

Acceleration of Query Response in Distributed System using Materialized View Approach

Pooja G.Dhawane¹, Prof. Garima Singh Makhija²

Department of Computer Science and Engineering, RTMNU University, W.C.E.M., Nagpur, Maharashtra, India.^{1,2}

Abstract: Data warehousing is a collection of data from multiple, very large, distributed, heterogeneous databases and other information sources. This information is stored in autonomous and distributed systems. Quick response time and accuracy are the vital things for the database to be successful. In distributed databases query response time is the basic need. Performance of query can be improved by different approaches like hybrid integration system where most recent data is stored at mediator i.e. common access point to distributed system, optimization of query, use of proper data structure, reduce complexity of query etc. leaving all alternatives, we are planning to use materialized view approach. Materialized view is used to store the result of queries. We are going to store the result of queries based on the frequency of queries fired by the user. The queries having high frequencies i.e. based on the user demandable queries materialized view is designed. This will save the time of revisiting the base table whenever any query is fired frequently and also saves the time of cost of operations like join, aggregations etc.

Keywords: Materialized view, Query Response, Access Frequency, data warehousing.

I. INTRODUCTION

Data warehouse is defined as subject oriented, integrated, nonvolatile and time variant collection of data to support management's decision.[1]

Data warehousing is the collection of huge amount of data for querying and analysis. Fundamental requirement for the success of a data warehouse is ability to make decision with both accurate and timely consolidated data also fast query response time.

In existing system [2] the attributes which are most probably to extract where placed at the common access point called mediator to the distributed system. This has reduced the access time for the getting the data from original source of data. But if the attributes placed at the mediator are not proper then performance of the query will not be up to the mark. The primary intent of this research is to develop a system which will reduce the execution time for query i.e. getting the response from database as early as possible which in turn improves the performance of queries. We are going to take the help of materialized view.

To avoid accessing the original data sources and increase the efficiency of the queries fired on data warehouse, some results in the query processing are stored in data warehouse. These results in data warehouse are known as materialized view. On abstract level data warehouse can be seen as collection of materialized view for quick access of data.

To eradicate the problem in existing system, our idea is to place the materialized view i.e. result of queries which have probability to fire mostly at the mediator of distributed system instead of attributes.

II. RELATED WORK

Various approaches have been proposed for improving performance of data warehouse.

In Hybrid integration system [2] some part of data were placed on mediator i.e. common access point to data warehouse. The attributes of interest are extracted from various data sources then possible values are assign to them and this information is placed on mediator. But if the attributes chosen are in appropriate then query performance will not be at the mark. Thus, we are trying to eradicate this problem using materialized view which directly stores the result of queries executing frequently Hema Botre and M. S. Choudhari [3] presented a Priority based algorithm for materialized view selection. this algorithm selects the queries based on the priority associated with queries for the materialization.

Harinarayan proposed greedy algorithm [4] using data cubes to select the materialized view but this work does not consider the storage space required.

Sanket Patel used Tree based algorithm [5] for the selection & maintenance of materialized view. The decision of which data set should be materialized must be made on the basis of the system workload, which is a sequence of queries and updates that reflects the typical load on the system.

Y.D.Choudhary [6] proposed a novel CBFSMV algorithm for selection of materialized view using query clustering strategy that reduces the time as compared to response time for actual database.

In [7], authors have implemented dynamic adjustment for static materialized view selection algorithm that is EMVSDIA (Efficient Materialized view selection Dynamic Improvement algorithm). Experiments demonstrates EMVSDIA has excellent implementing efficiency and reach to expected efforts.

B. Ashadevi, Dr.R. Balasubramanian [8] implemented an algorithm that is projected for choosing the views to materialize on basis of their weight acquired in the query

set. A greedy algorithm is used to incorporate the maintenance cost and storage constraint in the selection of data warehouse Materialized View. It reduces comp-lexity of algorithm compare to previous algorithm [9].

Hybrid algorithm is combination of Evolutionary algorithm and Heuristic algorithm. The result of algorithm show that applying an evolutionary algorithm to either global processing plan optimization or materialized view selection for a given global processing plan can reduce total query and maintenance cost. Excessive computation time of pure Evolutionary algorithm and unsatisfactory quality of solution of Heuristic algorithm can be overcome by using Hybrid algorithm[10].

III. PROPOSED SYSTEM

The architecture of system is represented in figure1. This approach is to give user the illusion of querying a homogeneous and centralized system, by avoiding him to have finding relevant sources for its queries and saves the time to revisit the base table again and again.

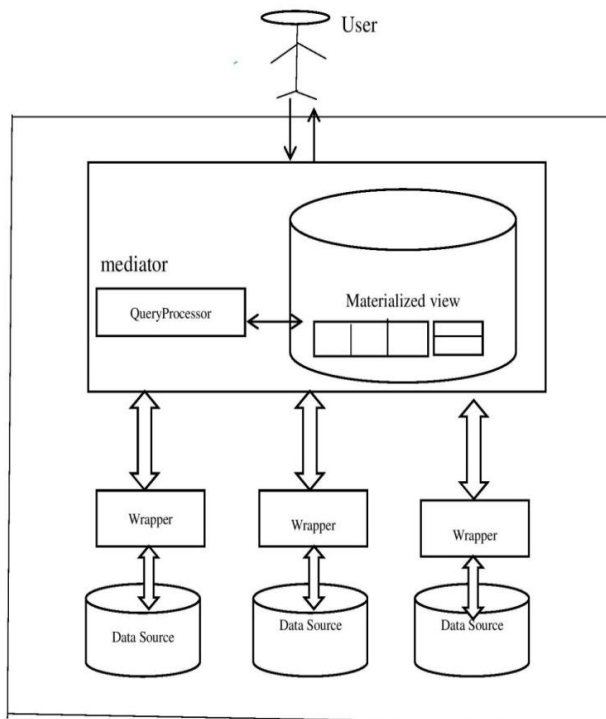


Figure.1. Proposed System Architecture

IV. CONCLUSION

One of the most important issues in designing a data warehouse is accuracy and quick response time for user query. By implementing the materialized views in distributed system, we are trying to get the combination of good query response time from which query processing cost should be minimized.

REFERENCES

[1] W.H. Inmon, Building the data warehouse, 2nd ed, John Wiley and Sons, Canada, ISBN:0471 -14161-5, 1996.
[2] Samir Anter, Ahmed Zellou, Ali Idri "Hybrid Integration System", 2013, 5 th IEEE International Conference on Computer Science and Information Technology (CSIT).

[3] Hema Botre, M.S. Choudhari, "Design an Algorithm For Materialized view Selection",2013, IJERA
[4] V. Harinarayan, A Rajaraman, and J. Ullman. "Implementing data cubes efficiently". Proceedings of ACM SIGMOD 1996 International Conference on Management of Data, Montreal, Canada 1996.
[5] Sanket Patel and Deepak Dembla, "An Approach for Selection and Maintenance of Materialized View in Data Warehousing", International Journal of Information Sciences and Application. ISSN 0974-2255 Volume 4 Number 1 (2012).
[6] Yogeshree D. Choudhari, Dr. S. K. Shrivastava, "Cluster Based Approach for Selection of Materialized Views, Volume 2, Issue 7, July 2012.
[7] Zhou Lijuan , Ge Xuebin and Wang Linshuang, Shi Qian, 'Efficient Materialized View Selection Dynamic Improvement Algorithm' , 6th International Conference on Fuzzy Systems and Knowledge Discovery, Vol.7, pp.294-297, 2009.
[8] B.Ashadevi and Dr.R.Balasubramanian, "Optimized Cost Effective Approach for Selection of materialized Views in Data Warehousing", JCS&T Vol. 9 No. 1, April 2009.
[9] Gupta, H. & Mumick, I. Singh, "Selection of Views to Materialize in a Data Warehouse" IEEE Transactions on Knowledge and Data Engineering, 17(1), 24-43, 2005.
[10] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "A Framework for Efficient Data Anonymization Under Privacy and Accuracy Constraints," ACM Trans. Database Systems, vol. 34, no. 2, article 9, 2009.
[11] Chaun zhang, Xian Yao "An Evolutionary Approach to Materialized Views selection in a Data Warehouse Environment" IEEE vol.31, No.3, Aug. 2001.