Query Optimizer for the ETL Process in Data Warehouses

Bhadresh Pandya¹, Dr. Sanjay Shah²
¹Professor, Department of Computer Science, Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat, India
²Director, S. V Institute of Computer Studies, Kadi, Gujarat, India

ABSTRACT

ETL (Extraction-Transformation-Loading) process is responsible for extracting data from several sources, cleansing, transforming, integrating and loading into a data warehouse. Extraction process accesses large amount of data by executing several complex queries in source databases. These queries are repetitive and executed at regular interval to refresh the data warehouse. Extraction of data from source must be completed in a certain time window; hence it is necessary to optimize its execution time. In this paper, we delve into the optimization of queries by recommending indices which reduces cost of the queries and improves performance of the queries.

Keywords: Extraction Transformation Loading, Data Warehouses, Query Optimizer, Business Intelligence, Execution Plan of Query, Database Tuning, Query Tuning, Performance Tuning

I. INTRODUCTION

In the area of business intelligence, data warehouse plays an important role. Designing and implementing a data warehouse requires using different tools and techniques to be used and applied to effective implementation and maintenance of data warehouse. The category of tools which are responsible for extraction of data from several source systems, their cleansing, transformation and inserting them into a data warehouse are called ETL tools. Execution time of these ETL process need to be optimized so that it can be completed in specified time window [1].

The functionality of ETL tools involve prominent tasks which include a) the identification and extraction of relevant information from source systems b) transformation and integration of information extracted from several source systems into a common format c) cleaning these data as per the database and business rules d) loading quality assured data to the data warehouse. The design and implementation of these tasks in the ETL process is labor-intensive activity which consumes large portion of the data warehouse projects [6].

Data extraction process involves execution of complex queries on the relational databases of operational source systems having large amount of data. In this paper, we propose the heuristic algorithm which works on certain key parameters of the query such as tables accesses, columns accesses, conditions applied on the columns, and using statistics of tables and columns it determines optimal index set to be built which generated better execution plan and reduces the cost of query.

II. METHODS AND MATERIAL

Data warehouses have become integral part of decision making process for the businesses. To provide enterprise level view of data for analysis of different functions, data from all the operational systems need to be cleaned, transformed and integrated at enterprise level. There is no de facto standard for architecture and construction of data warehouses. However quality and completeness of data coming from different heterogeneous sources play a
major role in success of data warehouse along with flexible and scalable architecture.

Enterprise database applications are often characterized by a large volume of data and high demand with regard to query response time and transaction throughput. Beside investing in new powerful hardware, database tuning plays an important role for fulfilling the requirements. However, database tuning requires a thorough knowledge about system internals, data characteristics, applications and the query workload. Among others index selection is a main tuning task. The problem is to decide how queries can be supported by creating indexes on certain columns [13].

The problem of low performance in data extraction of ETL process in the data warehouse can be critical because of the major impact in using the data from data warehouse, the project can be compromised. In this case there are several techniques that can be applied to reduce queries’ execution time and to improve the performance [12].

Reducing I/O

Locating some selected records from the large tables based on the selection criteria is a common task of extraction process. Deriving the result set efficiently from the operational databases is often difficult due the complex nature of the data and query. The simplest way of evaluating a query is to do a full scan of the tables and apply specified conditions which induces higher disk I/O. The most commonly used indexing method is the B-Tree which is very effective for on-line transaction processing (OLTP). Almost every database product has a version of this indexing method. As disk I/O operation is time consuming database operation, focusing on the efficiency of disk I/O is an effective means for improving performance and scalability [10].

The task of improving performance of query and building suitable indexes is done by database administrator, according to his knowledge and expertise. This is both subjective and quiet hard to achieve when number of queries is very large [11]. Proposed algorithm simplifies this task by recommending indexes which result in better execution plan reducing I/O significantly.

The Index Recommending Heuristic Algorithm

For the given query, capture details of tables accessed, columns accessed, and conditions applied on the columns

Collect statistics:
- Table statistics: Number of tuples, tuple size in bytes
- Column statistics: Number of distinct values

Assign ranking to the operators/conditions for execution

Assign column selection priority based on the assigned ranking and heuristic knowledge

List suggested indexes best suited for the query

Figure 1: Heuristic algorithm for index recommendation
III. RESULTS AND DISCUSSION

The algorithm was tested on real-time large databases on different modules in real-application environment. Examples of execution plan before putting recommended indexes in place and after creating are as below:

![Figure 2: Original Execution Plan of Query – A](image)

![Figure 3: Execution plan of Query – A after creating recommended indexes](image)
Figure - 4 : Original Execution Plan of Query – B

Figure - 5 : Execution plan of Query – B after creating recommended indexes
### Comparative Cost

<table>
<thead>
<tr>
<th>Query Label</th>
<th>System</th>
<th>Original Cost</th>
<th>Cost After Indexing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>ECQ</td>
<td>319</td>
<td>1</td>
</tr>
<tr>
<td>Q2</td>
<td>ECQ</td>
<td>922</td>
<td>513</td>
</tr>
<tr>
<td>Q3</td>
<td>ECQ</td>
<td>938</td>
<td>21</td>
</tr>
<tr>
<td>Q4</td>
<td>ECQ</td>
<td>1458</td>
<td>1</td>
</tr>
<tr>
<td>Q5</td>
<td>ECP</td>
<td>2009</td>
<td>1</td>
</tr>
<tr>
<td>Q6</td>
<td>ECQ</td>
<td>4752</td>
<td>561</td>
</tr>
<tr>
<td>Q7</td>
<td>CRQ</td>
<td>6077</td>
<td>3</td>
</tr>
<tr>
<td>Q8</td>
<td>CRP</td>
<td>8450</td>
<td>2466</td>
</tr>
<tr>
<td>Q9</td>
<td>CRQ</td>
<td>14673</td>
<td>15</td>
</tr>
<tr>
<td>Q10</td>
<td>CRP</td>
<td>440554</td>
<td>60627</td>
</tr>
</tbody>
</table>

### Figure 6: Comparative of Original Cost and Cost After Indexing

### IV. CONCLUSION

ETL process is a vital component of Data Warehouse responsible for its successful implementation. Extraction process extracts data from various operational source systems having large amount of data by executing several complex queries which need to completed in a specified time window. Database tuning and query tuning plays a major role in performance tuning, apart from adding new hardware. Results of proposed algorithm of recommending creation of indexes to tune performance queries are encouraging, which reduces cost of execution significantly and thus reducing execution time. Future work in this direction is planned to develop an automated tool to recommend set of indexes for ETL process tuning by including required details in the design of metadata repository as a part of total data warehouse architecture.

### V. REFERENCES

2. E. Malinowski, E. Zima nýi, Hierarchies in a multidimensional model: From conceptual modeling to logical representation, Data & Knowledge Engineering, 2005 Elsevier
4. Michel Schneider, Integrated vision of federated data warehouses, Data Integration and the Semantic Web, 2006
5. Songting Chen, Cheetah: A High Performance, Custom Data Warehouse on Top of MapReduce, 36th International Conference on Very Large Data Bases, September 13-17, 2010, Singapore.
6. Unmesh Dayal, Malu Castellanos, Alkis Simitsis, Kevin Wilkinson, Data Integration Flows for Business Intelligence, EDBT, 2009
7. Xuan-Thi Dung • Wenny Rahayu • David Taniar, A high performance integrated web data warehousing, Cluster Computing, 2007 - Springer
8. Benoit Daveville, Dinesh Das, Karl Dias, Khaid Yagoub, Mohamed Zait, Mohamed Zaiauddin, Automatic SQL Tuning in Oracle 10g, VLDB Conference, Canada 2004
13. Kai-Uwe Sattler, Eike Schallehn, Ingolf Geist, Autonomous Query-driven Index Tuning, in International Database Engineering & Applications Symposium, Portugal, 2004