International Journal of Contemporary Technology and Management
Volume VI, Issue IX, September - 2017 ISSN (Online) 2278-8034

# An Implementation of Data Migration Approach of MS SQL Server to MongoDB Database

Pragya Soni[#1], Devendra Singh Koushal[*2]
Department of Computer Science and Engineering,
Jawahar Institute of Technology, Borawan
pragyasoni555@gmail.com[#1],
devendrasinghkoushal@gmail.com[*2]

*Abstract* - **As a part of achieving specific targets, business decision making involves processing and analyzing large volumes of data that leads to growing enterprise databases day by day. Considering the size and complexity of the databases used in today's enterprises, it is a major challenge for enterprises to re-engineering their applications that can handle large amounts of data. In this paper, we propose a data migration model where we migrate the source database in to destination schema. In this work, we take 2 database i.e. MS SQL Server as source database and MongoDB as Destination Database. Considering the size and complexity of the databases used in today's enterprises, it is a major challenge for enterprises to re-engineering their applications that can handle large amounts of data. The successful migration of the project ensures those converted databases is fully scalable and have high availability. Hence proposed novel automated migration model which utilizes the power of service oriented architecture to help these companies easily migrate to NoSQL databases of their choice. The performance of the proposed solution is evaluated using result generation for SQL server and mangoDB.**

***Keywords:*** *Data Migration, DBMS, SQL, Data Transfer Services, MongoDB, NoSQL, Connection, RDBMS*

## I. INTRODUCTION

The advances in Cloud computing and in modern Web applications have raised the need for highly available and scalable distributed databases to accommodate the big data being created and consumed. Along with the explosion in data growth come the necessity to rapidly evolve databases and schemas to meet user demands for new functionality. A special attention is being paid to the vast amounts of semi-structured and un-structured data, and the data management tools should reflect the support for these needs.

There is a revolution going on in database systems. For years, database systems have been primarily based on the relational model, queried using standardized SQL, and accessed using common interfaces such as ODBC and JDBC. The suggested solution is able to deliver more precise database conversion effectively by using Microsoft SQL server and mongoDB of distributed connectivity. But before introducing the solution need to study the process and different steps of multiple databases environment.

### *A. Data Migration*

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 [1].

Typically data migration occurs during an upgrade of existing hardware or transfer to a completely new system. Examples include: migration to or from hardware platform; upgrading a database or migrating to new software; or company-mergers when the parallel systems in the two companies need to be merged into one. There are three main options to accomplish data migration [2]:

- ✓ Merge the systems from the two companies into a brand new one
- ✓ Migrate one of the systems to the other one.
- ✓ Leave the systems as they are but create a common view on top of them - a data warehouse.

### *B. Database Migration*

Database are growing very fast and becoming more complex in the volume (terabyte to petabyte), variety (structured, unstructured and hybrid), and velocity (high speed in growth). Management of database has become the global challenge. The data collection is currently managed and exploited mostly by using conventional data management tools such as classic relational database management systems (RDBMS) or conventional search engines. Figure 1 depicts the architecture of database migration.

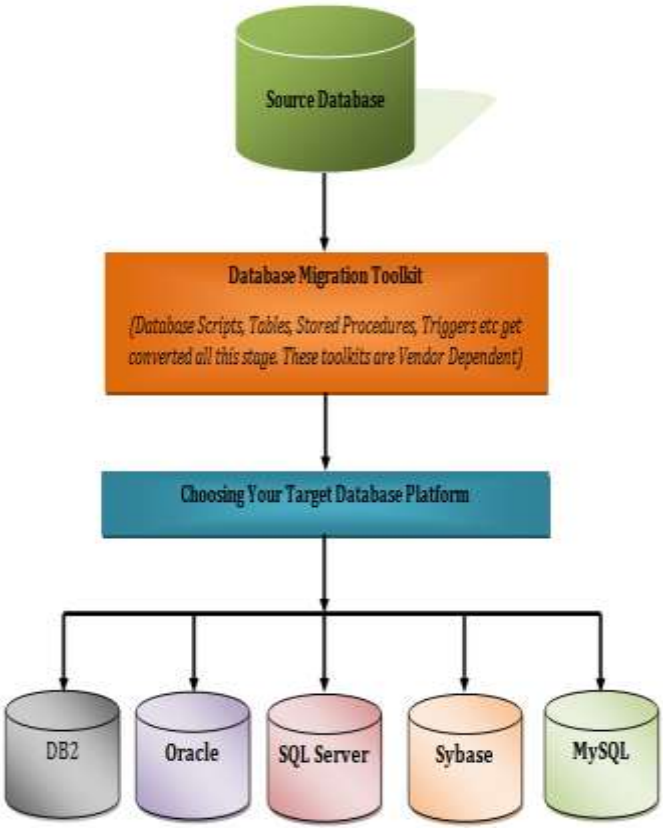


**Figure 1: Database Migration Architecture**

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. Database independent applications necessarily shoot for the lowest common denominator of all of your database platforms, thus lowering the bar on what high-performance features you might take advantage of on the platform you are currently using [3] [4].

### *C. Characteristics of Database Migration*

On first thought, it may seem that when two systems have to hold similar data they would relate to one another seamlessly. But this is most certainly not the case. The differences in their architectures, technology and implementation lead to innumerable issues during database migration. For this reason, database migration projects (and hence data migration projects) have a tendency to fail.

- ✓ **Schema Migration-** Replicating the database schema of the current system in the new system
- ✓ **Data Migration-** Extraction of data from the current system and loading it on to the new system
- ✓ **Application Migration-** Making necessary changes in the applications associated with the current system so that its behaviour remains unchanged when made to interact with the new system

## II. PROPOSED WORK

System of broad standards or principles by which particular techniques or strategies might be inferred to translate or take care of various issues inside the extent of specific study. Unlike an algorithm, a methodology is not a formula but a set of practices covered in this study.

Therefore first the straightforward solution is provided and then the entire modeling of the system is defined.

Figure 3.1 depict the working overview of the data migration process different entities. The data which is have to be converted into final database named as source data base while as well the data base to be converted into final database named as destination database.

Following are the brief description of the given diagram.

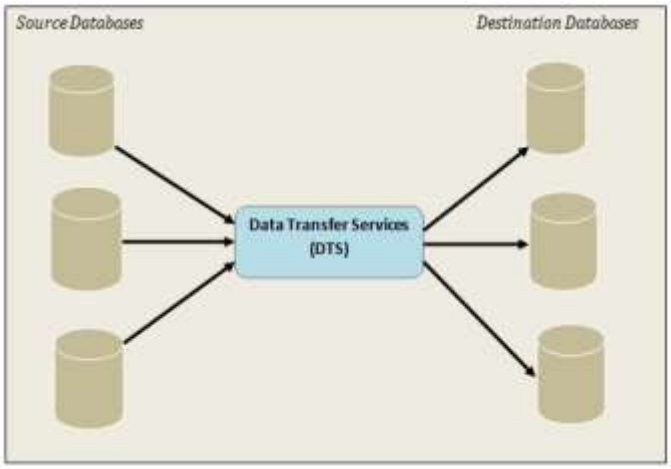


**Figure 2: Working Scenario**

- ✓ **Source Database:** In this working model we take MS SQL Server as source database for migrating towards the output database. Here we emphasize on MS SQL server database briefly:
- ✓ **Data Transfer Services:** This is the intermediate process of proposed data migration model of data model. This is the main step of overall working procedure which indicate the how data is transformed from one form to other form. Here also list the short description of the data transfer services.

  From the point of view of data migration process, we understood the working scenario of the proposed data model in figure 2. Furthermore, the core entire process of the project is shown in figure 3. In this section we demonstrate the step-by-step process of the data migration; therefore, we describe the each step in briefly including appropriate terminology.

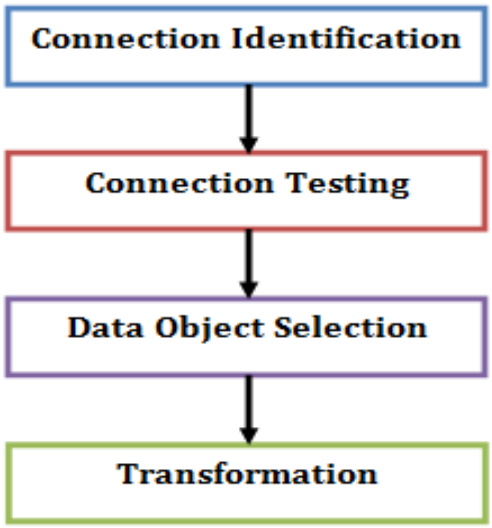


**Figure 3: Steps of Data Migration**

**Connection Identification:**

Connection identification is the term which denote the two entities are communicating through the suitable inter mediator. Once a connection is identified, the process will go to the resulting stage. Hence, in this scenario, for connection, user input the databases named source schema and target (destination) schema firstly. After that we identify appropriate drivers for both schemas which integrate the processes of source and target schema.

**Connection Testing**

For establishing connection of our data model first ping both of the services and tries to test both current establish connection among the tool and data source. This will return status code in terms of Boolean values if the test is succeeded. We got the true values and vice-versa. After test the connection next phase is appeared i.e. data object selection.

**Data Object Selection**

Data objects are the variables of the rule expression language. Here, we select the some of the object of SQL server for mapping in to the mongoDB schema. Here we describe the data object in simple term.

A data object is a collection of data attributes contained within the vocabulary, based on simple types (for example, number, text) and their associations to other data objects. A data object can be scalar, or a multi-value list (a column). An object's structure and attribute names should be described in non-technical, business language. A data object represents the current value of a persistent element of the vocabulary, using a convenient and readable natural language notation.

Hence, we list the mapping table 1 of the SQL database which is mapped into mongoDB database. The following table presents the various SQL terminology and concepts and the corresponding MongoDB terminology and concepts.

**Table 1: SQL to MongoDB mapping**

| SQL terms/Concept | MongoDB Tetm/Concept |
|---|---|
| Database | Database |
| Table | Connection |
| Row | document or BSON document |
| Column | Field |
| Index | Index |
| Table joins | $lookup, embedded documents |
| Primary Key | Primary Key |

**Data Transformation**

This phase demonstrate how the selected object is transformed into a new database structure. Thus the table 2 contains the step process of object transformation:

**Table 2 Transforms Function**

| |
|---|
| Input: Source Schema S, Target Schema T<br>Output: transformed Data to Target schema $T_f$ |
| Process:<br>1. $for(i = 0; i < S.objectLength; i++)$<br>a. $Sobj_i = getSourceObject(i)$<br>b. $T_o = TransformObject(Sobj_i)$<br>c. $T.Append(T_o)$<br>2. $end\ for$<br>3. Return T |

## III. RESULT ANALYSIS

The performance of the proposed migration model over MS SQL server to MongoDB databases for distributed databases is demonstrated in this section:

### *A. Data Processing Time*

Database takes a time to convert the source database to target database in a given schema. Therefore, to find the data processing time following formula is used to calculate how much time consume for conversion

The time consumption of the implemented algorithms is reported using the figure 4. In this figure X axis shows the different experiments performed with the system and the Y axis shows the amount of data processing time for migrating database from SQL Server to mongoDB scheme. The processing time measures in seconds. According to the given performance the implemented migration algorithm consumes time according to different experimentations and blue line demonstrating Data migration model of the system. Hence, the data processing time depends on the processing of step during data transfer service is called.

Processing Time = End Time of Target Database − Initial Time of Sou

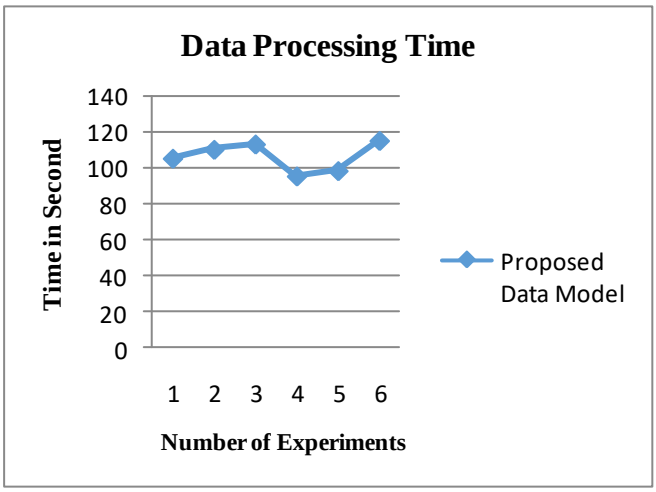


**Figure 4: Processing time of Data Migration**

### *B. Memory Consumption*

The amount of main memory required to perform data conversion that reside in system using proposed algorithm is termed here as memory usages or space complexity. The estimated consumed memory of both databases is implemented and shown in figure 5.2 and numeric values of memory consumption shown in table 5.2 Following are the formula by which we can estimate consumed memory:

Consumed Memory = Total Memory − Free Memory

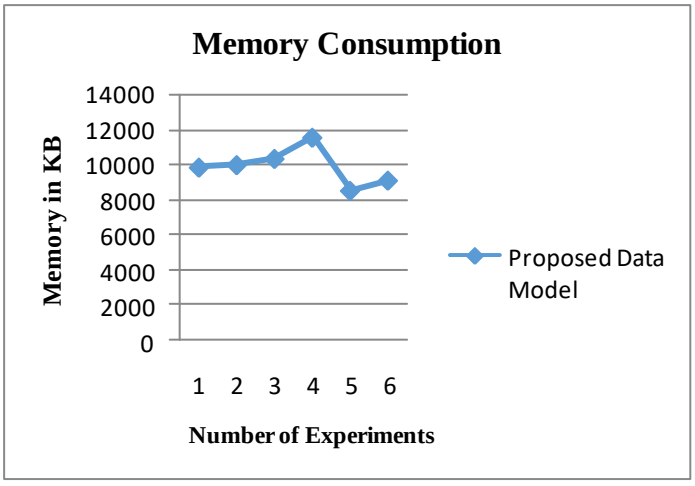


**Figure 5: Memory Consumption**

Here, we implemented the data migration model of SQL to mongoDB that efficiently use of end user applications. The memory required by the databases is demonstrated in figure 5. In this diagram blue line indicated proposed conversion model. In this graph, X- axis contains different experiments performed with system and Y axis shows amount of main memory consumed in terms of KB (kilobytes). By depiction of the given graph it clearly show that there is variation for memory consumption if we performing multiple runs of developed system

## IV. CONCLUSION

MongoDB is widely used in the field of large databases. The demand for MongoDB databases is increasing because of their diversified characteristics that offer rapid smooth scalability, great availability, distributed architecture, significant performance and rapid development agility. The main of the research work is to provide a methodology for migrating rapidly growing enterprise data from back end relational model to NoSQL (MongoDB) data store.

From different choices of databases proposed work selected open-source Ms SQL Server from relational databases group and MongoDB from the NoSQL document databases group as the test case. As the document database MongoDB is formed with collection of JSON objects and .NET C# provides completely asynchronous driver to interact with MongoDB, the object oriented approach was taken for migrating data utilizing the underlying technological advantages from C# language available in the .NET platform. This work accomplished a successful implementation of data migration process and performance is evaluated on the basis of the selected parameters.

## REFERENCES

[1] Arun Swaminathan, “Database Migration- How hard can it be?” 2nd International Conference on Information Communication and Management (ICICM 2012), Volume 55, IACSIT Press, Singapore

[2] “Data Migration”, Data Integration Info: Quick view on world of data, available online at: http://www.dataintegration.info/data-migration

[3] Rintamäki, Lina, "Data migration, a practical example from the business world", Master of Science Thesis in Software Engineering and Technology, August, 2010.

[4] Anil Mahadev, “Database Migration: An in Depth look!!” available online at: https://db2hitman.files.wordpress.com/2010/03/database-migration-word.pdf