УДК 004.65

D.B.Bui[1], N.N. Ruban[1]
[1]Taras Shevchenko National University of Kyiv
60, Vladimirskaya st., Kyiv, Ukraine, 01033

COMPARATIVE ANALYSIS OF NON-RELATIONAL AND RELATIONAL DATABASES

Д.Б. Буй[1], М.М. Рубан[1]
[1]Київський національний університет імені Тараса Шевченка
вул. Володимирська, 60, м. Київ, Україна, 01033

ПОРІВНЯЛЬНИЙ АНАЛІЗ НЕРЕЛЯЦІЙНИХ ТА РЕЛЯЦІЙНИХ БАЗ ДАНИХ

Relational databases (DB) provide good support for data with a predetermined structure. But the latest trends in the IT sphere have shown that the tools for working with massive data (Big Data), which have different presentations are need. Therefore, non-relational DBs (NoSQL DBs) appeared. One of the key characteristics of NoSQL DBs is that they can work with a huge range of different data structures. The article compares the most common relational and non-relational database.
Key words: databases, SQL, NoSQL, non-relational DB, sharding.

Реляційні бази даних (БД) забезпечують хорошу підтримку даних з наперед визначеною структурою. Але останні тренди в IT сфері показали, що необхідні інструменти для роботи з великими обсягами даних (Big Data), які мають різні структури представлення. Тому і з'явилися нереляційні БД (NoSQL БД). Однією з ключових характеристик NoSQL БД є те, що вони можуть працювати з величезними наборами даних різної структури. У статті порівнюються найпоширеніші реляційні і нереляційні БД.
Ключові слова: бази даних, SQL, NoSQL, нереляційні БД, шардінг.

Introduction

The relational data model is based on the mathematical theory (the theory of sets, the theory of the relations, logic theory) whereas non-relational data bases (DB) have no uniform mathematical theory. The relational model is convenient when the database have the following characteristics – reliability, flexibility, stability. But to correspond to needs of the modern applications where data are huge also usually unstructured, it is necessary to resort to use the one of non-relational DB which will give much more convenience in case of data handling. Further instruments of data operation are considered both in relational, and in non-relational approaches.

Reliability of the DB model shall be checked by means of ACID properties.

- ✓ **A**tomicity. Atomicity designates 'everything or nothing'. If any part of transaction isn't executed then all transaction is read unfortunate.
- ✓ **C**onsistency. Consistency guarantees that any transaction saves coherence of a DB.
- ✓ **I**solation. Insulation guarantees that the repeated transactions which are executed at the same time don't influence execution of each other.
- ✓ **D**urability. The durability guarantees that as soon as transaction was executed, data will remain in the same status.

When the data set is organized by means of formally described tables, the received DB call relational. It is possible to get to elements of such DB easily access, to create new data, to expand. Today this type of information storage is the most widespread for data storage.

On the other hand, when traditional tables aren't used for data storage, such DB call non-relational. These DB can store data in the form of key-value storages a, in the HTML format (eXtensible Markup Language – an extensible markup language), etc.

Non-relational DB also are known as NoSQL (Not Only SQL) [1-2]. These DB provide flexible scaling (sharding) meaning that they increase transparently, adding a new node, and are usually developed with not expensive hardware. In difference from relational DB NoSQL DB are often based on BASE properties (**B**asic **A**vailability, **S**oft state, **E**ventual consistency).

- ✓ Basic Availability. Basic availability – each request with guarantee comes to the finish (successfully or unsuccessfully).
- ✓ Soft state. The flexible status – a status of system can change even without input of new data for achievement of coherence of data over time.
- ✓ Eventual consistency. Coherence eventually – data can be mismatched some time, but come to consistency state after a while.

Also some advantages of non-relational DB are: simple support of replication, simple API (Application Programming Interface), processing of massive data (Big Data) [3].

Relational databases

In 1970 E. Kodd laid the foundations of the relational DB theory. A DB – the application that allows storing and obtaining data very quickly. The relational DB – collection of the data elements organized in formally described tables in which it is possible to get access to data by the different methods. The relational DB – the set of interrelated tables, each of which contains information of certain type objects. Every row contains a unique copy of the data for the respective categories. The principal instrument of data handling in relational DB is the structured query language SQL (Structured Query Language) which allows to get access and to change the data, which are stored in a DB. The SQL language is based on relational calculation, relational algebra, predicates, requests and operators.

We will provide some advantages of a relational DB:

- ✓ the most part of information is stored in a DB, but not in the application,
- ✓ it is simple to add, update or delete data,
- ✓ advantages of data aggregation, search and reports creation,
- ✓ changes in the DB scheme are quite simple.

Two modern most intensively used relational DBMS (**DB M**anagement **S**ystem) – MySQL and Oracle. MySQL increasingly used in the web environment. This system is extremely fast and simple in use, and Oracle is often used for operations with big DB, which are used in banking, insurance, ERP (Enterprise Resource Planning – an enterprise management system), and investment companies. Oracle is used for the solution of complex problems and supports the difficult OLTP (**O**nline **T**ransaction **P**rocessing – transaction processing in real time). Though MySQL and Oracle work similarly, nevertheless there are several basic differences between them.

1. Oracle – expensive system, which can be used only by the large organizations unlike MySQL that is open source system. But MySQL concedes in stability which is maintained by Oracle.
2. Oracle can work with massive data, supports role control, snapshots and packets unlike MySQL.
3. The syntax of Oracle user language is more flexible that is provided with such integrated programming language as PL/SQL (**P**rocedural **L**anguage/**S**tructured

Query **L**anguage – procedural extension of the SQL language) which in turn has larger list of commands in comparison with the MySQL language.

4. Oracle provides the high level of safety. MySQL provides only three security levels: user name, password and layout; whereas Oracle has more perfect methods of protection by creation of access profiles, local and external authentications.
5. Oracle is insensitive to the register, and MySQL is sensitive to the register that is inconvenient when naming a DBs, tables and attributes.
6. For data transfer Oracle uses the XML language, which isn't supported by MySQL.
7. MySQL has only two character types of the data – CHAR and VARCHAR in difference from Oracle that operates with four character types: CHAR, NCHAR, VARCHAR2 and NVARCHAR2.
8. Oracle – very comprehensive DBMS which maintains advanced functionality, using such tools as the active data protection, reserve storage, division and data analysis (Active Data Guard, Audit Vault, Partitioning, Data Mining). MySQL doesn't support reserve storage on the server.
9. Temporary tables are processed differently in Oracle and MySQL. In MySQL such tables are deleted automatically after completion of the current session and are seen only for the user of the current session. In Oracle temporary tables are available to all sessions, but data are seen only for the user of the current session inside.
10. MySQL have backup utilities, such as mysqldump and mysqlhotcopy. Oracle has more perfect tool – the manager on recovery (RMAN). RMAN allows to do automate backup process, using different existing methods and scenarios.

Oracle have a set of different functions, using which the user easily can control the DB, at the same time using simple XML language syntax. In view of high functionality it reduces need for the external software or other tools that in turn allow to support applications of large business.

MySQL is used for web scenarios and games, allows to support small data stores and systems of OLTP. Thus, this DBMS can be used in projects in the smaller companies and at smaller cost. However, it requires the external software that involves shortcomings of reliable functionality that is provided by Oracle.

We will describe some shortcomings that are characteristic for the relational DB.

1. Relational DB don't maintain high scalability.
2. As data in a relational DB are saved in the table form, using of such structure potentially can result in high complexity if data can't be presented in a table style adequately.
3. The most part of the functionality provided with relational DB can simply not used that can result in excessive complexity.
4. Relational DB use the SQL language to work with structured data, but SQL can be very difficult in use by operation with semi-structured data.
5. When the data volume becomes the considerable (huge, massive), the DB will be physically partitioned into several servers. This division involves a problem because connection (JOIN) of tables in the distributed servers is, generally speaking, a complex operation.

Thus, there was a need to store and get access to huge data arrays, using the modern tools; thereby there was a need of data storage for the new shape kept by non-relational DB.

Non-relational databases

Non-relational DBMS – a class of systems which control DB which essential difference is that tables aren't used as structures of storage. Other distinctive features of such systems: SQL isn't used as requests language, join operations aren't supported; it doesn't guarantee execution of ACID properties, but there is a possibility of simple horizontal scaling (sharding). Today there are many classifications for NoSQL DBs, one of classification is based on theorem CAP [4]. Non-relational DBs can be classified depending on a data storage method.

1. Key-value. This approach allows to store data without especially expressed structure. These data consist of a key, which is provided, in the line, and the actual data that is value in the pair key-value. Data can be any primitive of a programming language (sequence, an integer number or an array) [5, 6]. Thus, it weakens the requirement of data formatting for storage, eliminates the need for a fixed data model.

2. Storages of documents (the document-oriented DB). The storages of documents also known as the document-oriented DBs, are generally the programs used for storage, receiving, up-dating of data from a DB.

The basic storage structure used in such DBs – the document. Each storage of documents can differ on the data implementation, however it is supposed that data are coded in some standard format, for example, XML, BSON (**B**inary Java**S**cript **O**bject **N**otation – the interchange digital data format, based on JavaScript), PDF or the Microsoft Office document. Each document is presented by unique key, which is sequence (URI – **U**niform **R**esource **I**dentifier). API or request language support fast extraction of documents based on its contents. For example, request which receives all documents in which a certain field has some certain value.

3. Graph DBs. Graph DBs – databases that as a data structure use the graph, which nodes and edges are allocated with certain properties of represented data. Nodes are represented as objects (people, business or any other element) just as objects are provided in any programming language. Properties of such objects are presented in the form of edges, which connect the appropriate graph nodes. By sequential passing from one node to another the necessary information is find.

4. Columnar DBs. Columnar DB store the data in the form of columns in difference from the standard storage form as in the rows. This approach is justified by the fact that often there is a need for updating of one column for the table, at the same time the other data in rows not need to be modified, i.e. when you change one data column other columns remain unchanged.

5. Object-oriented DBs. Object-oriented DBMS also usually known as OODBMS (**O**bject-**O**riented **D**ata**B**ase **M**anagement **S**ystem). Data are stored as objects, inheritance and, therefore, a possibility of repeated use is supported (like in object-oriented programming).

6. Grid and cloudy DBs. Grid and cloudy DB share grid and cloud computing. Grid computings are used for control the heterogeneous and geographically distributed DBs while cloud computing provides easy access to remote hardware and storage resources.

7. XML DBs. XML DBMS is used to save data in a XML language format. In all XML DB the XML document is the fundamental storage format. However, in some XML DB data of a XML document are partitioned into parts and these parts are saved in tables, thereby using so-called XML-mapping.

8. Multidimensional DBs. Multidimensional DB store data in the form of n-dimensional matrixes. All appropriate elements beforehand are compiled and saved, allowing convolutions and evolvents to be received in an interactive mode. Many products

use this approach, for example, of Arbor Essbase (Arb) and IRI Express (IRI). Multidimensional DB can use a relational DB as a basis in which multidimensional requests are translated in the equivalent relational requests.

9. Multimodel DBs. Multimodel DB – combining of different non-relational DBs, it provides connection of the advantages of different DBs mentioned above.

Non-relational DB propose solutions for storage of massive data, which are used in the Internet today. With the advent of NoSQL cloud computing now can control more effectively massive data and their effective processing. The new architecture of data representation, such as CDSA (**C**loud **D**ata **S**torage **A**rchitecture) improves data handling productivity [7].

Except different advantages, such as high capacity, horizontal scalability, non-relational DB have some shortcomings [8-10]. The most considerable are given below.

1. The majority of non-relational DBs use the open source software and though it well proved, but nobody bears responsibility for failures.
2. Many non-relational DBs provide BASE properties and sacrifice standard ACID properties to increase productivity.
3. Because of neglect ACID properties a level of the reliability provided with non-relational DBs, is less than in a relational case. Developers need to programming for ensure compliance with the ACID that might be easy in a relational database.

Conclusion

Massive data, heterogeneous data, requirements of high performance and scalability of the modern applications of the relational DBMS became the reason that programmers began to look for new decisions that are more flexible. Such decisions have become NoSQL DBMSs. Integrity of data, data availability are the main problems, defining what DBMS must be used for application.

We must note that for achievement the level of relational DBMS reliability and a maturity the some important research questions remain unresolved for NoSQL DBMSs yet.

References

1. Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Э. Редмонд, Д. Уилсон, 2013. – 384 с.
2. Фаулер М. NoSQL. Новая методология разработки нереляционных баз данных / М. Фаулер, П. Садаладж, 2013. – 171 с.
3. Ullman J. Mining of Massive Datasets / J. Ullman, J. Leskovec, A. Rajaraman, 2014. – 513 p.
4. Survey on NoSQL database / H.Jing, E. Haihong, L. Guan, D. Jian // Pervasive Computing and Applications (ICPCA). – 2011. – P. 363–366.
5. Chang F. Bigtable: A Distributed Storage System for Structured Data / F. Chang, J. Dean, S. Ghermawat // Proceeding OSDI '06 Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, 2006 – P. 15.
6. Conolly T. Begg C. Database Systems: A Practical Approach to Design / T. Conolly, C. Begg / Pearson; 6 edition, 2014. – 1440 p.
7. Jing H. A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing / H. Jing, S. Meina, S. Junde // ICIS '11 Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science. – 10. – P. 351–355.
8. Security Issues in NoSQL Databases / L.Okman, N. Gal-Oz, Y. Gonen,, E. Gudes. // «Trust, Security and Privacy in Computing and Communications» (TrustCom), 2011 IEEE 10th International Conference. – 2011.
9. Leavitt N. Will NoSQL Databases Live Up to Their Promise? / Neal Leavitt // IEEE Computer Society Press Los Alamitos, CA, USA. – 2010. – P. 12–14.
10. Date C. Database in Depth: Relational Theory for Practitioners / C. J. Date., 2005. – 232 p. – (O'Reilly Media).

References

1. Redmond E. Sem baz dannyih za sem nedel. Vvedenie v sovremennyie bazyi dannyih i ideologiyu NoSQL / E. Redmond, D. Uilson, 2013. – 384 s.
2. Fauler M. NoSQL. Novaya metodologiya razrabotki nerelyatsionnyih baz dannyih / M. Fauler, P. Sadaladzh, 2013. – 171 s.
3. Ullman J. Mining of Massive Datasets / J. Ullman, J. Leskovec, A. Rajaraman, 2014. – 513 p.
4. Survey on NoSQL database / H.Jing, E. Haihong, L. Guan, D. Jian // Pervasive Computing and Applications (ICPCA). – 2011. – P. 363–366.
5. Chang F. Bigtable: A Distributed Storage System for Structured Data / F. Chang, J. Dean, S. Ghermawat // Proceeding OSDI '06 Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, 2006 – P. 15.
6. Conolly T. Begg C. Database Systems: A Practical Approach to Design / T. Conolly, C. Begg / Pearson; 6 edition, 2014. – 1440 p.
7. Jing H. A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing / H. Jing, S. Meina, S. Junde // ICIS '11 Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science. – 10. – P. 351–355.
8. Security Issues in NoSQL Databases / L.Okman, N. Gal-Oz, Y. Gonen,, E. Gudes. // «Trust, Security and Privacy in Computing and Communications» (TrustCom), 2011 IEEE 10th International Conference. – 2011.
9. Leavitt N. Will NoSQL Databases Live Up to Their Promise? / Neal Leavitt // IEEE Computer Society Press Los Alamitos, CA, USA. – 2010. – P. 12–14.
10. Date C. Database in Depth: Relational Theory for Practitioners / C. J. Date., 2005. – 232 p. – (O'Reilly Media).

RESUME

Д.Б. Буй, М.М. Рубан

Порівняльний аналіз нереляційних та реляційних баз даних

Реляційні бази даних (БД) забезпечують хорошу підтримку даних з наперед визначеною структурою. Але останні тренди в IT сфері показали, що необхідні інструменти для роботи з великими обсягами даних (Big Data), які мають різні структури представлення. Використання традиційних реляційних БД обмежено тим фактом, що у них є строгі обмеження на структуру даних. Тому і з'явилися нереляційні БД (NoSQL БД). Однією з ключових характеристик NoSQL БД є те, що вони можуть працювати з величезними наборами даних різної структури. Зокрема, завдяки їхньої розподіленої природи, а також з огляду на інші засоби зберігання даних на фізичному рівні вони ідеально підходять для обробки величезної кількості інформації, як це часто буває в разі інтелектуального аналізу даних (Data Mining). Іншими словами, канонічні структури даних реляційного підходу перешкоджають ефективному використанню реляційних БД. У статті порівнюються найпоширеніші реляційні і нереляційні БД. БД багато в чому пов'язані з управлінням великими даними, тому необхідно враховувати такі властивості як стабільність і швидкість.

Надійшла до редакції 3.07.2016

