

Database Migration Over Heterogeneous Network

Rukhsar Parveen Tafazzul Husain

Department Of CSE

Jhulelal Institute of Technology, Nagpur.

Nisha Balani

Department Of CSE

Jhulelal Institute of Technology, Nagpur.

Abstract:- The propos of this research is to execute transactions in heterogeneous distributed databases. Instead of using the traditional approach of executing global transaction by remotely accessing distributed data, they propose that transactions be executed locally, and data is dynamically migrated to the appropriate sites. Thus, they eliminate the need for global transactions. Since there are no global transactions, the problem of distributed commitment does not arise. This is an important issue related to database recovery that is often ignored by protocols for transaction processing in heterogeneous databases. A special protocol is executed for migrating data objects. They present a protocol for localizing the access of a data object.

INTRODUCTION

Data migration is the process of moving data between storage units or entire computer systems. In order for this process to be efficient, powerful data extraction and data loading designs are critical. These designs help in mapping data, which is present on the current system, to the new system which is being implemented. One of the categories of Data migration is Database Migration. Database migration is the process of moving the business logic, schema, physical data and database dependencies from a current system to a different/new system. Database Migration is used when it is required to shift from one database vendor to another. This may be because of various reasons such as cost, capabilities, functionalities, requirements etc. This paper deals with Database Migration exclusively and not Data Migration as a whole.

The structure of a database is evolving as we develop and maintain a database-driven application. For example, during development, we may want to add a new table; or after the application is put into production, we may realize the need of adding an index on a column. It is important to keep track of these structural database changes (called **migration**) like we do with our source code. If the source code and the database are out of sync, it is very likely the whole system may break. For this reason, database migration tool that can keep track of database migration history, apply new migrations, or revert existing ones.



Fig1.Database Transfer

HETEROGENEOUS NETWORK

A heterogeneous network is a network connecting computers and other devices with different operating systems and/or protocols. For example, local area networks (LANs) that connect Microsoft Windows and Linux based personal computers with Apple Macintosh computers are heterogeneous. The word heterogeneous network is also used in wireless networks using different access technologies. For example, a wireless network which provides a service through a wireless LAN and is able to maintain the service when switching to a cellular network is called a wireless heterogeneous network.

Reference to a HetNet often indicates the use of multiple types of access nodes in a wireless network. A Wide Area Network can use macro cells, picocells, and/or femtocells in order to offer wireless coverage in an environment with a wide variety of wireless coverage zones, ranging from an open outdoor environment to office buildings, homes, and underground areas. Mobile Experts defines the HetNet as a network with complex interoperation between macrocell, small cell, and in some cases WiFi network elements used together to provide a mosaic of coverage, with handoff capability between network elements. A recent research study from ARC chart estimates that HetNets will help drive the mobile infrastructure market to account for nearly \$57 Billion in spending globally by 2017.

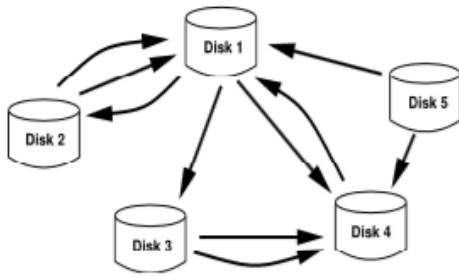


Fig.2 heterogeneous Network

PREPARING A MIGRATION PLAN:

This paper describes the process of how to create a migration project plan. It identifies the sections you must include in the migration plan, describes how to determine what you must include for each section, and how to avoid the risks involved in a migration project. This chapter includes:

- Task 1: Provide Authentication to user.
- Task 2: Converting database SQL to Microsoft Access.
- Task 3: Converting data from Microsoft Access to SQL
- Task 4: Converting data from SQL to XML.
- Task 5: Converting data from SQL to excel.

Task 1: Provide Authentication to user.

In this, user open the home page when it register then the admin provide authentication for user to login. Providing authentication is become secure for user to unauthorized access of data.

Role of the Database Administrator (DBA) During the Migration:

The database administrator (DBA) plays a crucial role in the paper. The DBA is responsible for the success of the migration. The DBA oversees every step of the project except the ones that involve application transfer and application testing on the new platform.

The tasks of the DBA are:

- Assigning well-defined roles to each person involved in the migration plan
- Scheduling the test and production migrations
- Performing test migrations

- Performing back-ups of the production database
- Completing the upgrade of the production database
- Performing back-ups of the new database

Task 2: Converting data from SQL To Microsoft Access.

In this task it use the source as SQL server and convert it to the Microsoft Access. When we convert to the data from SQL it performs some following:

It fetch the data from the database.
 Create the temporary variable and store the data in the temporary variable.
 Then it take data from temporary variable and migrate to the Microsoft Access.

Task3: Converting data from Microsoft Access to SQL.

In this task, the source is Microsoft Access and target is SQL. The result of Microsoft Access is then converted to the SQL server or vice versa.

It fetch the data from the database.
 Create the temporary variable and store the data in the temporary variable.
 Then it take data from temporary variable and migrate to SQL.

Task4: Converting data from SQL to XML.

In this task, the source is SQL and target is XML. The result of SQL is then converted to the XML or vice versa.

It fetch the data from the database.
 Create the temporary variable and store the data in the temporary variable.
 Then it take data from temporary variable and migrate to XML.

Task5: Converting data from SQL to excel.

In this task, the source is SQL and target is EXCEL. The result of SQL is then converted to the EXCEL or vice versa.

It fetch the data from the database.
 Create the temporary variable and store the data in the temporary variable.
 Then it take data from temporary variable and migrate to EXCEL.

MOTIVATION FOR DATABASE MIGRATION

➤ *Up gradation:*

Organizations deem it fruitful to move the large amounts of data stored in legacy systems to newer and more reliable current systems. This may be done to make use of the advanced features and functionalities that current systems offer. Further, legacy systems are continually on the decline and moving to contemporary systems make sense when seen in terms of service support, use of latest technology etc.

➤ *Total Cost of Ownership (TCO):*

Because legacy systems require experienced DBA's and Application Developer, the total cost incurred is high. Therefore to bring down cost, organizations move on to a single contemporary system with greater functionalities and low total cost of ownership (TCO).

➤ *Technology:*

One of the major reasons for organizations to decide to migrate is to capitalize on latest technology. There is always a risk with legacy systems that vendors may withdraw its support and make the system obsolete. This motivates organizations to move to new suitable systems. Another motivation could be business demand. The business may expand beyond the capabilities of the current system forcing the need to move to a new system. For example, a leading insurance company used a character based application driven by an 8 year old Informix database. The business demands changed and they decided to opt for an e- business and ERP integration, with a single vendor providing for all the requirements. To exploit the extant technology and its capabilities, the organization migrated from the Informix database to Oracle 8i.

➤ *Database Unification :*

It is most often the case that organizations have their data stored across multiple databases. Different applications run on different databases. But managing and tracking multiple databases is a logistical nightmare. Multiple licenses, multiple vendor support, source feed synchronizations etc. takes up the cost and effort many fold. Thus consolidating the data as far as possible makes sense. For example, an airline company had its data and corresponding applications running on 9 different databases. When running costs and data management came into question, they decided to migrate their data and applications to just two database platforms.

CONCLUSION

Database migration cannot be overlooked as a simple step of retiring an existing platform and moving on to a new one. It is a complex process with many phases and every migration process is unique. This makes it prone to failures. Thorough knowledge of what is required out of the migration, proper migration design and predicting the possible issues that might come up during the migration can bring down the chances of failures drastically.

ACKNOWLEDGEMENT

This research has been funded by the future ruptures PhD program from institute telecom.

REFERENCES

For the completion of this project we are using following references :

- [1] 2011 31st International Conference on Distributed Computing Systems -Chadi Kari , Department of Computer Science and Engineerin University of Connecticut Storrs, CT 06269Email:
- [2] S. Kashyap, S. Khuller, Y.-C. Wan, and L. Golubchik, "Fast reconfiguration of data placement in parallel disks, in ALENEX, 2006.
- [3] L. Golubchik, S. Khuller, Y. Kim,S. Shargorodskaya, and Y. J. Wan, "Data migration on parallel disks," in 12th Annual European Symposium on Algorithms (ESA), 2004.
- [4] H. Tang and T. Yang, "An efficient data location protocol for self organizing storage clusters," in Proceedings of the International Conference for High Performance Computing and Communications (SC)2003.