Migration from relational to NoSQL database

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042055 doi:10.1088/1757-899X/263/4/042055

Migration from relational to NoSQL database

Sunita Ghotiya, Juhi Mandal, and Saravanakumar Kandasamy

VIT University, Vellore-632014, Tamilnadu, India.

E-mail: sunitaghotiya12345@gmail.com

Abstract. Data generated by various real time applications, social networking sites and sensor devices is of very huge amount and unstructured, which makes it difficult for Relational database management systems to handle the data. Data is very precious component of any application and needs to be analysed after arranging it in some structure. Relational databases are only able to deal with structured data, so there is need of NoSQL Database management System which can deal with semi -structured data also. Relational database provides the easiest way to manage the data but as the use of NoSQL is increasing it is becoming necessary to migrate the data from Relational to NoSQL databases. Various frameworks has been proposed previously which provides mechanisms for migration of data stored at warehouses in SQL, middle layer solutions which can provide facility of data to be stored in NoSQL databases to handle data which is not structured. This paper provides a literature review of some of the recent approaches proposed by various researchers to migrate data from relational to NoSQL databases. Some researchers proposed mechanisms for the co-existence of NoSQL and Relational databases together. This paper provides a summary of mechanisms which can be used for mapping data stored in Relational databases to NoSQL databases. Various techniques for data transformation and middle layer solutions are summarised in the paper.

1. Introduction

There are number of applications which produce a large amount of data which might not be structured [3]. The amount of data produced by various social media sites, real time applications is very huge and not possible to get handled by traditional databases which works only on structured data[14]. The traditional databases are unable to handle complex data generated by current applications so need of such a database which is suitable for current applications has been emerged [8][21]. These problems encountered important research about databases which can handle huge amount of data which might be complex and semi-structured. The researches introduced a new database model called, NoSQL databases which offers easy scalability, faster I/O operations and incur lower cost than traditional databases [5-9]. There is growing demand to implement NoSQL databases to meet the requirements of current applications and which raise the problem of data migration from Relational to NoSQL databases [11-16]. The main issue with migration is that the applications which have been developed with Relational databases as base database have to change the code in order to communicate with the new NoSQL databases [10]. The data stored in relational databases is in the form of tables with a number of records which makes the rate of data accessing slower and inconvenient [1]. NoSQL databases do not follow any particular structure and do not support foreign key bounding which makes it easy to access the data [1-5]. The addressed problem is that the less efficiency of relational databases and growing demand of current applications have emerged the necessity to propose a system which can migrate the data from relational databases to NoSQL databases while maintaining the integrity of data and let the application code remain unchanged (that the queries can be implemented in MySQL from the existing applications. A number of systems have been proposed which provides solution to the addressed problem. The paper analyses all the recent approaches [1-7] which have been proposed for data migration from Relational databases to NoSQL databases. A

literature review of all the current solutions to the addressed problem has been done. This paper also presents a comparative study of the proposed systems to address the problem of data migration from Relational to NoSQL databases.

2. Related Work

Relational database and other traditional databases follow rigid structure to organize the data generated from various applications but [8] NoSQL databases provide flexibility in organizing the data which makes it easy to access the data. The data generated from social networking sites and real time applications needs flexible and scalable system [9] which increases the need for NoSQL and multidimensional model has been proposed for data migration. The biggest challenge is to migrate existing data residing in data warehouses to NoSQL databases while maintaining the characteristics of the data [10]. The growing use of web applications has raised the demand to use NoSQL because traditional databases are unable to handle the rapidly growing data [11]. Forensic investigation framework has been proposed for NoSQL databases due to the horizontal scalability and reliability [12]. Scalable three-tier architecture which can support inter database transaction has been proposed which uses distributed middle -ware protocol [13] and also NoSQL databases [14] reduces the cost and time in data accessing and support scalable applications. To handle semi-structured data from various applications so that it can be used for further analysis NoSQL is needed [15]. The features like high performance, flexibility, reliability and scalability has been provided by Hbase schema to migrate data from relational to NoSQL [16]. Cloud based data migration techniques have been used because cloud now provides services to store data in NoSQL databases [17]. A data model based approaches for data migration and reverse engineering techniques were discussed to give a solution for RDBMS to MongoDB in [18]. The semantics of NoSQL and Relational databases has been compared and a model has been proposed to optimize the NoSQL MongoDB [19]. A schema conversion model has been proposed for data migration which can support complex join operations [20].

3. Major Approaches for Data Migration from Relational to NoSQL databases

3.1 Mid-model approach using data and query features [1]

3.1.1 Technique Used. Mid-model for model transition and for migration of data from relational databses to NoSQL databases. The model proposed in this paper is based on two basic concepts: Data features and query features. This mid-model is transferred to the physical model which is there at the destination database and when it is done the data can be migrated from Relational to NoSQL databases.

3.1.2 Experiments and results. The main challenge to this system was to find out the differences that exist between the models of relational and NoSQL databases. The mid-model has Object as the basic model which is somewhat like an entity in the relational databases, having a no of attributes describing the characteristics of the entity and relationships with other objects. The model also contain a module strategy lib which contains the strategy which is going to be used for various NoSQL databases and also specify how the data model for a specific NoSQL database is generated. (1)The data from the relational database is given to the model as an input (2) the mid-model create an Object for each entity and data features and query features are generated for the source data (3) the model is then transferred to a physical model according the pre-defined strategy as an output (4) the data is migrated easily after model mapping.

3.1.3 Advantages. The most basic feature of the model is that it is uniform model which can generate model for any of the NoSQL database as required. The proposed system is not specific to a number of NoSQL databases but universal for all the target NoSQL databases. This model not only migrate data from relational to NoSQL but also maintains the data integrity which was present in the Relational database. The main two features of Mid-model: Data feature contains some of the characteristics which are present in source data in relational database whereas the Query feature specifies the queries

which are commonly or frequently accessed on the source data and according to these queries the data is arranged in the destination database (NoSQL databases) for making the data access easy.

3.2 NoSQLayer approach [2]

3.2.1 Technique Used. NoSQLayer which ensures the data to be migrated from Relational to NoSQL databases without emerging the need of change in the application code. The framework works as an interface which is present between the database and application.

3.2.2 Experiments. The proposed solution for the addressed problem is implemented using two phases: migration and mapping. The framework contains two modules, migration module which ensures seamless data migration from Relational to NoSQL databases using the metadata information contained in data dictionary and mapping module which provides the facility for executing the MySQL queries to let the application code remain unchanged. It keeps the semantics of source database and makes it easy for the programmers to query the database by allowing them to code with a query on relational database.

The queries coming from the application are captured by the NoSQLayer and converted into the specific NoSQL database format, the result generated by the NoSQL database is again captured by the framework and converted into the application specific format. Mediator is an open source tool which is used for communication between the application and database and Convert is a conversion sub-module which converts all the request coming from the mediator to queries which are supported by the NoSQL database MongoDB.

3.2.3 Evaluations. The framework has been tested by two major techniques; Qualitative evaluation and Quantitative Evaluation. In the Qualitative evaluation various user applications which access relational databases are accessed using the NoSQLayer with a number of operations. Quantitative evaluation evaluates the overhead incurred by the NoSQLayer in the data accessing.

3.2.4 Results. The results of the evaluation stated that the framework is working with the layer and the overhead incurred by the layer is significant when the data involved in the operations is low but it is less when the amount of data is increased and the NoSQLayer gives efficient results. The proposed frameworks works better with large volume of data when compared with MySQL databases. The evaluation compares the runtime of various queries with varying data volume and the proposed framework is found an efficient solution for data migration from Relational to NoSQL database MongoDB.

3.2.5 Limitation. It works only with a specific target NoSQL database MongoDB which makes its scope limited.

3.3 Content Management System approach for Schema De-normalization [3]

3.3.1 Technique Used. CMS (Content Management System) software for SQL-to-NoSQL Schema De-normalization and Migration. Content management systems (CMS) are popular Internet system to publish web contents and able to extend new functionalities. CMSs are generally used to confirmed e-commerce online shops, community portals, personal blogs, or organization websites. CMSs acts the role of gathering data and then can be easily extended to become bright Internet systems. Most popular CMSs, e.g., WordPress and Joomla, pass along relational databases or SQL databases that is MySQL, PostgreSQL, etc.

3.3.2 Experiments and Results. The paper shows how to migrate the original CMS software from SQL databases to NoSQL databases becomes one rising and sarcastic research issue. As the hot stuff of big data, many website administrators migrate their websites from physical machines into the cloud environment. The goal of migrating the SQL database service to the cloud environment is the flexibility while scaling up or out for greater computing power. The SQL database has the well -

structured data and promotes the cross-table query. However, the well-structured characteristic of the SQL database also limits horizontal scaling.

3.3.3 Advantages. The key contribution of the paper is to denormalize SQL database schemas and then migrate into a NoSQL database independently.

3.4 Hbase database Technique [4]

3.4.1 Technique used. In this paper they have used Hbase database technique to implement the hush database. They also use traditional web-based Content Management System (CMS).

3.4.2 Experiments and Results. This paper views how an automatic SQL-to-NoSQL schema transformation mechanism is working on the MySQL and Hbase databases. With a view to avoid cross-table query in the NoSQL database, their design is to follow NoSQL's DDI principles. This paper yields Hbase, it is the most popular column-oriented procedure. The database called hush from the Hbase basically provides two version schemas. Among them one is the SQL version. Another one is NoSQL version.

Query Approach: Only SQL database does not provides different mechanism rather than the tabular relationship of relational database for storing data. Whereas NoSQL database is divided into four categories: i) Column, ii) Key-Value, iii) Document and iv) Graph, in which a column-oriented NoSQL database also composed with tables.

3.4.3 Advantages. NoSQL database may not get better performance always, but Hbase technique provides better performance, flexibility and scalability. An automatic SQL-to-NoSQL schema transformation reduces the migration overhead from SQL to NoSQL data.

3.5 Data Adapter approach [5]

3.5.1 Technique used. Data Adapter system for using hybrid database. Data Adapter system integrates RDBMS (Relational Database Management System) and NoSQL databases, which handles database transformation. In this paper they also used MapReduce process. By using this method we can use maximum number of mappers.

3.5.2 Experiments and Results. The result shows that the data adapter consumes very little time to parse each query to get important data and information, so that it is less time consuming and it takes less time. As we all know a rapid number data is increasing day by day, for controlling the growth of application the data adapter technique for converting SQL data to NoSQL database. In this paper they focuses on the Big Data Applications. The main features of data adapter are SQL interface to RDB and NoSQL database, DB converter, Query approach. They also used MapReduce process. By using this method we can use maximum number of mappers by ourselves. Amazon Elastic MapReduce test data is used as their RDB data source.

Query Approach: The data adapter provides a mechanism that application acquire both relational and NoSQL database whether data transformation is conducting or not. An important design is designed to check the process of query executing.

3.5.3 Advantages. The user can understand the system easier. User can choose different query approach like BT, BD, DA for different scenario according to their need. BT mode is useful for batch application instead of real-time application. The advantage of BD mode is it allows faster patching process. DA mode provides almost delay-free data transaction.

3.6 Automatic Mapping Framework [6]

3.6.1 Technique Used. A framework which provides automatic mapping of relational databases to a NoSQL database MongoDB.

3.6.2 Experiments and Results. NoSQL is efficient for both relational and object-oriented databases which solve the problem for recent applications that uses object oriented databases and are not compatible with SQL queries. The NoSQL database MongoDB is a collection of key-value pairs. The data in a document are somehow related to each other. While mapping the relational databases to MongoDB collections, the tuples are mapped as documents. The mapping of Relational databases to NoSQL involves the consideration of relationships that exists between various tables of relational database. In the 1:1 relationships the best way of mapping involves embedding. In 1:M relationships embedding, linking or bucketing strategy can be used whereas in N:M relationships one or two way embedding can be used. The framework uses the metadata of the source data residing at the relational databases which contains the name of the tables, type of data stored in the tables and the access privileges etc. The mapping steps involves: Creating the MongoDB database, creating new tables in the databases according to various cases of the relationships with other tables.

3.6.3 Advantages and Limitations. The framework proposed by the researchers is capable enough to handle the existing relational databases which have a large amount of data residing into it as the schema-less NoSQL databases. The algorithm used in the proposed framework uses the metadata stored in the relational databases. The paper only gives mapping output for MongoDB so the work can be enhanced for other NoSQL databases. The framework can be extended to handle a large amount of data.

3.7 Unified transparent approach ZQL [7]

3.7.1 Technique Used. A unified transparent query engine as middleware known as ZQL by using MySQL and hive.

3.7.2 Experiments and Results. The main features of ZQL are: It supports cross-database operations including the both NoSQL and RDBMS. ZQL language is translated to the underlying languages by a syntax tree with its high efficiency. ZQL language is simple, thus the development process is easy to handle. ZQL language has high efficiency and strong extensibility to support large number of databases.

3.7.3 Advantages. ZQL aims to support the applications of developing by hiding the specific details of both NoSQL databases and RDBMS. ZQL automatically translates the input query languages into the underlying languages of NoSQL and RDBMS, respectively.

4. Comparative Study of Data Migration Approaches

The proposed mid-model approach is efficient to migrate data to any of the NoSQL database with uniformity. It is a uniform model which is suitable for any type of output NoSQL database but it requires to maintain Data and Query features of the data stored in the relational database which can be extracted from Meta-Data and this incur extra overhead, whereas the NoSQLayer provide the facility to the programmers to code independently from the underlying database. The layer is evaluated and proved cost and time efficient for larger data not for small amount of data, but it is efficient only for MongoDB NoSQL database. The Content management approach provides data migration by De-normalizing the Relational databases and then migrating to NoSQL. Hbase technique provides better performance, flexibility and scalability then all other approaches and less overhead is incurred. Data Adapter query is better from all the other approaches when compared with the time taken to process the query. The transaction in this approach are delay-free and provide quick data access. The automatic mapping framework uses the features of relational database and rigid to MongoDB NoSQL database only. ZQL is suited for the applications which works with both the databases Relational and NoSQL because it provides cross-database operations and hides all the underlying details. So the NoSQLayer and ZQL provides application code independence whereas Hbase provides high performance and data adapter takes least time in query processing. Mid-model produces uniform output data for any of the NoSQL database.

Table 1. Comparison of data migration approaches

S. No.	Author Name	Technique Used	Advantages	Efficiency
1	Liang, D., Lin, Y., & Ding, G. [1]	Mid-Model Design	Integrity of data Uniform for Any NoSQL databases	Requires metadata information
2	Rocha, L., Vale, F., Cirilo, E., Barbosa, D., & Mourão, F. [2]	NoSQLayer	No changes required in the application code. Quantitative and qualitative evaluation ensures efficiency.	Only useful for MongoDB.
3	Chao-Hsien Lee, and Yu-Lin Zheng [4]	Correlation-aware Technique	Transform analyzed data into the same Hadoop data node.	Cardinality should be the combination of all primary keys with the longest chained length.
4	Chao-Hsien Lee and Yu-Lin Zheng [3]	Traditional web based content management systems (CMS)	Flexibility while scaling up Greater computing power.	May limit the scalability
5	Liao, Y. T. et al., [5]	Data Adapter system, MapReduce	Support hybrid database architecture Can handle database transformation.	Data inconsistency problem may occur.
6	Stanescu, L., Brezovan, M., & Burdescu, D. D. [6]	Automatic mapping framework.	Maintain the features of relational database. Data access easier than MySQL.	Works only for MongoDB. Causes overhead. Metadata required.
7	Xu, J. et al. [7]	Transparent query engine, ZQL	Hide the specific details of both NoSQL databases and RDBMS	Replication problem may occur

5. Conclusion

To meet the growing demands of currently used application of managing a huge amount of data in an efficient manner has emerged the necessity of schema-less NoSQL databases which are capable of handling large amount of data and makes the access to this data easy and efficient. All the data resides mostly on the traditional databases in the form of tables which is not suitable for real-time applications and social media sites etc. and this data needs to be migrated from Relational to NoSQL databases. A number of models, framework and layers have been proposed in recent years to migrate the existing data which is stored in relational databases and queries in MySQL to be appropriately executed without emerging the need of changes in the underlying application code. This paper gives a review of some of the recent researches conducted to migrate data from relational to NoSQL databases. The technology which is used for migration and various advantages and limitations of the researches has been listed out in the survey table. The paper helps the researchers to take a review of the current researches which have been accomplished for data migration which can serve as a base for their further study. This paper summarizes some of the most recent techniques which can be used for data migration from Relational to NoSQL databases.

References

[1] Liang, D., Lin, Y., and Ding, G. (2015, December). Mid-model Design Used in Model Transition and Data Migration between Relational Databases and NoSQL Databases. In Smart [1] City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on (pp. 866-869). IEEE.

[2] Rocha, L., Vale, F., Cirilo, E., Barbosa, D., and Mourão, F. (2015). A Framework for Migrating Relational Datasets to NoSQL. Procedia Computer Science, **51**, 2593-2602.

[3] Lee, C. H., and Zheng, Y. L. (2015, October). SQL-to-NoSQL Schema De-normalization and Migration: A Study on Content Management Systems. In Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on (pp. 2022-2026). IEEE.

[4] Lee, C. H., and Zheng, Y. L. (2015, June). Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases. In Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on (pp. 426-427). IEEE.

[5] Liao, Y. T. et al., (2016). Data adapter for querying and transformation between SQL and NoSQL database. Future Generation Computer Systems, **65**, 111-121.

[6] Stanescu, L., Brezovan, M., and Burdescu, D. D. (2016, September). Automatic mapping of MySQL databases to NoSQL MongoDB. In Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on (pp. 837-840). IEEE.

[7] Xu, J. et al., (2016, August). ZQL: A Unified Middleware Bridging Both Relational and NoSQL Databases. In Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2016 IEEE 14th Intl C (pp. 730-737). IEEE.

[8] Atzeni, P., Bugiotti, F., Cabibbo, L., and Torlone, R. (2016). Data modelling in the NoSQL world. Computer Standards and Interfaces.

[9] Yangui, R., Nabli, A., and Gargouri, F. (2016). Automatic Transformation of Data Warehouse Schema to NoSQL Data Base: Comparative Study. Procedia Computer Science, **96**, 255-264.

[10] Bicevska, Z., and Oditis, I. (2017). Towards NoSQL-based Data Warehouse Solutions. Procedia Computer Science, **104**, 104-111.

[11] Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The NoSQL landscape. Information Systems, **63**, 1-23.

[12] Yoon, J., Jeong, D., Kang, C. H., and Lee, S. (2016). Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study. Digital Investigation, 17, 53-65.

[13] Dharavath, R., and Kumar, C. (2015). A scalable generic transaction model scenario for distributed NoSQL databases. Journal of Systems and Software, **101**, 43-58.

[14] Karnitis, G., and Arnicans, G. (2015, June). Migration of Relational Database to Document-Oriented Database: Structure Denormalization and Data Transformation. In Computational Intelligence, Communication Systems and Networks (CICSyN), 2015 7th International Conference on (pp. 113-118). IEEE.

[15] Goyal, A., Swaminathan, A., Pande, R., and Attar, V. (2016, April). Cross platform (RDBMS to NoSQL) database validation tool using bloom filter. In Recent Trends in Information Technology (ICRTIT), 2016 International Conference on (pp. 1-5). IEEE.

[16] Zhao, G., Li, L., Li, Z., and Lin, Q. (2014, November). Multiple Nested Schema of HBase for Migration from SQL. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014 Ninth International Conference on (pp. 338-343). IEEE.

[17] Bansel, A., González-Vélez, H., and Chis, A. E. (2016, February). Cloud-Based NoSQL Data Migration. In Parallel, Distributed, and Network- Based Processing (PDP), 2016 24th Euromicro International Conference on (pp. 224-231). IEEE.

[18] Alae El Alami; Mohamed Bahaj, 2016 (ICMCS). Migration of a relational databases to SQL: The way forward, 2016 5th International Conference on Multimedia Computing and Systems (pp: 18 – 23).IEEE.

[19] Zhao, G., Huang, W., Liang, S., and Tang, Y. (2013, September). Modelling MongoDB with relational model. In Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on (pp. 115-121). IEEE.

[20] Zhao, G., Lin, Q., Li, L., and Li, Z. (2014, November). Schema conversion model of SQL database to nosql. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014 Ninth International Conference on (pp. 355-362). IEEE.

[21] Aaron Schram and Kenneth M. Anderson. MySQL to NoSQL: Data modelling challenges in supporting scalability. In ACM SPLASH '12, pages 191–202, 2012.