РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНЖЕНЕРИЯ ЗНАНИЙ

УДК 004.041

Shaldanbayeva N.K.
International Information Technology University
Almaty, Kazakhstan
Scientific superviser: Kalpeyeva Zh.B.

A GENERALIZED SYSTEM FOR MAPPING RELATIONAL AND NON-RELATIONAL (NoSQL) DATABASES

Abstract: *Nowadays systems for mapping relational and non-relational (NoSQL) databases are getting more and more popularity. Because today most businesses faced a reality, where they have to work with a big amount of data in order to make decisions that can affect to their place in the market. The data produced in todays world is growing day by day, which makes it hard to process and analyze it. Traditional relational databases cannot handle the work with big amount of data, while non-relational database is suiting more to work with it. Therefore it is important to make a generalized system which can map relational database to non-relational database. In this paper we introduce a novel methodology for migrating relational database to non-relational database based on two existing works.*

Keywords: *Relational database, non-relational database, migration, horizontal scaling, Hadoop, Cassandra, Hbase, MongoDB, structured, unstructured, MySQL*

Introduction

There is no claim that one type of database is better than the other. Sometimes it is better to use relational database, while in some cases the non-relational database is fitting more. For example, if the system is doing a lot of database transactions, the relational database is better, because it is more reliable. But if your system should work with a big amount of data the non-relational database will be the solution. With the growth of data and with the need to use that data, there will be cases when it is required to move the data from relational database to non-relational one.

Problem statement

The main problem of migration of the data from relational database to non-relational database is their structural incompatibility. In relational database there is a strict and definite structure, while non-relational database stores data without explicit and structured mechanism, which makes it hard to move data between databases.

Moreover, there is no generalized system tool for migration between them. Many of the solutions are more specific to some organization and are developed only for their system. But having a generalized system helps to save development time, and to save required resources of the organization. Therefore it is important to develop such a system.

Related works

Based on the concept of Object Relational Mapping and traditional Extract, Transform and Load data migration technique thesis [1] proposed a methodology for migrating data from RDBMS to NoSQL. The main idea of the methodology is based on the analyzing properly the existing database and create a join criteria, which further is used for scheme design and coding.

The authors in [2] work proposed an approach, named as R2NoSQL, which defines conceptual mappings to enhance the data conversion process. In this paper, [2] presented their approach and some implementation and experimental results, which show that, by using the defined conceptual mappings, they obtain a consistent target NoSQL database with respect to a source Relational one. The concept of this idea mostly lies on the idea of metadata of the existing database.

This paper [3] provides a literature review of some of the recent approaches proposed by different researchers in the sphere of migration of data from relational to NoSQL databases. A number of researchers proposed some approaches for the coexistence of NoSQL and Relational databases together.

This article [4] describes proposed migration mechanism from a relational database to a database-oriented columns type HBase and Cassandra. The first approach that the authors have implemented is migrating MySQL to HBase, for which they have exported the data from the MySQL table in a format readable by HBase such as CSV, and make sure there is a field that represents the key to the HBase table row. Then the [4] used MapReduce to import this data, running an instance of MapReduce in a Cluster, and then mapping information between the column family of the HBase table and the data columns in the CSV file. This is one of the approaches proposed by the authors to migrate MySQL database tables to an HBase table. To do the migration [4] have exported the data from the table of a RDBMS in a format readable by Cassandra such as CSV using the CQL language. CQL is a declarative language similar to SQL and it has the same basic structure as SQL with some differences.

This work [5] provided a new model of migration process which basically consists of three phases, the first of which allows to obtain a copy of these metadata using the principle known as semantic enrichment and which extract the different features of the objects, like aggregation, inheritance and composition, the second phase of the process concerns the automatic generation of a New Optimized Data Model (NODM) which stores all required relational database information in a flattened way. And final step was the mapping of the target relational database into column-oriented form of the Hadoop ecosystem. And by uniting those approaches the authors developed a migration solution from a relational database to a NoSQL column-oriented database.

A generalized system methodology to migrate data from relational database to non-relational database

When working with large volumes of data, like processing and analyzing it, enterprise applications use relational data model that does not support improved performance relative to NoSQL. To support statistical data analysis data migration is required as a part of performance of enterprises. The main differences of NoSQL from Relational model according to their structure and the way they store the information. The structure of the relational databases are more complex in terms of their concept of normalization as compared to NoSQL databases. Based on the rules of normalization they divided their information into different tables with join relationship. On the other hand NoSQL databases store their information denormalized way which is unstructured or semi structured[1].

In order to handle the problem of structure between databases, we should first get the metadata of source database (relational DB) and use it as a starting point in destination database(NoSQL DB), and second we should define all of the relationships of tables within the source database to be able to migrate it to destination database.

Below are the steps of the methodology for migration processes:

Step1: Obtain a copy of metadata using the principle of semantic enrichment, and this to extract the different principles of the objects, including aggregation, inheritance and composition[5].

Step2: Analyze data with detail relationship defined in the database schema, and subsequently design and develop join criterion according to the relationship in order to get complete information.

In relational database tables are defined in a schema using primary key (PK) and foreign key (FK) concept in order to make relationships among them. We should define the relationship between tables: one-to-many, one-to-one or many-to-many. In order to form complete information about an order, tables should be connected using different joining criterion (left join/right join/inner join/outer join). Based on the relationship of different tables shown in their schema we can define the joining criterion[1].

Step 3: Design and develop an implicit schema for MongoDB data storing. Based on the previous step we create a MongoDB schema, where the relationships are given as a nesting of data. The given data will be in the form of JSON document[1].

Step 4: Design and develop class diagrams based on the data analysis and implicit schema. In this step according to the step 1 we define all classes for each table and their relationship[1].

Step 5: Writing codes for classes defined in the class diagrams (refer to Step 1). Based on the class diagram we write code for classes and define their properties, relationships and methods[1].

Step 6: Writing code for Data Migration. This step provides some coding samples that comprises of getting or creating MongoDB data collection, extracting data from different SQL tables in order to form complete information using join criterion identified in Step 1, mapping the extracted data to the BSON objects instantiated from classes (refer to Step 4) and subsequently uploading these collection of objects to the MongoDB collections as BSON document[1].

Conclusion

All of these articles provided different kind of new methodologies, models, architecture and knowledge. But most of them works only with specific organization database. These works can be categorized depending on which NoSQL database researches have used: MongoDB, HBase, Cassandra. Also articles can be categorized by the goal of their work: to make a methodology, to make a model, to implement a real application. In this paper we introduced a novel approach of data migration which uses two existing methodologies and combines them in order to achieve a new generalized system methodology.

REFERENCES

1. Alam, F. et al. DATA MIGRATION: NOSQL. RELATIONAL RDBMS TO NON-RELATIONAL. - 2015.
2. de Freitas, M.C., Souza, D.Y. and Salgado, A.C. et al. Conceptual Mappings to Convert Relational into NoSQL Databases // ICEIS. - 2016. - pp. 174-181.
3. Ghotiya, S., Mandal, J. and Kandasamy, S. et al. Migration from relational to NoSQL database. // Materials Science and Engineering Conference Series. - 2017. - Vol. 263. - No. 4. - p. 042055.
4. Bouamama, S. et al. Migration from a Relational Database to NoSQL // International Journal of Knowledge-Based Organizations. - 2018. - pp. 63-80.
5. Youness Khourdifi, Mohamed Bahaj, Alae Elalami. et al. A new approach for migration of a relational Database into column-oriented NoSQL Database on Hadoop. - 2018.

Шалданбаева Н.К.
Научный руководитель: Кальпеева Ж.Б.

Обобщенная система для отображения реляционных и нереляционных (NOSQL) баз данных

Аннотация: В настоящее время системы отображения реляционных и нереляционных (NoSQL) баз данных становятся все более и более популярными. Потому что сегодня большинство предприятий сталкиваются с реальностью, когда им приходится работать с большим количеством данных, чтобы принимать решения, которые могут повлиять на их место на рынке. Данные, производимые в современном мире, растут день ото дня, что затрудняет их обработку и анализ. Традиционные реляционные базы данных не могут

обрабатывать работу с большим объемом данных, в то время как нереляционные базы данных больше подходят для работы с ней. Поэтому важно создать обобщенную систему, которая может переводить реляционную базу данных в нереляционную. В этой статье мы представляем новую методологию переноса реляционной базы данных в нереляционную базу данных на основе двух существующих работ.

Ключевые слова: реляционная база данных, нереляционная база данных, миграция, горизонтальное масштабирование, Hadoop, Cassandra, Hbase, MongoDB, структурированная, неструктурированная, MySQL.

Шалданбаева Н.К.
Ғылыми жетекші: Кальпеева Ж.Б.

Реляциялық және реляциялық емес (NOSQL) мәліметтер базасын көрсетудің жалпыланған жүйесі

Аңдатпа: Қазіргі уақытта реляциялық және реляциялық емес (NoSQL) мәліметтер базасын көрсету жүйелері танымал бола бастады. Себебі, бүгінгі таңда көптеген кәсіпорындар нарықтағы орнына әсер етуі мүмкін шешімдер қабылдау үшін көптеген мәліметтермен жұмыс істеуге мәжбүр болады. Қазіргі әлемде өндірілген мәліметтер күннен-күнге өсуде, бұл оларды өңдеу мен талдауды қиындатады. Дәстүрлі реляциялық деректер базасы үлкен көлемде жұмыс істей алмайды, ал реляциялық емес мәліметтер базасы онымен жұмыс істеуге ыңғайлы. Сондықтан реляциялық емес дерекқорды реляциялық емес дерекқорға аудара алатын жалпыланған жүйені құру маңызды. Осы мақалада біз екі жұмыс негізінде реляциялық дерекқорды реляциялық емес дерекқорға ауыстырудың жаңа әдістемесін ұсынамыз.

Кілт сөздер: Реляциялық деректер базасы, реляциялық емес деректер базасы, қоныс аудару, көлденең масштабтау, Hadoop, Cassandra, Hbase, MongoDB, құрылымдалған, құрылымданбаған, MySQL

Сведения об авторах:

Кальпеева Жулдыз Бейшеналиевна, PhD, ассистент-профессор кафедры «Компьютерной инженерии и информационной безопасности» Международного университета информационных технологий.

Шалданбаева Назерке Курманжанкызы, магистрант кафедры «Компьютерной инженерии и информационной безопасности» Международного университета информационных технологий.

УДК 004.4'274

Бегарыс А.Б.
Х. Досмұхамедов атындағы Атырау мемлекеттік университеті
Атырау, Қазақстан
Ғылыми жетекші: Кубашева А.Н.

ЭЛЕКТРОНДЫҚ ОҚУ КУРСТАРЫН ҚҰРУДА ISPRING SUITE ПРОГРАММАСЫН ҚОЛДАНУ

Аңдатпа. *Бұл мақалада iSpring Suite программасы Microsoft PowerPoint қолданушыларына қандай қосымша мүмкіндіктер ұсынатыны туралы - слайдтарға орналастыруға болатын iSpring QuizMaker тестілері, iSpring Kinetics интерактивтілігі, SWF форматты флеш-роликтер, YouTube бейне-фильмдері және веб-объектілер туралы айтылған.*