Using Distributed Database Technology to Simplify the ETL Component of Data Warehouse

Abid Ahmad, Muhammad Zubair
Institute of Management & Information Sciences
CECOS University of Information Technology & Emerging Sciences
Peshawar, NWFP, Pakistan

Abstract: The increasing need for Decision Support Systems for business enterprises has led to tremendous growth of Data Warehouses. Within large enterprises number of data sources is increasing; due to which Data Warehouses are getting more and more complex. The ETL component- one of the basic components of data warehouse- can be made simpler with the use of Distributed Database technology in the development of data warehouse. This will also result in the establishment of an infrastructure for information sharing within the enterprise.

Keywords: Data Warehouse, ETL (Extract, Transform, Load), Business Intelligence (BI), Distributed Database Management System (DDBMS), Change Detection Component (CDC)

1. Introduction
Data warehousing is a collection of decision support technologies, aimed at enabling the knowledge worker (executives, managers, analysts) to make better and faster decisions. [1]. Typical architecture for data warehouse as discussed in [1] is shown in the figure 1. Here this architecture shows that data is extracted from OLTP Systems databases and other data sources. After the extraction, cleansing and transformation operations are performed and data is loaded to DWH database. Data marts are then created and OLAP servers are populated with data to serve analysis services. Of course bottom up approach can be used in which first data marts are created and the data marts are then integrated to form the data warehouse; or the top down approach can be used in which the data warehouse is created first and then its subsets (data marts) are created.

2. Literature Review
Data warehousing is a collection of decision support technologies, aimed at enabling the knowledge worker (executive, manager, and analyst) to make better and faster decisions. [8]. A simple Architecture for developing Data warehouse is proposed in [8]. This research also highlight that data warehouse can also be distributed, but this distribution is only for load balancing and scalability of the database.

As stated in a research study, business intelligence is typically performed by extracting relevant data from operational systems, transforming it into a common representation, and then loading it into a data warehouse. Business analysis through warehousing and Extract-Transform-and Load (ETL) techniques is widely used in large enterprises at the current time [9]. For developing a comprehensive business solution using a data warehouse, one or more middleware frameworks may be implemented to provide integration solutions, as indicated in a research study, such middleware frameworks may include publish-and subscribe and object request broker, remote procedure call (RPC), message-oriented middleware (MOM) and extract-transform-load (ETL) which moves, copies and/or merges data between databases. [3]

Here in this architecture the ETL Tool is responsible to locate the data sources, connect to the data sources and execute queries against that data sources to extract data. The component is also responsible for cleansing, transformation and loading the data into the DWH database. The ETL component also has to keep track of the changes made to the data sources and updates the DWH database accordingly. This shows that too much functionality is integrated into the ETL component. The ETL tool has to execute complex queries against the data sources to extract and transform data, which make the ETL component very complex. Section 3 of this paper provides a solution to made the ETL component more simpler and easily manageable.
operational source system, extract data relevant to the data warehouse, transform the source data according to the target convention of the data warehouse, cleanse the transformed data and periodically load the data into the data warehouse. Modern ETL systems are not limited to the use of only the known APIs (Application Programming Interface). Integration technologies such as UDI (Universal Data Integration), JDBC (Java Database Connectivity) and SOAP (Simple Object Access Protocol) are also used. Looking at the total expenditure of a data warehouse, ETL’s expense comprises between 50 and 80%. This makes ETL the principal success factor [4]. Current market analysis shows that data warehouse-based Business Intelligence (BI) solutions are increasingly used in medium-sized enterprises. Given that BI requires the data warehouse and that the data warehouse calls for ETL, BI, when employed, is a consistent platform for the integration of the company’s data world. With it, an excellent foundation is installed on which later efficient application and process integration can be created. The ETL-packages put together for companies support this process.

Thus the component of the data warehouse which has to perform ETL is a foremost component and serves as a basis for data warehouse. A fundamental fact about the ETL process, as stated in a research study is that the extract-transform-load (ETL) process is typically complex and its cost and difficulty are usually underestimated. [7]. Many research studies emphasize the importance and complexity of ETL process but relatively small amount of work is done to minimize the complexity of ETL component of a data warehouse. Some literature discussing the distributed nature of data warehouses as in [2], poses that source specific data extractor retrieves the desired source data and converts it into a uniform relational format. In this approach the ETL Component is still responsible to keep track of individual data stores. Much work is done on Data Mining as discussed in [5], on aggregate query processing as in [1], view Management as in [11], Physical database design [10] and other important issues in developing large and efficient Data Warehouses. But integrating operational data stores using distributed database technology to reduce the complexity and enhance the performance of ETL Component is relatively a new idea.

Distributed database technology is one of the more important developments of the past decade. A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network. A distributed database management system (distributed DBMS) is then defined as the software system that permits the management of the DDB and makes the distribution transparent to the users. According to this study in a distributed database data is stored at a number of sites. Each site is assumed to logically consist of a single processor. The processors at these sites are interconnected by a computer network rather than a multiprocessor configuration. The DDB is a database, not some “collection” of files that can be individually stored at each node of a computer network. The system has full functionality of a DBMS; it is neither a distributed file system, nor a transaction processing system [6].

Figure 2: Distributed database architecture

3. The proposed Solution

We suggest a new architecture for developing large data warehouses that integrate the operational data stores using the distributed database technology (shown in figure 2). The major aim of the proposed architecture is to reduce the complexity of the ETL Component of the Data warehouse.

Figure 3: Proposed architecture

Various components of the proposed architecture are discussed below.

3.1. Distributed DBMS Component
ETL component queries the DDBMS Component. These queries are relatively simpler. In Distributed database systems terminology we term these queries as ‘Global queries’. The DDBMS component has to transform these queries into the queries which are to be executed against individual OLTP system’s databases. These queries are then termed as ‘Fragment queries’ as these are to be executed against each data site. For the ETL component all the OLTP systems are integrated as a single data source due to which the data extraction queries for the ETL component are simpler. More over much of the transformation is done by the Distributed Database component which again simplifies transformation in ETL queries.

3.2. ETL (Extract, Transform & Load) Component

ETL component has to communicate with a single Distributed Database Management System. The queries issued by ETL component are relatively simple ones and the Distributed DBMS is now responsible to locate the data sources and execute queries against them. Using this approach will decrease the complexity of ETL component. The main logic of ETL component is regarding transformation & loading of data. The data is actually distributed over several data sources but the ETL component has the perception that data is placed locally under a single DBMS. Locating data sources and sending connection requests and then data

3.3. Change Detection Component (CDC)

A change detection component (CDC) can also be integrated to the proposed architecture. The CDC keep track of changes made to the OLTP systems databases since last extraction. This component can be integrated with the distributed DBMS component. This component can have a meta-data repository of its own or it can use the global conceptual schema of the distributed database component.

3.4. Information Sharing Infrastructure

Quick communication of useful information is a key factor in the success of business enterprises. Using this approach can also provide an infrastructure for information sharing within the enterprise. As the OLTP systems at different branches will be connected through a distributed database thus various branches and departments of the enterprise can share information among them. A very little effort will be required to establish such an information sharing system after using this approach.

4. Results and Analysis

Design and implementation of both the algorithms and then assessing the execution cases yield the following results. Let’s see both the traditional and proposed approach with the help of a diagram and then derive a relation representing performance gain of the proposed approach over the traditional approach.

![Diagram](image-url)
Considering figure a, EC = EC_1 + EC_2 + ... + EC_n

Thus we can say that
EC = \sum_{i=1}^{n} EC_i

Thus Total complexity T.C can be derived as
T.C = \sum_{i=1}^{n} C_i

Considering figure b,
EC = EC_i as i=1

Thus we can say that total complexity T.C can be determined as,
T.C = C_i

Comparing both the approaches to determine percentage performance gain,
P.G = \left( \frac{E.C \times C_i}{T.C} \right) \times 100

Simplifying the relation
P.G = \left( \frac{T.C_a - T.C_b}{T.C_a} \right) \times 100

We can find out the percentage performance gain using the equation.
P.G = \left( \frac{(20 - 8)}{20} \right) \times 100

P.G = 60%

Table 1: Best & Worst case execution

<table>
<thead>
<tr>
<th>Execution Situation</th>
<th>Example</th>
<th>Performance gain</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>For any T.C_a</td>
<td>T.C_b = 0</td>
<td>P.G = ((20-0)/20) x100</td>
<td>BEST CASE</td>
</tr>
<tr>
<td>For any T.C_a</td>
<td>T.C_b = 5</td>
<td>P.G = 75%</td>
<td>AVERAGE</td>
</tr>
<tr>
<td>For any T.C_a</td>
<td>T.C_b = 10</td>
<td>P.G = 50%</td>
<td>SATISFACTORY</td>
</tr>
<tr>
<td>For any T.C_a</td>
<td>T.C_b = 15</td>
<td>P.G = 25%</td>
<td>WORST CASE</td>
</tr>
</tbody>
</table>

For any T.C_a | T.C_b = 100% of T.C_a | Performance gain | Cases |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>For any T.C_a</td>
<td>T.C_b = 20</td>
<td>P.G = 0%</td>
<td>WORST CASE</td>
</tr>
</tbody>
</table>

5. Conclusion
Developing data warehouses is a must to develop Decision support systems in business enterprises. This paper proposes architecture to incorporate distributed database technology into the development of data warehouses. Using this approach will simplify the ETL Component of the data warehouse. The developers while developing ETL components will mainly concentrate on the extraction and transformation logic. This approach also enables the enterprises to establish...
an infrastructure for information sharing among several branches of the enterprise

6. Future Work

An ETL component will be developed using the traditional way as described in the architecture in [1] and an ETL component for the same system will be developed using the proposed architecture. The complexity and performance of both the approaches will be compared using several metrics & parameters like processing time, line of code, number of disk & network accesses.

7. References


[5] Ming-Syan Chen, Jiawei Han, Philip S. Yu, “Data Mining: An Overview from database perspective.”


