Data Warehouse Performance Optimization Implementing DHE Algorithm in Mortgage Backed Security using Mondrian

Imran Momin[1], Imran Amin[2]

[1]MSCS Student, SZABIST
Karachi
immomin786@yahoo.com

[2]Faculty of computing, SZABIST
Karachi
Imran.amin@szabist.edu.pk

Abstract—*OLAP (Online Analytical Processing) means analyzing large quantities of data in real-time. It requires massive amount of processing time to extract information from data warehouse cubes. Business requires online reporting\information these days even for historical data that spans years if not decades. Data warehousing helps in making the retrieval of that data easier by aggregating large datasets in a summarized format. One of the key issues is aggregation of the data. There are different techniques that can be used to overcome the query processing and obtain quick answers for their queries. The design of data warehouse itself plays a virtual role.*

This research investigates different performance optimization techniques that apply at data warehouse design. This paper also discusses the implementation of DHE (Dimension Hierarchical Encoding) that significantly reduces the disc I/Os and improves the OLAP queries processing, using open source Mondrian OLAP system. The comparative analysis and experimental results demonstrate that using DHE algorithm in MBS (Mortgage Backed Securities) financial domain helps to achieve high performance.

Index Term— *Data warehouse, performance optimization, Mondrian, DHE algorithm.*

1. INTRODUCTION

Today the database size is more than a terabytes. Querying on such a massive dataset to perform analysis has become a challenging task. In this type of analysis the data warehouse plays an important role and usually consists of historical data that is derived from transactional data. It separates analysis workload from transaction workload and enables a business to consolidate data from heterogeneous sources in supporting management decision making process. A number of approaches are used to reduce query response time such as indexing [1], view [2], pre-calculated summary tables, and optimization query.

OLAP uses a technique called multidimensional analysis. Relational database stores all data in the form of rows and columns, whereas a multidimensional data consists of axes and cells.

In the financial application where huge transactions are performed on a day-to-day basis, there is a need for such types of system that help management to take key decisions. From the management point of view, the cost of implementation in such kinds of systems is one of the problems. Mondrian is an open source OLAP tool that solves the aforementioned problem.

2. DATA WAREHOUSE CONCEPT

Data warehouse stores the data related to a particular problem e.g. sales information, HR, inventory. The data is tightly coupled with the relationships that cover all the angles that can be used to analyze particular scenarios. Also, the store keeps periodic data that helps in comparing/analyzing different periods. In the end the store provides vital information on top of which management can perform analysis and take decisions of a strategic nature. Also the warehouse can be broken into smaller parts for organizing huge chunks of data into an organized form.

Data warehousing uses ETL (Extractions, Transformation and Loading) framework to integrate heterogeneous information sources in organizations, and to enable online analytic processing. Once the data comes from

OLTP (Online Transactional Processing) to OLAP it is never updated or deleted; in other words data warehouse is read only.

There are three types of storage modes in OLAP:

- MOLAP: In this type, data is stored in multidimensional cubes. The storage is not in the relational database but in proprietary format.
- ROLAP: In this type, data is stored in relational database, and it does not require pre-computation and storage of information.
- HOLAP: It is a combination of both MOLAP and ROLAP, where most recently used data is stored in multidimensional cubes and detailed data is stored in the relational database.

3. DHE ALGORITHM

DHEGA (Grouping Aggregate Based on the Dimension Hierarchical Encoding) is a grouping aggregation algorithm. It substitutes the original key and saves storage and reduces the multi-table join between fact and dimension tables [3].

In the star schema the new attribute in DHE attribute is added to the corresponding dimension member for each dimension table and it refers to the fact table [3].

Dimension Hierarchical Encoding: According to the author, 'In dimension Di, DHE of each dimension member is encoded by compounding the binary code of all the dimension hierarchical attributes (Li1, Li2, Lih) from top to down.' [3]

TABLE 1 : DHE OF TIME DIMENSION

TIMEID	EFFECTIVE YEAR	EFFECTIVE QUARTER	EFFECTIVE MONTH	BTIMEID
1	2000	1	1	00010010001
2	2000	2	2	00010100010
3	2000	3	3	00010110011
….	….	….	….	….
365	2008	4	12	10011001100

We have created a new DHE field of time dimension; Table 1 shows the BTimeID which represents binary code that will be used in the fact table. To generate the binary key for a single tuple we used h following formula.

Binary code for year:

⇨ 2008 – 2000 + 1 = 9
⇨ Binary code for 9 = 1001

Binary code for quarter:

⇨ Binary code for 4 = 100

Binary code for month:

⇨ Binary code for 12 = 1100

The DHE of 2008.4.12 is 10011001100, it is a combination of Byear=2008, Bquarter=4, Bmonth=12

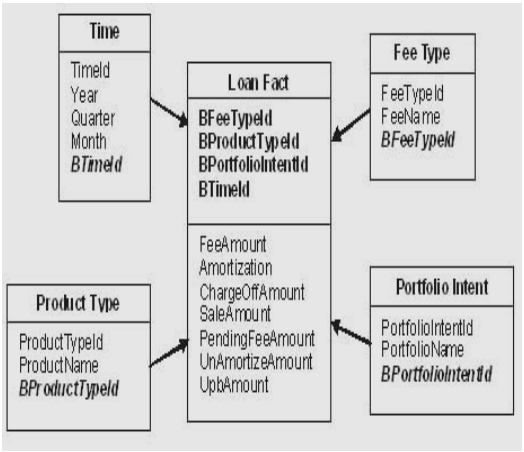

Figure 1 : Star Schema with DHE

Data warehouse with star schema, both multi table join and grouping aggregation are involved in OLAP queries.

Figure 1 shows the star schema of mortgage backed securities that implements DHE algorithm, in which DHE BFeeTypeId, BProductTypeId, BPortfolioIntentId, BTimeId to dimension table Time, Product, Fee and PortfolioIntent.

4. MONDRIAN ROLAP

Mondrian is an open source BI product of Pentaho. It is written in Java which being a platform independent language gives it the flexibility to target a wide variety of operating systems. Being open source also gives its clients the flexibility to poke around the code, add new

features, fix bugs etc. Mondrian is specifically a ROLAP. It has the philosophy to use existing infrastructure as much as possible and so it relies heavily on RDBMS to provide the storage and access capabilities [4].

It enables users to design the data warehouse to analyze a very large dataset store in RDBM (Relation Database Management) without writing SQL queries. It supports MDX (Multi Dimensional Expression) queries and XMLA (XML for Analysis) specification to extract data from data warehouse and return to end user to perform analysis

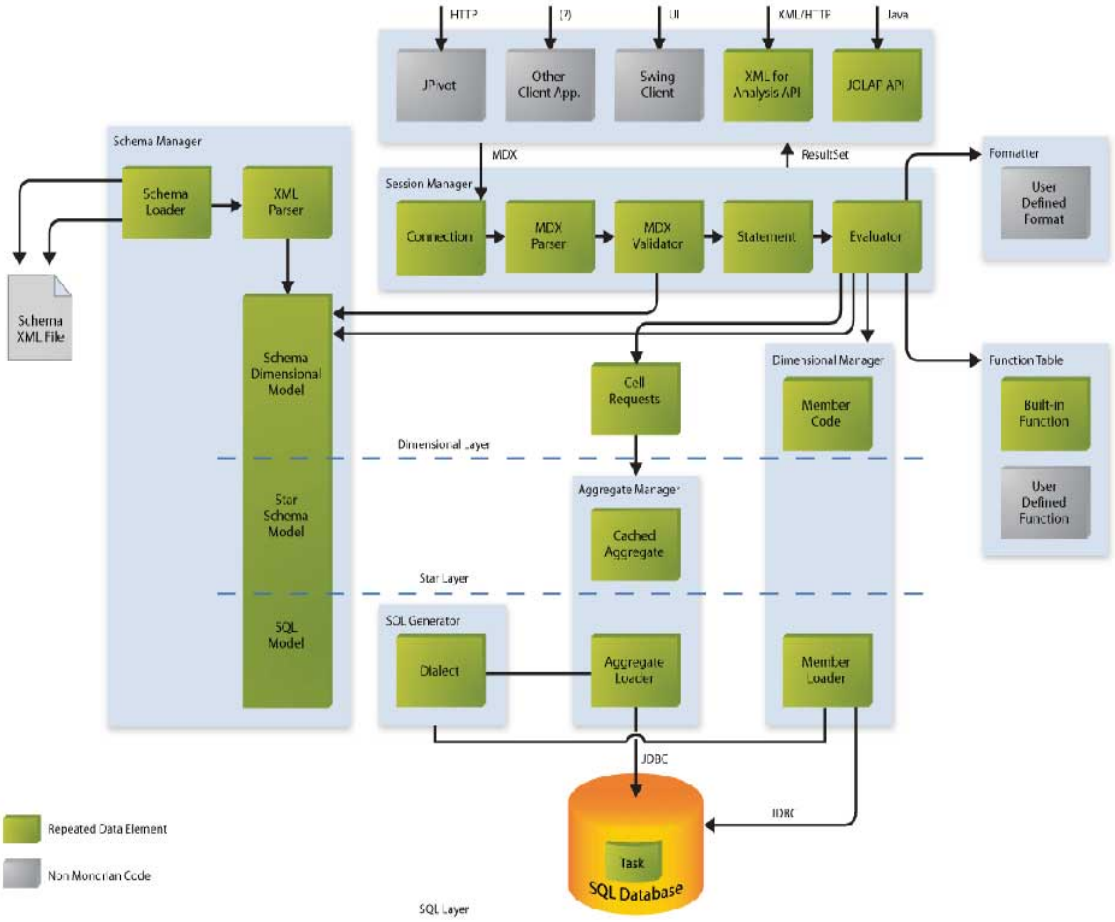

Figure 2 : Mondrian Architecture [4]

Figure 2 illustrates the architecture of Mondrian OLAP system.

It consists of four layers (presentation, dimensional, start and storage). The presentation layer is related to end-user who performs analysis based to the data returned from the OLAP query.

The second layer is responsible for validating and executing the MDX query. Maintaining an aggregate cache is the responsibility of the start layer, i.e. the third layer.

The last layer is an RDBMS and its responsibility is to provide aggregated data from dimension data.

Figure 3 illustrates the working of Mondrian. It takes MDX query as input and breaks into smaller pieces to extract data from dimensions and fact table.

Once all the data has been extracted, it aggregates queries' result into final output that returns it to the end user for decision making.

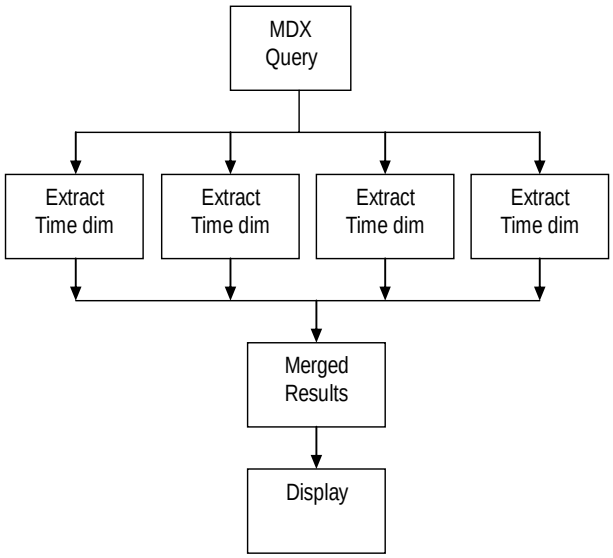

Figure 3 : Mondrian Query Processing

5. DW OPTIMIZATION TECHNIQUES

As data grows on analytical application, the query takes more time to fetch results because it does not support OLTP workload. There are different techniques the developer can adopt to tackle these issues: [5]

MOLAP: In this type data is stored in multidimensional cubes. The storage is not in the relational database but in proprietary format. De-normalizing the model that is to put data into single table to eliminate joins. Adding indexes to scanning large volumes of data.

First, the developers must understand the analysis and reporting requirement in addition to the above mentioned optimizations techniques[5]. The main reason for building pre-computed aggregate tables is improving performance.

Figure 4 shows seven distinct steps that are involved in the process of going from detail level to a summary level. The first step toward increasing performance is to implement views that make navigation easier.

Views typically do not affect the performance itself but help to create better SQL as a result of reduction in joins and it depends on database for instance Oracle which provides materialize view.

The next step is to add indexes, which will meet the performance requirement. The main advantage of using indexes is to allow the system to maintain the concurrency with the base data tables.

If performance is not achieved through de-normalization, then the next step is to build the summary table which contains pre-computed information.

From a developer's point of view, creating summary table is a cumbersome activity for example in a cube if there are 3 dimensions and 4 levels, the possible aggregate tables are M * N, that require higher cost in the time required for data management and resources for the disk storage.

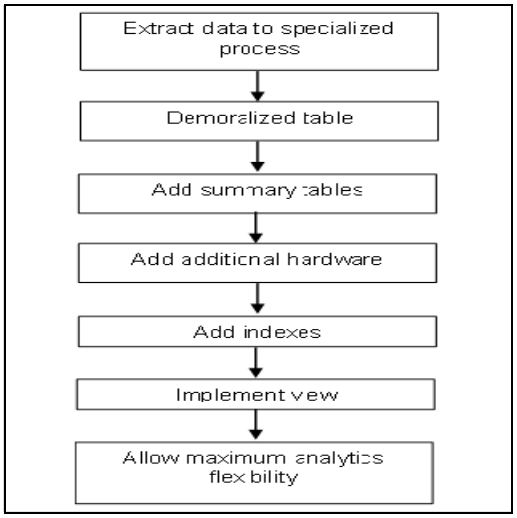

Figure 4 : Seven steps in data optimization [5]

6. REQUIREMENT ANALYSIS AND IMPLEMENTATION

To implement the data warehouse model, we selected Mortgage Backed Securities domain. The following are the requirements that were used to design the warehouse model:

- Subject Area: For MBS the subject area which we have selected is 'Loan Amortization Method'.
- Dimension: Time, Product Type, Portfolio Intent, Fee Type and Inventory.
- Fact : Loan Fee
- Measure: Beginning unamortized balance, beginning pending fee, total new fee, sale balance, ending balance.

The Mondrian Tool sets up OLAP cubes, fact table and dimension tables. All the configurations are written in Mondrian schema and it contains all logical model, cubes, share dimension, hierarchies and calculated members. Mondrian schemas are written in XML file that contains the implementation code for MBS data warehouse model.

Loan Detail Report

					Measures						
Date	Loan Type	Fee Type	Portfolio Intent	Inventory Id	Beginning Unam	Beginning Pending Fee	Total New Fees	Total Amortization	Proration Facility Balance	Charge Off Deferred Balance	Sale Balance
All	All	All	All	All			-26,156,378.3	7,880,569.08	0	0	5,571,143.826
1995	All	All	All	All			-220,755.36	4,660.414	0	0	0
1996	All	All	All	All	-216,094.946	0	-6,120	75,680.486	0	0	0
1997	All	All	All	All	-146,534.46	0	-32,042.13	145,371.502	0	0	0
1998	All	All	All	All	-33,205.088	0	-278,858.86	31,491.213	0	0	0
1999	All	All	All	All	-280,572.735	0	-279,280.82	58,268.998	0	0	0
2000	All	All	All	All	-499,920.557	-1,664	-589,134.303	89,597.582	0	0	0
2001	All	All	All	All	-999,457.278	0	-458,838.296	163,170.95	0	0	235,037.458
2002	All	All	All	All	-1,038,453.166	-21,634	-2,008,324.426	139,870.867	0	0	49,829.254
2003	All	All	All	All	-2,725,667.471	-131,410	-2,378,887.723	513,069.916	0	0	412,597.41
2004	All	All	All	All	-4,071,873.868	-107,014	-3,557,271.667	1,178,712.738	0	0	332,861.242
2005	All	All	All	All	-6,022,264.418	-95,307.136	-4,739,969.156	2,016,243.649	0	0	1,073,552.846
2006	All	All	All	All	-5,994,081.883	-1,678,355.196	-11,606,895.56	3,464,430.764	0	0	3,467,265.616

Figure 5 : Output Analysis Report

Figure 5 shows the output report that is used by the management for analysis. The results are extracted using the MDX query.

7. DISCUSSION

In this section we performed a comparative analysis between DHE algorithm and Normal Star schema. The analysis was executed on Core™ 2 Duo, clocked at 1.9 GHz with 2GB memory. The processor runs windows XP and implemented Java platform.

TABLE 2 : NUMBER OF RECORDS USED FOR ANALYSIS

DESCRIPTION	NO. OF RECORDS DATA SET 1
LOAN FEE FACT	41969
TIME DIMENSION	139
FEE TYPE DIMENSION	36
PRODUCT TYPE DIMENSION	8
PORTFOLIO DIMENSION	2
INVENTORY DIMENSION	1533

Table 2 shows the number of records in each fact and dimensions table used for the analysis of query processing with and without implementing DHE algorithm.

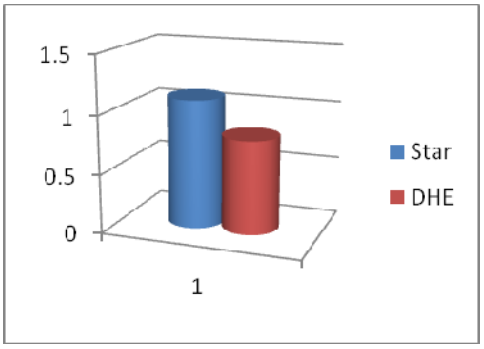

Figure 6 : Measurement of Time Dimension

From Figure 6, it is seen that the query to extract time data for all levels takes more time as compared to DHE. It shows that implementing DHE on dimension also saves some time. The analysis was performed on the same amount of data but in real time data warehouse dimension contains huge amounts of data, therefore some gain in performance can be obtained on populating dimension information.

Analyses are performed on multidimensions, meaning there is more than one way to obtain the analysis results. In the model there were 4 dimension tables and maximum time was consumed while analyzing the data at Inventory level. Converting the Inventory resulted in further performance improvement.

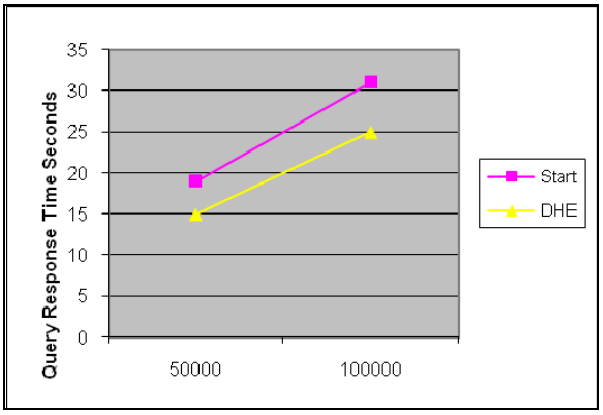

Figure 7 : Measurement of MDX Query Response Time

Figure 7 shows the overall measurement of MDX query that is being executed on two different datasets. It is evident that the DHE algorithm takes less time on 50K records as compared to non-star model. As data grows the query response time is reduced. The chart shows the gap between two points at 100K.

8. CONCLUSION

This paper discussed a way to implement cheap and robust data warehouse solution for the financial mortgage market. This paper also discusses the implementation of DHE (Dimension Hierarchical Encoding) that significantly reduces the disc I/Os and improves the OLAP queries processing using open source Mondrian OLAP system.

The future research in this area includes (1) Distributed data warehouse solution (2) Advanced algorithm to improve performance, and (3) Ways to implement MOLAP in Mondrian tool.

REFERENCES

[1] P. E. O'Neil, D. Quass. 'Improved query performance with variant indexes', Volume 26, issue 2, Jun 1997, pp. 38-49.

[2] H.Mistry, P. Roy et al. 'Materialized view selection and maintenance using multi-query optimization', Volume 30, issue 2, Jun 2001, pp. 307-318.

[3] Zhen-zhi Gong, Kong-fa Hu. 'A grouping aggregation algorithm based on dimension hierarchical encoding in data warehouse', cisim, pp.135-142, 2007 6th International Conference on Computer Information Systems and Industrial Management Applications, 2007

[4] (2008) The Mondrian website. [Online]. Available http://mondrian.pentaho.org/

[5] Rob Armstrong NCR. 'Seven steps to optimizing data warehouse performance', Computer, Vol. 34, Issue 12, Dec 2001, pg. 76-79.

[6] Amin Y. Noaman, Jen Barker. 'A Horizontal Fragmentation Algorithm for Fact Relation in a Distribute Data Warehouse', WSEAS Transactions on Computer Research, Vol. 3, Issue 3, Mar 2008.

[7] Chia-Cheng Hsu. 'The Pricing of Mortgage-Backed Securities', Master's Thesis, National Central University, Taiwan, June 2004.

[8] Dennis Vink. 'ABS, MBS AND CDO Compared: An Empirical Analysis', The Journal of Structured Finance, Vol. 14, No. 2, pp. 27-45, August 2008.

[9] Gary W. Hutto. 'Handbook of Mortgage Banking Financial Management', Mortgage Bankers Association of America (1999), 2nd Edition.