DATA MIGRATION ACROSS HETEROGENEOUS NoSQL DATABASES IN A BIG DATA CONTEXT

Doctoral Dissertation of:
Marco Scavuzzo

Supervisor:
Prof. Elisabetta Di Nitto

Tutor:
Prof. Stefano Ceri

The Chair of the Doctoral Program:
Prof. Andrea Bonarini

The recent growing interest on highly-available data-intensive applications sparked the need for flexible and portable storage technologies, e.g., NoSQL databases.
NoSQL databases have emerged as the solution to handle large quantities of user-generated content still guaranteeing fault-tolerance, availability and scalability. NoSQL databases offer differentiated properties and characteristics as well as different data models, architectures and data access interfaces.
As a result of these heterogeneities, the development of applications exploiting such kind of technology is strictly dependent on the specific NoSQL solution being adopted.

Thus applications are not portable across different databases and, usually, the adaptation to a different technology requires applications re-engineering and code modifications, which, in turn, result in an increase of the application maintenance costs. This, together with the lack of effective data migration solutions for NoSQL databases, has required, until now, the development of ad-hoc code managing the transfer of data between different NoSQLs.

Applications portability is key when requirements can vary over time; in fact, the introduction of a new requirement might invalidate the original database choice. In this case, if a new database is selected, the data stored by the application in the old database has to be migrated to the new database.

Hence, this thesis aims at providing guaranteed fault-tolerant techniques and supporting architectures to migrate data across heterogeneous NoSQL technologies. In particular, we focus our work on column-family-based NoSQLs, as they are among the most interesting class of NoSQL for their high level of scalability, and elaborate on our approach and supporting architecture, that we call *Hegira4Cloud*, to perform offline and online data migration across heterogeneous NoSQL databases.

The proposed approach is based on the idea of extracting data from the source database, transforming them into an intermediate format, and, finally, translate and store them into the target database. The proposed approach is designed in such a way to cope with the different NoSQL data access interfaces.
Moreover, by means of the intermediate transformation format, that we call Mediation Data Model (MDM), Hegira4Cloud is capable of coping with NoSQLs heterogeneities at the data model level, and it is able to preserve different databases data types, secondary indexes and different atomicity and isolation properties.

The fact that the NoSQL database market currently provides hundreds of different technologies, and that their number is constantly growing, requires an extensible approach towards the migration of data between different representatives of the above mentioned market. Hence, our approach aims at providing an extensible database migration system, which allows developers to easily add support for new databases, without requiring them any additional knowledge about the other databases already supported by the system.

The adoption of a NoSQL database, in place of a traditional relational database management system, is often motivated by the high-availability and low latency that the former is able to provide with respect to the latter. However, many data migration tools, in order to avoid data inconsistencies, require applications to stop using the databases while a data migration is in progress; this service interruption may not be tolerable from the perspective of modern applications, which require high-availability. Hence, our approach provides both offline and online data migration. In particular, when the data migration is performed in an online fashion, our approach guarantees that applications operations are synchronized with the ongoing data migration process, so as to produce two mutual consistent databases.

Long running data transferring processes, typically occurring when migrating huge quantities of data, are more subject to failures. Hence, our database migration system has been built to tolerate failures of any of its components, thus being able to recover and correctly complete failed data migrations even in the presence of a failure within the migration infrastructure.

To prove the effectiveness of our approach we evaluate it through several experiments featuring a real-life case-study.
We conclude that our method and supporting architecture offer an efficient, fault-tolerant and data-preserving mechanism for NoSQL portability and interoperation.