BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

NoSQL data models: contributions to structure representation and performance evaluation

Doctoral thesis summary

PhD student: Camelia-Florina ANDOR
Scientific supervisor: prof. dr. Bazil PÂRV

2021

1. Introduction

In the current context, selecting a database management system (DBMS) unsuitable for an application's specific demands has a significant impact on performance. Modern applications generate an impressive amount of data, which requires adaptable and high-performance DBMSs that can easily operate in a distributed environment. This data generated at an accelerated rate is in various formats and requires complex processing. Relational DBMSs were not designed to function optimally in such conditions. This has led to the emergence of a new generation of DBMSs, adapted to the current requirements of data storage and processing. These new DBMSs are part of the NoSQL category, are generally open source projects and work successfully in a distributed environment. There is a high number of NoSQL implementations available on the market, each with its own data model, query language, and automated data replication and distribution capabilities. The interaction with these DBMSs is more difficult, due to the particularities of each implementation but also to the highly specialized nature in solving certain problems related to big data. Data modeling rules differ from implementation to implementation, and getting acquainted with them takes a lot of time and effort. Some NoSQL implementations do not have a graphical user interface, thus interaction takes place only through a console application. Developers of NoSQL DBMSs are generally very concerned with adding new features and enhancing existing ones, so these DBMSs are evolving rapidly and have significant differences between versions. High performance and data structure flexibility are two of the biggest advantages of NoSQL DBMSs.

Selecting the right NoSQL DBMS for a given application is a daunting task. The wide variety of products available and the big differences between them make it significantly more difficult to make a correct comparison. The technical documentation of the products is also very complex and requires a long study. NoSQL DBMS performance evaluation experiments provide important information about their performance in specific contexts, being useful in the process of selecting the right DBMS.

In this thesis, we aim to simplify the process of selecting the right NoSQL DBMS for a given application or type of application and to facilitate user interaction with NoSQL databases. Specialists in all fields are currently working with large amounts of data that require complex processing. Applications that facilitate database interaction are very helpful, especially in the case of users who do not specialize in computer science.

The context and the description of the problems that we intend to solve in this thesis are presented in chapters 1, 2 and 3.

The original contributions of the thesis are presented in chapters 4, 5 and 6.

2. The structure of the doctoral thesis

The doctoral thesis consists of seven chapters.

Chapter 1 presents the shortcomings of the relational data model, which led to the emergence of NoSQL DBMSs, but also the four fundamental NoSQL data models. NoSQL DBMSs use non-relational data models, are designed to work seamlessly in a distributed environment, and offer a greater degree of flexibility regarding data structure. Each fundamental NoSQL data model (*key-value*, *document*, *column-family*, *graph*) is described in a separate section. Other important topics in the area of NoSQL DBMSs are also addressed, such as data replication and data distribution.

Chapter 2 addresses the problem of selecting the right NoSQL DBMS for a given application. The big differences between NoSQL DBMSs, due to different data models, query languages and other features cause great difficulties in making a correct comparison between different NoSQL implementations. NoSQL DBMS performance evaluation experiments are a solution in this case, as they reveal important information about the performance of each NoSQL implementation tested under specific conditions. Taking into account both common use cases and the complexity of the data model, two NoSQL data models that are considered more versatile (*document* and *column-family*) were chosen and a representative implementation for each is described (MongoDB and Apache Cassandra respectively). Next, the topic of performance evaluation applications is addressed. Both the types of performance evaluation applications and their role in a performance evaluation experiment are described. The most popular performance evaluation application, *YCSB*, is also presented. Performance evaluation experimental studies from the literature in which *YCSB* was used are also succinctly described.

Chapter 3 addresses the issue of difficult interaction with NoSQL databases, mainly due to the fact that each NoSQL DBMS has its own query language. A solution to this problem could be *Visual Query System* applications, which use a conceptual representation of database structure and queries and allow users to interact with the database in a visual manner. Different approaches from the literature that use conceptual graphs to represent the structure of the database and queries are presented. These approaches target relational databases and data stored in XML format, and are implemented in *Visual Query System* applications.

Chapter 4 presents five case studies based on experiments evaluating the performance of MongoDB and Apache Cassandra DBMSs. The performance metrics analyzed in these studies are *throughput*, *average latency*, and *total runtime*. New versions of MongoDB and Apache Cassandra DBMSs are used in the last two case studies. Analyzing the results of the last two case studies, it was found that there are significant performance differences between different versions of the same DBMS. The performance evaluation application used in these five case studies is *YCSB*. Following the use of the *YCSB* application, we noticed some of its shortcomings, such as the lack of a graphical user inter-

face based on a conceptual model for representing the structure of the database and the operations involved in tests. Performing performance evaluation experiments that require custom data sets and custom operation sets with *YCSB* requires changes or extensions of the source code.

In chapter 5 we propose two methods for representing the structure of the database and database queries based on conceptual graphs for MongoDB, respectively Apache Cassandra. These methods can be implemented in both *Visual Query System* applications and in a DBMS performance evaluation application.

Chapter 6 presents two implementations of the MongoDB database structure and query representation method. The *Conceptual Graphs for MongoDB* web application falls into the *Visual Query System* category and offers the ability to represent the structure of databases and database queries in a visual manner using conceptual graphs. The application transforms the conceptual representations into equivalent MongoDB instructions, which can be executed afterwards. The *DBMark* desktop application is a performance evaluation application that provides a graphical user interface based on conceptual graphs. Currently, the *DBMark* application can be used to evaluate the performance of the MongoDB DBMS. With this application, performance evaluation experiments that use custom data sets and custom operation sets can be performed. The graphical user interface based on conceptual graphs facilitates the personalization of experiments, and the application can automatically generate the conceptual representation for the structure of an existing database. Of course, new conceptual representations can be created in the application, in which case the equivalent MongoDB instructions will be generated automatically.

The last chapter presents the conclusions of this thesis and the future research directions. In the future, we intend to develop methods for representing the structure of the database and database queries based on conceptual graphs for other NoSQL DBMSs, but also to conduct other case studies based on experiments evaluating the performance of NoSQL DBMSs.

3. Original contributions

The two problems addressed in this thesis are the difficulty and complexity of selecting the right NoSQL DBMS for a given application and the difficulty of interacting with NoSQL DBMSs, which is mainly caused by the significant differences between query languages. The original contributions aim to simplify the process of selecting the right NoSQL DBMS for a given application and to facilitate interactions with NoSQL DBMSs for users who are not computer science experts.

Performance evaluation experiments play an important role in simplifying the process of selecting the right NoSQL DBMS for a given application. They provide important information about the performance of the DBMSs evaluated under the specific conditions of the considered use case. Chapter 4 presents five case studies based on performance evaluation experiments. In these case studies, the performance of two of the most popular NoSQL DBMSs, MongoDB and Apache Cassandra, was analyzed. The performance metrics analyzed are *throughput*, *average latency*, and *total runtime*. The first three case studies (SC1, SC2 and SC3) analyze the performance of MongoDB and Apache Cassandra DBMSs in terms of throughput, average latency and total runtime. The statistical analysis of the obtained results was also performed. NoSQL DBMSs are evolving rapidly, and there are significant differences between different versions of the same DBMS that can impact performance. In the last two case studies (SC4 and SC5) the experiments in SC1 and SC3 were redone using new versions of DBMSs, and the analysis of the results showed that the impact on performance is significant.

NoSQL DBMS performance evaluation experiments are performed using performance evaluation applications, also called *benchmarking tools* or *benchmarking frameworks*. Chapter 2 presents the two types of performance evaluation applications and their role in the performance evaluation process. The most popular application for evaluating the performance of NoSQL DBMSs, *YCSB*, was used to perform the experiments underlying the five case studies presented in chapter 4. A description of this application but also of the MongoDB and Apache Cassandra DBMSs has been made in chapter 2. Following the use of the *YCSB* application in all five case studies, we identified some of its shortcomings regarding the customization of performance evaluation experiments, such as the lack of a graphical user interface that allows a conceptual representation of the database structure and operations performed in a performance test.

In chapter 5 we have proposed two methods for representing the structure of the database and database queries using conceptual graphs for MongoDB and Apache Cassandra DBMSs. These can be implemented in a DBMS performance evaluation application with support for MongoDB and Apache Cassandra that simplifies performance evaluation experiments which use custom data sets and custom operation sets. The MongoDB database structure and query representation method was implemented in the *DBMark* application, which is described in chapter 6. The *DBMark* application provides a graphical user interface that allows MongoDB database structure and queries to be rep-

resented using conceptual graphs and can be used to perform performance evaluation experiments involving custom data sets and custom operation sets.

Interaction with NoSQL DBMSs, which is hampered by the wide variety of query languages, can be simplified with *Visual Query System* applications. These applications facilitate interaction with the database, especially for users that are not computer science experts. In a *Visual Query System* application, the user can represent the structure of the database and database queries in a visual manner, using a conceptual modeling language. The methods for representing the database structure and queries for MongoDB and Apache Cassandra DBMSs proposed in chapter 5 can also be implemented in *Visual Query System* applications. The MongoDB database structure and query representation method was implemented in an application called *Conceptual Graphs for MongoDB*, described in chapter 6. The application *Conceptual Graphs for MongoDB* simplifies database interaction for the MongoDB DBMS, transforming conceptual graph representations of database structure and queries into instructions that can be executed in MongoDB.

Regarding future research directions, we intend to implement the representation method for database structure and database queries developed for Apache Cassandra in a *Visual Query System* application, but also in the performance evaluation application *DBMark*, in order to offer support for performance evaluation experiments that target Apache Cassandra DBMS. Regarding the performance evaluation experiments of NoSQL DBMSs, we intend to perform new experiments that include other NoSQL DBMSs and use a *cloud-based* infrastructure. Another future goal is to develop methods for representing the structure of the database and database queries using conceptual graphs for other NoSQL DBMSs. These could later be used to extend the functionality of the *DBMark* performance evaluation application and to create new *Visual Query System* applications.

Keywords: NoSQL databases; performance evaluation applications; MongoDB; Apache Cassandra; YCSB; conceptual graphs; data models; big data.

Thesis contents