

UDC 004.652

Prospects For the Development of Modern Database Technology

¹ Ivan A. Doluev² Mihyail G. Petrov¹ Sochi State University, Russia

Sovetskaya street 26a, Sochi city, Krasnodar Krai, 354000

Student

² Sochi State University, Russia

Sovetskaya street 26a, Sochi city, Krasnodar Krai, 354000

PhD (technical), associate professor

E-mail: kchalovskyi@mail.ru

Abstract. The article discusses comparative characteristics of relational and non-relational approach to database management

Keywords: non-relational databases; database; NoSQL; UnQL.

База данных – неотъемлемая часть современной информационной системы, в том числе и web-приложения. В ней хранится вся информация, необходимая для работы веб-приложения. Исключения составляют файлы: изображения, документы. Они обычно и хранятся в виде файлов в предназначенном для них каталоге на сервере. Однако существует возможность и такую информацию хранить в базе данных.

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям. На данный момент их можно разделить на два вида: Реляционные и нереляционные базы данных. Реляционная база данных – эта база данных, основанная на реляционной модели данных. Термин «реляционный» означает, что теория основана на математическом понятии отношение (relation). Нереляционные базы данных – это базы данных, основанные на сетевой, иерархической, объектно-ориентированной моделях данных.

В последнее время появилось много нереляционных баз данных. Реляционные системы разрабатывались для удовлетворения требований OLTP (online transaction processing, оперативная обработка транзакций), рабочей нагрузки, характеризующейся наличием непредусмотренных запросов, существенного трафика записи, а также для гарантированного обеспечения транзакционности и целостности. Но далеко не всем приложениям необходимы транзакции. Например, web-приложениям. Рассмотрим MySQL – реляционную базу данных, которая поддерживает транзакции, но только в InnoDB. Проблема в том, что InnoDB редко используется: либо InnoDB вообще отсутствует, либо просто в использовании InnoDB нет необходимости. Необходимость использования транзакций отпадает из-за того, что практически каждая операция добавления информации на сайт (в интернет-магазин, интернет-библиотеку и т.д.) является атомарной и не нуждается в транзакциях. Именно для обслуживания базы данных web-приложений и был создан NoSQL^{[1],[2]}.

NoSQL (англ. not only SQL, не только SQL) – термин, обозначающий ряд подходов, проектов, направленных на реализацию моделей баз данных, имеющих существенные отличия от используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

Термин NoSQL стал известен относительно недавно и был введен для описания различных технологий баз данных, возникших для удовлетворения требований, известных как "Web-scale" или "Internet-scale".

Есть три основных требования к Web-scale приложениям:

1. Много данных: самые большие из веб-приложений обрабатывают объёмы данных на порядки больше тех, что предполагались для управления базами данных (Facebook – 50 терабайт для поиска по входящим сообщениям; eBay – «всего» 2 петабайта).

2. Огромное количество пользователей (исчисляются миллионами, доступ к системам осуществляется одновременно и постоянно).

3. Сложные данные: как правило, это приложения не простой обработки табличных данных, которые можно найти во многих коммерческих и бизнес-приложениях.

Технологии реляционных баз данных, которые доминировали в ИТ-индустрии с 1980 года, начали показывать свои слабые места при переходе к web-масштабам именно в этих трёх аспектах, поэтому все большее число людей начали искать им альтернативу. Так стали появляться NoSQL базы данных. Хотя разнообразие моделей растёт, все они имеют общие черты:

1. Обработывают огромные объёмы данных, разделяя их между серверами.
2. Поддерживают огромное количество пользователей.
3. Используют упрощённую, более гибкую, не ограниченную схемой структуру базы данных^[3].

Если сравнивать NoSQL с реляционными базами данных, то можно сделать следующие выводы по основным характеристикам:

• **Отказоустойчивость.** NoSQL поддерживают журналирование запросов и операции восстановления базы данных. Восстановление базы данных не окажется сложной операцией, даже если web-приложение большое и имеет «огромную» базу данных, на её восстановление уйдёт около минуты.

• **Популярность.** Под реляционные базы создано множество замечательных продуктов, которые необходимо поддерживать и развивать. В эти продукты уже вложены большие деньги и заказчики готовы ещё вкладывать деньги в их развитие. Напротив, с использованием NoSQL разработано сравнительно мало серьёзных коммерческих продуктов, т.е. существует мало мощных NoSQL-систем.

• **Язык запросов и его стандартизация.** Ещё в 1986 году был принят первый стандарт SQL-86, который определил всю судьбу реляционных баз данных. После принятия стандарта все разработчики реляционных баз данных обязаны были следовать ему. Для NoSQL язык запросов только формируется, хотя уже известны примеры удачной реализации – язык UnQL.

• **Иерархии, деревья.** В реляционных базах плоские таблицы, т.е. в них нельзя хранить сложные иерархии и деревья. NoSQL позволяет создавать такие иерархии.

• **Математический аппарат.** Для реляционных баз данных Эдгар Кодд заложил фундамент математического аппарата реляционной алгебры. Этот математический аппарат объясняет, как должны выполняться основные операции над отношениями в базе данных, доказывает их оптимальность. С другой стороны для NoSQL нет такого аппарата, но работы в этой области ведутся.

• **Дополнительные свойства, отсутствие структуры.** Одна из самых насущных проблем в разработке и сопровождении web-приложений - добавление дополнительных свойств для каких-либо элементов. Реляционные базы данных основаны на строгой описываемой структуре таблиц. Для добавления нового свойства в реляционной базе данных необходимо добавлять новое свойство в саму таблицу или создать таблицу «дополнительные свойства». NoSQL позволяет легко добавлять новые свойства к любому элементу благодаря отсутствию чётких структур и иерархий.

• **Производительность.** Исходя из вышесказанного, можно сделать вывод, что NoSQL за счёт другого подхода к хранению данных, позволят оптимизировать запросы, снять лишнюю нагрузку с сервера. Необходимо заметить также, что скорость вставки и обновления в NoSQL часто намного больше, чем в реляционных базах данных^[2].

Исходя из вышеперечисленного, следует отметить рост количества СУБД, использующих NoSQL. Их число постоянно растёт, даже компания-гигант Oracle совсем недавно выпустила свой новый продукт «Oracle NoSQL Database», чтобы занять своё место на рынке NoSQL. Из-за возросшей потребности в NoSQL был создан язык UnQL, аналогичный SQL, но ориентированный на NoSQL.

UnQL (Unstructured Data Query Language) - язык запросов, ориентированный на работу с неструктурированными данными. Основной целью разработки UnQL было создание для NoSQL-систем привычного и стандартизованного языка для определения и

манипулирования данными. Язык UnQL выступает в роли неструктурированного эквивалента SQL и призван заполнить образовавшуюся нишу, связанную с отсутствием единой формы задания модели данных и языка запросов.

В заключение можно сделать вывод, что реляционные базы данных будут использоваться ещё достаточно долго. NoSQL не сможет заменить реляционные базы данных в полном объёме, т.к. в некоторых задачах все же удобней и правильней хранить данные в таблицах.

Но со временем NoSQL-системы отвоюют у реляционных баз данных часть рынка коммерческих систем, однако добиться сразу такого тотального преимущества, которое сейчас имеют реляционные БД, они не смогут.

Тем не менее, свою нишу NoSQL уже занял и постепенно набирает популярность благодаря ключевым преимуществам:

1. Возможность работы с большими массивами данных.
2. Поддержка одновременной работы большого количества пользователей.
3. Отсутствие транзакций.
4. Отказоустойчивость.
5. Лёгкая поддержка и обновление базы данных, актуализация за счёт отсутствия чётких структур.
6. Производительность (соизмеримая или даже лучшая, чем у реляционных баз данных).
7. Практически полное отсутствие возможностей для SQL- инъекций.

Все эти преимущества сделали NoSQL просто незаменимым при разработке web-приложений либо приложений, которые должны работать с огромными массивами данных.

Примечания:

1. Seltzer M. Beyond Relational Databases: There is more to data access than SQL [Электронный ресурс] // Associations for Computing Machinery URL: <http://queue.acm.org/detail.cfm?id=1059807> (Дата обращения: 02.04.12).
2. Зайцев Д. NoSQL, MongoDB и другие [Электронный ресурс] // Google: Blogger URL: <http://nosql-dev.blogspot.com/2011/07/blog-post.html> (Дата обращения: 04.04.12).
3. Дмитриевич О. Универсальное NoSQL - введение в теорию [Электронный ресурс] // Блогератор.ру URL: <http://blogerator.ru/page/nosql-vvedenie-v-teoriju-bd> (Дата обращения: 04.04.12).
4. Katz D. UnQL Specification [Электронный ресурс] // The UnQL Specification home URL: <http://www.unqlspec.org/> (Дата обращения: 05.04.12).

УДК 004.652

Перспективы развития современных технологий баз данных

¹ Иван Александрович Долуев

² Михаил Геннадьевич Петров

¹ Сочинский государственный университет, Россия
354000, г. Сочи, ул. Советская, 26 а
Студент пятого курса

² Сочинский государственный университет, Россия
354000, г. Сочи, ул. Советская, 26 а
Кандидат технических наук, доцент
E-mail: kchalovskyi@mail.ru

Аннотация. В статье рассматриваются сравнительные характеристики реляционного и нереляционного подходов к управлению базами данных.

Ключевые слова: нереляционные базы данных; базы данных; NoSQL; UnQL.