Role of Materialized View Maintenance with On-line analytical Processing

Assist. Prof. Dr. Murtadha M. Hamad
Dean of Computer Science College,
University of AL-Anbar.

M.Sc.Bilal Adil Mahdi
Computer Science College,
University of AL-Anbar.

ABSTRACT

The OLAP operation are very useful for application and query different dataset ,the operators aggregate data from variant levels which can be efficiently used for data presentation in data warehouse environment. Materialized view another important issues fore DW and OLAP operates, materialized views are found useful for fast query processing. The process of updating materialized view in response to change is a great challenge in data warehousing. Maintenance of materialized views efficiently with OLAP operators is also a challenge today in the field of RDBMS as well as data warehouse. In this paper, we discuss capabilities of OLAP operator, materialized view maintenance; achieve the principle of consistency MV with DW, view maintenance action with OLAP operators. The proposed system is implemented using Microsoft visual studio C#.NET2010programming language with embedded SQL server management studio 2008 R2, and all test results are found as close as they were expected. The results proved the correctness of system design and its reasonable considerations and choices.

Keyword:

Data Warehouse (DW), Materialized Views (MV), Relational Database Management System (RDBMS), Unified Modeling Language (UML), On-Line Analytical Processing (OLAP).

1- INTRODUCTION

Data warehouse have been developed to overcome the weakness of traditional databases. A data warehouse is a very large database system that collects, summarizes, and stores data from multiple remote and heterogeneous information sources [1]. A DW is a collection of materialized views, which are pre-computed and summarized from multiple operational data sources. These pre-computed materialized views are used to answer OLAP aggregate queries. This technique improves the OLAP query processing efficiency [2]. A data warehouse is an information base that stores a large volume of extracted and summarized data for On-Line Analytical Processing and Decision Support Systems [3].Materialized views play central role in the data warehouse, therefore recently database research community paying attention to the materialized view selection and maintenance. The

attention is towards selecting a set of materialized views to pre-compute under specific resource constraints, such as disk space and maintenance time, in order to minimize the total query processing cost [4].One of the major performance problems in DW is the maintenance process. Whenever source data is changing, the warehouse does not reflect the correct state of this source and has to be maintained. Fortunately only parts of the warehouse are affected by a modification to a single source. So the obvious solution would be to reload these parts. But the huge data inventory and the great amount of changes forbids reloading even parts of the DW from the scratch. For this reason only the changes due to the source modifications are computed to the warehouse [5]. The materialized views need to be maintained so as to reflect changes in the source information. The data may be loaded into the system on a monthly, weekly or daily basis depending on the organization's requirements. During the start of the uploading cycle queries already being processed are generally allowed to execute to completion. This leads to a distinction between two kinds of warehouse maintenance [6]. In this paper, we present framework for OLAP operation design and algorithm for maintenance views based on the change base table. Maintenance view developer to maintain integrity of data and reduce cost of maintenance.

2- Related Work

A materialized view gets "dirty" whenever the underlying base tables are modified. The process of updating a materialized view in response to changes to the underlying data is called view maintenance or view refreshing. The problem of materialized view maintenance has received increasing attention in the past few years due to its application to data warehousing. Traditionally, a view is a derived relation defined in terms of source relations. The data sources may be heterogeneous and remote from the warehouse. Presents a survey on view maintenance, there are some of these studies and their results are referred to. B.Ashadevi. [7], propose framework for selecting views to materialize so as to achieve the best combination of good query response, low query processing cost and low view maintenance cost in a given storage space constraints. The framework takes into 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. Ashish P. Mohod.[8] propose various techniques that are implemented in past, recent for the selection of materialized view. Second, the most critical issues related to maintaining the materialized view, the effective query maintenance strategy and compute total cost of processing a set of queries and maintenance of the materialized views are minimized. A.N.M. Bazlur Rashid. [9], Suggested Materialized views in OLAP applications in the data warehousing environment are an important issue to get query response quickly. View maintenance aims to incrementally maintain the view extent under a source data update. The idea is to issue maintenance query based on the data update to calculate the delta change on the view extent and in this regard the operators

PIVOT and UNPIVOT are playing important role for the commercial database as well as data warehouse. Jingren Zhou. [10] Present a novel way to lazily maintain materialized views that relieves updates of this overhead. Maintenance of a view is postponed until the system has free cycles or the view is referenced by a query. View maintenance is fully or partly hidden from queries depending on the system load. Ideally, views are maintained entirely on system time at no cost to updates and queries. The efficiency of lazy maintenance is improved by combining updates from several transactions into a single maintenance operation.

3- Proposed Work

This paper is focused on development a framework for OLAP operation, which plays an important role in the maintenance process of views and collection changes from base table reflect on materialized views in order to maintain consistency and integrity of the database by the compute changes to view in response to changes to base sources, the idea is maintenance of view with OLAP operation, which focused on aggregation function, method used to perform incremental maintenance in the data warehouse and describes how the integration methods. This method divided into three stage compute stage, delta stage and refresh stage the compute stage computes grid effect of changes to each materialized view because add or remove new rows to the base table, the second stage store these change in the delta stage for the view this delta table have same schema as the view and each tuple in the delta table describes the effects of base data changes on a corresponding tuple in the view, which view having the same value for all attributes as in the delta table. There is delta stage action with the base table and delta table which does not needed arrival to the materialized view for performance the change. The finally stage of the maintenance refresh stage at this stage is updated each materialized views existed in the data warehouse base on the changes that compute and saved in the previous stage (compute and delta stage) when we process refresh to perform the update the materialized view is not available for arrive queries (padlock stage), can be refreshed on demand or on a schedule. The proposed framework is illustrated in the following figure (1).

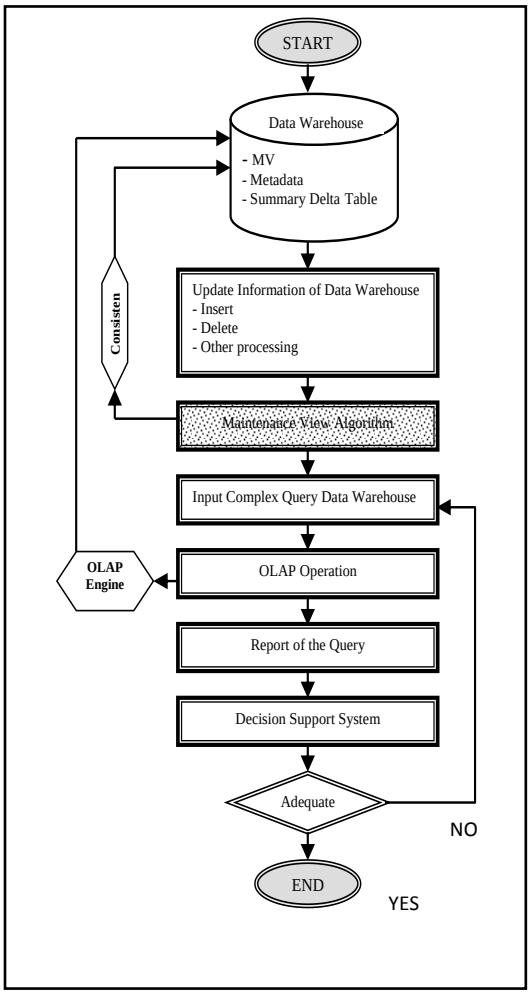

Figure (1): Framework for Proposed

4- On Line Analytical Processing

Relational database engines have been extended to support OLAP operations like drill down or rollup in order to achieve better performance. Materialized views are stored in data warehouse to enable users to quickly get search results for OLAP analysis. Will be adaptive architecture of the on line analytical (OLAP), show to the following algorithm (1).

Input: set of the constraint, OLAP query

Output: Report of retrieval information

- Begin
- User assign OLAP query through application tools.
- Constraint determines.
- Assign parameter that we need for analysis by user.
- OLAP engine specified data to OLAP query associated of metadata from repository.
- Select the dimension from dimension hierarchies and measure data for design data cube
- OLAP engine sent the result to user interface
- User receiving request (REPORT) from the interface application.
- The report content attributes and hugs of the column
- END.

Algorithm (1) Propose algorithms for OLAP Processing

There is more to building and maintaining a data warehouse than selecting an OLAP server and defining a schema and some complex queries for the warehouse. Different architectural alternatives exist. Many organizations want to implement an integrated enterprise warehouse that collects information about all subjects. The OLAP assist many queries over data warehouses require summary data, and, therefore, use aggregates. Another, in addition to indices, materializing summary data can help to accelerate many common queries.

5- View Maintenance

Data warehousing is an emerging and already very popular technique used in many applications for retrieval and integration of data from independent information sources. One can Beliefs of a data warehouse as defining and storing integrated materialized views over the data from multiple independent information sources, an important issue is the pay and correct propagation of update at the sources to the views at the warehouse number methods have been developed for materialized view maintenance in conventional database systems. The process of updating a materialized view in response to change to the underlying data is called view maintenance. When source are updated materialized view may become inconsistence. There are two methods of view maintenance the re-computation method the warehouse sends the query to the source data asking it to re-compute the view from

the scratch after certain number of updates and incremental view maintenance method the warehouse compute changes to view in response to changes to base sources, in this paper utilization second method in order maintenance view. It consists of three phases, namely compute stage, delta stage and refresh stage.

6- Compute Function

At this stage is compute the change in the base data which compute the tuples in the delta table for base view from the tuples in the base table and compute the delta table for other materialized views in the lattice from delta table for the base view, compute statement to compute sum of the row in the delta table and adding in the derived table for maintaining consistency. Can be used delta table for each view to compute delta table to another view in the lattice. After the compute and store the tuple transform the next stage apply (refresh stage) the grid change in the delta table to view each tuple in the delta table due to change to single match tuple in the view .the match tuple in the view is update if it exist in the view or if not found. The maintenance of structural update modification in data warehouses is a crucial point for keeping track of structural modifications. We have proposed a solution in which we used the concept of some exiting approaches like store stage. Our solution is based on incremental view maintenance using the functionality of store version. Assume a user of the system want updates some data in the source. In the source send the update (U) notification to the data warehouse, which Conduct an update on the source for informing. The data warehouse receiving notification from the source and begins prepares a query (Q_J) and send it to the source. The source receives the (Q_J) and returns back the answer (P_J) to that query and translated to compute stage, delta stage for sum change from data warehouse for Jump the last stage refresh stage in order to add changes derived tables to remain useful and consistency with base table. Synchronization between the warehouse and source is maintained. The following figure (2) and algorithm (2) appear how refresh stage in the system.

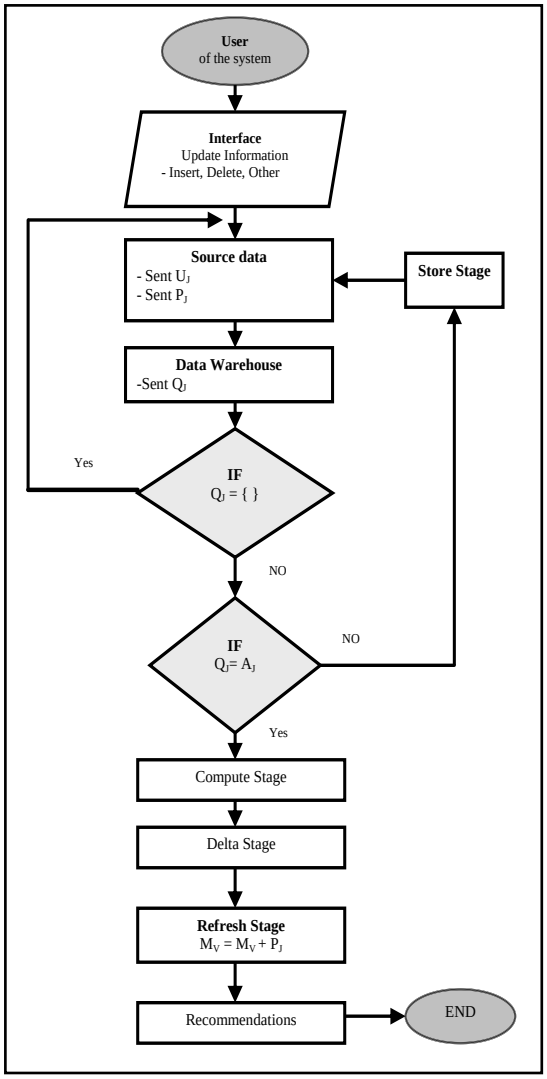

Figure (2): Computation stage &delta stage in the warehouse

Input: source data, data warehouse, set of the change (update query)

Output: view update

- Begin
- ΔT = { } or Ø \\ initially empty delta table.
- Comp = { } or Ø \\ initially empty computation stage.
- SC = {s: s ϵ Source} \\ Containment all change in the source data.
- DW= { } \\ fit on change according to the source.
- MV= {all view materialized in warehouse}.
- SSS \\ storage booking.
- PJ \\ represent source back answer after received query from warehouse.
- For each row change (rr) in the source
- Insert (rr) into DW.
- SC =SC + UI \\ the data source sent notification to DW.
- DW = DW + QJ \\warehouse after receiving the notification prepares query
- If QJ = { } then SSS =SSS+1.
- While (QJ = AJ) do
- Comp = Comp + QJ.
- ΔT = ΔT + Comp.
- Refresh stage MV = MV+ PJ \\ insert change to the lattice materialize view
- Recommendation.
- end while
- end for
- END.

Algorithm (2) maintenance View in the data warehouse

The incremental maintenance algorithm firstly calculates the incremental data of the materialized view through the incremental maintenance expression, then implements the modify operation. The incremental maintenance has many ways and strategies, and adopting different methods will lead to different workload, which affect the resources and efficiency of the system. A materialized view V is defined based on the basic relations between R1 and R2, and R1 and R2 changes have the impacts on the number of V, recorded as V<R1, R2>.

The computation of V changes results from R1 changes and then computation the changes of V using changes in R1 and R2 to implement compute operation.
In order to minimum incremental maintenance for each view V based on relation {R1, R2… Rn}, we obtain modification data from various sources. Order value change Ascending, added into the queue, removes modification data from the queue. Load on incremental changes and then deal with the next operation. On the other hand minimum incremental calculation can be getting it by the ascending of relation changing value. The following table (1) illustrates the minimum incremental maintenance and other maintenance (general).

Table (1): maintenance views to several relations

NO. of Relational	Our way maintenance	Other works	Frequency
2	95	160	number of times
4	180	490	number of times
5	290	650	number of times
7	500	900	number of times

After the compare maintenance views of the update information through conversion relation obtained from collection changes from different sources of the warehouse, figures (3) and (4) shown according to the number of relation.

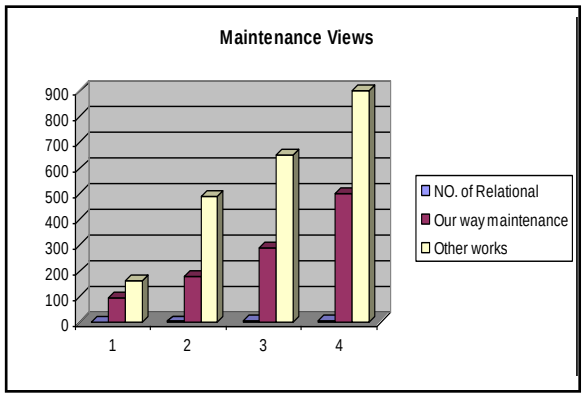

Figure (3): Incremental Maintenance Views

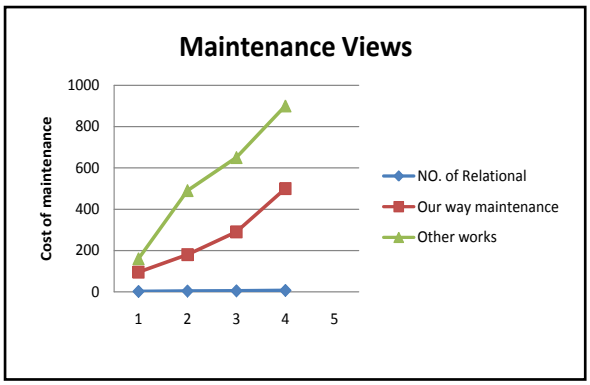

Figure (4): Incremental Maintenance Views

7- Consistency Data Warehouse

Data warehouse is constructed by integrating multiple heterogeneous sources. The data preprocessing are applied to ensure consistency. The integrator is important responsibility when view maintenance, integrator is responsible collecting data source from diversified collecting and store in the data warehouse and periodically

update the materialized views using incremental method. View defined using subset of view specification by the Project manager the view check before store and passes metadata, which created a view manager to primary and maintain the view. The manager start creates view by sending query to the source data, which query processor, the manager passed the results to bestowal warehouse to initialized views and the monitor determine the change at the source and passes to the data warehouse for applies change to warehouse. In our system the manager collection new rows for loading into base table and specify the properties each attribute in which table through the design stage for avoided inconsistency state in each attribute and refused the null value in attribute insert for integration table and control the follow data in to warehouse the integration transfer the change to the materialized view for become consistency with base table through maintain time of maintenance view applies method transmit change to the derived table in the data warehouse to achieve the level of consistency defined in their work as guarantee that the state of the view at any time correspond to a family of states of the source data. The materialized views must be updated to keep consistency with the underlying base tables which the materialized view are defined in the system.

8- Unified Modeling Language

At this stage, the use of UML standardized general purpose modeling language in the field of software engineering. The Unified Modeling Language includes a set of graphic notation techniques to created visual models of object oriented. That helps software developers to identify the features of the implementation of the resolution of errors safely. Using UML for evaluating action the following figure (5) has shown our work.

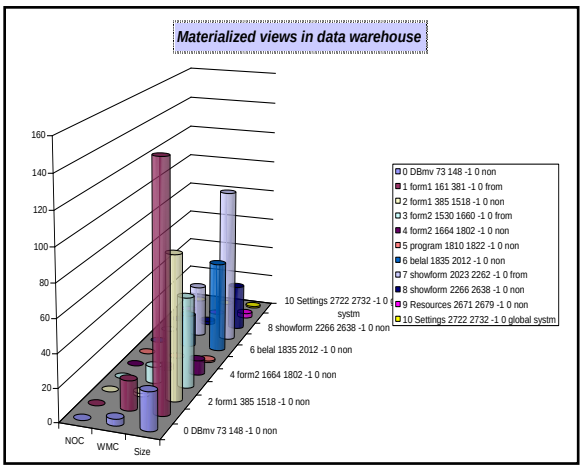

Figure (5): Unified modeling language to Our Work

9- Conclusions

The maintenance of the materialized views is one of the most important issues in designed data warehouse. This paper gives the idea of the materialized view maintenance techniques base on OLAP operation playing important role for the data warehouse. The complete system design has been implemented and tested, the results are found as they were expected. The results proved the correctness of system design and its reasonable considerations and choices. This work arrives at the following conclusions:

1. The main theme behind the design of this system is implemented to get the optimizing queries performance and minimum response time's ratio to answer complex, ad-hoc queries having aggregations.
2. View maintenance aims to support maintain the view under a source data update (changes reflect).
3. Efficient maintenance of materialized view with additive aggregations function.
4. Materialized views have consistency preserving maintenance policies.
5. View maintenance overhead can be significantly reduced by condensing delta table.
6. Experimental results have shown an excellent improvement efficiency of maintenance materialized views with OLAP operation.

REFERENCES

[1] Gang Gou, “A* Search: An Efficient and Flexible Approach to Materialized View Selection”, May 2006.
[2] Miranda Chan," Incremental Update to Aggregated Information for Data Warehouses over Internet", 2000.
[3] B.Ashadevi," Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment", October 2008.
[4] Goretti K.Y. Chan," Design and Selection of Materialized Views in a Data Warehousing Environment ", RixPumps Limited Computer & Information Systems2005.
[5] Michael Teschke," Using Materialized Views To Speed Up Data Warehousing".2003.
[6] Ellis Miller," Analyzing Materialized Views for Fast Refresh", 2006.
[7] B.Ashadevi," Cost Effective Approach for Materialized Views Selection in Data Warehousing Environment ", October 2008.
[8] Ashish P. Mohod Improve Query Performance Using Effective Materialized View Selection and Maintenance: A Survey", April 2013.
[9] A.N.M. Bazlur Rashid," Role of Materialized View Maintenance with PIVOT and UNPIVOT Operators ",2009 .
[10] Jingren Zhou," Lazy Maintenance of Materialized Views",2007 .