A Comparison of NoSQL and SQL Databases over the Hadoop and Spark Cloud Platforms using Machine Learning Algorithms

Chao-Hsien Lee, *Member, IEEE*, and Zhe-Wei Shih
Department of Electronic Engineering, National Taipei University of Technology
Taipei City, Taiwan (R.O.C.)
Email: chlee@ntut.edu.tw (Correspondence)

***Abstract*—Machine learning (ML) algorithms have been widely applied to data analytics and prediction. This paper extends our previous work, which proposed how to automatically convert data from the traditional SQL database to the new NoSQL database, to support the entity-relationship (ER) model. Based on our experimental results over two different cloud platforms, the NoSQL database can always provide better performance than the SQL database while executing ML algorithms.**

I. INTRODUCTION

The Internet of Things (IoT) technique interconnects all kinds of end devices, including smart devices and embedded sensors. A mass of big data from end devices is converged toward cloud platforms that have the characteristics of parallel processing and high scalability. Thus, data science discusses and utilizes interdisciplinary artificial intelligence (AI) algorithms, e.g., machine learning (ML), to extract knowledge or insights from structured or unstructured data, and then do prediction and decision making in response to people or environments. Nowadays, smart environments, i.e., smart home, smart factory or smart city, are rapidly established and realized around the world.

Regarding the data stored in cloud platforms, two kinds of databases are (1) traditional SQL database, which organizes data into more than one table, and (2) new NoSQL database, which provides more flexible data composition. However, most engineers and programmers are familiar with the SQL database than the NoSQL database. Hence, our previous work has proposed the concept to transform from the SQL database to the NoSQL database automatically without re-designing the corresponding schemas [1, 2]. This paper extends our previous work to support the entity-relationship (ER) model that is one common tool to describe how data are structured and implemented in the SQL database. In order to evaluate the transformed NoSQL database, we utilize two ML algorithms, i.e., random forest and k-means, which is provided by Mahout and MLlib, and operated on two cloud platforms, i.e., Hadoop and Spark.

This paper is organized as follows. Section II introduces the proposed architecture of our ER-model based SQL-to-NoSQL transformation. Section III has the corresponding evaluation and discussion. Finally, Section IV concludes this paper and possible future work.

II. ER-MODEL BASED SQL-TO-NOSQL TRANSFORMATION

In the SQL database, normalization is the explicit process to discover the relationships among tables. Once the normalization is finished, data are organized into tables without redundancy. Users can access data across tables by the JOIN operation. On the contrary, column-based NoSQL database, e.g., HBase, allows data to be stored in one big table. Thus, no cross-table query is required. In our previous work [1, 2], the concept of automatic SQL-to-NoSQL transformation is proposed and measured with content management systems (CMSs). However, it lacks the theoretical basis for all conditions of various SQL database schema design. This paper extends our previous work [1, 2] and adopts the ER model to analyze possible relationships among SQL tables.

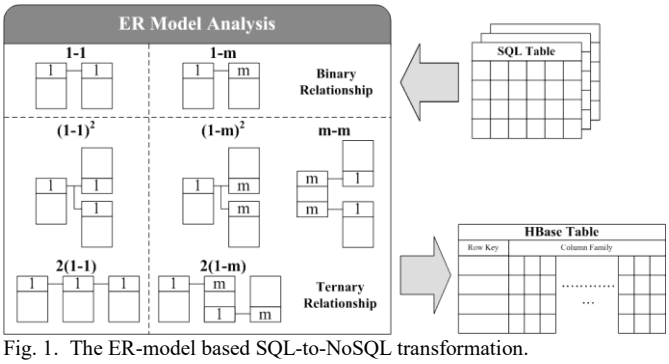

Fig. 1. The ER-model based SQL-to-NoSQL transformation.

Fig. 1 depicts the proposed ER model analysis. According to the ER model, there are two types of relationships, i.e., (1) binary relationship that means the relationship is established between two tables, and (2) ternary relationship that means the relationship is established among three tables. Two basic binary relationships include (i) one-to-one (denoted by 1-1), in which one element of A is only linked to one element of B, and vice versa, and (ii) one-to-many (denoted by 1-m), in which one element of A may be linked to many elements of B, but one element of B is linked to only one element of A.

The basic ternary operation is many-to-many (denoted by m-m), in which one element of A and C may be linked to many elements of B, but one element of B is linked to only one element of A and C. However, when we use two times of the binary relationship, it would induce more types of ternary relationships, including (a) $(1\text{-}1)^2$ that means A can be linked to B and C with the one-to-one relationship, (b) 2(1-1) that means A can be linked to B and B can linked to C with the one-to-one relationship, (c) $(1\text{-}m)^2$ that means A can be linked to B and C with the one-to-many relationship, and (d) 2(1-m) that means A can be linked to B and B can be linked to C with the one-to-many relationship. Therefore, there are totally 7 relationships among tables based on our ER model analysis.

Fig. 2. The performance of the GAME database.

After analyzing all relationships among tables, we should determine the row key of the big table inherited from our previous work. Due to the page limitation, the derivation of the row key creation rules is not described in this paper. When tables with the 1-1 series relationships, i.e., 1-1, $(1\text{-}1)^2$, and 2(1-1), the row key is determined as the original primary key of tables. When tables with other relationships, i.e., m-m, 1-m, $(1\text{-}m)^2$, and 2(1-m), the row key is determined as the concatenation of all tables' primary keys.

III. Performance Evaluation

In order to evaluate whether the transformed NoSQL database is well-functioned, we utilize two ML algorithms, i.e., (1) k-means and (2) random forest, two cloud platforms, i.e., Hadoop with Mahout and Spark with MLlib, and two databases, i.e., MySQL and HBase. Three desktop computers, that equip Intel i7 950 CPU, 16 GB RAM and 3 TB HDD, are composed of our Cloudera Platform. One open dataset called OpenDota is a data dump of Dota 2 matches from Mar. 2016 [3]. We randomly extract 10 million matches (about 4 GB) to do the transformation from the SQL schema to the NoSQL schema. Once the transformation is done, we configure two databases as the data source and then execute two ML algorithms over different cloud platforms. Fig. 2 depicts the experimental results. Basically, all cases using the NoSQL database have shorter execution time (26%-54% improvement) than the one using the SQL database.

IV. Conclusion and Future Works

This paper extended our previous work to support the ER model which enables to analyze all kinds of SQL databases and helps to select better row key for automatic NoSQL transformation. According to our experimental results, the transformed NoSQL database over different combinations of ML algorithms and cloud platforms provide better performance than the SQL database. In the future, we will further discuss and measure the details of our improvement.

Acknowledgment

The research is supported by the Ministry of Science and Technology of the Republic of China (Taiwan) under the grant number MOST 106-2221-E-027-005 and 106-2218-E-027-002.

Reference

[1] C. H. Lee and Y. L. Zheng, "Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases," *Proceedings of IEEE International Conference on Consumer Electronics – Taiwan (ICCE-TW)*, Taipei, Taiwan, Jun. 2015, pp. 426-427.

[2] C. H. Lee and Y. L. Zheng, "SQL-to-NoSQL Schema Denormalization and Migration: A Study on Content Management Systems," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Kowloon, Hong Kong, Oct. 2015, pp. 2022-2026.

[3] Dota 2 matches from OpenDota (formerly yasp.co), http://academictorrents.com/details/1a0c5736bb54610ad00a45306df2b33628301409