International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 23 (2017) pp. 13927-13932
© Research India Publications. http://www.ripublication.com

Approaches for Query Optimization using Materialized Views in Dynamic Distributed Environment: A Review

Abhijeet R. Raipurkar
Research Scholar, Department of Computer Science and Engineering
Shri Ramdeobaba College of Engineering and Management,
Gittikhadan, Katol Road, Nagpur-440013, Maharashtra, India.
Orcid Id: 0000-0003-2327-8358

Manoj B. Chandak
Professor and Head, Department of Computer Science and Engineering,
Shri Ramdeobaba College of Engineering and Management,
Gittikhadan, Katol Road, Nagpur 440013 Maharashtra, India.
Orcid Id: 0000-0002-0103-4224

Abstract

Data warehouse in present scenario contains data from multiple heterogeneous data source. Data sources are dynamic in nature as information schema are continuously evolving and changing. In loosely coupled environment data sources are independent hence updates from various data sources are synchronous.

Frequency of addition, deletion and updation operations on based relation have increased because of dynamic nature of data source. Analyzing query response time in distributed heterogeneous environment consists of communication part to establish a connection with heterogeneous data source and data transmission part to transfer data across distributed sites. Selection and maintenance of views is vital task in order to provide maximum efficiency by cutting down query processing and maintenance cost. Integrating data from distributed dynamic environment is big challenge. Querying data from heterogeneous sources is an important research issues in distributed environment. Materialized Views can be used to precompute and store results in data warehouse. Data warehouses, decision support applications and mobile computing environment widely use materialized views for reducing query response time. In distributed environment materialized views are replicated at multiple sites and concurrent updated are performed at multiple sites whenever underlying base relations are modified. Various conflict resolution methods are employed while replicating views at distributed sites. View maintenance cost is enlarged if all the views are materialized in distributed environment at multiple sites but it gives best performance for user queries. Using materializing views in data warehouse includes challenge of identification of which views to materialize, performance consideration, view adaption, view maintenance. We present various approaches for query optimization on materialized views in distributed dynamic environment proposed by researchers.

In this paper comparative studies on research work of various authors have been surveyed based on important query optimization parameters.

Keywords: Materialized views; view maintenance; data grids; query optimization; distributed environment

INTRODUCTION

The requirement of data in heterogeneous database system must be effective integration with the widespread use of network platform in order to better data sharing and data processing. Data integration and maintenance from distributed dynamic source has become important for various applications. The integrated data is usually stored as materialized views to allow better access, performance, and high availability in data warehouse. Data warehouse (DW) generalizes and consolidated data in multidimensional space. Data warehouse also provide online analytical processing (OLAP) tools for interactive analysis of data. Data warehouse is accessed through queries which generally interact with views which are stored in data warehouse. These views are called as materialized views. While designing data warehouse important decision is to formulate materialized views which are most suitable for OLAP queries. Benefit of materialized view is faster response time and enhanced performance.

The most vital task in data warehousing environment is to keep materialized views up to date by using various view maintenance technique. The most important challenges while designing data warehouse is to select which views to materialize.

If OLAP (on-line analytical processing) queries are run against huge data volume in data warehouse then it may result in unacceptable query performance. OLAP queries are very complex and involve aggregation. OLAP queries are critical as it access million of records and response time is effective measure. In order to improve efficiency of OLAP queries, materialized views can be stored in data warehouse which are pre-computed and helps to speed up queries.

Materialized view selection that minimizes total cost is vital decision while designing data warehouse. Because of independents, heterogeneous nature of data source materialized view maintenance is important decision in order to maintain consistency. Immediate, periodic and on demand update view maintenance strategies are used for view maintenance based on requirement of application.

View selection and view maintenance is big challenge in materialized views which are maintained in real time data warehouse. These materialized views must be maintained in response to actual relation updates in the remote sources

If the base relations are modified then the view (often) becomes inconsistent. Updating the inconsistent view in order to make it consistent is called view maintenance (refreshing). Three are different view maintenance policies which are Immediate view maintenance, deferred view maintenance each of which having its own advantages and disadvantages.

We, therefore, have reviews various approaches for materialized view selection, maintenance in dynamic distributed environment where data source is heterogeneous and schemas are evolving because of dynamic nature of information sources. The paper deals with comparative study of various approaches. It is organized as follows. In section 2 we introduce concept of materialized views. In section 3, we highlight and define the view materialization Section 4 presents study on various research works on view selection and view maintenance. Section 5 gives comparative study of these approaches based on various parameters in tabular format. Lastly, we conclude and suggested future scope in section 6.

MATERIALIZED VIEWS

Materialized views are derived from base relations which are stored as auxiliary views in data warehouse. Consistency and integrity of materialized views are maintained by updating materialized views when base relations are modified. The process of updating materialized view is called as view maintenance with which view maintenance cost is always associated. It is not possible to make all views materialized in data warehouse because of disk space, time and integrity constraints. Appropriate set of views needs to be selected for materialization in order to answer user queries. These queries should have reduced query response time. Problem of selecting proper set of views for materialization is called as view selection problem.

Materialized views are used to speed up queries in dynamic distributed environment. Environment in which schemas are continuously changing or evolving is called as dynamic environment. Computing views from scratch in such environment is very expensive and time consuming. Materialized view improves query performance by pre-computing expensive operations prior to execution and storing results in data warehouse as auxiliary view. The motivation for materializing view in data warehouse is to improve response time for OLAP queries which helps in decision making to users.

Materialized view definition includes number of joins and complex aggregate operators and also it consumes storage space. View freshness is one of the issues to be considered while designing materialized view. View needs to be updated whenever underlying base relations are modified.

Materializing views need corresponding disk storage space, and leads to spatial cost. Query cost, maintenance cost and spatial cost must be considered while selecting views for materialization in data warehouse. In present scenario Materialized views are becoming useful tool to cache data in data warehouse because of dynamic nature of real time data warehouse.

The goal is to select materialized views that minimize query response time and cost of maintaining selected materialized views.

VIEW MATERIALIZATION PROCESS

Query performance can be improved with materialized views but overhead associated with materialized view management can be problem. There is trade-off between enhance query performance and overhead associated with view management.

Materialized view management in data warehouse involves following activities.

- Classifying which views to materialize from available views
- Giving indexes to selected materialized views
- Updating materialized views when bases relations are updated/
- Ensuring refreshment of materialized view indexes when base relations are updated.
- Examining which materialized views have been used
- Deciding how effective each materialized view has been on workload performance
- Computing space being used by materialized views
- Deciding which existing materialized views should be dropped
- Taking back up of old detail and materialized view data that is no longer useful

View Adaption and Synchronization

View adaption means rewriting of views that lead to changes in the original view. This problem is addressed as View Adaption. View Adaption can be done either in incremental fashion or by performing full recompilation. If the structure of base relation changes then view definition needs to be changed accordingly. This technique is View Synchronization.

View Selection

OLAP queries gives best performance with materialized views but selection of materialized views to different levels in real time data warehouse environment must be done carefully. There is trade off among performance, scalability and view maintenance restriction that needs to be considered while selecting views. Performance of OLAP queries can be optimized if pre-computation of materialized view has been done carefully. But limitation of this methodology is less scalability of the system and longest time required for view maintenance.

Right set of views needs to picked up for materialization which is a nontrivial task [4], since a materialized view can be queried in order to calculate other views quickly. It is possible to materialize a view corresponding to an infrequently asked query if it helps to answer many other queries fast. It is important to point out that identifying which views to materialize is equivalent to identifying which of the frequently queries to materialize. Query refers to commonly asked representative users' queries, while view refers to aggregation of data from multiple heterogeneous source in the data warehouse organization

View selection is to choose a set of views to materialize in order to achieve best query performance for a given workload query. View selection is to store the most appropriate set of materialized view so they optimize query processing cost and materialized view maintenance cost. View selection algorithm identifies common sub expressions in queries which serve as candidate for MV. Materialization of all possible views is not recommended due to disk memory space and time constraints. View selectivity, query complexity [5], database size, query performance and update performance are the factors to be considered while selecting views to materialize.

The view selection problem [3] is to choose a set of views to materialize in order to achieve the best query performance for a given query workload.

View Maintenance

Materialized views become inconsistent every time the data of the base relations are altered. In a data warehousing environment, the information sources may continue to change their data after the creation of the DW. Such changes must be detected and propagated to the environment to be incrementally incorporated into the DW.

Comparing conventional database system with data warehousing environment, it is difficult to maintain consistency of materialized view in data warehousing environment. It is necessary to integrate changes made in the data source from other source before storing the data in data warehouse. Insertion, updation and deletion anomalies may happen during this process. Other updates may have happened at the sources, generating inconsistencies in the integrated data

These anomalies arise because of dynamic nature of real time data warehouse. Decoupling between information sources is another reason for such anomalies. These anomalies also arise in environment wherein view updates are performed.

Re-computation of materialized views every time base relation is changed is not recommended due to timing constraint. View Maintenance updates only part of the MV which are affected by changes in the base relation. If views are maintained efficiently then overhead incurred while performing expensive join and aggregate queries are eliminated to larger extent. Incremental view maintenance efficiently computes the rows of the view in response to the changes of base relation by modifying some tupules in existing MV. View maintenance policies are Immediate, periodic and on demand updates.

Challenges in Materialization of Views

- Deciding which set of virtual views to be materialized
- Taking decision for re-computation for every change in base relation.
- Incremental adaption of view from existing materialized views
- Building indexes on materialized views to obtain efficient and fast user queries.
- Determining whether selected materialized views are useful given queries
- Handling materialized view staleness problem
- Checking whether selected materialized views answers aggregate, parametric and recursive queries

LITERATURE SURVEY

In this section, we describe various approaches designed for view maintenance for query optimization in materialized view in the past. Certain limitations are associated with the concept of materialization of views. Because of storage and time constraint materialization of all view is not suggested. Quality of service is an important parameter while selecting and maintaining materialized view.

Various approaches have been proposed and few of these literatures are discussed below.

In [1] author proposed algorithm for improving system performance in real time data warehouse. It enhances the quality of service in real time data warehouse environment by refreshing all materialized views stored in data warehouse. Dynamic selection of materialized views algorithm (DynaSeV) have been proposed. To maintain materialized views update policy is proposed based on access frequency and update frequency. Dynamic adaption of materialized views based on access ration, update frequency and update adaption threshold is proposed.

In [2] authors have discussed various approaches developed for handling stream of data from heterogeneous data source. Challenges for dealing with stream of data are infinite length and concept drift. Query processing from stream of data is important research issue.

In [3] author proposed view maintenance technique which can handle concurrent data updation and schema changes. They address view maintenance anomaly problem under both concurrent data updates and schema changes but have extra cost on updates and cannot maintain mixed update in single process.

In [4] author implemented algorithmic approach for incremental view maintenance. They have incorporated concept of version stores so that older version of relation can be preserved and retrieval correct data in desired state is available every time. This process becomes bit lengthy and more disk space is needed with this approach. Version number should be handled more carefully.

In [5] author proposed a system for integrating and querying data resources in large scale distributed environment. The data grid middleware OGSA-DAI have been used for virtualization. This system can improve accessibility, integration and management of heterogeneous data source.

In [6] author proposed framework called QOP to support changes in view maintenance at data warehouse. Query processing and query optimization algorithms are proposed at view maintenance level. Warehouse data is modeled by relation model terminology. Various levels have been proposed in this paper to address view maintenance problem.

In [7] author identified view adaption problem for view evolution in the context of information source schema changes. This problem is called as view synchronization. They propose Evolvable View Environment (EVE) approach for solving the view synchronization problem. E-SQL is used to direct view evolution process. Authors have successfully built EVE system using JAVA and JDBC component.

In[8] author have taken holistic approach and considered optimization of both global processing plan and materialized view selection. Author have shown that by applying an evolutionary algorithm to either global processing plan optimization or materialized view selection for a given global processing plan can reduce the total query and maintenance cost significantly. Hybrid algorithms perform better than the heuristic algorithm in terms of cost savings, they often require longer computation time.

In [9] author has proposed view maintenance strategies to handle general join views by extending batching and grouping maintenance techniques. Cost-based view maintenance optimization framework is proposed that is capable of generating optimized maintenance plans. Author has concluded that grouping maintenance performs much better than batching. This is because a grouping maintenance plan requires much less maintenance queries than a batching plan.

In [10] author has developed a theoretical framework for the general view-selection problem of selection of views in a data warehouse. They have presented polynomial-time greedy heuristics that provably deliver a solution within a constant factor of the optimal for some important special cases, namely, OR view graphs and AND view graphs, under the disk-space constraint. They have addressed the maintenance-cost view selection problem where the resource constraint is the total maintenance cost of the views selection for materialization. Author have developed greedy heuristic that provide solution for the OR graph. View selection problem is NP- hard in AND graph.

[11] author has proposed materialized view maintenance scheme using Markov's analysis for heterogeneous applications. This method is flexible, dynamic and independent of application areas. Implicit and explicit assumptions are not considered in this methodology. Hence this methodology is independent of the size or type of database and ensures consistent performance. Markov's analysis is chosen to predict steady state probability.

Comparative study of various materialized view selection approaches have been summarized as below by comparing work of various authors on different parameters along with the limitations.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 23 (2017) pp. 13927-13932
© Research India Publications. http://www.ripublication.com

Table 1: Comparison of Various Materialized View Selection Approaches

Parameters **Author**	**Issues Considered**	**Proposed work**	**Benefits**	**Limitations**	**Implementation Methodology/ Tools**
Issam Hamdi, Emma Bouazizi and Jamel Feki,	Dynamic selection of materialized view	New update policy that dynamically determine view maintenance method (DynaSeV)	Improves performance in real time data warehouse and improved QoS	Only focus on dynamic selection of materialized view. Storage space constraint not addressed.	DynaSeV algorithm proposed
Chen, Zhang and Elke	View maintenance	Handles information source schema modification and synchronous data updation.	View maintenance issues addressed	Updated is costly and mixed updated from distributed source is constraint.	Not Addressed
Almazyad and M. Siddiqui	Incremental View maintenance	Implemented concept of version store to retrieve correct data in desired state.	Query processing and view maintenance cost is optimized.	Disk space requirement is high and version number to be maintained carefully.	Not Mentioned
M. Nedim Alpdemir, Arijit Mukherje	Query processor for Grid	OGSA-DAI framework is proposed for virtualization	Improve accessibility, integration and management of heterogeneous data source	View freshness issue not addressed.	SQL Server 2008
Garima Thakur and Anjana Gosain	Addresses issue of query optimization in view maintenance phase	Proposed framework to support changes in view maintenance at data warehouse	Improves functionality by integrating query processing and optimization component.	View maintenance is one of the major challenges during query optimization.	QOP framework proposed
Amy J. Lee, Anisoara Nica, Elke Rundensteiner	View adaption problem when schema changes addressed	Propose Evolvable View Environment (EVE) approach for solving the view adaption problem. E-SQL is proposed.	Dynamic changes in schema changes can be managed.	View Selection problem not addressed.	E-SQL
Chuan Zhang, Xin Yao and Jian Yang	Multiple query optimization and materialized view selection issue	Heuristic algorithm and evolutionary algorithms are proposed.	Query performance optimized when multiple queries are executed on data warehouse	Relationship between computation time and cost saving not handled.	SUN operating system and simulation softwares.
Gupta, Mumick	View selection issues under limitation of disk space	Heuristic algorithm and AND/OR graph model proposed.	Heuristics takes polynomial time and disk space constraints are addressed.	Approximation concerns not addressed	Not cited
Partha Ghosh, Soumya Sen	Implemented for Heterogeneous applications	Markov's Analysis is used for view maintenance.	Evolving, dynamic and distributed applications are taken into account	Replacement of attributed in materialized view not included.	Implemented in JAVA

CONCLUSION AND FUTURE SCOPE

Different query optimization approaches on materialized views have been proposed by researchers. In this paper we discussed designing of data warehousing environment using materialized view to reduce query response time and enhance performance. View selection, maintenance and synchronization problem is studied in real time data warehouse environment. Query processing cost and materialized view maintenance cost are vital parameter while selecting materialized views in data warehouse environment. We have surveyed various methodologies on various parameters and provide comparative study on various parameters suggested by many researchers. We have analyzed that approach of materialized view selection and maintenance has its own benefits and limitations.

In our future work, we will develop new approach that will be helpful in query optimization while accessing data from multiple heterogeneous data source in dynamic distributed environment. We will propose approach that will be helpful in overcoming disadvantages that have been survey in comparative analysis.

REFERENCES

[1] Issam Hamdi, Emma Bouazizi and Jamel Feki, “Dynamic Management of Materialized Views in Real-Time Data Warehouse”, IEEE Trans. on Soft Computing and Pattern Recognition, 2014

[2] M. B. Chandak and Rimjhim Singh, “Survey on various strategies for classification and novel class detection of data stream”, International Journal of computer science and application, Vol.8, No.1, pp. 60-64, Jan-March-2015

[3] Songting Chen, Xin Zhang and Elke A. Rundensteiner, IEEE Trans on Knowledge and Data Engg., vol. “A Compensation-Based Approach for view Maintenance in Distributed Environments,” 18, no. 8, Aug 2006.

[4] A. Almazyad and M. Siddiqui, ―”Incremental View Maintenanace: an algorithmic approach”, In International Journal of Electrical & Computer Sciences, Vol.10,2010.

[5] M. Nedim Alpdemir, Arijit Mukherje, ‘OGSA-DQP: A Service-Based Distributed Query Processor for the Grid’**,** 2005.

[6] Garima Thakur and Anjana Gosain, “A Comprehensive Analysis of Materialized Views in Data warehouse Environment”, 2011

[7] A.M. Lee, A. Nica, and E.A. Rundensteiner, “The EVE Approach: View Synchronization in Dynamic Distributed Environments,” IEEE Trans. Knowledge and Data Eng., vol. 14, no. 5, pp. 931-945,2002.

[8] Chuan Zhang, Xin Yao and Jian Yang, ‘An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment’, IEEE transactions on systems, man, and cybernetics—Part C: Applications And Reviews, Vol. 31, No. 3, pp.282-294, 2001

[9] Bin Liu And Elke A. Rundensteiner, “Optimizing Cyclic Join View Maintenance Over Distributed Data Sources” IEEE Transactions On Knowledge And Data Engineering, Vol. 18, No. 3, March 2006.

[10] A.Gupta and I.Mumick, “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] Partha Ghosh and Soumya Sen, “Materialized View Replacement using Markov’s Analysis”, IEEE International Conference on Industrial Technology (ICIT), Feb. 26 - Mar. 1, 2014, Busan, Korea, 2014

[12] Thomas Schwentick. XPath Query Containment. ACM SIGMOD Record, Vol.33, No. 1, March 2004

[13] Hai Liu, Yong Tang, Qimai Chen, The Online Cooperating View Maintenance Based on Source View Increment. CSCWD 11th International con ference, pp. 753-756, April 2007

[14] Lijuan Zhou, Qian Shi, Haijun Geng, ―”The Minimum Incremental Maintenance of Materialized Views in Data Warehouse”, 2nd International Asia Conference (CAR), March 2010.11th International con ference, pp. 753-756, April 2007

[15] M. Nedim Alpdemir, Arijit Mukherje, ‘OGSA-DQP: A Service-Based Distributed Query Processor for the Grid’**,** 2005.

[16] M. Nedim Alpdemir, Arijit Mukherje, ‘OGSA-DQP: A Service-Based Distributed Query Processor for the Grid’**,** 2005.Donald Kossmann and Konrad Stocker. Iterative Dynamic Programming: A new Class of Query Optimization Algorithms. ACM Transactions on Database Systems, Vol. 25, No. 1, March 2000, Pages 43-82.