Incremental Data Migration in Multi-database Systems Using ETL Algorithm

Rajesh Yadav, Prabhakar Patil, Uday Nevase, Kalpak Trivedi, Prof. Bharati Patil

Department of Computer Engineering
Imperial College of Engineering and Research, Pune

Abstract— Nowadays Information is becoming an increasingly valuable corporate asset, the demand for a right tool to store, manage, and move this information in a cost efficient and reliable manner do arises. As part of an Information Lifecycle Management (ILM) best-practices strategy, the organizations require innovative solutions for migrating data among heterogeneous database systems and environments. To support this need, we as engineering students planned to design a powerful tool that enables high-performance and affordable data migration in a wide range of storage environments. This paper covers the unique challenges of data migration in dynamic IT environments and the key business advantages that we have designed to provide over traditional tools used for migration.

Index Terms— Databases, networks, middleware, multi-database and data migration

I. INTRODUCTION

Nowadays Database systems have become an essential part of any computer software, almost every computer system be it personal or corporate do have a database system. Low introduction and running cost being the main reason for this popularity. Having a look at the current trend when an organization starts, it usually starts as a Small-Scale industry and hence Microsoft Access Database is quite capable of handling its records and databases. But perceptibly as the company expands its conditions[5][6]

target data is unaffected what so ever be the data is unlocked. During the whole process of migration the transfer of data takes place where the target data is inserted to the destination database after the process of cleaning and deletion function are used for data migration. In this classical data migration technique that uses the locking, insertion and deletion function for processing data migration. In this technique, queries on the target data cannot be processed until the data migration is completed. Here, the turn-around time of other queries using this technique is more degraded than the snapshot technique wherein the turn-around time for data migration using this technique is shorter than the snapshot. Hence we consider only this classical data migration technique [4].

II. BRIEF HISTORY

Data migration techniques have always been a topic of research. Many reorganization techniques have already been proposed [2] [3] and the research on many is on the way. Some of these typical techniques proposed use the snapshot function to target data for the migration process. By using the snapshots, other queries can be processed while processing the data migration. But these techniques are not always adaptive hence other technique for data migration is necessary.

One way to solve the data migration issue is by using the classical data migration technique that uses the locking, insertion and deletion function for processing data migration. In this technique, queries on the target data cannot be processed until the data migration is completed. Here, the turn-around time of other queries using this technique is more degraded than the snapshot technique wherein the turn-around time for data migration using this technique is shorter than the snapshot. Hence we consider only this classical data migration technique [4].

III. PROPOSED SYSTEM

The system proposed is designed on the basis of the classical data migration scheme where locking, insertion and deletion function are used for data migration. In this scheme, target data in the source database is locked to prevent the execution of any queries on the target data until the data migration is completed. Some part of the destination database is also locked if necessary, there after transfer of data takes place where the target data is inserted to the destination database after the process of cleaning and transformation. Finally after successful migration the target data is unlocked. During the whole process of migration the target data is unaffected what so ever be the condition[5][6].
1. lock the target data in the source database system
2. get some locks in the destination database system if necessary
3. transfer the target data
4. insert the target data to the destination database system
5. unlock (according to 1 and 2)

Figure 4 Proposed scheme

The migration tool developed based upon the above scheme helps migrating database structures and data across various relational databases. The system will take an existing database in one format from the user and will convert it to a database in another format, which again will be specified by the user. The system will felicitate migrating data among MS-access, Sql server and Oracle.

Figure 5 Relation databases

IV. PROPOSED ARCHITECTURE

The system architecture designed for the data migration allows the validated user to select the source and destination database from the given options. The migration of database is carried out by migrating the tables first and then going for the actual migration. Before starting the migration process the connections for the source and destination are established and the detailed summary of the selected items is displayed to the user and he is then prompted to select the desired constraints. And finally the actual migration process is initiated.

Figure 6 System Architecture

V. ETL ALGORITHM

To achieve an effective data migration process, data on the old system is mapped to the new system providing a design for data extraction and data loading. This design relates old data formats to the new system's requirements and formats. Programmatic data migration may involve many phases but it minimally includes data extraction where data is read from the old system and data loading where data is written to the new system.

The algorithmic strategy used for the purpose is that of ETL where data from a relational database is read and after applying some transformation or business rules it is loaded to another database. Here it is assumed that the source data structures are generally not similar to target data structures.

The Plan

Basically extraction, transformation, and load require three main steps:
(i)Read the source data
(ii)Apply business and transformation rules
(iii)Load the data

Figure 7 ETL Architecture
After the process reads the data, it must transform the data by applying business and transformation rules to it. Applying business and transformation rules to target data means transforming codes, generating keys, converting data types, parsing, merging, and many other operations. Once the target data is in an appropriate business and technical format, the ETL process can load it into target tables.

**Extract**

The first part of an ETL process involves extracting the data from the source database. In most of the cases this is the most challenging aspect of ETL, since correct extraction of data sets the stage for how subsequent processes go further. In general, the goal of the extraction phase is to convert the data into a single format appropriate for transformation processing. The extract step should be designed in a way that it does not negatively affect the source system in terms of response time, performance or any kind of locking [8][9].

**Transform**

The transform stage applies a series of rules or functions to the extracted data from the source to derive the data for loading into the destination. Few data sources may require very little or even no manipulation of data. In some cases, one or more of the following transformations may be required to meet the business and technical needs of the target database:

(i) Selecting only certain columns to load (or selecting null columns not to load). For example, if the source data has four columns (also called attributes), for example roll_no, name, age, and salary, then the extraction may take only salary and roll_no. Similarly, the extraction mechanism may ignore all those records where salary is not present (salary = null) or just select some particular names.

(ii) Translating coded values (e.g., if the source database stores 1 for male and 2 for female, but the destination database stores M for male and F for female)

(iii) Encoding free-form values (e.g., mapping "Male" to "M")

(iv) Deriving a new calculated value (e.g., sale_amount = qty * unit_price)

(v) Applying any form of data validation. In case the validation fails, it may result in a partial, full or no rejection of the data, and hence none, some or all the data is handed over for the next step, which basically depends on the rule design and exception handling.[10]

**Loading**

The load phase loads the data into the target database. Here if the user as chosen to extract and load the full database then the whole database would be extracted and after applying transformation it would be loaded to the destination database else the user and specify the tables, views and stored procedures that need to be migrated.

**VI. APPLICABILITY**

**A. Individuals**

The database migration tool also becomes useful for individuals. Imagine a scenario where an individual has data divided into two different databases and he wants to combine the two, he can convert one part into the other’s format and use it. Thus database migratory comes handy here. The below figure explains how database migratory becomes handy for individuals. If the data is divided into two different formats then database in format 1 is used in the same format and the database in format 2 is given to the migrator to convert the database in format 2 to the database in format 1. This converted database is then combined with the previous database.

**B. Organizations**

According to the above mentioned scenario i.e. when a company starts it starts as a small-scale industry and hence MS Access database is quite capable of handling its records. But gradually with time as the company expands its database has to expand too thus the scenario of switching to a more efficient database do arises. Hence the proposed tool becomes very useful for the organizations when the size of their data expands and the necessity to migrate to another more efficient database arises.

**VII. CONCLUSION**

As the technology changes, working with the same old technology would surely bear some losses which any organization wouldn’t prefer. Hence the scenario of upgrading from the existing system to the new one do arises. And to do so the user needs to understand the new technology in order to work with it. Though importing, exporting or migrating data between different databases is very complicated and time consuming process, especially if these data sources store data in different formats, that’s where this tool comes into existence providing a interactive GUI to work with.
REFERENCES


[6] Jiawei Han, Michelinne Kamber, “Data Mining: Concepts and Techniques.”


[8] Panos Vassiliadis, Alkis Simitits Conceptual Modelling for ETL process”

[9] “DATA Quality in health Care Data warehouse Environments” Robert L.Leitheiser, University of Wisconsin – Whitepaper

[10] www.mysql.com