International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST

Data Migration from SQL to No SQL using Snapshot- Live Stream Migration

Girish Bhagwat, Yadav Akash Rajiv, Paurnima Kawale
Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

Article Info
Volume9, Issue 1
Page Number: 436-443

Publication Issue
January-February-2022

Article History
Accepted : 15Jan2022
Published :25Jan2022

ABSTRACT

Due to massive use of new technologies, the huge amount of data is available and needs to be processed appropriately. Such tremendous amounts of data add a great number of challenges to the traditional database paradigm. Data migration is the procedure of transforming data from one type of database to another. Companies are moving their database from one database (e.g., RDBMS) to another new one (e.g., NoSQL) because of many reasons such as new database can handle more data, that could be fast, scalable etc. Relational database has been used by many organizations for storing and analyzing enterprise database since last few decades. Relational database stores information in a structural or relational way because it follows the structured model but it has many restrictions as compared to the non-structured model. NoSQL uses document model, graph model, keyvalue etc., as compared to RDBMS databases. NoSQL was primarily designed for storing and retrieving a large amount of data. Using NoSQL database in new enterprises is not a major issue because the new application design will be based on NoSQL database. But the issue appears when existing systems that are built on relational database are restructuring database system to implement NoSQL database. They need to reanalyze the system requirements to build up the new database schema. In this research, we will study an approach for migrating real time as well as old data from SQL to NoSQL database using database snapshot and live-stream of database changes. This research also helps us to understand how snapshot and live data migration shows the higher performance as compared to other methods.

Keywords: Data migration, RDBMS, SQL, NoSQL, database schema.

I. INTRODUCTION

1.1 Introduction

Data migration is a process of transforming data from a source database to target database. Data is migrated from one database (e.g., RDBMS) to another new one (e.g., NoSQL) because of many reasons such as new database can handle more data, that could be fast, scalable etc. Companies are moving their database from one database to another new one because of many reasons such as new database can handle more data, that could be fast, scalable etc. But the issue appears when existing systems that are built on relational database are restructuring database system

to implement NoSQL database. In this research, we will study an approach which concentrates on parallelly migrating snapshot data and live stream data of RDBMS to the NoSQL database. We will also study various pre-existing models for data migration.Relational database has been used by many organizations for storing and analyzing enterprise database since last 3-4 decades. Relational database stores information in a structural or relational way because of relational database follows the relational model. Relational database is used in an environment where data is growing slowly and adheres some structure.

1.2 Motivation

Data migration is one of the important technologies in recent times. It has also gained importance in cloud platforms.Companies are moving their database from one database to another new one because of many reasons such as new database can handle more data, that could be fast, scalable etc. But the issue appears when existing systems that are built on relational database are restructuring database system to implement NoSQL database. Having knowledge of databases like MySQL and MongoDB and also basics of cloud, motivated me to choose this topic as my seminar topic.

1.3 Aim and Objectives

1.3.1 Aim

In recent times, organizations are moving from RDBMS to NoSQL due to benefits of NoSQL. The aim of this topic is to understand data migration and study how data can be migrated from SQL to NoSQL system and also find an efficient method for the same.

1.3.2 Objectives

- Understand basic concepts like Relational and non-relational databases, data migration, etc.
- Understand complex concepts of database snapshots, live stream data, etc.
- Study and compare various data migration methods to find out the most efficient technique.

1.4 Introduction to the Topic

Relational database has been used by many organizations for storing and analyzing enterprise database since last 3-4 decades. Relational database stores information in a structural or relational way because of relational database follows the relational model. Relational database is used in an environment where data is growing slowly and adheres some structure.

Now, the data is increasing exponentially and it has been generated from different sources. Such huge data is called big data, which has three types of key concepts, i.e., volume, variety, and velocity (3 V's of big data). This data also can be structured or unstructured. Big data is stored in a new type of database, called NoSQL (mostly said non-relational database or not only SQL). Software companies such as, Amazon, Facebook, Google etc., realize that relational database was not able to store and analyze such type of data. They developed new type of database as per their requirements; e.g., Amazon developed DynamoDB, Google developed Bigtable as their NoSQL databases.

Data migration is a process of transforming data from a source database to target database. Companies are moving their database from one database to another new one because of many reasons such as new database can handle more data, that could be fast, scalable etc. But the issue appears when existing systems that are built on relational database are restructuring database system to implement NoSQL database. In this research, we will study an approach which concentrates on parallelly migrating snapshot data and live stream data of RDBMS to the NoSQL database. We will also study various pre-existing models for data migration.

II. LITERATURE SURVEY

2.1 Introduction

In the Literature Survey we will discuss various concepts involved in Data migration. Relational

database is used for storing and retrieving information in a structural or relational way while NoSQL uses other types of models for example the document model, graph model, key-value etc. NoSQL was primarily designed for storing and retrieving a large amount of data. NoSQL is a new breed of databases that do not based on the relational model and do not use SQL for data manipulation. Several previous researches had been conducted in relation to data transformation process from SQL database to NoSQL database.

Various approaches for data migration which are already used. Data preprocessing is required to ensure appropriate data migration from a relational database to NoSQL database. Graph transforming algorithm is one of the simple and easy transformation algorithms. However, it must be kept in mind that the algorithm cannot transform data optimally in the case of relational databases that have subtype-supertype structures. [1].

Big Data solution in handling data has gained good proficiency, such as NoSQL because of which the developers in the past decade to started suggesting and using the big data databases, such as Oracle NoSQL, etc. This approach has two modules: data transformation and data cleansing. The first phase is the transformation of a relational database to Oracle NoSQL database through model transformation. The required elements for model transformation are input model, transformation description, transformation engine and transformation rules. Data cleansing is provided by the second part to improve data quality and prepare and make it really very useful for the for big data analytics [2].

A linear transformational approach is suggested to be useful in order to achieve a solution that can map between a wider range of schemas. This common IR model can support the productivity of transformation agents and reduce their complexity and thus optimizes their performance. The hierarchical composition is applied to the internal design of each translation system. Where the translation rules are distributed across different translation agents, based on the structure of the target schema (model) [3].

A snapshot of database generally represents instance data of database at a particular time. This is a very common technique, which is mostly used in relational databases for migrating data from one relational database to another relational database. Nowadays, this technique is also used in relational to NoSQL databases migration such as, mongify tool. In live data replication technique, the live data changes are processed by stream processor for transforming and copying into new database by capturing the live data changes through any changed data capture software [4].

2.2 Different Approaches of Data Migration

The various data migration approaches are as follows:

2.2.1 Enhanced Graph Transforming

The enhanced graph transforming algorithm can transform data from SQL database to NoSQL database. A simulation where data is migrated from relational database to NoSQL database. The dataset is converted to relational tables. The completion of the multiple relationships does not require comparing the number of instance entities between the two tables. The initial step in solving this case is to convert the relational database into a graph. The next step is to remove edges that cause transitive dependencies in the graph, followed by combining the vertices and removing the next edges starting from the leaf node [1].

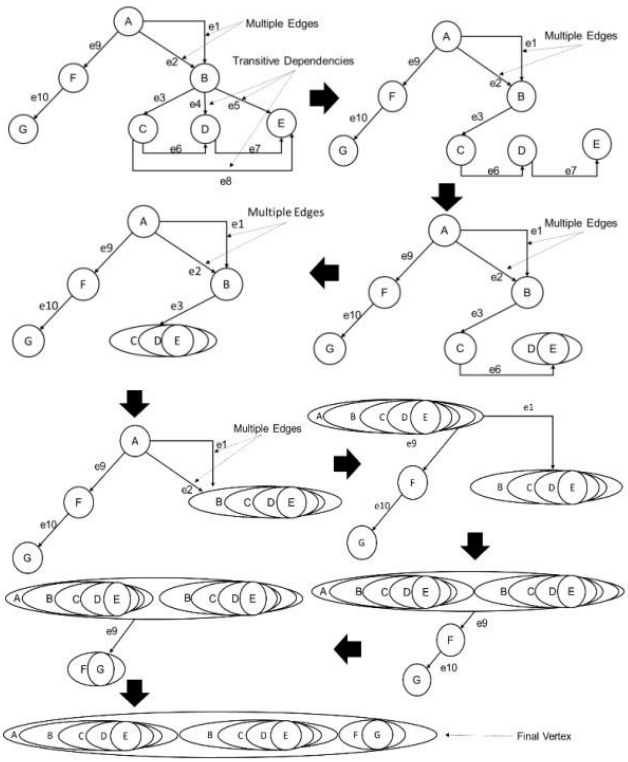

Figure 2.1: Enhanced Graph Transformation process [1]

Afterwards, the NoSQL schema is generated according to the sequence of combined vertices in the final vertex. The row keys in the schema are adopted from the primary key of the root table in the relational database. After that, data is migrated from relational database to NoSQL database schema. This experiment also runs transformation process using graph transforming algorithm, multiple nested schemas, and DDI concepts [1].

2.2.2 Intelligent Data Engineering for Migration

In this paper, they have proposed an approach that has two modules: transformation module and data cleansing module. The first one is to transform RDB to Oracle NoSQL database through model transformation. The latter provides data cleansing methodology to improve data quality.Model driven Architecture is an approach that deals with models to develop software. The required elements for model transformation are input model, transformation description, transformation engine and transformation rules. In this approach SiTra is used as a transformation engine to transform the input model to output model and maps the concepts of input model (SQL Server, RDBs model) to output model (Oracle NoSQL model) for the generation of transformation rules that are used by SiTra for transformation. Model to model transformation is our main concern that transforms a model into another model. The MDA approach maps concepts of source metamodel into corresponding concepts of target metamodel. The Java based MDD (Model-Driven Development) Approach is used to implement model transformations. The two interfaces are used to implement the transformation rules i.e. rule interface and transformer interface. The rule interface is implemented for each transformation rule and it has two methods, check method and build method. [2].

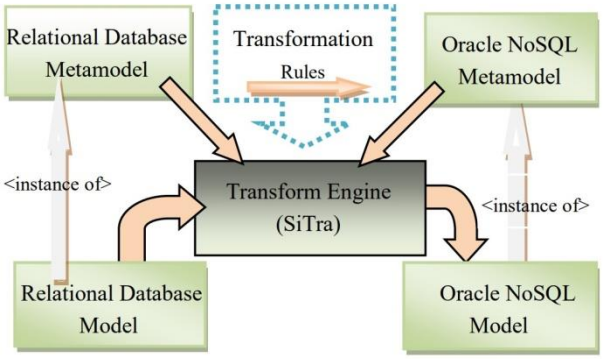

Figure 2.2: Intelligent Data Engineering for Migration [2]

2.2.3 Linear Composition of Transformation agents

A linear transformational approach is used in order to achieve a solution that can map between a wider range of schemas. Thus, it can cover types of translations between different database vendors, SQL vendors like MySQL and Oracle, and NoSQL vendors like MongoDB and Cassandra, and even the graph data model such as Neo4J database. From this point of view, the demand for introducing the more common and less technical intermediate representation (IR) model using a textual Domain-Specific Modelling Language (DSML) has emerged. This common IR model can support the productivity of transformation

agents, reducing their complexity and optimizing their performance. The hierarchical composition is applied to the internal design of each translation system. Where the translation rules are distributed across different translation agents, based on the structure of the target schema (model). It is worth mentioning that it is still an open question whether the benefit of considering the structure of the target schema when designing the translation system outweighs the structure of the source one. It is worth mentioning that defining the mapping between NoSQL schemas, such as graph- or documentbased and RDBMS ones is not a straightforward task. Based on the transformation language and framework used (declarative/ imperative), the mapping process normally involves a series of transformation iterations to form the final structure of entities, its properties and data integrity and constraints, including relationships. The detailed design and implementation of the suggested transformation framework is out of the scope of this paper [3].

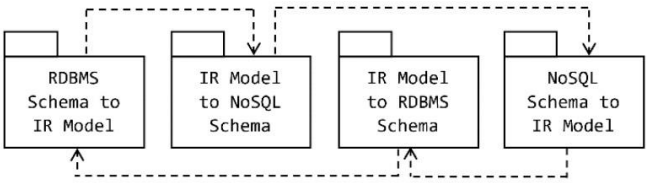

Figure 2.3: Linear Composition [3]

2.2.4 Snapshot Migration

While snapshot data is migrated, various settings are migrated from the source snapshot to the destination. These actions are all performed as one transaction. If the source snapshot was the default, that property is also transferred to the target. Runtime environment variable and server configuration settings are migrated from the source to the target snapshot. Default values set at design time are not migrated. The changes to the environment variable or server configuration that occur during run time are migrated from the source if they have a modified date that is more recent than any change that occurred at run time in the target snapshot. A change at run time always overrides the snapshot default settings [4].

2.3 Various Algorithms for Data Migration

Some of the pre-existing data migration algorithms are as follows:

2.3.1 Intelligent Data Engineering Transformation Algorithm

In this algorithm, each concept of SQL Server database is mapped to corresponding concepts of Oracle NoSQL. Oracle NoSQL have no join operation; it uses parent child relationship instead of join operations. Relational databases have concept of permissions whereas Oracle NoSQL has the concept of privileges [2].

Transformation algorithm

1. Create table object of the source.
2. Generating a list of target table objects from sourcetable's objects.
3. Mapping the source table objects to target table objects.
4. Create object of each column of source table.
5. Mapping the source column object to target field object.
6. If value is true
7. Select the transformation rule
8. // rule is applicable or not, when more than one rule isavailable for same type object.
9. endif
10. The target table object is created.
11. The target object of each column is created.
12. Select primary key.
13. Keep a history of transformed objects.

// to avoid duplicate creation of target objects.

2.3.2 Enhanced Graph Transforming V2 Algorithm

The algorithm is implemented to transform data from relational database with simple and non-simple graph characteristics. The first step of this algorithm is performed by converting the relational database structure into a graph. If the number of Vertex in the graph is only one and loop occurs ($|V|=1$ and $I[v]>0$), the graph is added with a vertex used to store

attributes identified by a foreign key that forms the loop (V←V+v1; v1←{x|xεV∧ v1 ⊆ v ;}. Next, vertices integration is completed and edges that form a loop is eliminated ({v} ∪ {v1}; E ← E-<f(u),v>;V ← V-v1;). When the final vertex has outdegree and indegree that equal zero, the iteration is terminated. If the number of vertices in the graph is more than one (|V|>1), identification of leaf node in the graph is done (P←{v | v ε V∧ O[v] = 0}). If the graph has the leaf node (|V|>1; and P > 0;), the vertices integration process and edges elimination are completed. However, when there is no leaf node (|V|>1; and P=0), it is possible that there are multiple edges and/or loop in the graph. When there are multiple edges in the graph, the comparison of the number of instance entities between vertices related to multiple edges is made (X = {x | x←∀ instance entities u}; Y = {y | y← instance entities f (u)}
;|X|>|Y|?) [1].

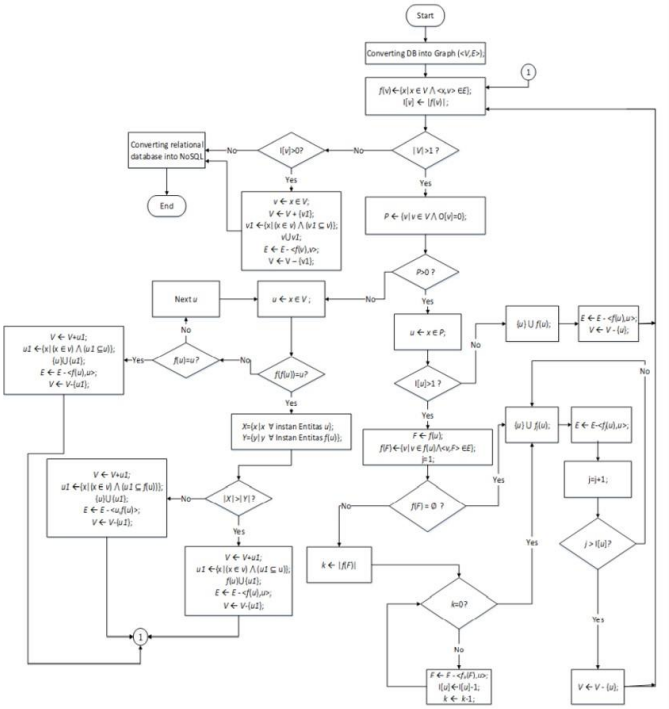

Figure 2.4: Enhanced Graph Transforming V2 Flowchart [1]

The next process is the integration of additional vertex to the one that has smaller number of instance entities ({u} ∪ {u1};or f(u) ∪ {u1}E←E-<u,u1>; V← V-u1;). When there is a loop on a graph, a vertex is added to the graph to accommodate attributes identified by the foreign key forming the loop V← V+u1; u1←{x|xεV∧ u1 ⊆ u};). Then, it is integrated into the vertex that contains a loop, followed by the removal of edge that results in a loop ({u} ∪ {u1};E← E<u,u1 >; V←V-u1;). This iteration of transformation is kept running until the final vertex has outdegree and indegree that equal zero [1].

2.4 Workflow

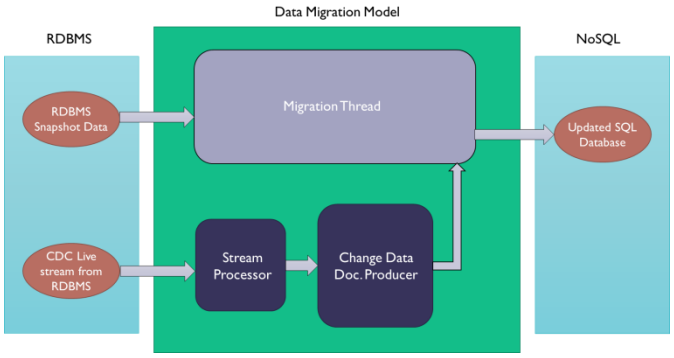

Figure 3.1: Database Snapshot Live Stream DB Workflow

Database Snapshot Live Stream DB Migration approach provides a solution for database migration for NoSQL systems. This approach focuses on parallelly migrating snapshot data and live stream data of RDBMS to the NoSQL database component (Documents) in NoSQL database.

A function can be developed for creating snapshot data of RDBMS tables, and then it will starts the CDC for live stream of databases changes. Our main focus is on copying all the data, because in a big data application data is updated/ produced in a high speed. Migrating snapshot data also consumes some time in real life application, so the changes (insert/update/delete) occurred during snapshot database migration, will also be reflected in new NoSQL database.

The approach comprises of three components i.e., migration thread, stream processor changed data document producer. The model considers two inputs; namely, snapshot data of RDBMS mode live stream of changed data capturer. Snapshot can be created by creating JSON files from SQL queries. For example,

MySQL database provides extension to SQL queries for creating JSON files.

The migration thread component in this approach firstly creates a pipeline for each document of MongoDB (NoSQL). Thereafter, it reads the data of individual document (each table data) from snapshot data of database in parallel and copies it into individual document of NoSQL database. The stream processor component of model reads the data from changed data capturer. The change data capturer reads the RDBMS databases log for any database changes.The changed databases are forwarded to stream processor for grouping and filtering the data for particular database.

The filtered data is then transferred to third component of model. The change data document producer will create inserted/ updated/ deleted data record of each document parallelly. Now, first component of model, migration thread, will reads this new data which is created by change data document producer parallelly.

2.5 Algorithm

In this algorithm, our primary focus is on the migration of two things; namely, snapshot data of RDBMS, and live stream of changed data of RDBMS. Algorithm considers listOfCollsas an input, which is a collection or array of JSON files of snapshot data of RDBMS tables for each collection of MongoDB. It reads each collection from listOfColls, and creates a new thread, then pass this collection to a thread startMigrationThread. This creates number of threads same as those are collections in listOfColls. This startMigrationThreadperforms two tasks; first it copies the snapshot data in NoSQL and it checks and copies the temporary data created by other startCDCThread. This output is then copied to temporary database for processing by startMigrationThread.

Transformation Algorithm

Input: var listOfColls- A list of JSON files for each collection of NoSQL (Snapshot Data).

Output: Update NoSQL database with Snapshot and Live stream of Changed data.

Begin for each Collection x in listOfColls
Begin
Call startMigrationThread(x)
End
Call startCDCThread End Algorithm;
function startMigrationThread(Collection x)
//Runs in a different Thread
Begin copyDataInNoSQL x
copyCDC_DataInNoSQL from Temporary DB for Collection x.name End startMigrationThread;
function startCDCThread() //Runs in a different Thread
Begin
ReadDataFromChange_Data_Capturer
CopyChanged Data in Temporary DB
End startCDCThead;

III. APPLICATIONS

3.1 Probable Applications

The applications of the Database snapshot and live stream database migration approach are as follows :

- NoSQL database has many benefits over RDBs. Hence, this technology can help organizations who wish to move old as well as update real-time data changes from SQL to NoSQL.
- The companies or organizations moving from local storage system to a cloudbased system to optimize operations can use this approach.
- Installation of new systems to previously existing applications sharing the same dataset.
- For replacement, upgrade, and expansion of storage systems this approach can be very useful.
- AWS DMS(Database Migration Service) and AWS SCT(Schema Conversion Tool) are already existing applications using such data migration technology.

IV. CONCLUSION

Migration of data from RDBMS to NoSQL database is a very complex task where data is regularly updating and increasing in size very fast. The proposed approach for data migration from RDBMS to NoSQL is presented, which uses both the data migration techniques, snapshot and live stream of change data parallelly. This model has three components; namely, migration thread, stream processor and changed data document producer. It takes two inputs; namely, snapshot data of relational database mode and live stream of changed data capturer.

This approach helps us to efficiently migrate the snapshot as well as live data stream of changed data in NoSQL. It is also found that proposed model performs faster than other existing models and it also migrates the live data.

V. REFERENCES

[1]. SutediSutedi, Noor Akhmad Setiawan, Teguh Bharata Adji. "Enhanced Graph Transforming V2 Algorithm for Non-Simple Graph in Big Data PreProcessing", IEEE Transactions on Knowledge and Data Engineering, volume: 32, issue: 1, INSPEC Accession Number: 19291983, Publisher: IEEE, page numbers : 67 - 77, November 2018.

[2]. Shabana Ramzan, Imran Sarwar Bajwa, Bushra Ramzan, Waheed Anwar. "Intelligent Data Engineering for Migration to NoSQL Based Secure Environments", IEEE Access, volume: 7, INSPEC Accession Number: 18731935, Publisher: IEEE, page numbers : 69042 - 69057, May 2019.

[3]. Ahmad F. Subahi. "Edge-Based IoT Medical Record System: Requirements, Recommendations and Conceptual Design", IEEE Access, volume: 7, INSPEC Accession Number: 18843554, Publisher: IEEE, page numbers : 94150 – 94159, July 2019.

[4]. Basant Namdeo, Ugrasen Suman. "A Model for Relational to NoSQL database Migration: Snapshot-Live Stream Db Migration Model", 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), volume: 1, INSPEC Accession Number: 20677445, Publisher: IEEE, June 2021.