | Volume 5 Issue 4 | ISSN: 2349-6002

Materialized View Selection and Preservation on Data centric Environment: A Survey

Kanchan Warkar[1], Ms. Prajakta Bhoyar[2]
[1]*Assistant Professor. , MTech, CSE, BDCOE, Wardha*
[2]*Mtech Student, BDCOE, Wardha*

Abstract- **Data warehouse (DW) can be defined as a huge amount of data accumulated from a wide range of sources within a corporation and often used to guide management decisions. To avoid complex query evaluation based on already existing master table to increase the speed of queries posted to a data warehouse, we can use an already created snapshot results from the query processing stored in the data warehouse called materialized views. Appropriate Materialized views selection is one of the fragile and important decisions in designing a data warehouse for better efficiency and plays a vital role for successful business application. Materialized view creation and preservation are found extremely useful in quick query processing for data centric environment.**

In this paper, our main focus is on a variety of techniques that are implemented in past, recent for the selection and preservation of materialized view. Second, the most serious issues related to maintaining the created materialized view for effective query maintenance strategy are also discussed in this paper.

Index Terms- Data Warehouse, Materialized View, View Selection Problem, Query processing cost, View Maintenance, Access Frequency, Threshold, Fuzzy logic.

1. INTRODUCTION

Database systems of various Organizations therefore there should be efficient provision for Data warehouse (DW). DW is a repository that can bring together selected data from multiple database or other different information sources into a single repository. To avoid accessing from base table to increase the speed of queries posted to a DW, we can use some intermediate results from the query processing stored in the DW called materialized views. Therefore, materialized view selection involved query processing and storage cost along with materialized view maintenance cost. Selecting cost effective views to materialize for the purpose of supporting the decision making efficiently is one of the most crucial decisions in designing data warehouse. Selecting a set of derived views to materialize which minimizes the total query response time & maintenance of the selected views is defined as view selection problem. Thus, to select an appropriate set of view is the major target that reduces the entire query response time and also maintains the selected views is also important. So, the materialized views are designed based on the user's requirements (e.g., frequently used queries, processing & storage cost).

The use of materialized view is definitely beneficial since index structures can be built on materialized views. Consequently, database retrieval to the materialized view is just like a cache, which is copy of the data that can be retrieved quickly. Materialized views are significant for query optimization.

In short, when a view is defined, over the database that normally stores only the query defining the view whereas, in case of a materialized view it is a snapshot or replica of a targeted base table whose contents are dynamically computed and stored. It is cheaper in many cases where the query is complex (e.g., involve many tables and complex calculations) and base tables contain a huge amount of records to compute. Materialized views are important for improving performance in many business applications in the context on data warehouse therefore recently database research community paying attention to the materialized view selection and maintenance process.

In this paper various methodologies that are implemented in past, recent for selection of materialized view are discussed. Section 2, 3 gives brief overview of various materialized view maintenance techniques. Section 4 gives the comparison between all the discussed systems basedon the various parameters that are considered during materialized view selection/maintenance process.

2. RELATED WORK

The problem of finding appropriate views to materialize to answer frequent queries has been studied under the name of Materialized view selection process.

Dr. T.Nalini et al. [1]: proposes an Index-Mining algorithm for the selection of materialized views so that query evaluation costs can be optimized as well as materialized view maintenance and storage was addressed in this piece of work.

Ashadevi, B and Balasubramanian.[2] Proposed framework for selecting views to materialize(i.e., View selection problem), which takes in to account all the cost metrics associated with the materialized views selection, including query execution frequencies, base relation update frequencies, query access costs, view maintenance costs and the system's storage space constraints Selects the most cost effective views to materialize and thus optimizes the maintenance storage, and query processing cost

Himanshu Gupta and Inderpal SinghMumick [3] developed a greedy algorithm to incorporate the maintenance cost and storage constraint in the selection of materialized views for data warehouse.

Yang, J et al.[4] Proposed a heuristics algorithm based on individual optimum query plans. Framework is based on specification of multiple views processing plan (MVPP), which is used to present the problem formally.

Harinarayan et al. [5] proposed a greedy algorithm for the materialized views selection so that query evaluation costs can be optimized in the special case of "data cubes". This paper provides good tradeoffs between the space used and the average time to answer a query. Here, the costs for view maintenance and storage were not addressed in this piece of work.

Amit Shukla et al.[6] proposed a very simple and fast heuristic algorithm, PBS, to select aggregates for pre computation. PBS runs several orders of magnitude faster than BPUS, and is fast enough to make the exploration of the time-space tradeoff feasible during system configuration.

Wang, X et al.[7] View maintenance techniques are classified into four major categories : self maintainable recomputation, not self-maintainable recomputation, self maintainable incremental maintenance and not self maintainable incremental maintenance. Self-maintainable incremental maintenance is significant in terms of both space usage and number of rows accessed.

Materialized View Selection and Cost Model

The problem of selecting an appropriate set of views to materialize is called the materialized view selection problem. There are many general as well as research issues related to DW [2], one of them is materialized view selection. Appropriate materialized views speed up query processing. On the other hand, they need to be refreshed when changes occur to the data sources. Therefore, there are two costs involved in materialized view selection: the query selection cost and materialized view maintenance cost. The main objective of materialized view selection problem is the minimization of a constraint or a cost function. A constraint can be system oriented (space constraint) or user oriented (query response time constraint). Most of the selection approaches are designed for minimization of a cost function. Gupta, H (1997), and Barlis. E. et al. (1997) defined view selection problem and take as input the queries that the data warehouse has to satisfy for an initial or an incremental design.

The basic goal of view selection problem is to find a set of views that minimizes the expected cost of evaluating the queries that are frequently used. While designing a data warehouse, it is extremely important to minimize the cost of answering queries because the data warehouse is very huge. The selection of most favorable collection of views for available storage space and minimum query cost is primarily referred to as the view selection problem. There are huge numbers of the base tables (with schemas in hundreds attributes) from multiple data sources, it would be very difficult to decide which views should be materialized. To solve the view selection problem, mathematical formulation is the required step. In view selection problem, data structures are used to represent the view selection. For this, the following subsections are generally used.

A. Relational Algebra

It is similar to algebra, except it uses relation as value instead of number. It is procedural query language most commonly used for outer join. A set of operations are used to express a query. Each operation takes one or more relations as arguments and produces a new relation as the result. This property makes it easy to compose operations to form a complex query. The fundamental set of Relational Algebra operations are Selection (sigma σ), Union (□), Set-difference (-), Cartesian – product (X), Projection (pi Π), Rename (rho ρ). These fundamental operations are used in the query processing for the query optimization process.

B. Directed Acyclic Graph

In mathematics a directed acyclic graph , is a graph having direction and no directed cycles, which is formed by a collection of vertices and edges having direction, each edge connecting one vertex to another, such that there is no way to start at some vertex V and follow a sequence of edges that eventually loops back to V again. For example, if an edge u<=v indicates that v is a part of u, such a path would indicate that u is a part of itself, which is impossible.

C. AND / OR Graph

A form of graph or tree used in problem solving as well as problem decomposition. The nodes of the graph represent states or goals and their successors are labeled as either AND or OR branches. The AND successors are sub goals that must all be achieved to satisfy the parent goal, while OR branches indicate alternative sub goals, any one of which could satisfy the parent goal. A problem: Find path a-z can be solved by either solving a-z via f or a-z via g. A problem a-z via f can be solved by both the sub problem a-f and f-z and a problem a – z via g can be solved by both the sub problems a-g and gz. Groups of sub problems are joined together by an arc.

3. COMPARATIVE STUDY

We have analyzed the various research works on several parameters and presented their comparison in the table below.

Table1. COMPARISON OF VARIOUS RESEARCH WORKS

Features Author	Technique	Issues Address	Proposed Work	Advantages	Disadvantages	Query Lang. Support	Tool Support/ Implementation
Agrawal,Chaudhari & Narasaya (2000) [8]	View Selection	Automated view and index selection	Framework for index & view selection + Candidate selection & enumeration techniques	Robust tool support + Both indexes & view selected	Only a part of physical design space addressed	SQL based	SQL Server 2000
Gupta & Mumick (2005) [3]	View Selection	View selection under disk space & maintenance cost constraints.	AND/OR view graphs + Greedy heuristics based algorithms	Optimal solution for special cases (AND/OR views) + Polynomial time heuristics	Approximation in view-selection problem not addressed + Problem in AND view graphs not NP-hard + Solution fairly close to optimum	SQL based	Not addressed
Yang & Chung (2006) [9]	View Selection	Attribute value density + Clustered tables + Selection of views based	ASVMRT algorithm for view selection	Faster computation time + Reduced storage space + 1.8 times performance	Maintenance of reduced table not addressed + Updating reduced tables needs attention	SQL based	In Pubs database + ETRI

		on clustered /reduced tables		better than conventional algorithms			
Ashadevi & Balasubramanian (2008) [2]	View Selection	Cost effective view selection under storage space constraints	Framework for selecting views + Algorithm for the same + Cost metrics	All cost metrics considered	Query response time not considered + Threshold value not indicated clearly	Not addressed	Algorithms implemented in JAVA
Elena Baralis, Tania Cerquitelli, and Silvia Chiusano (2009)[10]	View Selection	Cost effective view selection under storage space constraints	i-mine algorithm for selecting views	Faster computation time	More memory space	SQL based	Not mentioned
Qingzhou zhang & xia sun, ziqiang wang (2009) [11]	View Selection	Cost effective view selection under storage space constraints	MA algorithm for selecting views	Faster computation time + Comparison of GA & HA algorithm	Only optimal research	Not addressed	Not mentioned
Karde & Thakare (2010) [12]	View Selection	Query cost, maintenance cost, storage space &	Algorithm for creation and maintenance of views + Algorithm for node selection	Query performance improved	Only distributed environments highlighted	Not mentioned	Not addressed
Almazyd & Siddiqui (2010) [13]	View maintenance	Incremental view maintenance + synchronization between DW and source + lost update notifications	Framework with version store	Synchronization between source and DW + Detection of update notification messages	Process becomes a bit lengthy + large space needed + Version numbers should be handled properly	Not mentioned	Not addressed
T.Nalini & A. Kumaravel (2011) [14]	View selection	Cost effective view selection under storage space constraints	i-mine algorithm (modification)for selecting views +using multiple constraints to reduce storage space	Faster computation time + Reduced storage space	Selection of Threshold value is not calculated	SQL based	Algorithms implemented in JAVA + SQL Server 2008

Dr. T. Nalini, Dr.A. Kumaravel (21012) [1]	View selection & maintenance	Cost effective View selection based on best combination of low storage cost, low query processing cost and high frequency of query + Updation of materialized view using LSI(Latent Semantic Index)	IM-LSI (Itemset Mining using Latent Semantic Index) algorithm.	Faster computation time + Reduced storage space	Selection of Threshold value is not calculated	SQL based	Algorithms implemented in JAVA + SQL Server 2008
Y.D. Choudhari. Dr. S. K. Shrivastava(2012) [15]	View selection	Cost effective view selection under storage space constraints	CBFSMV Algorithm for selection of view	Faster computation time + Reduced storage space	View maintenance problem not addressed	SQL based	Not Addressed
Amit Kumar and T. V. Vijay Kumar(2017)[17]	View Selection	Selection under storage space	a discrete genetic operator based particle swarm optimization (DGPSO) has been used to select Top-K views from a multidimensional lattice.	Computational time is more +Reduce storage space	View preservation and maintenance problem not addressed	Not mentioned	Not Addressed

4. CONCLUSION

The effective materialized view selection and preservation with appropriate maintenance is truly advantageous for quick and accurate query processing. Materialized views stores precomputed data(snapshot), but the proper selection and maintenance of materialize views is a major issue in designing a effective data centric environment. This paper provides the key idea regarding the important materialized view selection, preservation and maintenance parameters that plays a crucial role in selection and preservation of appropriate set of materialized views so that the average cost of processing a set of complex but frequent queries are minimized. The query frequencies, query space, query processing time ,materialized view processing time materialized view frequency are the constraints that are the most important factors while selecting the views to be materialized and preserve.

REFERENCES

[1] Dr.T.Nalini,Dr.A.Kumaravel , Dr.K.Rangarajan, "A Novel Algorithm with IM-LSI Index For Incremental Maintenance of Materialized View" JCS&T Vol. 12 No. 1 April 2012
[2] B.Ashadevi, R.Balasubramanian," Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.10, October 2008
[3] Gupta, H. & Mumick, I., Selection of Views to Materialize in a Data Warehouse. IEEE Transactions on Knowledge and Data Engineering, 17(1), 24-43, 2005.
[4] Yang, J., Karlapalem. K., and Li. Q. (1997). A framework for designing materialized views in a data warehousing environment. Proceedings of the Seventieth IEEE International Conference on Distributed Computing systems, USA, pp:458.
[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] A. Shukla, P. Deshpande, and J. F. Naughton, "Materialized view selection for the multidimensional datasets," in Proc. 24th Int. Conf. Very Large Data Bases, 1998, pp. 488–499.
[7] Wang, X., Gruenwalda. L., and Zhu.G. (2004). A performance analysis of view maintenance techniques for data warehouses. Data warehouse knowledge, pp:1-41.
[8] S. Agrawal, S. Chaudhari, and V. Narasayya, "Automated selection of materialized views and indexes for SQL databases", In Proceedings of 26th International Conference on Very Large Databases, 2000.
[9] J. Yang, and I. Chung, "ASVMRT: Materialized view selection algorithm in data warehouse", In International Journal of Information Processing System, 2006.
[10] Elena Baralis, Tania Cerquitelli, and Silvia Chiusano," I-Mine: Index Support for Item Set Mining" IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, april 2009
[11] Qingzhou Zhang, Xia Sun, Ziqiang Wang," An Efficient Ma-Based Materialized Views Selection Algorithm", 2009 Iita International Conference On Control, Automation And Systems Engineering
[12] Mr. P. P. Karde, Dr. V. M. Thakare. "Selection & Maintenance of Materialized View and It's Application for Fast Query Processing: A Survey". Proceedings of International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010
[13] A. Almazyad and M. Siddiqui, "Incremental view maintenance: an algorithmic approach", In International Journal of Electrical & Computer Sciences, vol. 10, 2010.
[14] T.Nalini, Dr.A.Kumaravel, Dr.K.Rangarajan," An Efficient I-Mine Algorithm For Materialized Views In A Data Warehouse Environment", Ijcsi International Journal Of Computer Science Issues, Vol.8, Issue 5, No 1, September 2011 Issn (Online): 1694-0814
[15] Y.D. Choudhari and Dr. S. K. Shrivastava, "Cluster Based Approach for Selection of Materialized Views", International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 2, Issue 7, July 2012
[16] Ashish Mohod and Manoj Chaudhari "Efficient Algorithms for Materialized View Selection in DataWarehousing Environment" International Journal of Computer Science and Network, Volume 2, Issue 3, June 2013
[17] Amit Kumar and T. V. Vijay Kumar "Materialized view selection using discrete genetic operators based particle swarm optimization" 2017 International Conference on Inventive Systems and Control (ICISC) IEEE explorer 19-20 Jan. 2017
[18] W. Chen J. Zhang H. Chung W. Zhong W. Wu Y. Shi "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems" IEEE Trans. Evolutionary Computation vol. 14 no. 2 pp. 278-300 2010.