

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ NOSQL ДЛЯ РЕАЛИЗАЦИИ БАЗ ДАННЫХ

Маркин Е.И., Рябова К.М., Артюшина Е.А.

Пензенский государственный технологический университет, Пенза,

e-mail: Alexey314@yandex.ru

В настоящий момент большое количество существующих программных приложений в процессе своего функционирования используют базы данных для организованного хранения информации, а также её чтения, записи или обработки. Операционные системы NoSQL являются относительно новыми, поэтому не все еще разработчики ясно понимают их возможности и отличия от традиционных реляционных систем управления базами данных. Приведенная ниже сводка характеристик четко отличает эти два класса систем и освещает направления для будущей работы. По мере роста потребностей в обработке Больших Данных появляются новые модели управления данными, которые позволяют ежесекундно выполнять миллиарды запросов. Также изменяются и традиционные реляционные модели, чтобы соответствовать современному состоянию рынка программного обеспечения. В данной статье рассматриваются современный инструментарий СУБД и критерии для выбора оптимального решения, наилучшим образом удовлетворяющего конкретным нуждам заказчика.

Ключевые слова: архитектуры для обработки транзакций, системы поддержки принятия решений, инструменты Больших Данных

THE MODERN NOSQL TECHNOLOGY TO IMPLEMENT THE DATABASE

Markin E.I., Ryabova K.M., Artyushina E.A.

Penza State Technological University, Penza,

e-mail: Alexey314@yandex.ru

At the moment, a large number of existing software applications in the course of their operation use databases for organized storage of information, as well as its reading, recording or processing. Operational NoSQL systems are relatively new in the data-management ecosystems, and there is much confusion about their capabilities and how they differ from traditional relational database systems. This summary of characteristics clearly distinguishes the two system classes and provides a glimpse into directions for future work. Big data requirement are motivating new database management models that can process billions of data requests per second, and established relational models are changing to keep pace. In this article are considered practical tools for navigating this shifting product landscape and finding candidate systems that best fit a data managers applications needs.

Keywords: architecture for transaction processing, decision support systems, Big Data Tools, NoSQL

В настоящий момент большое количество существующих программных приложений в процессе своего функционирования используют базы данных (БД) для организованного хранения информации, а также её чтения, записи или обработки. Взаимодействие с данными осуществляется с помощью систем управления базами данных (СУБД), которые представляют собой специализированное программное обеспечение или же наборы библиотек для поддержки разнообразных форматов БД.

На протяжении последних четырех десятилетий реляционные СУБД [2] оставались основным инструментом управления данными на предприятиях, которым необходимо ежедневно обрабатывать миллионы запросов. Реляционные СУБД не только позволяют хранить информацию в удобном для структурирования табличном виде, но и поддерживают различные средства контроля целостности данных,

аутентификацию, ролевой контроль доступа, резервное копирование и восстановление данных, а также вполне соответствуют требованиям ACID (atomicity, consistency, isolation, durability – атомарность, согласованность, изоляция и долговечность) [1]. Реляционная схема включает в себя описание содержания и структуры таблиц, ограничений целостности на уровне таблиц, а также межтабличные ссылки (рис. 1).

Реляционную модель традиционно используют либо для оперативной обработки транзакций (OLTP – Online Transaction Processing), либо в системах поддержки принятия решений (DSS – Decision Support System). Соответственно, для OLTP-систем рабочими являются задачи по исполнению кратких запросов на чтение или обновление небольшого числа кортежей. Задачи DSS заключаются в исполнении более сложных запросов, требующих просмотра, объединения или агрегации данных из нескольких таблиц [3].

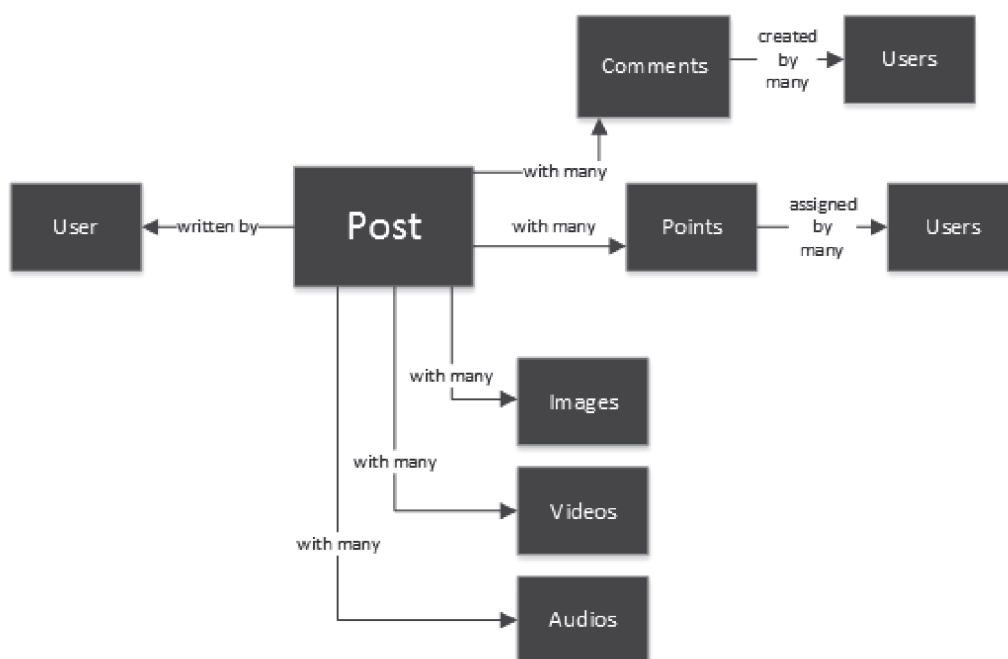


Рис. 1. Пример реляционной схемы БД

Однако современным приложениям, ориентированным на работу с Big Data – Большими Данными, нужна особая функциональность, не свойственная реляционным системам. В первую очередь – это возможность горизонтального масштабирования БД, поддержка изменяемых схем и специфических типов данных, а также экстремально высокий уровень быстродействия при обработке гигантских объемов информации, поступающих из сети в режиме реального времени. Попытка устранения вышеуказанных недостатков и привела к появлению на рынке СУБД программных продуктов NoSQL.

Термин «NoSQL» появился в июне 2009 года и был расшифрован как «Not Only SQL» – «не только SQL». Термин служит для обозначения нереляционных БД, в которых внутренние связи между разобщенными гетерогенными источниками информации устанавливаются путем создания метаданных и их сохранения в системе NoSQL. Основным методом поиска по базе в таком случае становится поиск в метаданных по ключевым словам.

Технология NoSQL устраняет основные ограничения, свойственные реляционной модели, например, трудоемкость горизонтального масштабирования системы и невысокий уровень производительности в кластере, а также существенно упроща-

ет способы хранения и доступа к данным. При попытке классификации СУБД NoSQL по аналогии с реляционными базами также можно разделить на системы транзакционного и аналитического типа. К первым относят такие СУБД как Cassandra, Aerospike, Couchbase, Oracle NoSQL, MongoDB, Redis, Riak, DinamoDB и др. Ко второму типу – СУБД MapReduce, Spark, Hadoop. Оба типа востребованы и в достаточной мере представлены на независимом рынке программного обеспечения.

Ранние системы NoSQL не вполне соответствовали стандартам ACID, используя вместо них менее строгие требования BASE (basic availability, soft state, eventual consistency – базовая доступность, негарантированное сохранение состояния, возможная согласованность). Однако позднейшие разработки имеют гораздо меньшее расхождение с ACID, а в некоторых случаях практически полностью выполняют основные требования, сформулированные к системам транзакционного типа.

В процессе проектирования БД NoSQL используется подход, при котором организация данных специфических типов возможна за очень небольшой интервал времени. Одновременно предлагаются различные типы доступа к разнородной информации (рис. 2).

```

{
  "id": "ew12-res2-234e-544f",
  "title": "post title",
  "date": "2016-01-01",
  "body": "this is an awesome post stored on NoSQL",
  "createdBy": "User",
  "images": ["http://myfirstimage.png", "http://mysecondimage.png"],
  "videos": [
    {"url": "http://myfirstvideo.mp4", "title": "The first video"},
    {"url": "http://mysecondvideo.mp4", "title": "The second video"}
  ],
  "audios": [
    {"url": "http://myfirstaudio.mp3", "title": "The first audio"},
    {"url": "http://mysecondaudio.mp3", "title": "The second audio"}
  ]
}

```

Рис. 2. Пример структуры данных NoSQL в формате JSON

Для построения или динамического переопределения гибких схем могут применяться разнообразные модели представления данных в зависимости от назначения и класса системы NoSQL. Например, для транзакционных OLTP-систем используются следующие модели:

- модель «ключ-значение» – СУБД Redis, MemcacheDB, Riak, DinamoDB и др.;
- поколоночные (column-oriented) СУБД Cassandra, HBase и т.д.;
- документная модель – СУБД MongoDB, Couchbase, MarkLogic и т.д.;
- графовая модель – СУБД OrientDB, Neo4J, Titan и др.;
- нативные XML-базы – СУБД BaseX, TeraText Database System, Sedna, eXistdb.

Модель «ключ-значение» хранит данные в виде соответствующих пар: значение любого типа рассматривается как двоичные данные и хэш-функция, преобразующая ключ в индекс [1]. Программное приложение при поиске данных в базе, подключаясь к серверу, задает ключ и его значение. Оптимально использование таких систем для управления сеансами web-приложений, при обмене сообщениями, персональном обслуживании пользователя, в online-играх, серверами рекламных объявлений и пр.

Поколоночная БД NoSQL является улучшенной версией хранилища «ключ-значение». Она представляет собой двумерный массив, где каждый ключ (запись) содержит одну или несколько привязанных пар «ключ-значение». Модель позволяет хранить и обрабатывать огромные объемы разреженной неструктурированной информации разного типа. Модель применяется,

когда недостаточно простых пар «ключ-значение».

Базы на основе документной модели поддерживают иерархические структуры с большими уровнями вложенности, чем «ключ-значение» и поколоночные БД. Это дает возможность описывать сколь угодно сложную структуру документа, например – электронную медицинскую карту. Модель имеет серьезный недостаток: в случае запроса конкретных полей из данного документа в ответ приходит полностью весь документ, что негативно отражается на производительности приложений для работы с документно-ориентированными БД.

БД на основе графов используют древовидные структуры с узлами и связями, объединяющими их. Как и в математике, определенные операции с данными намного удобнее выполнять благодаря связям между ними и их группировке. Такие БД используются при обработке геопространственной информации или в приложениях социальных сетей. Например, хранить связи между «друзьями» в социальной сети проще в случае использования БД на графах.

Наиболее развитыми являются нативные XML-базы, которые обеспечивают хранение и поддержку максимально широкого набора типов данных по сравнению с любыми другими NoSQL-базами. Термин «нативные» в названии подкласса предназначен для отличия этих СУБД от реляционных систем, в которых также реализована работа с XML-данными, хранимыми в виде больших символьных объектов. В нативных XML-базах документы содержатся в естественных иерархических структурах. Область примене-

ния таких систем – это медицина, страхование, а также сайты с контентом, который генерируется на основе данных.

Таким образом, в отличие от реляционных БД технология NoSQL обеспечивает эффективное хранение и обработку значительных объемов неструктурированных данных, требующих экстремально высоких скоростей для осуществления операций чтения и записи.

Список литературы

1. Гудивада В., Рао Д., Рагхаван В. Ренессанс СУБД: Проблема выбора // Открытые системы. СУБД. – 2016. – №03. – С. 12–17. – <https://www.osp.ru/os/2016/03/13050249/> (дата обращения: 05.08.2017).
2. Дейт К.Д. Введение в системы баз данных. 6-е изд. – К.; М., СПб.: «Вильямс», 2015. – 848 с.
3. Пател Дж. Операционные системы NoSQL: сегодня и завтра // Открытые системы. СУБД. – 2016. – №03. – С.8–11. URL: <https://www.osp.ru/os/2016/03/13050248/> (дата обращения: 05.08.2017).