Ideal Strategy to Improve Datawarehouse Performance

Fahad Sultan
Faculty of information technology
University of Central Punjab
Lahore, Pakistan
Fahadsultan@ucp.edu.pk

Dr. Abdul Aziz
Faculty of Information Technology
University of Central Punjab
Lahore, Pakistan
aziz@ucp.edu.pk

***Abstract*—Data warehouse is set up for the benefits of business analysts and executives across all functional areas. The primary goal of data warehouse is to free the information locked up in the operational database so that decision makers and business analyst can make queries, analysis and planning regardless of the data changes in operational database. Data analysis in a large database requires some innovative techniques; there should be some new methods and procedures to look at the data with different angles. While executing adhoc queries over a large database degrade the query performance at considerable level because query will have to go through from large volume of data after making lot of joins. Moreover, as the number of queries is large, therefore, in certain cases there is reasonable probability that same query submitted by the one or multiple users at different times. Each time when query is executed, all the data of warehouse is analyzed to generate the result of that query. In this paper we purpose an ideal approach which definitely minimizes response time and improves the efficiency of data warehouse overall, particularly when data warehouse is updated at regular interval. This approach has been validated by Formal Method.**

Keywords-Data Warehouse; Perfomance; Formal Method

I. INTRODUCTION

A data warehouse is generally defined as a collection of subject-oriented, integrated, nonvolatile, and time-varying data to support decisions makers. A data warehouse is the copy of transaction data especially structured for querying, reporting and analysis purpose. The data warehouse contains copy of transaction which can not be updated or altered by the transaction system. Data warehouse is the source of stable and integrated data designed to support decision makers and business analysts. Data are acquired from various operational data stores across the enterprise. After acquisition of the data, cleansing, transformation and possibly de-normalization is been applied for better performance.

Analysts use the data warehouse to answer unlimited variety of questions, which may be very difficult to answer in operational database. Data warehouse contains a number of databases regardless of the number of sources and volume of data. The resulted warehouse is more homogeneous compared to operational data repository [1]. Data warehouse is often used by the large companies to analyze the data over time, and to check day to day operations. The primary goal is creating strategic planning resulting from long tern data analysis. We can create reports, projection, and business model and can forecast by these analysis [2]. Because data stores in the data warehouse is read only and intended to provide reporting. You can not update the data in the data warehouse by altering the records. The warehouse can be updating by adding more data from various sources and it keeping updated after a certain time period. The lifecycle of the data warehouse is continuing activity, it start form initial investigation till the requirement is met. As one phase of the data warehouse is completed, other phase is started because of the new data requirements and data sources. This lifecycle of the data warehouse will not end until it is valuable source of providing decision support information. Data warehouse is not used to keep all the data but it is used to store the necessary data for specific analysis [3].

II. DATA WAREHOUSE ARCHITECTURE

A data warehouse model consists of three layers. The lower layer consists of operational database that support one or more legacy system. Operational databases reveal the current state of the organization and evolve as business events takes place through capturing and posting of transaction. The middle layer is data warehouse layer that combine the data across and even outside of the organization and combine them so that it can be used for various modeling and decision support activities. The contents in the data warehouse remains static between consecutive refreshes so that people uses in decision support activities may work with a relative static copy of the transaction [4]. At the third layer we have data marts. Data mart is scaled down, lower cost version of data warehouse. The data in the data mart does not need to cleans and transform because it is already done when data is placed in the data warehouse. A data mart is designed to support the need of specific

group or user or department needs. The data mart is set of tables for direct access by the user. These tables are designed for aggregation, but cannot be used for traditional statistical analysis.

There could be many approaches for reloading data for data warehouse and data marts. For example the data warehouse may be refreshed periodically by the data warehouse administrator. Whereas data mart can be refreshed according to the demand, whenever user perceive that difference between operation system and data marts is increased to some threshold value [5]. The data warehouse architecture is depicted in the Figure 1, in which the data warehouse derives the information from internal and even across the internal source, export data to data mart, and data is provided to user by the use of staging engine [6].

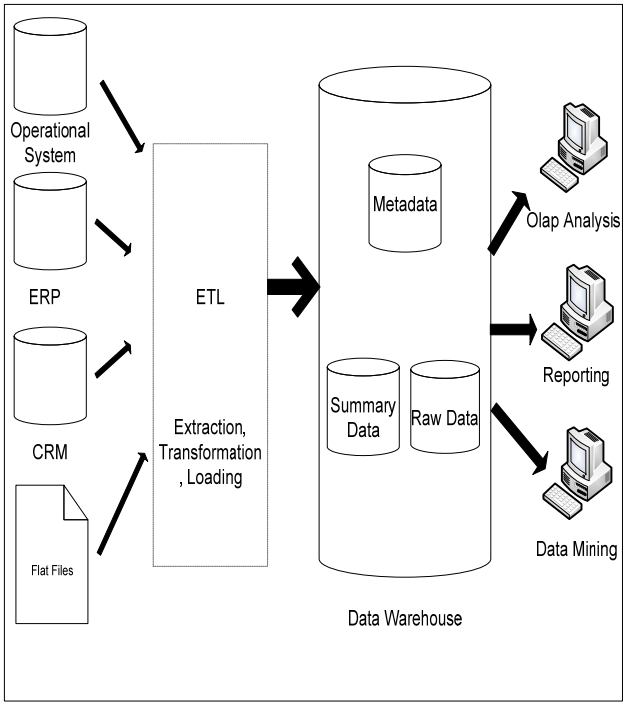

Figure 1. Data warehouse Arcitechure

The data warehouse is further advanced by the use of data mining. Data mining does not require the need of data warehouse but it could be the best platform to start data mining. Data mining is the search of useful business knowledge from a large amount of data which is previously hidden. Using information contained in the data warehouse a data miner can often provide answers to questions about an organization that a decision maker had previously not thought to ask [7].

III. DATA WAREHOUSE AND ITS IMPORTANCE

As we are getting more and more data, there should be some tools to analyze it. Another reason is that human brain has some limitation while processing multidimensional data. A third reason is that machine learning techniques becoming more affordable and efficient at the same time. Due to rapid development of computer technologies, organization realized the need to understand the data in depth. In doing so, they realized the need of collecting informational data for decision making and analyzing process [8]. However when they try to extract useful information from the operational data, they have experienced the following problems.

- They have to operationalize legacy and pre relational system.
- They have to run across historical, soft data and point in time data.
- They encounter with the varied data, inhibiting quick integration of data necessary for decision making process.

In start managers use traditional relational database management system, because they provide substantial data independence and a query language. These systems support both operational and decision making process. This approach was successful with the use of OLTP for the following reasons [9].

- Due to high complex query schema
- The extraction of useful information involve increasingly data analysis
- Locking content which are necessary for decision making support and operational workload degrades the performance of the system.

After realizing such problem the need of the data warehouse is realized. Data warehouse was developed to enable those organizations to integrate all sorts of data across server desperate sites. Data warehouse contains details, summarized and historical data allow drill down techniques to be applied to support decision making activities. Data is extracted from various operational data stores, which are then cleaned, transformed, reconciled, summarized and aggregated for being used by the data warehouse application [10].

IV. EXTRACTION TRANSFORMATION AND LOADING

Mostly, the information contained in a warehouse flows from the same operational systems that could not be directly used to provide strategic information. What makes a difference between the operation system and information contained in the data warehouse? It is the set of functions that fall under the broad group of data extraction, transformation, and loading (ETL). ETL functions reshape the relevant data from the source systems into useful information to be stored in the data warehouse. Without these functions, there would be no strategic information in the data warehouse. if source data taken from various sources is not cleanse, extracted properly, transformed and integrated in the proper way,

query process which is the backbone of the data warehouse could not happened [11]. The following tasks are involved in ETL:

- **Data Extraction** – The data contained in the data warehouse is extracted from the organization operational store. The data in the data warehouse can also be obtained from various external sources. Sources such as flat files, html, xml documents, text etc.
- **Data Transformation** – Data transformation techniques has been applied on the data to make it more uniform. After transformation the resulted data is more homogeneous and have less inconsistencies and errors. This enhanced the performance of data warehouse.
- **Data Loading** – After the cleansing and transformation of the data, uniformed data is placed in the warehouse.

The above tasks are often referred to as ETL – extraction, transformation, loading. They are frequently not well-defined distinct phases as suggested above. The extracted data will often be stored in a central staging area where it will cleanse and otherwise transformed before loading into the warehouse. An alternative approach to information integration is that of mediation: data is extracted from original data sources on demand when a query is posed, with transformation to produce a query result [12].

V. DATA WAREHOUSE TOOL

The process of selecting a data warehouse is totally relying on the existing infrastructure of the organization. Data warehouse use different tools than those used for the application development for example analysis tool, implementation tool, development tool, delivery tool etc. Organization use bit and pieces from different vendor and make them work together. They are only few vendors who are offering full integrated system such as IBM and Oracle [13]. If you are purchasing tools from different vendors, one thing should be considers that these tools should have high compatibility among other tools of the vendors. We will discuss tera data, one of the well known data warehouse tool.

A. Teradata

Tera data is suite of connectivity product which provides uses to access the transform view of the data. Teradata is product of RazorSql which contains visual tools, SQL editor, import and export tool, query builder and table editor to work with the data [14]. Some of the Teradata features are explain below.

- Visual and GUI Tools – A graphical user interface allow user to crate tables and views. Further these tables and views can be altered and dropped. Database Browser – A tool to explore database objects such as system table, views, tables, columns, primary key, foreign keys, procedures etc.
- SQL Editor – Sql editor is available to run sql script and queries. Editor supports 20 programming languages such as SQL, PHP, HTML, XML, Java, and more. Import Tool – import data facility is available. Various type of files can be imported such as delimited files, spreadsheets etc.
- Export Tool – Export facility is also available. Data can be export in various formats such as Csv, spreadsheet, xml etc.
- Table Editor – you can physical change/replace the tables contents and its characteristics.
- SQL Query Builder – You can crate various types of SQL queries depending upon your need such as select, insert etc. you can also create involving more than one table by the use of joins.

VI. DATA WAREHOUSE APPLICATIONS

Data warehouse is designed to support senior executives, policy maker and business analyst in making decision support activities. The use of data warehouse is not limited to any particular field. Some of the data warehouse applications are mentioned below.

A. Marketing/Sales

Large number of data is produced by the shopping malls. Such raw data can be used to find the hidden pattern underlying that data. Identify the potential distinct customers in customer database and find their buying behavior and characteristics. Email or SMS marketing can be done on such target people to increase the sale and revenue.

B. Finance and Banking

- Determine the effectiveness of current standards and policies and help to evaluate the risk associated while modifying the policies. Support senior management in planning, marketing and financial decision making.
- Identify the potential customers and highly targeted mailing list can be generated. What is the likelihood that a certain customer will default or pay back on schedule?
- We can compare actual budget which is allocated for a project on monthly, yearly as month to date basis. We can identify the future need by analyzing the previous cash flows. We can instantly identify what are the key expense factors. A set key financial ratio and indicators can be generated. Help to evaluate performance on various factors such as product, geography

etc. After evaluation you can create better policies and procedures.

- Use corporate data to analyze the Cardholder spending by their location, merchant to develop direct mail promotion withy key merchants. We can focus on potential marketing partners. And we can make loyalty by making innovative and successful new products.

C. Health Care

- Instantly determine the performance of physician by comparing to peer group. Analyze historical data to determine success and failure attributes. We can check the outcome of decisions by analyzing its margin and revenue.
- Can improve quality be providing feedback to doctors regarding the results and cost of realizing these results. The decision of ordering test can be refined by the study of the result given to doctors. According to our skills and sources and we can evaluate what services are to be provided in house and what to outsource.
- Rapidly identify the data pattern that can predict future individual health problem by the use of data mining. Quickly isolate the patients who will not respond to specific procedures and operations.
- Identify what is the appropriate medical diagnosis for a patient. Using data mining we can quickly identify pattern indicating that certain providers are using unnecessary procedures, such as performing unnecessary test, keeping the patients in hospital longer than necessary, prescribing expensive medication where as less expensive is also effective.
- Analyze the historical date to indentifying how well the hospital is providing services. This will not only improve the time factor but it minimizes cost as well.
- Use corporate data to analyze the Cardholder spending by their location, merchant to develop direct mail promotion withy key merchants. You have choice of selecting best of breed marketing partners. So that you can flourish you business by introducing new products.

VII. QUERY CACHE

Firstly, we have to augment the data warehouse architecture by the addition of query cache. Query cache will keep record of recently executed queries. Query cache will also be responsible for keeping result of recently executed queries. The primary goal is to make the system intelligence at the warehouse level, so that system will remember the recent work it has performed. This memory will later used to answer the result of adhoc queries which has been earlier performed by the users. The cache is maintained at the warehouse level and contains a tuple (Q, QR) [15]. Where Q is the query and QR represent the result of the query. Whenever a query is submitted by the user, either directly or through the data mart, the cache memory is first examined to check whether requested query is already store in the cache. If the query is stored, the cache is used to answer the result of the query. Otherwise query is evaluated and result is provided to the end user or data mart. This can save significant time and enhance data warehouse performance by not re-evaluating the queries which are already stored in the cache. This enhanced feature is displayed in the Figure 2. How this enhanced feature will work, this can be explained by an example. Suppose you have a data warehouse of Superstore chain.

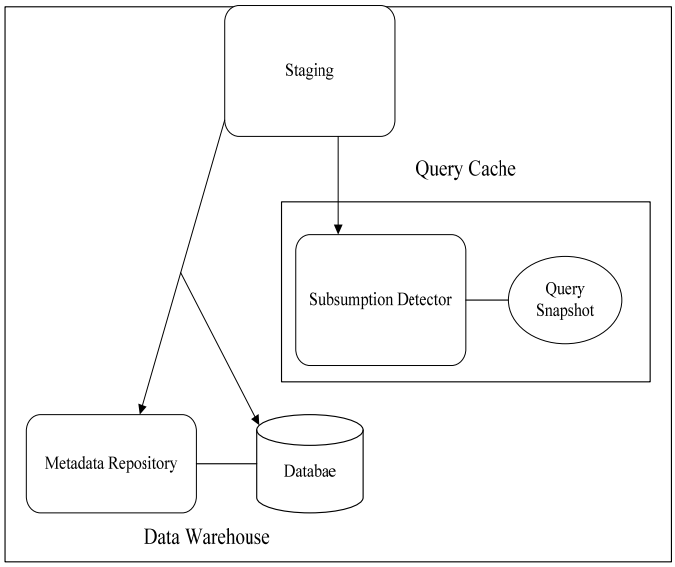

Figure 2. Query Cache in data warehouse

One of your business analyst place a query to show me the customer of store xyz, located in Pakistan of year 2000. The query will look like as follows: -

```
SELECT Name, Age, Income, ZipCode
        FROM Customers
        WHERE Store=xyz
              And
Country=Pakistan
              And Year = 2000
```

When the query is submitted, query cache will be examined to check whether this query is available or not. If it is not available, query will be evaluated and result will be store in the query cache. The results of the query are shown in the Table 1.

Table 1: Query 1 Results

Name	Age	Income	Zip code
Fahad	25	100$	54000
Saad	19	125$	54000
Sohail	30	105$	54000
Usman	29	106$	51000
Israr	26	107$	54000
Imran	27	100$	53000
Kiran	22	101$	53000
Sana	21	100$	54000

If any other user submitted the same query the result will be retrieved from query cache because that query is already stored in the cache. We will call this Query1. Let suppose another user wants to see the record of Store xyz, located in Pakistan of year 2000 and of city Lahore.

```
SELECT Name, Age, Income
       FROM Customers
       WHERE Store=xyz
              And
Country=Pakistan
              And Year = 2000
              And    Zipcode    =
54000
```

When the query is submitted, cache memory is examined. Same query is stored in the cache memory but there is only a difference of one addition constraint in query 2. If we apply drill down technique on Query 1 we can get the result of Query 2 as shown in Table 2.

Table 2: Query 2 Results

Name	Age	Income	Zip code
Fahad	25	100$	54000
Saad	19	125$	54000
Sohail	30	105$	54000
Israr	26	107$	54000
Sana	21	100$	54000

Cache query is capable of applying drill down techniques. Now result of Query 2 will be generated from the Query 1 result set instead of going through from all the data stored in the data warehouse. This process will save lot of time and effort required to go through all the records. There are a couple of points which need to be considered. What if the warehouse is updated with additional data? The query cache will provide the result of old data regardless of the new data is available in the warehouse. To cop up with this situation we will implement the use of Formal Methods.

VIII. Application of Formal Method

Formal method is used for the verification and validation of the system [16]. There are many formal method tools available, in this paper we will use Z-ves tool. Our problem is that we have a query and query result stored in the cache. But if the warehouse is updated with the new data the cache query result will reflect to old data. We will create a mechanism; if data is updated in warehouse the query will merge the old result plus the newly inserted data. For this the query will not have to go through from all of the data of warehouse, but it will evaluate query in newly added data. To achieve this functionality will construct a model using formal method. Figure 3 shows the state variables. The model consist abstract data type of Query, Query Result and indexes.

[Query].[QueryResult],[Indexes]

A query cache model has been designed using formal method in which we have defined the relation between queries and their results. Q is the set of all the queries that are submitted by the end users. QR is the set of all the records obtained by the query evaluation. Cache is a relationship between query and their associate result, which is used to keep track of all the queries and their results. Another relationship is maintained which keep track of all the quires and particular resulted row indexes. Domain of cache and QInd should be equal to the queries set. Where as the range of Cache should be the sub set of the QR.

QueryCache

Q: Π *Query*
*QR:*Π *QueryResult*
*Cache: Q*φ*QR*
QInd:Q ♣Π *Indexes*

dom *Cache* = *Q*
dom QInd = Q
ran Cache ζ QR

Figure 3. Query Cache Model

Whenever a query is submitted by the end user, the cache memory is examined in this relation (*Cache: Q*φ*QR*). If the query is available in the cache, the result will be submitted by accessing Cache. If it is not available then result will be evaluated and added to Query Cache as shown in figure 4.

Save Query Result

Δ QueryCache
Query?: Query
Result?: Π QueryResult_

Query? ∉ Q
Q'= Q Y {Query?}
QR' = QR Y { Result?}
Cache'= Cache Y {Query?□Result?}

Figure 4. Query Cache Model

Now the next step is how to calculate query result for which the data in the data warehouse is updated.

Query 1 is submitted by the user and his result is stored in the query cache. This is shown in figure 5. When next user submit the same query on updated data warehouse the query cache will check the flag bit if flag is true, it means the data warehouse is updated with new data. Now the query doesn't have to go through from all of the records. It will get the last index of the query result stored in the query cache. Then it will start searching the records which meet the query criteria from onward to that index. This can save lot of time and effort required to search the large amount of data.

The result obtained by the query are combined with the previous result by taking a union of both result set. The resultant data set will contains all the record which matches the query criteria. By doing so the query will only have to check newly added data and combining the result with old query.

Draw Query Result

Δ QueryCache
Query?: Query
Result?: Π QueryResult_
Flag?: N
OldCardinality:N
NewResult: Π QueryResult

OldCardinality=max{q:Query ∞ QInd (Query?)}
NewResult=A i:OldCardinality...#Result∞ (Query?)
Query ε dom Cache *f* Flag?=1 ⇒
Cache'= Cache Y {Query□NewResult}

Figure 5. Query Cache in data warehouse

IX. CONCLUSION

The computational cost for running a large data warehouse is very high. Accessing the information from sheer volume of data may degrade the performance, particularly when data warehouse is updated at regular interval. In this paper, we have introduced a strategy to improve the performance of data warehouse by minimizing the response time significantly. The primary purpose of this technique is to store queries and their corresponding results. If similar query is submitted by any other user the result will be obtained using cache memory. Working of the model has been shown using formal method. Formal method is used for the verification of the entire system. A model is created using formal method and some of the operations are created to show how data will be accessed and stored in query cache and how validation is performed. This dynamic strategy greatly improves the performance of over all data warehouse.

REFERENCES

[1] R. Winter and B. Strauch. A method for demand-driven information requirements analysis in data warehousing rojects. In Proc. HICSS, pages 1359–1365, Hawaii, 2003

[2] Albrecht, J.; Hümmer, W.; Lehner, W.; Schlesinger, L.: Using Semantics for Query Derivability in Data Warehouse Applications, appears in: Proceedings of the 4th International Conference on Flexible Query Answering Systems (FQAS'00, Warsaw, Poland, October 25 - 25), 2000

[3] Albrecht, J.; Günzel, H.; Lehner, W.: Set-Derivability of Multidimensional Aggregates, in: Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery (DAWAK'99, Florence, Italy, August 30 - September 1), 1999

[4] Albrecht, J.; Bauer, A.; Deyerling, O.; Günzel, H.; Hümmer, W.; Lehner, W.; Schlesinger, L.: Management of multidimensional Aggregates for efficient Online Analytical Processing, in: International Database Engineering and Applications Symposium (IDEAS'99, Montreal, Canada, August 1-3), 1999

[5] Cohen, S.; Nutt, W.; Serebrenik, A.: Rewriting Aggregate Queries Using Views, in: 18th Symposium on Principles of Database Systems (PODS'99, Philadelphia, Pennsylvania, USA, May 31 - June 2), 1999

[6] Yu, C.T.; Sun, W.: Automatic Knowledge Acquisition and Maintenance for Semantic Query Optimization, in: IEEE Transactions on Knowledge and Data Engineering (TKDE), 1989

[7] V. Markl and R. Bayer. Processing Relational OLAP Queries with UB-Trees and Multidimensional ierarchical Clustering. In Proceedings of DMDW 2000, June 5-6, 2000.

[8] Babad, Y.M., and Saharia, A.N. "Use of Stale Answers in Database Applications," Proceedings of the 13th International Conference on Information Systems, 1992

[9] S. Amer-Yahia and T. Johnson. Optimizing Queries on Compressed Bitmaps. In Proceedings of VLDB 2000, pages 329–338. Morgan Kaufmann, 2000.

[10] Barron, T., and Saharia, A.N. "Data Requirements in Statistical Decision Support Systems: Formulation and Some Results in Choosing Summaries," Decision Support Systems, Vol. 15, pp. 375-388, 1995

[11] Shim J.; Scheuermann, P.; Vingralek, R.: Dynamic Caching of Query Results for Decision Support Systems, in: Proceedings of the 11th International Conference on Scientific and Statistical Database Management (SSDBM'99, Cleveland, Ohio, USA, July 28-30)

[12] Nutt, W.; Sagiv, Y.; Shurin, S.: Deciding Equivalence among Aggregate Queries, in: 17th Symposium on Principles of Database Systems (PODS'98, Seattle,Washington, USA, June 1-3), 1998

[13] Larson, P.- A.;Yang, H.Z.: Computing Queries from Derived Relations, in: Proceedings of the 11th International Conference on Very Large Data Bases (VLDB'85, Stockholm, Schweden, August 21-23), 1985

[14] Cabibbo, L.; Torlone, R.: From a Procedural to a Visual Query Language for OLAP, in: Proceedings of the 10th nternational Conference on Scientific and Statistical Data Management (SSDBM'98, Capri, Italy, July 1-3), 1998

[15] Adiba, M.E., and Lindsay, B.G. "Database Snapshots," Proceedings of the 6thInternational Conference on VLDB, pp. 86-91, 1980

[16] J.A. Hall, "Seven Myths of Formal Methods," IEEE .SoftwareS, sept, pp 11-19, 1990

AUTHORS PROFILE

Fahad Sultan is MS student in Computer Science. He got his BS degree in CS from University of Central Punjab. Apart from studying he is working as Senior Software Engineer for a multinational company and has developed and deployed a large number of software. His research interest includes Data Mining, Advance Database Management Systems and Data warehousing.

Prof. Dr. Abdul Aziz did his M.Sc. from University of the Punjab, Pakistan in 1989; M.Phil and Ph.D in Computer Science from University of East Anglia, UK. He secured many honors and awards during his academic career from various institutions. He is currently working as full Professor at the University of Central Punjab, Lahore, Pakistan. He is the founder and Chair of Data Mining Research Group at UCP. Dr. Aziz has delivered lectures at many universities as guest speaker. He has published text books and large number of research papers in different refereed international journals and conferences. His research interests include Knowledge Discovery in Databases (KDD) - Data Mining, Pattern Recognition, Data Warehousing and Machine Learning.

He is member of editorial board for various well known international journals/ conferences including IEEE publications. (e-mail: aziz@ucp.edu.pk).