

СРАВНЕНИЕ РЕЛЯЦИОННЫХ И НЕ РЕЛЯЦИОННЫХ (NOSQL) БАЗ ДАННЫХ

Зенцов Даниил Александрович

*студент, кафедра геоинформатики и информационной безопасности, Самарский университет,
РФ, г. Самара*

Галеева Аделя Ильдаровна

*студент, кафедра геоинформатики и информационной безопасности, Самарский университет,
РФ, г. Самара*

*Научный руководитель **Додонов Михаил Витальевич***

*доц., кафедра программных систем, Самарский университет,
РФ, г. Самара*

Введение

Роль компьютеров и цифровой обработки информации в нашем веке трудно переоценить. Скорость обработки информации, размеры материальных носителей, объемы хранимых данных, себестоимость составных частей серверов – все эти параметры улучшились в тысячи раз буквально за десятилетие.

С таким резким развитием технологий появляется необходимость хранить имеющиеся и всё поступающие данные. Эту проблему успешно решают базы данных. Какие же существуют возможности для хранения информации и виды баз данных, особенности представления данных?

Концепция реляционных баз данных была предложена в 1970-х годах. Она основана на реляционной модели данных (данные представляют собой множество отношений). Таблицы, в которых хранится информация – лишь визуальное представление отношений.

Не реляционные БД начали использоваться с созданием первых компьютеров, но их широкое применение началось уже в XXI веке. Параллельные распределённые системы для масштабируемых интернет-приложений, таких как поисковые системы, требовали нового подхода к хранению данных и уменьшению количества внутренних операций и проверок внутри БД.

Цель

Целью научной работы является проведение сравнения реляционных и не реляционных (NoSQL) баз данных. Будут рассмотрены следующие характеристики БД: производительность, масштабируемость, консистентность хранимых данных, какая часть работы с данными выносится на уровень приложения.

Реляционные базы данных

Особенностью реляционных БД является предопределенная модель хранимой информации. Данные являются строго типизированными, модель содержит ряд ограничений и отношений для обеспечения целостности данных.

Главные плюсы и, в то же время, ограничения, реляционных БД(ACID):

1. Атомарность (Atomicity) – будут выполнены либо все подоперации внутри операции, либо ни одной.
2. Консистентность данных (Consistency) – все данные будут целостными и согласованными по завершении каждой транзакции.
3. Изолированность (Isolation) – параллельные транзакции не оказывают влияния друг на друга.
4. Устойчивость (Durability) – система сохранит свое состояние независимо от внешних факторов (транзакция прошла – изменение данных не будет утеряно).

Реляционные базы данных используются приложениями на протяжении многих лет и предлагают функции, которые остаются важными и по сей день:

1. Мощный язык запросов и индексы, играющие важную роль в быстром доступе к данным, их поддержка обеспечивается на уровне базы данных, а не приложения.
2. Сильная консистентность данных – приложения имеют возможность работать с данными, которые были только что записаны в БД. Гораздо сложнее создавать приложения вокруг непостоянно консистентных данных (NoSQL), что создает дополнительные трудности разработчикам.
3. Хорошая интеграция в приложения уровня предприятия (Enterprise Management). База данных - всего лишь одна часть приложения, и

она должна легко встраиваться в ИТ-стек компании. Организациям нужна БД, которая была бы безопасной, контролируемой, легко интегрируемая с существующими технологиями, процессами, была бы уже освоена администраторами и аналитиками.

Модель данных заранее определяет, что будет храниться в базе данных. Этот факт является существенным минусом, так как часто необходимо добавлять новые атрибуты в уже существующую схему. Следовательно, разработчику необходимо добавить новый столбец в базу данных, а затем перенести всю базу данных на новую схему. Если база данных большая, это очень медленный процесс, который связан с существенным временем простоя. Если это происходит часто, то время простоя увеличивается в разы. Также нет способа использовать реляционную базу данных для хранения данных, которые полностью не структурированы или неизвестны заранее.

Самые известные реляционные БД – Oracle, SQL Server, MySQL, PostgreSQL.

Не реляционные (NoSQL) базы данных

Современные приложения налагают требования, не решенные реляционными БД, и это привело к разработке баз данных NoSQL, которые имеют следующие преимущества:

1. Гибкая модель данных.

NoSQL базы данных удовлетворяют многие потребности современных приложений. Они предоставляют возможность вставлять данные без predefined схемы. Это упрощает процедуру внесения существенных изменений в режиме реального времени. Разработчикам обычно приходилось добавлять код на стороне приложения, чтобы обеспечить контроль качества данных, например, указать наличие определенных полей, типов данных или допустимых значений. Более сложные NoSQL базы данных позволяют применять правила валидации на уровне БД, что позволяет пользователям обеспечивать управление данными через сами данные, сохраняя при этом гибкость преимуществ динамической схемы.

2. Масштабируемость и производительность.

NoSQL базы данных не обеспечивают по умолчанию атомарность транзакций и консистентность данных (но некоторые позволяют активировать их по необходимости), что позволяет базе данных масштабироваться на используемом в реальном времени оборудовании, а это обеспечивает почти неограниченный рост с более высокой пропускной способностью и меньшей задержкой, чем реляционные базы данных.

3. Максимальное использование облачных вычислений и хранилищ.

Хранилище на основе облачных вычислений позволяет еще лучше расширять базу данных, является экономичным решением по сравнению с собственными серверами и самостоятельным обслуживанием оборудования.

Существуют следующие типы NoSQL баз данных:

1. Базы данных на основе документов.

В такой базе данных уникальный ключ соединяется со сложной структурой данных, известной как документ. Документы могут содержать много разных пар ключ-значение, пар ключей или даже вложенные документы. Пример – MongoDB.

2. Базы данных на основе графов

Такие базы используются для хранения данных с большим количеством связей. К таким NoSQL базам относятся Neo4J и Giraph.

3. Хранилище «ключ-значение»

Хранилища ключ-значение - это самые простые базы данных NoSQL. Каждый элемент хранится в базе пара: имя атрибута (или «ключ») вместе со своим значением. К таким базам данных NoSQL относят Riak и Berkeley DB. Некоторые хранилища значений ключа, такие как Redis, позволяют каждому значению иметь тип, например «integer», который добавляет функциональность.

4. Колоночные базы данных

Такие базы данных способны хранить большое количество данных с большим количеством атрибутов. Колоночные БД хранят данные в

колонках, которые хранятся в отдельных файлах. Такой способ хранения ускоряет запросы.

Вывод

Не реляционные базы данных выигрывают у реляционных по производительности, возможности масштабируемости и гибкости модели данных. Множество небольших приложений уже сейчас используют не реляционные базы данных за счет отсутствия потребности во встроенном по умолчанию обеспечению атомарности операций, постоянной консистентности данных (а не только после окончания транзакции) и жесткой модели данных.

Но для систем уровня предприятий реляционные базы данных являются выигрышным решением. Минимум логики хранения данных на уровне приложения, безопасность доступа к информации обеспечивается “из коробки”, полное выполнение принципов ACID еще долго будут использоваться в Enterprise продуктах.

Список литературы:

1. Дейт, К., Дж. Введение в системы баз данных. 6-е изд. – К.; М., СПб.: «Вильямс», 2015. – 848с.
2. A Comparison Of NoSQL Database Management Systems And Models [электронный ресурс] – Режим доступа. – URL: <https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models> (Дата обращения: 30.04.2018)