Performance Optimization Query in Data Warehouse by using MV approach

[1]Amish Bhiwapurkar, [2]Vaishnavi Khandar, [3]Sanket Maliye & [4]K. N. Hande
[1-4] Department of Computer Science and Engineering,
[1]Priyadarshini Bhagwati College of Engineering, Nagpur, India

Received: January 25, 2019 Accepted: March 09, 2019

ABSTRACT: *Data warehouse is the collection of various types of data collected from the heterogeneous data sources. Selecting a suitable set of views that minimizes the total cost associated with the materialized views is the key component in data warehousing. Data warehouse contain the materialized view which can provide the quick response to the user as compared to access the same data from the base table. Materialized views are very useful to perform complex queries as well as it can be the join result of a queries or the combination of more than one table. Those queries which are frequently accessing by the user will select for the materialization. It means materialized those queries which are accessing more frequently by the user & by which disk space can also be minimize. Bin Rank, a system that approximates Object Rank results by utilizing a hybrid approach inspired by materialized views in traditional query processing. It provides a fast search over previous queries giving the user an optimized result. This system also maintains ranking list of frequently used websites. Thus, implementation of materialized reduces the access time required to get the data from the data warehouse. Use of Bin Rank algorithm increases performance of system. Implementation of materialized view is one of the best options to improve performance of data warehouse.*

Key Words: : *Data warehouse, Materialized view, Query processing, Bin Rank.*

I. Introduction

Data is relevant information about any particular thing. Database system consists of collection of interrelated data known as database and software program to manage and access data. Data can be stored in many different kinds of databases and information repositories. A repository of multiple heterogeneous data sources organize at single site in order to support management decision making is called as Data Warehouse. Data Warehouse can be considered as repository of an organization's. It is designed to facilitate reporting and analysis of data, focuses on data storage. It is an approach to the integration of data from multiple possibly very large, distributed, heterogeneous databases and other information sources. Data Warehouse gives the quick response to complex data which never used to be possible in traditional database. Data warehouse (DW) is defined as a subject-oriented, integrated, steady and time varying data set which supports enterprises or organizations to make decisions. As the decision maker needs to query several values from one subject for real-time analysis processing, the multidimensional model of DW is usually implemented as star schemes to meet the requirements. This kind of hierarchical model is highly un-normalized and query oriented. There are two kinds of table in star schemes. One is fact table which contains basic quantitative measurements of a business subject the other is dimension table that describes the facts. If there are more than one fact table in a DW, it can be called galaxy model which is actually constituted of several star schemes. Complex queries are always requested in DW. When users need to process multi-dimensional analysis, multi-table joins may be involved. Although data can be stored in a multi-dimensional database, DW usually stores data in the form of relational database. As the number of dimension and the overall size of data sets increase, the size of DW often grows to gigabytes or terabytes. When the complex queries are implemented on mass multidimensional data, the query efficiency is far beyond satisfaction.

Materialized view is a kind of pre-computed structure it materializes the calculated results ahead of using. The pre-computed values are often mean, sum, average, etc. Queries on materialized views are fast responded, because no join needs to be made on successive requests, and the records in views are less than the original tables. Materialized views can be applied to OLAP, but due to the limit of storage space, it is infeasible to store results of all queries. Some heuristic algorithms have been used to find an approximate optimal solution. For example, greedy and genetic algorithms that based on requirement and probability are applied to generate views. But once queries are made on the records which are not materialized, the

efficiency cannot be improved, and it is unacceptable for any delay when users need the results urgently. So there is limitation of materialized method. Feature selection is a procedure to select a subset from the original feature set by eliminating redundancy and less informative features so that the subset contains only the most discriminative features. Applied to dimension reduction, a set of attributes that best represents the overall data set is found out by feature selection. But feature selection has the same problem with materialized view that when the queries involve the dimensions which are not selected, the efficiency of this method deceases. Materialized views are very useful to perform complex queries as well as it can be the join result of a queries or the combination of more than one table. This work selects those queries which are more useful to the user and this will be depends on the frequency of a queries. Those queries which are frequently accessing by the user will select for the materialization. It means materialized those queries which are accessing more frequently by the user & by which disk space can also be minimize. Maintenance of the materialized views is the most important factor to provide accurate data. Therefore this work will update the data time-to time, as the data will be changed in the base table corresponding data will also be changed in materialized views. This work will enhance the query processing as well as it will be useful in huge amount of databases. Materialized view (MV) is an approach used to increase performance of query in data warehouse. It is the pre-calculated (materialized) result of query. A MV is computed for SQL query since in Data Warehouse, It relates to a SQL statement, that is MV corresponds to the result of SQL statement execution. Data Warehouse can be seen as a set of materialized views over the data extracted from the distributed heterogeneous databases. Selecting a view to materialize for the purpose of supporting the decision making efficiently is one of the most significant decision in designing data warehouse.

Existing two keyword search algorithms, such as Object Rank and Personalized Page Rank provides high quality and high recall search in databases, and the web. These algorithms require a query time for iterative computation over the full graph which is too expensive for large graph, and not feasible at query time. Page Rank algorithm utilizes the Web graph link structure to assign global importance to Web pages. The Page Rank score is independent of a keyword query. Recently, dynamic versions of the PageRank algorithm have become popular. They are characterized by a query-specific choice of the random walk starting points. Personalized Page Rank (PPR) is used for web graph data sets. Object Rank uses a query term positing list as a set of random walk starting points and conducts the walk on the instance graph of the database. However, Object Rank suffers from the same scalability issues as personalized Page Rank, as it requires multiple iterations over all nodes and links of the entire database graph. Personalized Page Rank: In particular, two algorithms have got a lot of attention: Personalized Page Rank (PPR) for Web graph data sets and Object Rank for graph-modeled databases. PPR is a modification of Page Rank that performs search personalized on a preference set that contains Web pages that a user likes. For a given preference set, PPR performs a very expensive fix point iterative computation over the entire Web graph, while it generates personalized search results. Therefore, the issue of scalability of PPR has attracted a lot of attention. Object Rank: Object Rank has successfully been applied to databases that have social networking components, such as bibliographic data and collaborative product design. However, Object Rank suffers from the same scalability issues as personalized Page Rank, as it requires multiple iterations over all nodes and links of the entire database graph. Here we use a Bin Rank that employs a hybrid approach where query time can be traded off for preprocessing. Bin Rank query execution easily scales to large cluster by distributing the sub-graphs between the nodes of cluster. This way more sub-graphs can be kept in RAM, thus decreasing the average query execution time. Since the distribution of the query terms in a dictionary is usually very uneven, the throughput of the system is greatly improved by keeping duplicates of popular sub graphs on multiple nodes of the cluster. The Query term is routed to the least busy node that has the corresponding sub graph. Thus, Bin Rank algorithm finds application in efficient web mining and database searching.

II. LITERATURE SURVEY

Harinarayan et al[1] presented a greedy algorithm for the selection of materialized views so that query evaluation costs can be optimized in the special case of “data cubes”. However, the costs for view maintenance and storage were not addressed in this piece of work.

Issam Hamdi, Emma Bouazizi and Jamel Feki[2] 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.

Himanshu Gupta and Inderpal Singh Mumick[3] developed a greedy algorithm to incorporate the maintenance cost and storage constraint in the selection of data warehouse materialized views. "AND-OR" view graphs were introduced to represent all the possible ways to generate warehouse views such that the best query path can be utilized to optimize query.

Ziqiang Wang and Dexian Zhang[4] proposed a modified genetic algorithm for the selection of a set of views for materialization. The proposed algorithm is superior to heuristic algorithm and conventional genetic algorithm in finding optimal solutions.

III. PROPOSED METHODOLOGY

Data warehouse contains the collection of many materialized views which are used to answer some OLAP aggregate queries. This work has implemented selection & maintenance of materialized views. Those queries which are accessing frequently can materialize for good query performance. Therefore this work will select those queries which are accessing more frequently & will cross the threshold value select for materialization. Cluster based approach is used A cluster of data objects can be treated as one group. While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups. The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups. In this approach similar query will be clustered according to their query access frequency to select the MV that will reduce execution time and storage space. In next phase the code is written for execution of algorithm for that we employ Java language and eclipse tool will be used. For the data storage SQL server 8 have to us.

Bin Rank Algorithm

We use a Bin Rank system that provides user with personalized search and reduce query processing time and storage. For this, it employs a hybrid approach. This system maintains a separate record for each user and stores his/her search results according to the relevance of documents. User can maintain his/her personal search catalogue, can see his/her own search history, can get the results for queries which are stored in database within less time than time required to it search over web. It provides a fast search over previous queries giving the user an optimized result. User can select the category words for searching and he can add his own category words. This system also maintains ranking list of frequently used websites. The algorithm used for Bin Construction is given below.

Input: A set of workload terms W, with their posting lists
Output: A set of bins B
1. while W is not empty do
2. create a new empty bin b and empty cache of candidate terms C
3. pick term t∈W with the largest posting list size |t|
4. while t is not null do
5. add t to b, and remove it from W
6. compare a set of terms T that co-occur with t
7. for each t′∈T do
8. insert (or update) mapping < t′, null> into C
9. end
10. for each best I :=0
11. for each mapping < c , i > ∈ C do
12. if i=null then i:=|b|
13. update mapping < c, I > in C
14. end if
15. union :=|b| + |c| - i
16. if union > maxBinSize then
17. remove < c,I > from C
18. else if i > bestI then bestI := i, t :=c
19. end if
20. end for each
21. if bestI = 0 then pick t∈W with maximum |t| ≤ maxBinSize - |b|
22. if no such t exists, t :=null
23. end if
24. end while

25. add completed b to B
26. end while

Modules

I Phase: Data Loading and Data Cleaning

II Phase: Clusters are form depends on attribute values and they are distributed on different systems.

III Phase: Implementing materialized view and fire queries on data warehouse.

IV Phase: Comparison of result with materialized view and without materialized view.

Data Loading and Data Cleaning

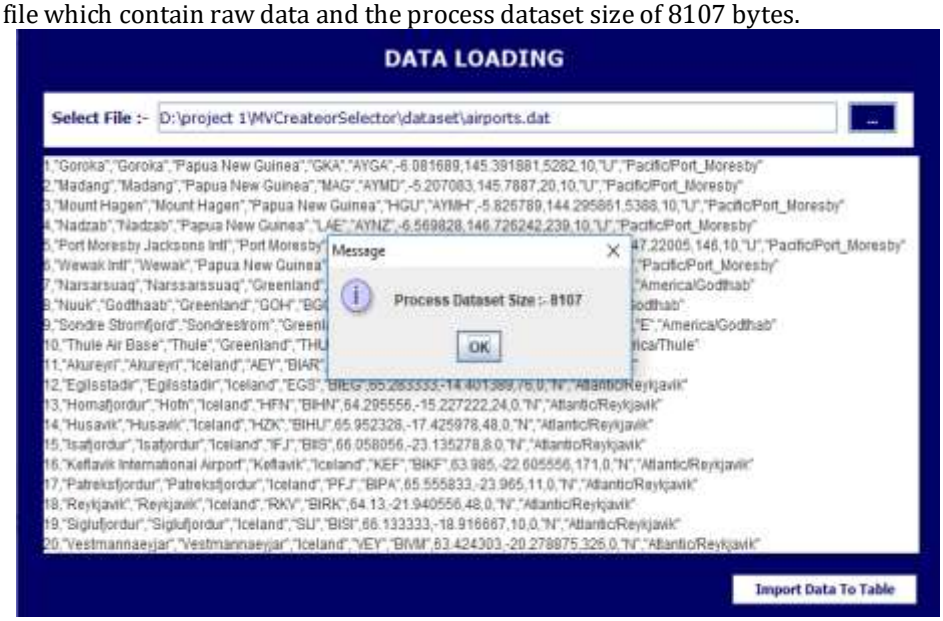
In this phase we collect raw data i.e.dat file is used. This raw data is loaded in table. After cleaning the data, data warehouse is created.

Data Clustering

Clustering is the process of making a group of abstract objects into classes of similar objects. A cluster of data objects can be treated as one group. While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.

IV. RESULTS

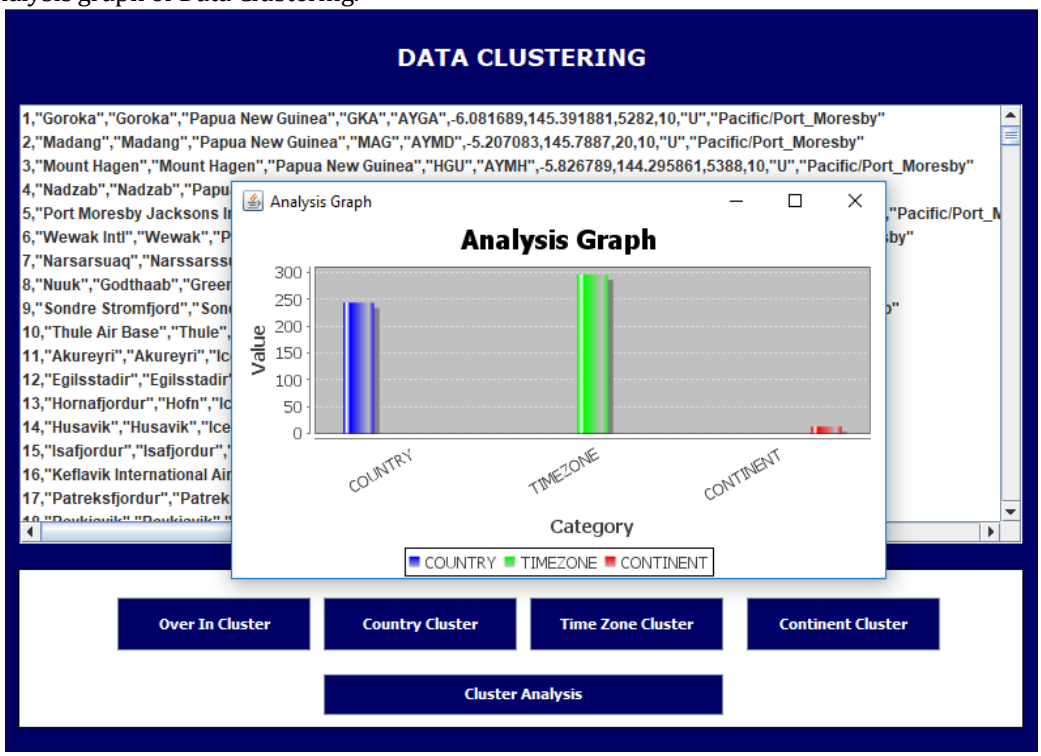
1. Select the file which contain raw data and the process dataset size of 8107 bytes.



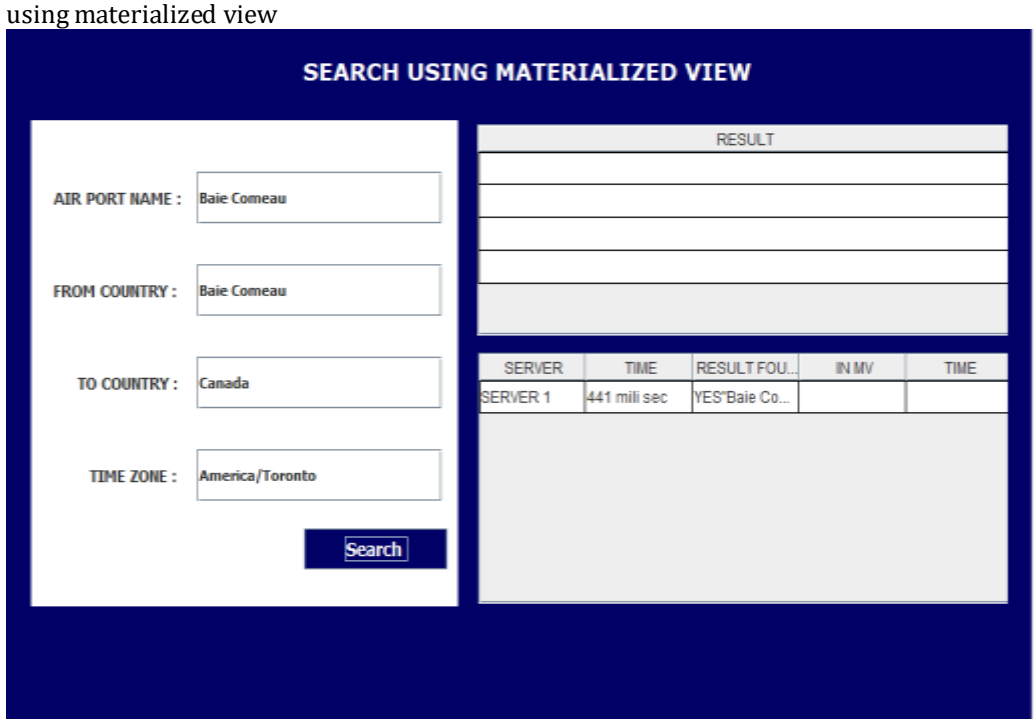
2. After clicking on the clean button the raw data is cleaned and arrange in the table form.

id	airport	fromcity	tocity	fromcode	tocode	latitude	longitude	altitude	timezone
1	Goroka	Goroka	Papua Ne...	GKA	AYGA	-6.081689	145.3918...	5282	Pacific/Po...
2	Madang	Madang	Papua Ne...	MAG	AYMD	-5.207083	145.7887	20	Pacific/Po...
3	Mount Ha...	Mount Ha...	Papua Ne...	HGU	AYMH	-5.826789	144.2958...	5388	Pacific/Po...
4	Nadzab	Nadzab	Papua Ne...	LAE	AYNZ	-6.569828	146.7262...	239	Pacific/Po...
5	Port More...	Port More...	Papua Ne...	POM	AYPY	-9.443383	147.22005	146	Pacific/Po...
6	Wewak Intl	Wewak	Papua Ne...	WWK	AYWK	-3.583828	143.6691...	19	Pacific/Po...
7	Narsarsu...	Narssars...	Greenland	UAK	BGBW	61.160517	-45.425978	112	America/...
8	Nuuk	Godthaab	Greenland	GOH	BGGH	64.190922	-51.678064	283	America/...
9	Sondre St...	Sondrestr...	Greenland	SFJ	BGSF	67.016969	-50.689325	165	America/...
10	Thule Air ...	Thule	Greenland	THU	BGTL	76.531203	-68.703161	251	America/T...
11	Akureyri	Akureyri	Iceland	AEY	BIAR	65.659994	-18.072703	6	Atlantic/R...
12	Egilsstadir	Egilsstadir	Iceland	EGS	BIEG	65.283333	-14.401389	76	Atlantic/R...
13	Homafjor...	Hofn	Iceland	HFN	BIHN	64.295556	-15.227222	24	Atlantic/R...
14	Husavik	Husavik	Iceland	HZK	BIHU	65.952328	-17.425978	48	Atlantic/R...
15	Isafjordur	Isafjordur	Iceland	IFJ	BIIS	66.058056	-23.135278	8	Atlantic/R...
16	Keflavik In...	Keflavik	Iceland	KEF	BIKF	63.985	-22.605556	171	Atlantic/R...
17	Patreksfjo...	Patreksfjo...	Iceland	PFJ	BIPA	65.555833	-23.965	11	Atlantic/R...
18	Reykjavik	Reykjavik	Iceland	RKV	BIRK	64.13	-21.940556	48	Atlantic/R...

3. The analysis graph of Data Clustering.



4 Search using materialized view



VI. CONCLUSION

By implementing the materialized view in distributed system, we are trying to get the combination of good query response time from which query processing cost should be minimized. Thus, implementation of materialized reduces the access time required to get the data from the data warehouse. Use of Bin Rank algorithm increases performance of system. Implementation of materialized view is one of the best option to improve performance of data warehouse.

VII. REFERENCES

1. Harinarayan, V.; Rajaraman, A.; Ullman, J.: Implementing Data Cubes Efficiently,in: Proc. of the 1996 ACM Int. Conf. on Management of Data(SIGMOD´96, Montreal, Quebec, 4.-6. Juni), 1996
2. 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
3. H. Gupta, I.S. Mumick, Selection of views to materialize under a maintenance cost constraint. In Proc. 7th International Conference on Database Theory (ICDT'99), Jerusalem, Israel, pp. 453–470,1999.
4. Ziqiang Wang, Dexian Zhang, "Optimal Genetic View Selection Algorithm Under Space Constraint," Proceedings of the International Conference on Data Warehousing and Knowledge Discovery , LNCS, vol. 1676, pp. 116–125, 999.
5. T.D. Khadtare, P.R. Thakare, S.A.J. Patel "An Efficient Personalized Web Search Mechanism using BinRank Algorithm" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-4, March 2013
6. Hema S. Botre, M.S.Chaudhari "Design & implementation of an algorithm for materialized view selection" International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 1 Issue 10, December- 2012
7. Gorettiv K.Y. Chan, Qing Li and Ling Feng, "Optimized Design of Materialized Views in a Real-Life Data Warehousing Environment," Internation Journal ofInformation Technology Vol.7, No 1 Sept 2008