Database Migration over Network

Kanimozhi N[1], Thresphine J. R[2]

[1]M.TECH (Computer Science & Eng), PRIST University, Pondicherry India

[2]Assistant Professor (Computer Science & Eng), PRIST University, Pondicherry India

Abstract: ***In today's IT organizations need the right tools to store, manage, and move that information in the most reliable and cost-efficient manner. To support this requirement, we as planed to design a powerful tool that enables affordable, high-performance data migration in a wide range of storage environments. This project aims to bring up the idea of automation of transfer of data structure designs as well as data from one DBMS to another (possibly different type of) DBMS across the network. The system has a source side and destination side. At the source side, the user selects a given RDBMS and interrogates to get a list of tables present in the system. He chooses a table, selects the columns of the table to migrate and sets up a filter condition for the records to pick. The user also specifies the destination RDBMS system. The destination side application receives the Object, parses them and creates the required tables and records in the destination RDBMS. The project becomes very useful for an administrator who wishes to switch from one DBMS system to another. This migration is independent of different platform and secured by strong cryptographic algorithm, in which the entire data conversion becomes reliable, fast and efficient.***

Keywords: Data Migration, Database, Design, RDBMS, xml

1. Introduction

It may be necessary to move from one database vendor to another, or to upgrade the version of database software being used. The latter case is less likely to require a physical data migration, but this can happen with major upgrades. In these cases a physical transformation process may be required since the underlying data format can change significantly. This may or may not affect behavior in the applications layer, depending largely on whether the data manipulation language or protocol has changed - but modern applications are written to be agnostic to the database technology so that a change from Oracle to My SQL, DB2 or SQL Server should only require a testing cycle to be confident that both functional and non-functional performance has not been adversely affected. A method of migrating a database from a first server to a second server while continuing to provide transaction service, the method comprising the steps of: providing transaction service on the first server; establishing a database copy on the second server; logging at least one transaction from the first server to create a transaction log; executing the at least one logged transaction on the second server; repeating the steps of logging at least one transaction and executing the at least one logged transaction on the second server until a set point is met; queuing at least one transaction request; executing the at least one queued transaction request on the second server; and providing transaction service on the second server; wherein a time duration of each repeating step is necessarily shorter than a preceding repeating step, and transaction service on the second server.

2. Related Work

Large amount of work has already been done on distributed database systems [3]. Many algorithms have been implemented to achieve dynamic fragmentation and object allocation in distributed databases [4]. Complexities arouse while considering heterogeneous database systems [1], [2] in which sites are unaware of each other, database software and schemas used by the different systems may be different. Data migration and task Scheduling problem called Data consolidation [5] has been evaluated which applies to data-intensive applications that need more than one pieces of data for their execution. A software framework has been designed to alleviate the problem associated with task migration [6], [7]. Some of the solutions are proposed which can transfer data from hierarchical database to relational database [8]. Our framework was used to transfer task, data and state information across the platforms.

3. Existing System

In previous system "Database migration" is also possible. But it can migrate the data only within the system only. This is the main drawback of this system. There is no chance to migrate the data into one system to another system. If you want to perform you can need some external devices like CD, Floppy for transferring the data into another system etc

4. Proposed System

The system architecture of our migrating tool explains the overall working of the application.

Step 1 - The user is provided with the login facility. The user will first enter its username and password which are encrypted in our coding. The user's entered information is validated.
Step 2 - The user is then asked to select the source database which is to be migrated. The source database consists of MS Access, SQL Server and Oracle. After selecting the source database the user has to fill the details of the selected source database to create the connection.
Step 3 - After the connection is tested the user is asked to select the destination database to which the migration will take place. After selecting the destination database the user has to fill the details of the selected destination database to create the connection.
Step 4 - The user is asked to select the tables, columns of the table, views which are to be migrated.
Step 5 - the user is also provided with the facility to select either the schema of the table or rows of the table along with the constraints which are applied to the tables.

Step 6 - Click start migration button to start the migration. After the complete and successful migration of source database to destination database it displays the status report of the complete migration process.

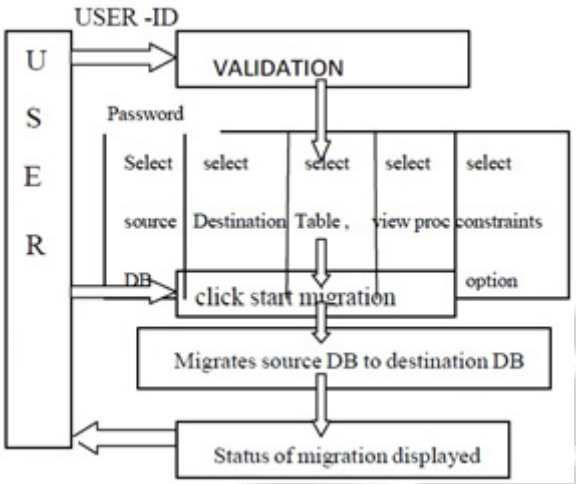

Figure 1: Architecture of the system

4.1 Design Specification

To design the system, we are considering following points: Front End: The front end of our application is designed using C# to make it look more attractive. Back End: The back end of our application includes coding for connections to our database and implements functionality of our application. The code should be optimized to make it more efficient. It avoids redundancy**.** It also reduces lines of code.

4.2 Project Plan

Purpose: "Database Migration Suite" is usually developed for individuals and organizations to save time for converting to a new database if a database already exists. The purpose of our project is to migrate data from an existing database to another database. Our migrator tool provides source databases as MSAccess, and destination databases as SQL Server and Oracle. The very purpose of our project is to provide flexibility to client to migrate his existing database into different database without any manual intervention.

Objectives: The objectives of our project are as follows:

1. To migrate Tables along with the following:
   - Primary key constraint
   - Foreign key constraint
   - Unique key constraint
   - Index key constraint
   - Check constraint
2. To migrate Views
3. To migrate selected columns.
4. Log

1. To migrate Table: The goal deals with migrating tables applying constraints, such as primary key, foreign key, unique key, index key and check constraints. Migrations of views are also supported. We have also provided facility for selecting and deselecting the fields of a table or an entire table itself for migration.
2. To migrate views: Views can be used to hide a part of database from certain user. Data access can be customized as per the needs of different users. Same data can be seen different way by the different user.
3. To migrate selected column: The goal deals with migrating selected column from source database to destination database.
4. Project Scope: The purpose of the project keeps the scope of project limited to following categories. They are:

The initial part deals with conventional migration keeping in mind i.e. we strictly adhere to the definition that "the source of data to be migrated is a database and the destination is also a database .While doing so the only part that undergoes any visible change is the structure of the source and destination database while all the data, attributes, table name etc remains unaffected. This is a classical case of database migration and every aspect regarding the migration is taken care of by tool called „ (Data migration suite) ". Our source database is an Access database, which is generally a norm for small databases and destination database is SQL Server which is an open source and capable of handling comparatively larger database efficiently.

The next part deals with migrating Tables applying constraints such as primary key, foreign key, unique key, index key and check constraints. Migrations of views are also supported. We have also provided facility for selecting and deselecting the fields of a table or an entire table itself for migration. The mapping rules have been defined in an MS Access file which is referenced by our migratory tool to map the various data types in the process of migration.

After the migration has occurred a status report will be displayed which will indicate if the table schema is migrated, records are inserted, constraints like primary key, foreign key, unique key, check constraints have been migrated successfully or pending or failed.

5. Modules Description

5.1 GUI Module

Actually every application has one user interface for accessing the entire application. In this application also we are providing one user interface for accessing this application. The user interface designed completely based on the end users. It is provide friendly accessing to the users. This user interface has attractive look and feel. Technically I am using the swings in core java for preparing this user interface.

5.2 Client Module

In this module we are prepare the code for client side activities. The client can get the data from the database and this data can be converted into one object than this object can be send to another client within the LAN. In this module the client have a facility to get the data from Oracle, CSV Files, Normal text files and Ms-Access etc. technically AWT, swings and network programming concepts are involved for preparing the client. JDBC concepts are used for getting the data from the different types of databases. In

this module TCP (Transaction Control Protocol) Protocol was involved for transferring the data from one host to another host.

5.3 Server Module

In this module we have to prepare the code to get the data from the client. The server can get the data from the clients and it can process the data based on the client requests. The server has facility to store the data into different types of storage areas like databases and files. Technically network programming concepts are involved for preparing the server. JDBC concepts are used for storing the data into databases like oracle, MS-Access, My SQL etc.

5.4 Migration Module

In this module we have to prepare the code to perform the process of transferring data between storage types, formats, or computer systems. Data migration is usually performed programmatically to achieve an automated migration, freeing up human resources from tedious tasks. This module can perform different types of migrations.

Type-1 Oracle to Oracle: The client1 system can get the data from oracle data base and the data can stored into the another oracle data base in another system within the LAN or within the system.
Type-2 CSV to Oracle: The Client1 system can get the data from .CSV files and the data can be stored into oracle database within the system or any other system within the LAN.
Type-3 Normal text file to oracle: The Client1 system can get the data from Normal Text files and the data can be stored into oracle database within the system or any other system within the LAN.
Type-4 Access to Oracle: The client1 system can get the data from Micro soft Access data base and the data can stored into the another oracle data base in another system within the LAN or within the system.

6. Applicability of System

6.1 Expanding Organizations

When a company starts, it usually starts as a Small-Scale industry. And hence Microsoft Access Database is quite capable of handling its records and databases. But gradually as the company expands its database has to expand too. Thus they have to switch to more efficient databases. Hence this software, “The Database Migration tool” is aimed at making the conversion of an MS access database, SQL Server. Using this tool the database migration becomes easy and efficient, without any manual effort and the need of technical knowledge.

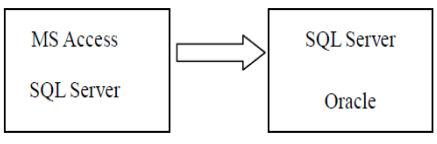

Figure 2: Expanding the Size of the database

The above figure simply shows the source databases and destination databases provided in our application. It provides four databases in source i.e. MS Access, SQL Server.

6.2 Individuals

The database migration tool also becomes useful for individuals. If an individual has data divided into two different databases and he wants to combine the two, then he can convert one part into the other formats and use it. Thus database migratory comes handy here.

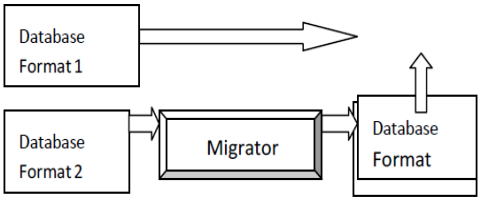

Figure 3: combining of different database

The above figure shows how the 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 migratory to convert the database in format 2 to the database in format 1. This converted database is then combined with the previous database.

7. Implementation Techniques

Cryptography, to most people, is concerned with keeping communications private. Encryption is the transformation of data into some unreadable form. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended. Decryption is the reverse of encryption; it is the transformation of encrypted data back into some intelligible form. Encryption and decryption require the use of some secret information, usually referred to as a key. The data to be encrypted is called as plain text. The encrypted data obtained as a result of encryption process is called as cipher text. Depending on the encryption mechanism used, the same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different. Same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different. In proposed approach we maintain server database with following fields-Unique name and id of sender and receiver, and encrypted key (Armstrong Number).

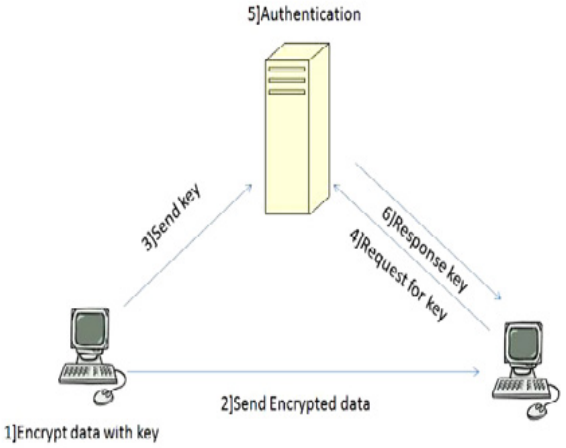

Figure 4: Server Architecture

Now, if sender A‟ wants to send data to receiver B‟, then he encrypts that data using randomly generated Armstrong number. That encrypted data is identified by unique timestamp given to it and sent to receiver. At the same time key (Armstrong Number) of encrypted data is sent to server with receiver „B‟ id and file name. Whenever receiver get that encrypted data he simply request for key to server. Now actual authentication is done by server, Server takes request from receiver with file name and receivers self id, and compare it with senders key name and receiver id. If both match then only that key is send to the receiver. Whenever receiver gets key now he can decrypt that data easily.

7.1 Illustration

Encryption: Step 1: Unimodular matrix is used to create encoding matrix given below. Take random Armstrong Number and add its total digits like. (n=1+5+3=9) and substitute it in Unimodular matrix as below

$$\begin{pmatrix} 8n^2+8n & 2n+1 & 4n \\ 4n^2+4n & n+1 & 2n+1 \\ 4n^2+4n+1 & n & 2n-1 \end{pmatrix}$$

After calculation Encoding matrix is

720 19 36
360 10 19
361 9 17

Step 2: (Encryption of the actual data begins here) Let the message to be transmitted be “ENCRYPT”. First find the ASCII equivalent of the above characters.

E N C R Y P T Extra Extra
69 78 67 82 89 80 84 -25 -25

Step 3: Now add these numbers with the digits of the Armstrong number Encrypted matrix as follows:

E N C R Y P T Extra Extra
69 78 67 82 89 80 84 -25 -25
+720 19 36 360 10 19 36 1 9 17
---
789 97 103 442 99 99 445 -16 -8

Step 4: Convert the above data into a matrix as follows:

$$A=\begin{pmatrix} 789 & 97 & 103 \\ 442 & 99 & 99 \\ 445 & -16 & -8 \end{pmatrix}$$

Step 5: Consider an encoding matrix...

$$B=\begin{pmatrix} 720 & 19 & 36 \\ 360 & 10 & 19 \\ 361 & 9 & 17 \end{pmatrix}$$

Step 6: After multiplying the two matrices (B * A) we get

$$C=\begin{pmatrix} 54262 & 56951 & 48860 \\ 27256 & 28495 & 24445 \\ 27075 & 28534 & 24482 \end{pmatrix}$$

The encrypted data is... 54262, 56951, 48860, 27256, 28495, 24445, 27075, 28534, 24482

The above values represent the encrypted form of the given message.

After storing this data into file it will be converted into byte array format as below:

-10, 119, -36, 120, 79, 125, -61, 118, -94.

Decryption:
Decryption involves the process of getting back the original data using decryption key.

*Step 1: (*Decryption of the original data begins here) The inverse of the encoding matrix is:

$$D=\begin{pmatrix} -1 & 1 & 1 \\ 43363 & -43508 & -43216 \\ -21682 & 21755 & 21608 \end{pmatrix}$$

Step 2: Multiply the decoding matrix with the encrypted data (C*D)
Step 3: Now transform the above result as given below:
789, 97, 103, -53830, -56733, -53405, 27581, 28400, 26872.
Step 4: Subtract with the digits of the Armstrong numbers & after converting the above data into byte array format and removing the extra parity bits we will get the original data as follows,

69 78 67 82 89 80 84

Step 5: Obtain the characters from the above ASCII equivalent:

E N C R Y P T
69 78 67 82 89 80 84

8. Conclusion

We know that various organizations use different type of databases to store their user's information. If we want to use the existing databases of such organizations then we have to convert the database in the format which is compactable to our application. This database migration is achieved by database migration tool. A "Database Migration Suite" is usually developed for individuals and organizations to save time for converting to a new database if a database already exists. The purpose of our project is to migrate data from an existing database to another database. Our migratory tool provides source databases as MS Access, destination databases as SQL Server and Oracle. We also have implemented migration from SQL Server to Oracle and vice versa. The purpose of our project is to provide flexibility to client to migrate his existing database into a different database without any manual intervention.

References

[1] J. Zhiquan, L. Chengfei, S. Zhongxiu, Z. Xiaofang, C. Peipei, and G.ianming, "Design and Implementation of a heterogeneous distributed database system," in *Journal of Computer Science and Technology,published by Springer Boston*, vol. 5, no. 4, pp. 363-373.

[2] Z. Gao, S. Luo, Y. Lin, and D. Ding, "A Grid-Based Integration Model of Heterogeneous Database Systems," in 2009 International Conference on Information Technology and Computer Science, vol. 2,pp. 126-129. 2009.

[3] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database Systems Concept, 4th ed, The Mc-Grill Companies, vol. 1, pp. 705-749. 2001.

[4] A. Sleit, W. AlMobaideen, S. Al-Areqi, and A. Yahya, "A DynamicObject Fragmentation and Replication Algorithm In Distributed Database Systems," in American Journal of Applied Sciences, vol. 4,no. 8, pp. 613-618. 2007.

[5] P. Kokkinos, K. Christodoulopoulos, A. Kretsis, and E. Varvarigos,"Data Consolidation: A Task Scheduling and Data Migration Technique for Grid Networks," in Eighth IEEE International Symposium on Cluster Computing and the Grid, 2008.

[6] M. Tungare, P. S. Pyla, and M. Sampat, P. Quinones, and M. A, "Syncables: A Framework to Support Seamless Data Migration Across Multiple Platforms" in Portable Information Devices PORTABLE07 IEEE International Conference, pp. 1-5.2007.

[7] X. Wang, L. Huang, and Y. Zhang, "A Grid Middleware--DISQ for Data Integration," in 2008 International Conference on Computer Science and Software Engineering, vol. 3, pp. 62-65, 2008.

[8] A. Meier, R. Dippold. J. Mercerat, A. Muriset, J.-C. Untersinger, R.Eckerlin, and F. Ferrara, Hierarchical to relational database migration, Vol. 11, no. 3, pp 21-27, 1994.