2018 IEEE 11th International Conference on Software Testing, Verification and Validation

Analysis of the logical consistency in Cassandra

Pablo Suárez-Otero
University of Oviedo, Spain
suarezgpablo@uniovi.es

***Abstract.* — Most methods to test the consistency of data in relational databases are based on checking the integrity of the data through testing of the schema. However, in NoSQL databases, there is no schema and the data is denormalized, so we need new methods to test the consistency of the database. One of these NoSQL databases is Cassandra, where each table is created to satisfy one query. As the same information could be retrieved by several queries, this information may be found in several tables. In these cases, whenever there is a change of the values of a row in a table, the consistency could be affected. In this thesis, we propose a preventive approach to this problem, where we test the consistency of the data by doing a static analysis of the Cassandra tables when there is a change of the data that could affect the consistency. To achieve this, we will use a conceptual model to help us in the static analysis.**

Keywords— Software testing, static analysis, consistency, databases, NoSQL, Cassandra.

INTRODUCTION

Testing methods related to relational databases have different aims. Some of these methods are focused on testing database applications such as selection of regression test cases for databases [8], the reduction of test databases [9] and the generation of data in order to carry out the testing process [10]. Other methods are focused on the analysis of the schema to test the consistency such as analyzing the mutants for database schemas [1] and the generation of data capable of satisfying and negating the referential integrities of the schema [3]. In these methods the consistency of the data is tested by testing the referential integrities defined in the schema or by mutation testing [3]. However, in NoSQL databases there are no explicit referential integrities or schema, so we need new testing methods to test the consistency of the data. This thesis addresses this problem, proposing a new testing method that detects when there are changes of the data that may cause an inconsistency in the Cassandra database. Additionally, we also want to provide the database statements needed to maintain the consistency of the data. This thesis is being developed in the context of a project to create new testing methods on databases such as [9] and [10].

Cassandra is a NoSQL database where its data modelling uses a query-driven approach, using the queries to organize the data. This means that generally each Cassandra table is designed to satisfy a single query [4]. The same information could be retrieved in several queries, so in each table that satisfies these queries the information is repeated, making the maintenance of consistency more difficult. This problem has been studied by the official development team of Cassandra developing the feature "Materialized views" [5] which are table-like structures used to query the information stored in a table (named base table) in several ways, needing only to maintain the consistency in the base table. However, each materialized view is synchronized with only one base table, it not being possible to combine information from more than one table in a materialized view.

When there is a change of data in the database, it is important to determine if this change produces inconsistencies in order that such inconsistencies may be prevented. Consider the situation of a Cassandra database that stores data relating to authors and their books. This database has two tables created to satisfy the queries "find books given an author" and "find information of a book given its identifier". Note that the information pertaining to a specific book is repeated in both tables. Suppose that a function that inserts books in the database is developed. In a relational model we would test one case where we insert the correct values of a book and another case that should be rejected by the database, like inserting a book without an author. However, when testing the consistency in Cassandra, we want to test if the developed function produces inconsistencies, like for example inserting the book in only one table.

Studies such as [7] create Cassandra tables using a conceptual model and queries, introducing the idea of using a conceptual model that is directly related to the Cassandra tables, which could help us in our proposal. Since NoSQL databases generally do not have this model, there has been researches such as [11] where a conceptual model is created based on the tables.

Usually, the term 'consistency' in Cassandra refers to the consistency of a row replicated throughout all the replicas in the Cassandra cluster [2]. However, in this thesis, we want to test the consistency of the information repeated among several tables, which we named "Logical Consistency".

RESEARCH QUESTIONS

RQ1: What affects the data consistency in Cassandra?

When there is a change that affects the data of the database, the database consistency can be affected. In this RQ we analyze the situations where these changes cause an inconsistency.

RQ2: How may we test the consistency of data in Cassandra?

In this RQ we seek to propose a testing method that tests the consistency of the data when there is a change in the data of the database. This method consists of a static analysis of the Cassandra tables building upon a conceptual model to determine if the change produces an inconsistency of the data.

RQ3: Is it possible to test the consistency of data in Cassandra without a conceptual model?

0-7695-6388-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICST.2018.00055

As NoSQL databases usually do not have a conceptual model, in this RQ we intend to propose a solution where the conceptual model can be obtained from the Cassandra tables.

APPROACH

In this section, we propose our approach to solve the RQs.

RQ1: What affects the data consistency in Cassandra?

We have identified two types of modifications where the consistency could be affected: 1) When there is a modification regarding the tables (creation of a new table or column) or 2) when there is a modification of data (change of the values of a row in the Cassandra tables or the change of the values of a tuple in the conceptual model). Our approach consists of analyzing the different situations that may arise in either of these two cases.

RQ2: How may we test the consistency of data in Cassandra?

Regarding the types described in RQ1, in this RQ we focus on testing the consistency with a static analysis of the Cassandra tables when there are modifications of data. Firstly, in order to detect the tables where there could be inconsistencies, we use a conceptual model that has a connection with the logical model (representation of the Cassandra tables) [7], where each database column is mapped to one attribute of the conceptual model.

We divide our approach in two: the top-down approach and the bottom-up approach. In the top-down approach, given a modification of data in the conceptual model (values of a tuple), we identify in the logical model the data (values of a row) that must be modified to maintain the consistency and how this must be done. In the bottom-up approach, given a modification of data in the logical model (values of a row), we identify in the conceptual model the attributes where there must be changes of data (values of a tuple) and how they must be changed. As the bottom-up approach determines changes of values of tuples in the conceptual model, which is the entry point of the top-down approach, we can combine both approaches, so we can also test the consistency after a modification of data in the logical model.

RQ3: Is it possible to test the consistency of data in Cassandra without a conceptual model?

In our approach for RQ2, we require a conceptual model. However, Cassandra databases are usually not designed using a conceptual model. In this RQ we aim to create a conceptual model based on the tables of the logical model, using the names of the tables and columns, as in the method presented in [11].

METHODOLOGICAL APPROACH

To define the research product of each RQ and their validation techniques we use the classifications defined in [6].

Research product

The research products for each RQ are: RQ1) A qualitative model where all the situations where the consistency of data is affected in a Cassandra database are identified. RQ2) Two techniques for the top-down and bottom-up approaches, which are aimed at the prevention of the generation of inconsistencies through static testing and their implementation. RQ3) A technique that automates our approach and its implementation.

Validation techniques

The validation techniques for each question are 1) Experience for RQ1, RQ2 and RQ3 analyzing real systems that use Cassandra. 2) Implementation for RQ2 and RQ3 that is carried out by implementing the approaches of both RQs. 3) Evaluation for RQ2 and RQ3 that is carried out by testing the detection of inconsistencies for RQ2 and testing the generation of conceptual models for RQ3.

CURRENT STATUS OF THE THESIS

Thus far, we have studied 48 different scenarios that serve as the basis to be able to test the consistency of the data after a modification of data. These scenarios have been automated through a tool that implements the top-down approach. We have used the model detailed in [7] to evaluate it. We have tested 102 insertions and 26 deletions of entities and relationships, obtaining successful results in preventing the inconsistencies. The average number of database statements were 7 for the insertions and 11 for the deletions.

ACKNOWLEDGMENT

This work was supported by the projects TESTEAMOS (TIN2016-76956-C3-1-R) and PERTEST (TIN2013-46928-C3-1-R) of the Ministry of Economy and Competitiveness, Spain. It has also been supported by the project GRUPIN14-007 of the Principality of Asturias and supported by the ERDF.

REFERENCES

[1] McMinn, P., Wright, C. J., McCurdy, C. J., & Kapfhammer, G. (2017). Automatic Detection and Removal of Ineffective Mutants for the Mutation Analysis of Relational Database Schemas. IEEE Transactions on Software Engineering.

[2] Datastax. Data consistency. 2018. [Online]. Available: https://docs.datastax.com/en/cassandra/3.0/cassandra/dml/dmlAboutDataConsistency.html [Accessed 30-Jan-2018]

[3] Kaphammer, G. M.; Mcminn, P,; Wright, C, 'J. Search-based testing of relational schema integrity constraints across multiple database management systems' in Software Testing, Verification and Validation (ICST), 2013. p. 31-40.

[4] Datastax. Basic Rules of Cassandra Data Modeling. 2015 [Online]. Available https://www.datastax.com/dev/blog/basic-rules-of-cassandra-data-modeling [Accessed 30-Jan-2018]

[5] Datastax. New in Cassandra: Materialized Views. 2015 [Online]. Available https://www.datastax.com/dev/blog/new-in-cassandra-3-0-materialized-views [Accessed 30-Jan-2018]

[6] Shaw, M. 2001. 'The coming-of-age of software architecture research'. In Proceedings of the 23rd international conference on Software engineering. (2001, July). p. 656

[7] Artem Chebotko, Andrey Kashlev, Shiyong Lu. 2015. 'A Big Data Modeling Methodology for Apache Cassandra' in IEEE International Congress on Big Data (BigData'15), 2015, p.238-245.

[8] D. Willmor and S. M. Embury, 'A safe regression test selection technique for database-driven applications', in 21st IEEE International Conference on Software Maintenance (ICSM'05), 2005, p. 421-430.

[9] J. Tuya, C. de la Riva, M.J. Suárez-Cabal, R. Blanco. 'Coverage-Aware Test Database Reduction' in IEEE Transactions on Software Engineering, 2016, p. 941-959

[10] M.J. Suárez-Cabal, C. de la Riva, J. Tuya, R. Blanco 'Incremental test data generation for database queries' in Automated Software Engineering, 2017, p. 719-755

[11] Ruiz, D. S., Morales, S. F., & Molina, J. G. 'Inferring versioned schemas from NoSQL databases and its applications' in International Conference on Conceptual Modeling (ER 2015), p. 467-480.