International Journal of Engineering & Technology, 7 (4) (2018) 3335-3339

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.11377
Research paper

A review on data transformation approaches for data migration processes from relational database to NoSQL database

Norwini Zaidi [1], Iskandar Ishak [2] *, Fatimah Sidi [2], Lilly Suriani Affendey [2]

[1] University Sains Islam Malaysia
[2] Department of Computer Science and Information Technology, Universiti Putra Malaysia
**Corresponding author E-mail: iskandar_i@upm.edu.my*

Abstract

In recent years, data migration has become an important exercise in data management due to the demand of scalable database management systems. Data transformation is an important part of data migration and researchers have put a lot of effort to produce better data transformation approaches. Based on the literatures, there are multiple ways of conducting data transformation between relation databases to NoSQL databases. However, among the issues in data transformation between the two databases are slow processing time and no definitive guideline for database administrator to guide them to conduct the transformation. This paper highlights the data transformation approaches that have been proposed by researchers in recent years as well as its strengths and limitations.

Keywords: *NoSQL Database; Data Migration; Data Transformation; Big Data*

1. Introduction

For many years, relational database management system (RDBMS) has been the most popular database for data management [1], [2]. However, with the fast increasing data growth shown recently, RDBMS has been found to be lacking in storing large volume of data and managing different types of structural data or flexible schema [3], [4].
The RDBMS is a data management system that uses structured data format by enforcing atomicity, consistency, isolation, and durability (ACID) to ensure database reliability [5]. The RDBMS is using rows and columns, primary key, indexing and normalization to maintain data integrity, data consistency and also using standard SQL language for querying. For example, relational database is essential for Banking System and Accounting System which transaction will be updated to all databases at the same time. But there are also challenges on RDBMS that forcing migration to NoSQL database.
RDBMS have limitation in processing of large volume of data for example; in managing more than one database in different server containing millions of records, RDBMS will take hours for processing the data for insert, update, and delete for ensuring data consistency [5]. RDBMS also have limitation on storing unstructured data. The unstructured data that do not have specific format are not compatible with the RDBMS that are rigid on structured format [6-7]. The RDBMS also having database scalability issue where the database is not able to support easy expansion and upgrades to ensure service uptime [3], [8], [9].
To overcome the RDBMS issues, NoSQL database is developed to support data management system which may easily fit big data processing and computation. NoSQL database is the new generation databases that features non-relational, distributed, open-source and horizontally scalable databases [10]. NoSQL database has a flexible structure format than RDBMS and making some data management operations faster in NoSQL.
There are four types of NoSQL database based on the literatures: key value store, column store, document store and graph database [6], [11]-[17]. Key Value store is a database that records the information based on key and value of the records. The key must be a unique key that associated with the data value for information retrieval. Some well-known NoSQL database that uses key value store are Redis, Aeropsike and Voldemort. Column store database is a database that stores the data in a set of columns and rows. The related data will be stored in column family and retrieved by using a rowkey. HBase and Cassandra are examples of well-known column store databases. Document store database has a unique key that represents the value of the data and can store structured or semi-structured format of document such as JSON and XML. Mongodb and CouchDB are the examples of document store databases. In Graph database, data is stored in representation of graph that has nodes and relationship. The nodes are attributes of each entity in the database, while relationships are relationships between the nodes. Neo4j is a known example of Graph databases.

2. Data migration

In order to move from RDBMS to NoSQL, data migration needs to be performed. Data migration is defined as a process to transferring or moving data from one database server; storage types or data formats to another. Data migrations always using ETL (Extract-Transform-Load) process as a migration step [18]–[21]. The first step in data migration is data extraction that implement for extracting and filtering data from source database.
In order to have a smooth data migration, data transformation needs to be performed properly. Figure 1 describes the data migration process in general. Data migration from existing RDBMS to

NoSQL databases is a very challenging task to perform since the data structure or data schema of NoSQL databases is different from RDBMS and the data mapping and transformation is needed to convert into NoSQL databases.

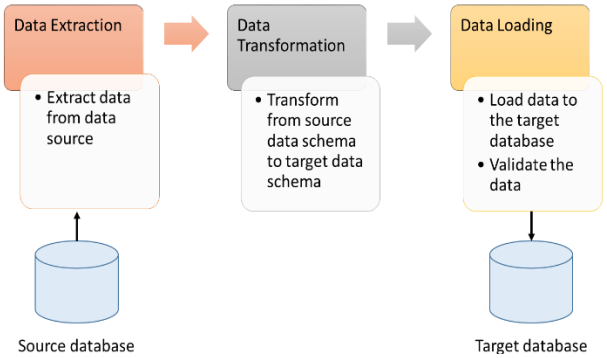

Fig. 1: Data Migration Process from RDBMS to No SQL Database

3. Data transformation in data migration from RDBMS to NoSQL

Data transformation is defined as a process to convert the data structure or data schema from source database according to the new data structure in target database [18]–[20]. The data transformation includes analyzing and mapping processes of the data schema between source database and target database. The data transformation is important to ensure that the metadata reflects correctly in the target database. The data is then loaded to the target database. The data in the target database needs to be validated to verify that the data is accurately translated. The following subsections discussed about multiple approaches in data transformation in data migration process between RDBMS to NoSQL database.

3.1. Data transformation using structured DE normalization

Data migration using structured denormalization needs to convert RDBMS that usually in a normalized form into non-normalized database. The normalized table will be combined to create a non-normalize table to eliminate joins within the tables. In [22], physical level and first logical level of data structure from existing RDBMS are used to get the metamodel structure. Data migration from RDBMS to the NoSQL document structure is generated based on the template that has been created from second logical level. The various algorithms that are based on the desired document format generated the templates.

The data migration are tested by implementing it to the relational database browsing tools named DigiBrowser. Users of this tool can create the logical data model and template using Digibrowser. Structured denormalization is used in [23] on column oriented NoSQL database. Its objective is to autonomously denormalize and re-aggregate SQL table into NoSQL table. The schema analysis checks all the primary keys from SQL table. The row key (RK) in column oriented NoSQL table are selected from several primary keys and the relationship among the table in SQL database. After RK is defined, migration aggregates all columns to NoSQL table. Data migration approach in [24] has an added layer called the persistence layer between application and the NoSQL database for querying. They built Mediator using RDBMS proxy that opens communication between RDBMS server and application.

3.2. Data transformation using query and data characteristics

Model transformation based on query characteristic and data characteristic known as description tags are proposed in [7]. These description tags are collected from the log records of RDBMS that help to estimate the limitations of RDBMS. Action tags will be generated by these description tags and are used as data transformation guideline. Conceptual model of RDBMS are used for model transformation algorithm by mapping concept between RDBMS and Mongodb. Based on the result of the model transformation, data migration process automatically migrated to Mongodb. Erwin HAWK is a tool that has been develop for model transformation and data migration and it helps user to use existing ER model for automatically migrate relational database to Mongodb.

3.3. Data transformation using SQL layer or framework

Another approach for data transformation in migrating RDBMS to NoSQL is using SQL layer or framework. A framework called NoSQLLayer has been proposed to automatically migrate to NoSQL database. This framework has two modules: The first module is Data Migration Module that automatically identify all metamodel structure or Metadata of RDBMS (eg: tables, attributes, relationships, indexes, etc.) for data migration. Second module is Data Mapping Module that uses the metadata collection that already defines for data migration. This module have persistence layer that responsible for interfacing between application and RDBMS (MySQL) for data migration that allows using all the existing queries and application programming without any changes. A component called Mediator is also created using MySQL proxy for establishing communication between MySQL and applications. The mediator handles the querying from application to NoSQL database that fetches all the SQL transaction from the application and translates to NoSQL.

In [25], an SQL layer named SQLtoKeynoSQL is used for translation structure of RDBMS to any key oriented NoSQL database (document store database, key value database and column family database). This layer maps the RDBMS schema to canonical model which is an intermediate data model between RDBMS and key oriented NoSQL database. The canonical model implements mapping strategies to map NoSQL and SQL command using REST API access methods (Get, Put and Delete) to NoSQL target schema. This layer has flexible option for user to choose any target key oriented database. This layer also has the ability to manipulate the RDBMS in any key oriented NoSQL database for users to manage the data. The SQLtoKeyNoSQL layer has an architecture that consists of seven modules for data migration process. SQL parser module receives SQL instructions and performs semantic verifications. It then sends the SQL instructions to the Query planner module for query execution optimizer. The Translate module generates the access method to the Execution engine module. The Execution engine module checks the login information, the target of NoSQL database and the table store from Dictionary module before filtering the data and then generates the result to be sent to the Access Interface module. The Communication module executes requested access method to any key oriented NoSQL database.

A framework-based approach for data migration is proposed in [26] by accessing data from MySQL, Mongodb or HBase using MySQL queries. The framework is use to map the MySQL query to HBase and Mongodb query format. The framework consists of:

i) GUI to control data migration
ii) JDBC as the connection application to the database
iii) Mongo migrate
iv) Hbasemigrate that act as a module that are programmed to auto detect tables in the MySQL which is match with related table of Mongodb or Hbase
v) Mongo Map as a migration module of Mongodb
vi) Hbasemap is a module that maps the query from MySQL to HBase query format.

The framework process the query in MySQL format using a GUI, then the decision maker will provide as input to MongoMap or HbaseMap to predict which query will execute more efficiently. Next, the data retrieval is displayed from the preferred NoSQL databases.

3.4. Data transformation using schema conversion

Schema conversion is another approach for data transformation in data migration process from RDBMS to NoSQL database. It is proposed in [4] using graph-transforming algorithm to improve performance of join query with table nesting process. The schema conversion is based on table nesting that generates one to one table mapping between RDBMS and NoSQL by using graph model of database schema. The graph model reviews the general database schema for converting RDBMS to NoSQL schema by using simple extension, vertical extension and horizontal extension conversion procedure. The graph-transforming algorithm then generates nesting sequence for RDBMS to validate correctness of the schema conversion.

An automatic SQL to NoSQL schema transformation from MySQL to HBase database is proposed in [27]. The schema transformation follows the NoSQL DDI design principles; aggregate relational tables into one big NoSQL table and then select most suitable row key in HBase database. The schema transformation also used tall-narrow design that allows a table with few columns but many rows for better query retrieval. The proposed schema transformation parses the SQL table schema automatically and converts the relationships among the table into several linked lists. After parsing all the tables, as a result the chained length of all linked list will be shown. The row key with highest cardinality should be the combination of all the primary keys with the longest chained length.

A migration approach using source data model, target data model and target data model is proposed in [5]. The schema of relational database becomes the source of data model and a column oriented NoSQL database becomes the target data model. A data migration algorithm is used for translation of data source model into target model. The translation approach generates the target schema, the translation pattern and also generates the data instances to the target database from the starting database. The data conversion uses three stages: data extraction consists of querying data, data processing involve the transformation data to the target database format and injection in the destination database.

Chongxin Li [28] proposed a schema transformation approach that requires data model in relational database to be converted according to the NoSQL database schema. These data is then grouped together in the same column family. Then, the data are mapped between data source schema and target schema using of nested schema mappings.

3.5. Data transformation using map reduce

Automated data migration is another data migration approach that focuses on the transformation of real time data [29]. This approach migrate database from RDBMS to document store using MapReduce. The data migration processes consist of three procedures. First, the log based change data capture extracts and collects all the data changes from RDBMS. Row based replication (RBR) from RDBMS is used to capture the changed data from binary logs and uses bin log API to get incremental data. Then the data changes are merged and only the last updated records are captured for merging. The last procedure is blocking and transformation for data migration process from RDBMS to NoSQL database. This process is implemented using MapReduce where all the changed record will get the different blocking key in the map function. Then the "map" outputs are distributed to the multiple "reduce" task for transformation of all reduced partition. They also used predicate logic of mathematical relation and QVT relation to define the mapping process. This data migration process is a seamless approach in live data migration for data retrieval.

3.6. Data transformation using log-based change data capture

Data replication approach that uses log based change data capture and stream processing framework is proposed in [30]. The replication process consists of three steps. First, data replication mapping maps the entire RDBMS tables, column, relationship, primary key and foreign key into a schema free collection document. This is a one to one transformation from RDBMS to schema free collection document.

The next step is log based change data capture process. This process is almost the same as the log based change data capture process in [29]. The output of the changed data is sent as input to the stream processing framework. The last step is column grouping, column merging and column versioning. This step is performed to calculate the target schema free collection and to avoid data loss in case of failure.

4. Discussion

In this section, we discussed on each of data transformation approaches mentioned in previous section in terms of its strengths, and its limitations related to data management. Structured denormalization approach focuses on denormalizing the table in which, multiple structured tables in relational table will be mapped into a big single table. It increases the speed of querying but the issues of having unnecessary replication of data means, storage need to be larger in the NoSQL database.

Model transformation based on the query and data characteristic focuses on understanding of the database context learned from its relational query and conceptual model. Therefore it depends on the correct translation of the relational logs and conceptual design which in turn to be the most crucial part of the data and any misinterpretation at this stage could jeopardize the whole migration process. Databases that are complex and large need manual intervention from Database Administrator to understand the conceptual design as well as data semantic. This is to ensure correct interpretation before the model can be transformed.

Data migration using SQL Layer focuses on the use of a special layer that helps understanding the relational database through its metadata before it can be mapped into NoSQL. This SQL Layer also can execute the same SQL query to query to NoSQL database. However, this approach need high understanding on RDBMS data model before develop a layer or framework that can execute using existing SQL query to both RDBMS and NoSQL database.

Schema conversion approach for database migration focuses on the translation and schema transformation of RDBMS to NoSQL database. Until today, there are no standard procedures to implement schema conversion to NoSQL databases [27]. This shows that in order to perform data migration or data transformation from RDBMS to NoSQL database, it needs the expertise of the database administrator to identify and use suitable approaches. Another issue upon this approach is it uses large amount of space at the expense of query efficiency [4].

Another approach for data migration from Relational Database to NoSQL database is through automated data transformation using Map Reduce. In this approach, the transformation is based on the tracking of changes recorded on the binary log, merging of the changed data and blocking and transformation. This approach can be conducted while the data is still running live as it tracks recent changes and it can be done smoothly when business operation is running. Therefore it is very suitable to transaction-intense database. However, the only downside about this approach is that it is technically more complex as it requires both database and Big Data computing knowledge and computation infrastructure.

We summarized the data migration approaches in Table 1. Based on the literatures, we summarized each approaches based on its advantages and disadvantages.

Table 1: Data transformation approaches

References	Data Transformation Approaches	Strength	Limitations
[22], [23], [24]	Structured Denormalization	Reduce number of tables Faster querying	Create unnecessary replicated data thus increases data storage
[7]	Model Transformation based on query and data characteristic	Understanding target database from its conceptual model	Need access to relational database design not data, lack semantic understanding
[2], [25], [26]	SQL Layer or Framework	Understanding target database through its metadata Query layer after migration	Need to maintain the SQL layer or framework for both SQL and NoSQL database accessing
[5], [4], [27]	Schema Conversion	Transform the schema structure Faster querying Using nested mapping	Increase data storage Did not take multi-nested mapping into consideration
[29]	Automated Data Migration	Suitable for real-time data transformation	Learning complexity is high due to the inclusion of Big Data computational approach
[30]	Data Transformation using Log-based Data Capture and Stream Processing	Suitable for real-time data transformation	Learning complexity is high due to the inclusion of Big Data computational approach

5. Conclusion

As a conclusion, data transformation is an important process before a complete data migration can be performed from RDBMS to NoSQL database. Based on the literatures, there are many ways data transformation can be performed and each approach has its own unique mechanism. Each of these approaches also shows its own strengths and limitations in terms of the impact in terms of knowledge needed, storage size, and data content understanding.
The variety of the data transformation approach and its impact of each approaches give problems for database administrator to decide which approach will be the most suitable to choose in performing data migration. Therefore, we can conclude that, there is no definitive general or standard guideline for database administrator to conduct data transformation from relational database to NoSQL database. It is an utmost important to have a very dedicated guideline for database administrator to perform the data transformation process. More researches need to be conducted in order to produce definitive guideline for database administrator to perform data transformation task.

Acknowledgement

We would like to thank Faculty of Computer Science and Information Technology, Universiti Putra Malaysia through MyRA PTJ Incentive Allocation 2016 for supporting this research work.

References

[1] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, and J. Bernardino, "Choosing the right NoSQL database for the job: a quality attribute evaluation," J. Big Data, vol. 2, no. 1, p. 18, 2015. https://doi.org/10.1186/s40537-015-0025-0.

[2] L. Rocha, F. Vale, E. Cirilo, D. Barbosa, and F. Mourão, "AFramework for Migrating Relational Datasets to NoSQL1," Procedia Comput. Sci., vol. 51, pp. 2593–2602, 2015. https://doi.org/10.1016/j.procs.2015.05.367.

[3] N. Ntarmos, I. Patlakas, and P. Triantafillou, "Rank join queries in NoSQL databases," Proc. VLDB Endow., vol. 7, no. March, pp. 493–504, 2014.

[4] G. Zhao, Q. Lin, L. Li, and Z. Li, "Schema conversion model of SQL database to NoSQL," in Proceedings - 2014 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2014, 2014, pp. 355–362.

[5] S. Ataky, T. Mpinda, L. G. Maschietto, and P. A. Bungama, "From Relational Database to Column-Oriented NoSQL Database : Migration Process," Int. J. Eng. Res. Technol., vol. 4, no. 5, pp. 399–403, 2015.

[6] A. K. Zaki, "NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and its Security Challenges," IJRET Int. J. Res. Eng. Technol., pp. 403–409, 2014.

[7] T. Jia, X. Zhao, Z. Wang, D. Gong, and G. Ding, "Model transformation and data migration from relational database to MongoDB," Proc. - 2016 IEEE Int. Congr. Big Data, BigData Congr.2016, pp. 60–67, 2016.

[8] A. Abdullah and Q. Zhuge, "From Relational Databases to NoSQL Databases : Performance Evaluation," Res. J. Appl. Sci. Eng. Technol., vol. 11, no. 4, pp. 434–439, 2015. https://doi.org/10.19026/rjaset.11.1799.

[9] S. Sharma, U. S. Tim, J. Wong, S. Gadia, and S. Sharma, "A brief review on leading big data models," Data Sci. J., vol. 13, no. December, pp. 138–157, 2014. https://doi.org/10.2481/dsj.14-041.

[10] R. Arora and R. R. Aggarwal, "An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB)," Int. J. Adv. Stud. Comput. Sci. Eng., vol. 2, no. 1, pp. 6–12, 2013.

[11] J. Ahmed and R. Gulmeher, "NOSQL Databases: New Trend of Databases, Emerging Reasons, Classification and Security Issues," Int. J. Eng. Sci. Res. Technol., vol. 9655, no. 6, p. 1, 2015.

[12] O. Hajoui, R. Dehbi, M. Talea, and Z. I. Batouta, "An advanced comparative study of the most promising NoSQL and NewSQL databases with a multi-criteria analysis method," J. Theor. Appl. Inf. Technol., vol. 81, no. 3, pp. 579–588, 2015.

[13] V. Abramova, J. Bernardino, and P. Furtado, "Experimental Evaluation of Nosql Databases," Int. J. Database Manag. Syst. Vol.6, No.3, June 2014, vol. 6, no. 3, pp. 1–16, 2014.

[14] A. P. R. Sharma and A. P. Y. Kashyap, "A study of nosql databases and working overviews," Int. J. Recent Trens Eng. Res., pp. 43–50, 2016.

[15] S. S. Pore and S. B. Pawar, "Comparative Study of SQL & NoSQL Databases," Int. J. Adv. Res. Comput. Eng. Technol., vol. 4, no. 5, pp. 1747–1753, 2015.

[16] M. V, "Comparative Study of Nosql Document, Column Store Databases and Evaluation of Cassandra," Int. J. Database Manag. Syst. (IJDMS) Vol.6, No.4, August 2014, vol. 3, no. 5, pp. 29–39, 2013.

[17] O. B. Mohamad Hanine, Andesadik Bendarag, "Data Migration Methodology from Relational to NoSQL Databases," Int. J. Comput. Electr. Autom. Control Inf. Eng., vol. 9, no. 12, pp. 2511–2515, 2015.

[18] T. Odia, S. Misra, and A. Adewumi, "Evaluation of hadoop/mapreduce framework migration tools," Asia-Pacific World Congr. Comput. Sci. Eng. APWC CSE 2014, pp. 1–8, 2014.

[19] P. Badlani, "NoSQL in Action-A New Pathway to Database," Int. J. Sci. Res., vol. 5, no. 6, pp. 872–877, 2016.

[20] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. ElBastawissy, "A proposed model for data warehouse ETL processes," J. King Saud Univ. - Comput. Inf. Sci., vol. 23, no. 2, pp. 91–104, 2011.

[21] A. B. M. Moniruzzaman and S. A. Hossain, "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison," arXiv Prepr. arXiv1307.0191, vol. 6, no. 4, pp. 1–14, 2013.

[22] G. Karnitis and G. Arnicans, "Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation," Proc. - 7th Int. Conf. Comput. IntellCommun. Syst. Networks, CICSyN 2015, pp. 113–118, 2015.

[23] C. H. Lee and Y. L. Zheng, "SQL-To-NoSQL Schema Denormalization and Migration: A Study on Content Management Systems," Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015, pp. 2022–2026, 2015.

[24] N. Kuderu and V. Kumari, "Relational Database to NoSQL Conversion by Schema Migration and Mapping," Int. J. Comput. Eng. Res. Trends, vol. 3, no. 9, pp. 506–513, 2016.

[25] G. A. Schreiner, D. Duarte, and R. dos Santos Mello, "SQLtoKeyNoSQL: A Layer for Relational to Key-based NoSQL

Database Mapping," Proc. 17th Int. Conf. Inf. Integr. Web-based Appl. Serv., p. 74:1--74:9, 2015.
[26] P. Nikam, T. Patil, G. Hungund, A. Pagar, A. Talegaonkar, and M. S. Pawar, "Migrate and Map : A Framework to Access Data from Mysql Mongodb or Hbase Using Mysql Queries," IOSR J. Comput. Eng., vol. 18, no. 3, pp. 13–17, 2016.
[27] C. H. Lee and Y. L. Zheng, "Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases," 2015 IEEE Int. Conf. Consum. Electron. - Taiwan, ICCE-TW 2015, pp. 426–427, 2015.
[28] C. Li, "Transforming Relational Database into HBase : A Case Study," IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS 2010, pp. 683–687, 2010.
[29] K. Ma and F. Dong, "Live data migration approach from relational tables to schema-free collections with MapReduce," Int. J. Serv. Technol. Manag., vol. 21, no. 4–6, pp. 318–335, 2015 https://doi.org/10.1504/IJSTM.2015.073942.
[30] K. Ma and B. Yang, "Live Data Replication Approach from Relational Tables to Schema-Free Collections Using Stream Processing Framework," Proc. - 2015 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. 3PGCIC 2015, pp. 26–31, 2015.