

УДК 004.65

Рубан Н.Н., аспирант

Сравнение нереляционных и реляционных баз данных

Киевский национальный университет имени
Тараса Шевченко, 03187, г.Киев, пр-т. Глушкова
4д
e-mail: rubannn@gmail.com

N. N. Ruban, postgraduate

Comparison of non-relational and relational databases

Taras Shevchenko National University of Kyiv,
03187, Kyiv, Glushkova st., 4d,
e-mail: rubannn@gmail.com

В работе приведено сравнение наиболее распространенных баз данных (БД), которые используются для хранения информации. БД в значительной степени связаны с управлением большим количеством данных, при этом необходимо учитывать стабильность и скорость работы. Рассмотрены основные характеристики реляционных и нереляционных БД.

Ключевые слова: базы данных, SQL, NoSQL, нереляционные БД, sharding.

Relational database (DB) provide good support data for grid before defined structure. But recent developments in the IT sphere need tools for big data (Big Data), which have very large volumes and various data structures. Using a traditional relational database compounded by the fact that they have a strict limit on the data structure. One of the key elements of all document-oriented databases is that they can work with much larger data sets and structures than usual. In particular, because of their distributed nature, and other means of physical data storage, they are ideal where you need to process huge amounts of information, as is often the case for data mining. In other words, canonical data structure relational approach is hindering their effective use. The paper compares the most common database, which are used for storing information. DB is largely associated with the management of large amounts of data, and the need to consider the stability and speed. Considered the main characteristics of the non-relational and relational databases.

Key Words: databases SQL, NoSQL, non-relational DB, sharding.

Статью представил д.ф.-м.н., проф. Буй Д.Б.

Реляционная модель основывается на математической теории (теория множеств, теории отношений, математической логике), тогда как нереляционные БД могут не иметь единой математической теории. Реляционная модель выгодна, когда БД должна иметь следующие характеристики – надежность, гибкость, устойчивость, масштабируемость. Но чтобы соответствовать потребностям современных приложений, где данные огромны и обычно неструктурированные, то следует прибегнуть к использованию одной из нереляционных БД, которая даст гораздо больше удобства при обработке данных. Далее рассмотрены инструменты работы с данными как в реляционном, так и в нереляционном подходах.

Надежность модели БД должна быть проверена с помощью свойств ACID.

✓ Atomicity. Атомарность обозначает 'все или ничего'. Если какую-либо часть транзакции

не выполнена, то тогда всю транзакцию считают неудавшейся.

✓ Consistency. Непротиворечивость гарантирует, что любая транзакция сохраняет согласованность БД.

✓ Isolation. Изоляция гарантирует, что многократные транзакции, выполняющиеся одновременно, не влияют на выполнение друг друга.

✓ Durability. Долговечность гарантирует, что как только транзакция выполнена, то данные останутся в том же состоянии.

Когда набор элементов данных организован с помощью формально описанных таблиц, полученную БД называют реляционной БД. К элементам такой БД можно легко получить доступ, создать новые данные, расширить. Данный вид хранения информации является наиболее распространенным для хранения

данных, имеющих иерархическую или сетевую структуры.

С другой стороны, когда традиционные таблицы не используются для хранения данных, то такие БД называют нереляционными. Эти БД могут хранить данные в форме хранилищ ключ-значение, формата XML и т.д.

Нереляционные БД, также известные как NoSQL (Not Only SQL), обеспечивают гибкое масштабирование (sharding), означающее, что они увеличиваются прозрачно, добавляя новый узел, и обычно разрабатываются с недорогими аппаратными средствами. Также одним из преимуществ нереляционных БД являются: простая поддержка репликации, простой API (Application Programming Interface), огромный объем данных (BigData).

Реляционные базы данных

В 1970 г. Э. Кодд заложил основы теории реляционных БД. БД – приложение, которое позволяет хранить и получать данные очень быстро. Реляционная БД – набор элементов данных, организованных в формально описанных таблицах, в которых к данным можно получить доступ различными способами. Реляционная БД – множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Каждая строка содержит уникальный экземпляр данных для соответствующей категории данных. Главным инструментом обработки данных в реляционных БД является язык структурированных запросов (SQL – Structured Query Language), который позволяет получить доступ и изменить данные, хранящиеся в БД. Язык SQL основан на реляционном исчислении и алгебре отношений, опирается на использование предикатов, запросов и операторов.

Представим некоторые преимущества БД, разработанной согласно реляционной модели:

- Большая часть информации хранится в БД, а не в приложении.
- Просто добавить, обновить или удалить данные.
- Это дает преимущества агрегации данных, поиска и создания отчетов.
- Любые изменения в схеме БД довольно просты.

Две наиболее интенсивно используемые реляционные СУБД (системы управления БД) – MySQL и Oracle. MySQL в большей мере используется в веб-среде. Это легкая система, которая чрезвычайно быстра и проста в

использовании, а вот Oracle зачастую используется при работе с большими БД, которые используются в банковском деле, страховании, ERP (Enterprise Resource Planning – система управления предприятием) и инвестиционных компаниях. Oracle используется для решения сложных проблем и поддержки сложных OLTP (Online Transaction Processing). Хотя MySQL и Oracle работают схожим образом, все же есть несколько принципиальных различий между ними.

У Oracle имеется множество различных функций, запуская которые пользователь легко может управлять БД, при этом используя простой синтаксис XML. Ввиду высокой функциональности это уменьшает потребность во внешнем программном обеспечении или другом инструментарии, что в свою очередь позволяет поддерживать приложения крупного бизнеса.

MySQL используется для web-сценариев и игр. Позволяет поддерживать небольшие хранилища данных и системы OLTP. Таким образом, эта СУБД может использоваться в маленьких проектах в меньших компаниях и по меньшей стоимости. Однако это требует внешнего программного обеспечения, что влечет за собой недостатки в надежной функциональности, которую обеспечивает Oracle.

Опишем некоторые недостатки, которые характерны для реляционных БД.

1. Реляционные БД не поддерживают высокую масштабируемость.
2. Поскольку данные в реляционной БД сохранены в виде таблиц, то использование такой структуры потенциально может привести к высокой сложности в случае, если данные нельзя адекватно представить в табличном виде.
3. Большая часть функционала, обеспеченного реляционными БД, может быть просто не использована, что может привести к излишней сложности.
4. Реляционные БД используют SQL, чтобы работать над структурированными данными, но SQL может быть очень сложным в использовании при работе с неструктурированными (слабоструктурированными) данными.
5. Когда объем данных становится значительным (огромным), БД должна быть физически разделена на несколько серверов. Это разделение влечет за собой проблему,

потому что соединение (JOIN) таблиц в распределенных серверах является, вообще говоря, сложной задачей.

Таким образом, возникла потребность хранить и получать доступ к огромным массивам данных, используя современные инструменты; тем самым возникла потребность хранения данных в новой форме, т.е. использование нереляционных БД.

Нереляционные базы данных

Нереляционные БД – класс систем, которые управляют БД, существенным отличием которых является то, что не используются таблицы в качестве структуры хранения. Другие отличительные особенности таких систем: SQL не используется в качестве языка запросов, операции соединения не поддерживаются; это не гарантирует выполнения свойств ACID, но есть возможность простого горизонтального масштабирования (sharding). На сегодняшний день существует много классификаций для NoSQL БД, одна из классификаций основывается на CAP теореме [1]. Нереляционные БД могут быть классифицированы в зависимости от способа хранения данных [3]

1. Ключ-значение.

Данный подход позволяет хранить данные без особо выраженной структуры. Эти данные состоят из ключа, который представлен строкой и фактическими данными, которые являются значением в паре ключ-значение. Данные могут быть любым примитивом языка программирования (последовательность, целое число или массив) [2]. Таким образом, это ослабляет требование форматирования данных для хранения, избавляя от необходимости использования фиксированную модель данных.

2. Хранилища документов (документно-ориентированная БД).

Хранилища документов, также известные как документно-ориентированные БД, являются в основном (компьютерными) программами, используемыми для хранения, получения, обновления данных из БД.

Базовая структура хранения, используемая в таких БД – документ. Каждое хранилище документов может отличаться по своей реализации данных, однако предполагается, что данные включены и закодированы в некотором стандартном формате, например, это может быть XML, BSON, PDF или документ Microsoft Office. Каждый документ представлен уникальным

ключом, который является последовательностью (URI – Uniform Resource Identifier). API или язык запросов поддерживают быстрое извлечение документов на основе его содержания. Например, запрос, который получает все документы, в которых определенное поле имеет некоторое определенное значение.

3. Графовые базы.

Графовые БД – безструктурные базы данных, которые в качестве структуры данных используют граф, вершины и ребра которого наделены определенными свойствами, которые представляют данные. Вершины представляются как объекты (люди, бизнес или любой другой элемент) подобно тому, как объекты представлены в любом языке программирования. Свойства таких объектов представлены в виде ребер, которые связывают соответствующие вершины графа. Путем последовательного перехода от одной вершины к другой находится необходимая информация.

4. Колоночные базы.

Колоночные БД хранят свои данные в форме столбцов в отличие от общепринятой формы хранения по строкам. Данный подход оправдан тем, что часто возникает потребность в обновлении всего столбца в таблице, при этом остальные данные строки в модификации не нуждаются, т.е. при изменении одного столбца данных остальные столбцы остаются без изменений.

5. Объектно-ориентированные базы данных.

Объектно-ориентированные БД также обычно известны как OODBMS (Object-Oriented Database Management System). Данные хранятся в виде объектов, поддерживается наследование и, следовательно, возможность многократного использования (по аналогии с объектно-ориентированным программированием).

Кроме различных преимуществ, таких как высокая пропускная способность, горизонтальная масштабируемость, у нереляционных БД есть некоторые недостатки. Наиболее значительные представлены ниже.

1. Большинство нереляционных БД используют программное обеспечение с открытым исходным кодом и, хотя оно себя хорошо зарекомендовало, но никто не несет ответственности во времена отказов.
2. Много нереляционных БД обеспечивают свойства BASE (Basic Availability, Soft state, Eventual consistency) и жертвуют стандартными свойствами ACID, чтобы увеличить производительность.

3. Из-за пренебрежения свойствами ACID степень надежности, обеспеченной нереляционными БД, ниже, чем в реляционном случае. Разработчикам необходимо прибегать к программированию, чтобы обеспечить ACID, которые, возможно, можно было легко обеспечить в реляционной БД.

Выводы

Представим основные существенные различия между двумя типами БД в виде таблицы (таблица 1).

Беспрецедентные объемы данных, гетерогенные данные, требования производительности и масштабируемости современных приложений реляционных СУБД

вызов как единственный подход для управления данными. Это положило начало в спектре вариантов. Уровни целостности данных, высокой доступности и толерантности разделов есть три основные проблемы, определяющие, какие данные управления Система подходит для данного приложения.

Некоторые вопросы исследования остаются нерешенными для систем NoSQL, чтобы достичь уровня надежности и зрелости СУБД. Это захватывающее время для исследователей баз данных для работы по этим вопросам и разработать ассортимент вариантов управления данными.

Таблица 1

Сравнение некоторых характеристик реляционных и нереляционных БД

	Нереляционные базы данных	Реляционные базы данных
1	Высокая пропускная способность	Низкая пропускная способность
2	Хорошо масштабируются	Менее масштабируются
3	Данные могут быть вставлены в любое время, не определяя схемы данных. Данные могут быть изменены в любое время. Гибкий подход к структуре данных	Данные должны храниться в предопределенных таблицах или структурах
4	Производительность может быть улучшена, кэшируя данные в системную память	Кэширование должно быть сделано с помощью специальной структуры
5	Поддержка BASE свойств	Поддержка ACID свойств
6	Данные не обязательно должны быть непротиворечивыми	Лучшая непротиворечивость (целостность), чем для нереляционных БД
7	Дублирование данных позволено, что угрожает целостности данных. Кроме того, обновление одной записи означает, что необходимо обновлять все другие экземпляры этой записи	Процесс проектирования реляционной БД устраняет проблему дублирования, препятствуя тому, что возникнут противоречивые данные

Список використаних джерел

1. Jing H. Survey on NoSQL database / H.Jing, E. Haihong, L. Guan, D. Jian. // Pervasive Computing and Applications (ICPCA). – 2011. – P. 363–366.
2. 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.
3. Date C. Database in Depth: Relational Theory for Practitioners / C. J. Date., 2005. – 232 p. – (O'Reilly Media).

References

1. JING H. et al. (2011). Survey on NoSQL database. 2011 6Th International Conference On Pervasive Computing And Applications, p. 363 - 366.
2. JING H. et al. (2011). A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing. 2011 10Th IEEE/ACIS International Conference On Computer And Information Science, p. 351-355.
3. Date C. (2005). Database in depth. Sebastopol, CA: O'Reilly.

Надійшла до редколегії 15.12.2015.