Journal of Advanced College of Engineering and Management, Vol.1, 2015

OPTIMAL MATERIALIZED VIEW MANAGEMENT IN DISTRIBUTED ENVIRONMENT USING RANDOM WALK APPROACH

Purushottam Bagale[1], Shashidhar Ram Joshi [2]

[1]Department of Electronics and Computer Engineering, Advanced College of Engineering & Management, Kopundole, Nepal
Email Address: purushottam.bagale@acem.edu.np

[2]Department of Electronics and Computer Engineering Institute of Engineering, Pulchowk Campus
Email Address: sashi@healthnet.org.np

Abstract

Materialized View selection and maintenance is a critical problem in many applications. In large databases particularly in distributed database, query response time plays an important role as timely access to information and it is the basic requirement of successful business application. The materialization of all views is not possible because of the space constraint and maintenance cost constraint. Materialized views selection is one of the crucial decisions in designing a data warehouse for optimal efficiency. Selecting a suitable set of views that minimizes the total cost associated with the materialized views is the key component in distributed database environment. Several solutions have been proposed in the literature to solve this problem. However, most studies do not encompass search time, storage constrains and maintenance cost. In this research work two algorithms are depicted; first for materialized view selection and maintenance in distributed environment where database is distributed, Second algorithm is for node selection in distributed environment.

Keywords: *Materialized View Selection and Maintenance, Query Optimization, Distributed Database, Optimization, Random Walk Approach, Gossip Protocol, Node Selection Approach.*

1 Introduction

Materialized view (MV), proven to be an excellent technique in decision support applications, would continue to be useful in this scenario to preserve the integrated data to ensure better access, performance and high availability. MV must be maintained when the sources change. This has been extensively studied in the past few years, however, it is not sufficiently explored in distributed environment. Materialized view can significantly improve the query performance of relational databases. MVs are a well known technique to reduce the response time of complex queries in database management systems (DBMS). MVs can improve the query performance by avoiding re-computation of expensive query operations. In a distributed DBMS, MVs can materialize query results near to the query issuer and reduce network transmissions. On the other hand, MVs have to be recomputed when the underlying relations are updated, and various resource constraints have to be considered. Thus, it is far from trivial to find the optimal set of MVs in complex, distributed environment [1, 2].

The motivation for using materialized views is to improve performance but the overhead associated with materialized view management can become a significant system management problem.

The common materialized view management activities include; identifying which materialized view to create; indexing the materialized view; ensuring that all materialized views and materialized view indexes are refreshed properly each time the database is updated; checking which materialized views have been used; determining how effective each materialized view has been on workload performance; measuring the space being used by materialized views; determining which existing materialized views should be dropped; archiving old detail and materialized view data that is no longer useful. The distributed model is quickly becoming the preferred medium for file sharing and distributing data over the internet. A distributed network consists of numerous peer nodes that share data and resources with other peers on an equal basis. Unlike traditional client-server models, no central coordination exists in a distributed system; thus, there is no central point of failure [3, 4].

The selection of views to materialize is the important issue in distributed environment. In this thesis work, outlined a methodology whether the views created for the execution of queries is beneficial or not by considering the various parameters: cost of query, cost of maintenance, net benefit and storage space. Here, proposed methodology for selecting views to materialized so as to achieve the best combination good query performance. These algorithms are found efficient as compared to other strategies.

2 Methodology

2.1 Complete System

The optimization model of this research involves the process of selection and update of materialized view with tree based approach and determining the best node for executing the query using node selection strategy.

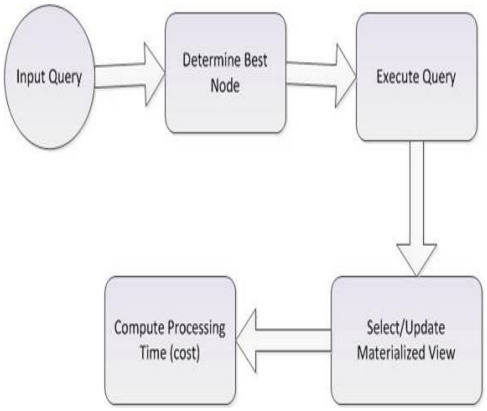

Fig 1: Complete Diagram of Proposed System

In the proposed system as shown in Fig: 1 queries are taken as input. Since our system is for distributed environment, best node is then determined where query is to execute. Then Query is executed using node selection strategy. Then the creation and maintenance of materialized view is done by using tree based approach. Then the Processing Time (Cost) is computed. This computed cost will be used for the evaluation of our system.

2.2 MV Selection and Maintenance, RWA

2.2.1 Materialized View Selection

The Materialized view selection is to select an appropriate set of views that minimizes total query response time and the cost of maintaining the selected views, given a limited amount of resource, e.g., materialization time, storage space, etc. There are mainly two classes of materialized view selection. First is materialized view selection under a disk space constraints and the second is materialized view selection under a maintenance time constraints. The problem of utilizing the limited resources disk space or maintenance time to minimize the total query processing cost comes under the materialized view selection with resource constraints [5].

2.2.2 Materialized View Maintenance

Materialized views are stored in data warehouse to enable users to quickly get search results for analysis. When the remote basic data source changes, the materialized views in data warehouse are also updated in order to maintain the consistency, this causes the need for handling the problem of materialized view maintenance [6].

Data sources in a distributed environment are typically owned by different information providers and function independently from one another. The relationship between materialized views and such autonomous data sources hence must be loosely coupled. That is, the source updates are committed without any concern of how and when the view manager will incorporate them into the view. This causes problems which we called maintenance anomalies: View maintenance, view synchronization, and view adaptation[7]. View maintenance maintains the materialized view extent under source data updates. In contrast, view synchronization aims at rewriting the view definition when the source schema has been changed. Thereafter, view adaptation incrementally adapts the view extent again to match the newly changed view definition.

2.2.3 Random Walk Approach

Random walk approach is a mathematical formalization of a path that consists of a succession of random steps. It is the stochastic (non-deterministic) process formed by successive summation of independent, identically distributed random variables. In this thesis work, basic of random walk approach is used for determining the best node in which the materialized view is created or updated. [8]

2.3 Algorithm

2.3.1 Node Selection Algorithm

This algorithm decides the nodes in the distributed environment for which materialized view should be created, updated or to be maintained.

The random walk algorithm is used as base for designing the node selection algorithm and gossip protocol is used to find the best set of the nodes. Where, M = Total number of nodes

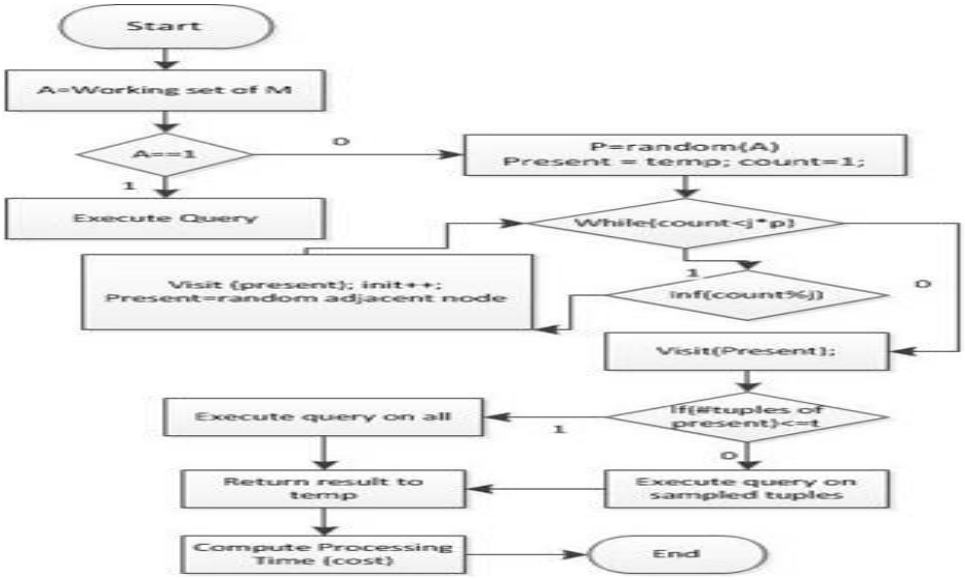

Fig 2: flow chart of node selection algorithm

in network A = Number of Active nodes Init = node where query is initiated P = number of nodes to visit Temp = Node where query is initiated Q= Query with selection condition j = jump size for randomly selecting nodes t = max tuples to be processed per node

The Random Walk Algorithm will be used for node selection. Initially it will check the available nodes. If there is only one node then the query will be executed on the same node otherwise the random nodes will be identified from the available active nodes on which the query will get executed [9]. This algorithm will decide the nodes in the distributed environment for which materialized view should be created, updated or to be maintained. RWA will be used as base for designing the node selection algorithm and gossip protocol will be used to find the best set of the nodes.

2.3.2 Materialized view creation and maintenance Algorithm

The tree based approach will be used for creating and maintaining materialized views. Flow chart of the materialized view creation and maintenance is given in fig.2 where T = Total records in database TR = threshold for number of records. V= Set of Materialized Views

The middle record will be selected as root element of tree. The records will be then splitted till the threshold doesn't reach so that the leaf of tree should contain the number of records that will be present in materialized view. Then the materialized view will be created for each leaf node indirectly each leaf represent materialized that has to be created and maintain. The materialized view will be selected as per the query the records for which the query is intended the materialized view for those records will be selected for the processing. This will minimize the total execution cost.

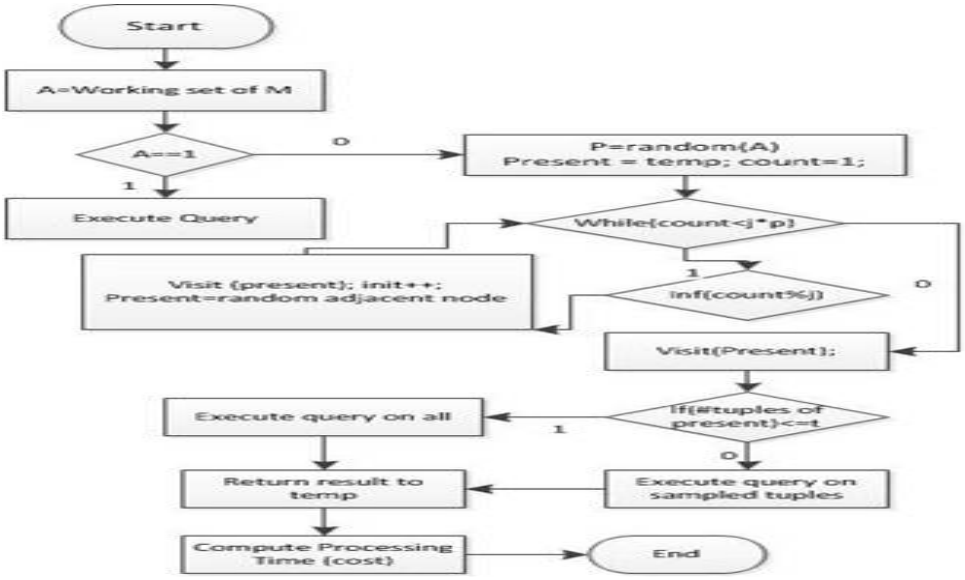

Fig 3: flow chart of materialized view creation and maintenance algorithm

3 Result and Discussion

3.1 Comparision of QPC, MC and SC for four strategies

The simulation model is designed to implement our Optimized Materialized view management in distributed environment using random walk approach. For our experiments, java prototype of the algorithms described in section 3 is programmed. The simulated distributed database environment including computing nodes, database tables and cost-based query optimizer is implemented for Optimized Materialized view management in distributed environment using random walk approach. The cost model for the optimizer will be described in the next section. The total cost is calculated on the basis of query processing, maintenance and storage cost for four strategies.Result is shown in Fig 4.

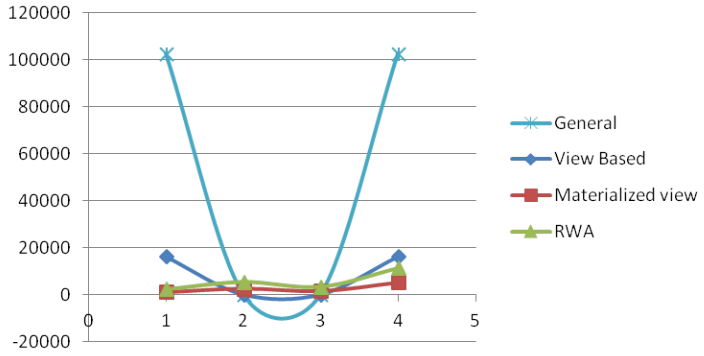

Fig 4: Comparision of QPC, MC and SC for four strategies

Here we found that the total cost computed from the query processing cost, maintenance cost and storage cost in four different stratigies shows that though the maintenance cost is not applied for general query and view based, the total cost is very high because it searches for the base table for the result. The total cost for RWA is higher than the general MV but lower than general and view based stratigies. The total cost for materialized view is lowest among all.

3.2 Total Cost Comparision

In Fig 5, the total cost comparision of all the four strategies along with proposed random walk approach materialized view is given.

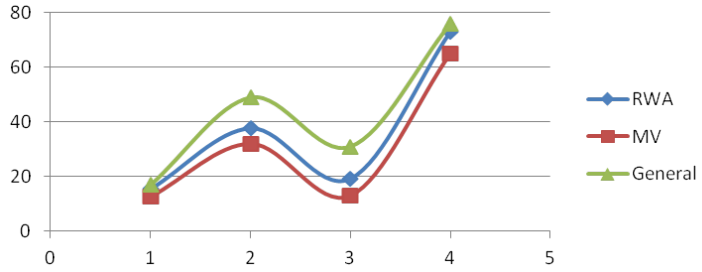

Fig 5: total cost comparision of all four strategies

3.3 Response time comparision

In Fig 6, the response time taken by all the four strategies along with proposed random walk approach materialized view is given. The response time is given in terms of milliseconds. Here the comparision is implemented using the proposed methodology with general, view based and matrialized view based strategy on the basis of response time and it is observed that proposed method requires a minimum time for execution and response and this minimizes the total cost of query for processing. Although the maintenance cost and maintenance cost for our proposed method is high, the response time taken by proposed method is best.

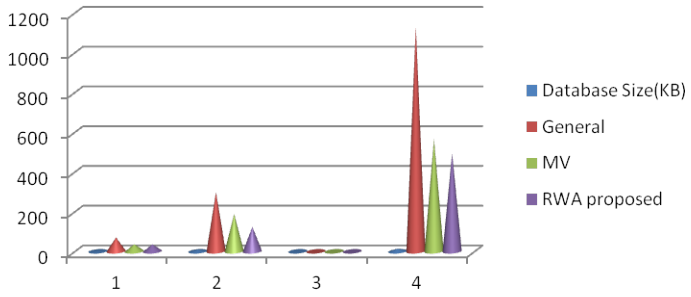

Fig 6: Response time comparision of all four strategies

4 Conclusion

This research given a optimal way to manage materialized view in distributed environment using random walk approach. Thus it is concluded that this approach provide simple and cost-effective guidelines to the engineers, scientists, and researchers in the field of materialized view in distributed environment.

The same concept can be deployed in the future by extending the domain size and environment to the cloud computing

References

[1] D. P. P.Kalnis, N. Maumoulis, "View selection using randomized search," *Internatioanl Journal of Information Technology*, 2002.

[2] A. R. V. Harinarayan and J. Ullman, "Implementing data cubes efficiently," *Proceedings of ACM SIGMOD 1996 International Conference on Management of Data, Montreal, Canada*, pp. 205–216, 1996.

[3] A. Bauer, W. Lehner, "On Solving the View Selection Problem in Distributed Data Warehouse Architectures." *IEEE(1099-3371)*, year = 2003.

[4] G. D. B. Babcock, S. Chaudhuri, "Dynamic sample selection for approximate query processing," pp. 539–550, 2003.

[5] C. Zhang, X. Yao, J. Yang, "An Evoluationary Approach to Materialized Views Selection in Data warehouse Environment," *IEEE*, 2001.

[6] F. H. LWF Chaves, E Buchmann, "Towards materialized view selection for distributed database," *International conference on extending database Technology*, 2009.

[7] H. Jiang, D. Gao, W. Li., "Exploiting Correlation and Parallelism of Materialized View Recommendation for Distributed Data Warehouse," *IEEE*, 2007.

[8] Wikipedia, "Random walk."

[9] C. Gkantsidis,M. Mihail,A. Saberi, "Random Walks in Peer to Peer Networks," *Proc. IEEE INFOCOM*, 2004.

[10] B. Liu, E.A. Rundensteiner, "Cost-Driven General Join View Maintenance over Distributed data source," *IEEE*, 2005.

[11] S. Chen, X Zhang, E.A. Rundensteiner, "A Compensation based approach for view maintenance in distributed environment," *Signal Processing: An International Journal*, 2006.

[12] A. Benjamin,D. Gautam,G. Dimitrios,K. Vana, "Efficient Approximate Query Processing in Peer-to-Peer Networks," *IEEE Trans on Knowledge and Data Engg.*, 2007.

[13] J. Shantanu, J. Christopher, "Materialized Sample Views for Database Approximation," *IEEE Trans of Knowledge and Data Engg.*, 2008.

[14] D. R. B. B. Ashadevi, "Optimized cost effective approach for selection of materialized views in data warehousing," *International Journal of Information Technology*, 2009.

[15] V. T. p.p.karde, "Selection and maintenance of materialized view and its application for fast query processing," *International Conference on extending database Technology*, 2010.

[16] Wikipedia, "Java remote method invocation."

[17] Wikipedia, "Materialized view."