

Optimized Generation and Maintenance of Materialized View using Adaptive Mechanism

Mr. Shailesh Kurzadkar

Department of Computer Science and Engineering
Nagpur Institute of Technology
Nagpur
skurzadkar@gmail.com

Prof. Abhijeet Bajpayee

Department of Computer Science and Engineering
Nagpur Institute of Technology
Nagpur
Abhijeet.Bajpayee1987@gmail.com

Abstract— Data Warehouse is storage of enormous amount of data gathered from multiple data sources, which is mainly used by managers for analysis purpose. Hence to make this data available in less amount of time is essential. Using Materialize view we can have result of query in less amount of time compared to access the same from base tables. To materialize all of the views is not possible since it requires storage space and maintenance cost. So it is required to select materialized view which minimizes response time of query and cost of maintenance. In this paper, effective approach is suggested for selection and maintenance of materialize view.

Keywords- *Materialized view, Cost of Maintenance, Response time of query*

I. INTRODUCTION

A data warehouse is mainly used for processing of queries and detailed analysis of data that is useful for decision makers. The method that is used in practice for best response time is the concept of quickly answered materialized views. A materialized view usually stores the result of a query and makes them available for future use which is used to answer the query in a data warehouse. For creation of materialized view, some storage space will be required as they contain result set of the query. Hence it is required to provide preservation technique to the materialize view.

Difference between views and materialize view is, views are logical and materialized views are physical i.e actual results gets stored into materialized view.

In the proposed work of this paper, Materialized view is created using adaptive method of clustering and then if materialized view is not in use there is a mechanism for deletion of the same.

II. RELATED WORK

Various methods have been suggested by different researchers for selection and maintenance of materialize views. The view selection problem is defined as to select view to materialize that is as per the user requirement and it should minimize the response time of the query and have less storage space.

A greedy algorithm is proposed in [11] for materialized view evaluation with the help of data cubes. Heuristic algorithm is proposed in [10]. View adaption technique is used in [25]. The usefulness of this method is, it reduces cost of query. Genetic algorithm is suggested in [22] to select materialize view. Sanjay Agrawal et al. [23] explained an end-to-end solution to the problem of selecting materialized views and indexes.

By maintaining local views, global view maintenance is proposed in PDMS[5] is one kind of super peer maintains meta data for mapping schema in inter-peers or inter-peer

changes in local PDMS but it has some draw back like Information sharing is complex & difficult in PDMS also Querying not addressed. In [8] outline how to maintain views differentially. The presented algorithm works by deleting a superset of the affected tuples in the materialized view and then rederiving some of the tuples. In [19], post-update deferred view maintenance is proposed. In this different algorithms are explained to maintain the view after the base relations have already been updated.

In this research paper we proposed adaptive mechanism which uses clustering approach to create materialized view and an approach is suggested for maintenance of materialize view.

III. PROPOSED METHODOLOGY

For generation of materialized view, large data is required. In this approach, huge amount of data is generated with the help of tool like Automatic Record Producer. In this tool, there is no need to create the database manually.

Next, Materialized views are created by considering 3 essential parameters like frequency of the query i.e number of times particular query gets executed into database. Using these parameter, only those queries that gets executed more number of times will get selected. Next is storage required for created materialized view. This is required because if certain query is getting more space for storage will not be considered for creation. Third parameter is response time of the query. This will check execution time for query and if query is taking less amount of time for execution then only it will be considered for generation of materialize view. While considering above parameters for generation, an important clustering adaptive mechanism is used. In this mechanism, similar queries will get clustered into one and on such different clusters, by considering all the above factors materialized view will get generated.

IV. ALGORITHM FOR MATERIALIZE VIEW GENERATION

I. Automatic Record Producer

In this step, huge amount of record is generated on which materialize view will get created. Permutation and combination of different values will be performed and accordingly data will get stored in database.

II. Generation of Materialize view

i) Calculating Rate of query frequency using following algorithm:

Algorithm 1

```
1: Calculate RQFreq
2: for each query in Q
3:   find FreqQ and MaxFreqQ
4:   RQFreq = FreqQ/MaxFreqQ
5: end
```

Where,

RQFreq =Rate of query frequency
FreqQ=Frequency of each query
MaxFreqQ=Maximum frequency of query

ii) Finding Storage space

Algorithm 2

```
1: Calculate MVst
2: for each query in Q
3:   find RMVc and CMVc
4:   MVst= RMVc*CMVc
5: end
```

Where,

MVst=Materialize view Storage space
RMVc =Rows required for Materialized view
CMVc=Columns required for Materialized view

iii) Calculating processing time for query

Algorithm 3

```
1: Calculate QPT
2: Find BTQ and ETQ
3: QPT=ETQ-BTQ
4: end
```

Where,

QPT=Query processing time
BTQ=Begin time of query
ETQ=End time of query

Finally, Cluster area Threshold value is calculated using following algorithm:

Algorithm 4

```
1: Calculate Area of Table (Ta) = TCol*TRows
2: Area of Cluster (Ca) = CRc*RRc
3: Maximum Cluster Area Threshold % ( MCAt) =Ac/At
4: Let Threshold_Area_Ratio = THRAr
5: If MCAt <= THRAr Then
6:   Create Materialize view
7: Else
8:   Do not create
9: EndIf
10: end
```

Where,

CRc=Columns required for creation of cluster
RRc=Rows required for creation of cluster
TCol=Total number of columns
TRows=Total number of rows

V. ALGORITHM FOR MATERIALIZE VIEW MAINTENANCE

After creation of materialize view, if found that created materialize view is having low access frequency and having high storage space, then there is a mechanism to delete that materialize view using following algorithm[7]:

Algorithm 5

```
1: Calculate MVPst
2: for each Materialized View
3:   X = 2 log (FMv) - log (SMv)
4:   if (X < Threshold) then
5:     Remove current Materialized view;
6:   end if
7: end for
8: end
```

Where,

FMv=frequency of materialized views get accessed
SMv=Storage required for materialize view

VI. EXPERIMENTAL RESULTS

Using Automatic record generator huge amount of data gets loaded into different tables in database.

By applying Clustering algorithm, suppose following queries are having more access frequencies low processing cost and less storage space.

```
Q1:select firstname,cust_id from custtable;
Q2:select firstname,middlename,lastname from custtable;
Q3:select firstname,telNumber from supinfo group by address;
```

If size required for creation of materialize view is less than that of size of base table then create materialize view else not.

After creation of materialize view, if found that created materialize view for above 3 queries are having low access frequency and having high storage space then it can be deleted.

Suppose for Q1 by considering access frequency and storage space.if processing cost is 4.213

And for Q2 it is 6.223 and for Q3: 7.121

Let threshold value is 5,then it will delete Materialize view created for Q1 and preserve Q2 and Q3.

Finally,we can show comparison if above 3 queries get accessed from base table and from Materialize view using following fig 1



Figure 1:Execution time comparison

VII. CONCLUSION

A greedy algorithm for selection of materialized view is assumed in [11] but here storage space and maintenance cost is not considered. In [15] an algorithm is proposed for selection of materialize view using reduced table. Very simple and fast heuristic algorithm is proposed in [9] to select aggregates for pre computation. Novel view maintenance strategy that takes the main-delta architecture and resulting merge process of columnar storage into account is explained in [2].

In this paper an optimized mechanism is used that uses clustering algorithm by considering all the essential factors like access frequency of queries, storage space and processing cost.

REFERENCES

- [1] Deepika Kirti and Jaspreeti Singh, "Assortment of Materialized View: A Comparative Survey in Data Warehouse Environment", International Journal of Computer Applications (0975 – 8887) Volume 96– No.7, June 2014.
- [2] Stephan Müller, Lars Butzmann, Kai H'owelmeyer, Stefan Klauck, Hasso Plattner, "Efficient View Maintenance for Enterprise Applications in Columnar In-Memory Databases", 17th IEEE International Enterprise Distributed Object Computing Conference, 2013
- [3] Ravindra N. Jogekar, Ashish Mohod, "Design and Implementation of Algorithms for Materialized View Selection and Maintenance in Data Warehousing Environment", ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 9, September 2013

- [4] 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
- [5] P. P. Karde and Dr. V. M. Thakare, "Selection & Maintenance of Materialized View and Its Application for Fast Query Processing: A Survey", International Journal of Computer Science & Engineering Survey (IJCSSES) Vol.1, No.2, November 2010
- [6] Abdulaziz S. Almazayad and Mohammad Khubeb Siddiqui, "Incremental view maintenance: An Algorithmic Approach", International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:03, June 2010
- [7] B. Ashadevi, R. Balasubramania, "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
- [8] 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
- [9] 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.
- [10] Yang, J., Karlapalem, K. and Li, Q., "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, 1997
- [11] 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.
- [12] Satyanarayana R Valluri, Soujanya Vadapalli and Kamalakar Karlapalem, "View Relevance Driven Materialized View Selection in Data warehousing Environment", Australian Computer Science Communications, Vol. 24, Issue 2, pp. 187-196, 2002.
- [13] C. A. Dhote and M. S. Ali, "Materialized view selection in Data warehouse: A survey", Journal of Applied Sciences, Vol. 9, No. 3, pp. 401-414, ISSN 1812-5654, 2009
- [14] W.H Inmon, Building the Data Warehouse, 4th Edition, John Wiley and Sons, Canada, ISBN: 978-0-7645-9944-6, September 2005
- [15] Jin-Hyuk Yang and In-Jeong Chung, "ASVMRT: Materialized View Selection Algorithm in Data Warehouse", International Journal of Information Processing Systems, Vol. 2, No. 2, pp. 67-75, 2006.
- [16] Garima Thakur and Anjana Gosain, "A Comprehensive Analysis of Materialized Views in Data warehouse Environment", International Journal of Advanced

- Computer Science and Applications, Vol. 2, No. 5, pp. 76-82, 2011
- [17] Yang Kehua, Abdoullahi Diasse, “A dynamic materialized view Selection in a Cloud-based DataWarehouse”, IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 2, No 1, March 2014
- [18] T Palpanas, R. Sidle, R. Cochrane, and H. Pirahesh, “Incremental Maintenance for non distributive aggregate functions”, in VLDB,2002,pp.802-813
- [19] P. P. Karde and Dr. V. M. Thakare, “An Efficient Materialized View Selection Approach for Query Processing in Database Management”, International Journal of Computer Science and Network Security, VOL.10 No.9, September 2010
- [20] S.Chaudhuri and U. Dayal, “An Overview of Data Warehousing and OLAP Technology,” SIGMOD Record, vol. 26, no. 1, pp. 65-74, 1997.
- [21] S Chen and E.A. Rundensteiner, “GPIVOT: Efficient Incremental Maintenance of Complex ROLAP Views,” 21st International Conference on Data Engineering (ICDE’05), pp. 552-563, 2005.
- [22] C. Zhang and J. Yang, “Genetic algorithm for materialized view selection in data warehouse environments,” Proceedings of the International Conference on Data Warehousing and Knowledge Discovery , LNCS, vol. 1676, pp. 116–125, 1999.
- [23]] S. Agrawal, S. Chaudhuri, and V. Narasayya, “Automated Selection of Materialized Views and Indexes in SQL Databases,” Proceedings of International Conference on Very Large Database Systems, 2000.
- [24] L. S. Colby, T. Griffin, L. Libkin, I. S. Mumick, and H. Trickey, “Algorithms for deferred view maintenance,” in *SIGMOD*, 1996, pp.469–480.
- [25] A. Gupta, I. Mumick, J. Run, and K. Ross, “Adapting materialized views after redefinitions: techniques and a performance study”, In Elsevier Science Ltd., pp 323-362, 2001.