

Современные NoSQL хранилища

Золотарев Александр Юрьевич – студент Донского государственного технического университета.

Калайда Алексей Васильевич – старший преподаватель Донского государственного технического университета.

Новикова Анна Ивановна – аспирант Донского государственного технического университета.

Аннотация: В данной статье проведен анализ новых решений хранения информации. Рассмотрены преимущества и недостатки реляционной (SQL) модели и не реляционной (NoSQL) модели. В данной работе авторы рассмотрели три наиболее известных и наиболее широко используемых документо-ориентированных СУБД: Couchbase, CouchDB и MongoDB. На основании проведенного исследования было выявлено, что на сегодняшний день лидером среди документо-ориентированных СУБД является MongoDB.

Ключевые слова: Базы хранилищ, SQL, NoSQL, СУБД, Couchbase, CouchDB, MongoDB.

Новые решения, хранения и доступа к информации, которым сейчас уделяется большое внимание мирового IT сообщества связано напрямую с систематичным ростом объемов накопленных данных в сети Интернет. В связи с этим процессом даже был предложен термин Big Data. Это понятие включает в себя некоторую стратегию эффективного взаимодействия с постоянно растущими массивами накопленной информации, которая помимо прочего оказалась еще и менее структурированной, что затрудняет её хранение и обработку. На фоне всех этих процессов появилась необходимость в иной не реляционной структуре хранения данных, которая была бы нацелена больше на масштабируемость и на скорость доступа к хранящимся данным. Для этого необходимо простое решение, которое бы не уступало существующей реляционной модели в целом ряде задач. Так появились NoSQL базы данных или не реляционные, которые, например, позволяют хранить пары ключ-значение и применять кеширование, что позволяет значительно увеличить скорость взаимодействия с базой данных [1, с. 23]. Помимо

хранилищ ключ-значение были так же разработаны документо-ориентированные базы данных, модели распределенного хранилища и хранилища на основе графов. Данная работа посвящена документо-ориентированным базам данных, которые в настоящее время динамично развиваются и совершенствуются. В связи с динамичным развитием и была определена цель данной работы, которая заключается в структурировании имеющейся информации по документо-ориентированным системам управления базами данных.

Когда становится вопрос выбора базы данных, самым важным решением является выбор модели данных. В подавляющем большинстве случаев выбор стоит между реляционной (SQL) моделью и не реляционной (NoSQL) [2, с. 175]. Хотя оба варианта являются применимыми ко многим задачам. Существуют определенные ключевые различия между этими двумя моделями данных, которые разработчики должны иметь в виду при принятии решения.

Системы управления базами данных SQL используют язык запросов (SQL) для модификации и управления данными. С одной стороны, это чрезвычайно удобно: SQL является одним из самых универсальных и широко используемых языков запросов, что делает его отличным решением и особенно подходит для сложных запросов. С другой стороны, применение SQL в базах данных требует, чтобы схема структуры данных была предопределена, прежде чем вы начнете работать с ними. Кроме того, все ваши данные должны соответствовать заданной структуре [2, с. 176].

NoSQL базы данных имеют динамическую схему для неструктурированных данных, а данные хранятся по-разному. Эта гибкость означает, что:

- Есть возможность создавать документы без предварительного определения их структуры;
- Каждый документ может иметь свою уникальную структуру;
- Синтаксис запросов проще чем у SQL баз данных и как следствие меньшее количество ошибок;
- Существует возможность добавления полей по мере их поступления;
- Для использования не требуются глубокие знания языков структурированных запросов.

Выполним сравнение SQL и NoSQL баз данных на примере двух ключевых игроков современного рынка СУБД: MySQL и MongoDB

В документно-ориентированной базе данные не хранятся в таблицах с полями, где для каждого поля четко определен тип хранимых данных. Вместо этого данные хранятся в произвольных "документах" с любым количеством полей и любым количеством вложенных структур. Такие документы обычно представляются в форматах JSON (JavaScript Object Notation), XML (eXtensible Markup Language) или BSON (Binary JavaScript Object Notation) и обновляются как правило посредством API (Application Programming Interface) [4, с. 178]. Большинство всех современных языков программирования поддерживает JSON, поэтому работа с документно-ориентированной базой больше похожа на работу с этим форматом данных.

Такая схема данных, имеет и свои ограничения. Разработчик должен сделать больше работы, чтобы обеспечить целостность хранимой информации, поскольку это не всегда гарантируется самой СУБД. А в реляционных базах данных модель данных заранее определена и является строго типизированной, содержит ограничения и отношения для обеспечения целостности хранимой информации. Но удобство, скорость, масштабируемость и универсальность документно-ориентированных баз данных трудно превзойти. Когда возникают задачи, которые требуют неоднородную структуру данных свободной формы, реляционная модель уже не применима.

В большинстве ситуаций базы данных SQL масштабируются по вертикали, это означает, что вы можете увеличить нагрузку на один сервер, модернизировав CPU, RAM или SSD [5]. С другой стороны, базы данных NoSQL масштабируются по горизонтали. Это означает, что вы обрабатываете больше трафика путем добавления большего количества серверов в свою базу данных. Это похоже на добавление большего количества этажей в одно и то же здание по сравнению со строительством большего количества сооружений в окрестности. Последнее в конечном итоге как правило позволяет реализовать куда более крупную и мощную систему, делая базы данных NoSQL более предпочтительными для крупных или постоянно изменяющихся наборов данных.

Теперь, когда мы установили ключевые структурные различия между базами данных SQL и NoSQL, давайте рассмотрим ключевые функциональные различия между ними, в частности, на примерах MySQL и MongoDB.

MySQL – реляционная система управления базами данных SQL.

Некоторые преимущества MySQL:

- Зрелость: MySQL представляет собой проверенную временем базу данных, а это означает, что существует огромное сообщество, было проведено обширное тестирование и как следствие данная СУБД имеет высокую стабильность работы.
- Совместимость: MySQL доступен для всех основных платформ, включая Linux, Windows, Mac, BSD и Solaris. Он также имеет драйвера для таких языков, как Node.js, Ruby, C#, C++, Java, Perl, Python и PHP.
- Экономическая выгода: База данных MySQL имеет открытый исходный код и является свободно распространяемым программным пакетом.
- Воспроизводимая: база данных MySQL может быть реплицирована на нескольких узлах, это означает, что рабочая нагрузка может быть уменьшена, а масштабируемость и доступность приложения могут быть увеличены.

MongoDB: – документо-ориентированная система управления базами данных.

Некоторые преимущества MongoDB:

- Динамическая схема: Как уже упоминалось, это дает нам возможность изменять схемы данных без изменения существующих данных.
- Масштабируемость: MongoDB масштабируется по горизонтали, что позволяет свести к минимуму человеческие и машинные ресурсы, затрачиваемые на масштабируемость проекта [3, с. 75].
- Управляемость: база данных не требует администратора базы данных так как данная СУБД достаточно проста в использовании. Администрированием может заниматься как разработчик, так и администратор.
- Скорость: высокая производительность для простых запросов, по сравнению с реляционными базами данных.

У MySQL и MongoDB есть свои сильные и слабые стороны. Если ваши данные требуют многочисленных транзакций или вам необходимо поддерживать устаревшее приложение, реляционная база данных станет хорошим решением для вашего проекта. Однако, если вам нужно более гибкое решение, которое может работать с большими объемами неструктурированных данных, MongoDB станет идеальным решением.

Мы рассмотрели ключевые различия между реляционными базами данных и не реляционными. Теперь рассмотрим и сравним несколько ключевых игроков не реляционных баз данных предназначенных для хранения иерархических структур данных (документов). В данной работе рассмотрены три наиболее известных и наиболее широко используемых документо-ориентированных СУБД: Couchbase, CouchDB и MongoDB.

Таблица 1. Сравнение трех основных документо-ориентированных СУБД.

Параметр

Couchbase

CouchDB

MongoDB

Тип

Документо-ориентированная

Документо-ориентированная

Документо-ориентированная

Разработчик

Couchbase, Inc.

Apache Software Foundation

MongoDB, Inc.

Устойчивость к разделению

+

+

+

Атомарность

+

+

+

Согласованность

+

+

+

Изолированность

+

+

-

Долговечность

+

+

+

Параллельный доступ

MYCC

MYCC

Master-slave

Кластеризация

Шардинг, репликация

Шардинг, консистентное хеширование

Шардинг, консистентное хеширование

MapReduce

Да

Да

Да

Запросы

Internal API (MapReduce)

Internal API (MapReduce)

Internal API

Поддержка сложных запросов

Поддерживаемые языки

.Net, C, Clojure, ColdFusion, Erlang, Java, JavaScript, Go, Perl, PHP, Python, Ruby, Scala, Tcl

C, C#, ColdFusion, Erlang, Haskell, Java, JavaScript, Lisp, Lua, Objective-C, OCaml, Perl, PHP, PL/SQL

C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp,

Операционная система сервера

Linux, OS X, Windows

Android, BSD, Linux, OS X, Solaris, Windows

Linux, OS X, Solaris, Windows

Вторичные индексы

Да

Да

Да

Клиентские приложения

LinkedIn, Nielsen, PayPal, Ryanair

Аkamai, Hothead Games, Vivint Solar

Adobe, BuzzFeed, CERN, NBCUniversal, The New York Times, The Washington Post, UPS, Ebay, Fa

Лицензия

Open Source

Open Source

Open Source

Couchbase была разработана с учетом опыта, накопленного в CouchDB и Membase, но не связана ни с одним из этих двух проектов. Это база данных иерархической структуры данных (документов). И является распределенным хранилищем типа ключ-значение, объединённых в одно целое, с расширенными функциями, такими как автоматическая отработка отказа и межцентровая репликация, предназначена для коммерческого использования [5].

Одной из особенностей, которая отличает Couchbase не только от остальных конкурентов NoSQL, но и от предшественника CouchDB, является его SQL-подобный язык запросов N1QL. N1QL не предлагает полный спектр команд, которые можно было бы ожидать от реализации стандарта ANSI SQL, но он предоставляет достаточно возможностей, таких, например, как оператор JOIN [5].

Система запросов Couchbase предназначена не только для разработчиков, но и для администраторов баз данных и бизнес-аналитиков, которые обычно имеют дело с

простыми базами данных. Такие функции, как объяснение ключевых слов по-видимому, были введены специально для использования именно этой группой пользователей [1, с. 27].

Комбинация документо-ориентированной базы данных и хранилища ключ-значение дает явное преимущество по сравнению с конкурентами. Couchbase хранит документы, используя их уникальные идентификаторы в качестве ключа. Документам также могут быть назначены значения времени жизни, чтобы функционировать как кэш ключ-значение. Тем не менее, специально разработанная система кэширования ключей и значений, такая как Redis, будет намного эффективней для базового хранилища ключей и значений, но Couchbase более гибка, а Redis и Couchbase могут быть эффективно объединены для еще большей эффективности [5].

Еще одна версия Couchbase, достойная внимания разработчиков приложений – это Couchbase Lite, встраиваемая версия Couchbase, которая может синхронизироваться с полной версией СУБД. Couchbase Lite является ключевым компонентом в Couchbase Mobile. Стеке разработок компании для мобильных приложений, которым требуется хранилище данных, которое автоматически синхронизируется с серверной частью. Couchbase Mobile доступен для iOS, Android, Java, .Net, MacOS и tvOS [5].

Проект CouchDB был начат в 2005 году бывшим разработчиком IBM. Иногда встречается информация, что CouchDB является основой Couchbase, но CouchDB и Couchbase являются независимыми проектами с разными целями [4, с. 192].

В отличие от Couchbase в CouchDB данные хранятся как иерархическая структура. И хотя Couchbase уже давно использует новые функции, такие как отказоустойчивость и SQL-подобный запрос, такие функции только начинают реализовываться в CouchDB.

CouchDB подчеркивает простоту развертывания и простоту использования. Получение данных из базы данных так же просто, как отправка запросов в формате JSON в конечную точку REST HTTPS с результатами, возвращаемыми в JSON. Большинство современных языков программирования могут выполнять эти операции.

Одной из уникальных возможностей реализованной в CouchDB является технология

согласования данных. Изменения, внесенные одним узлом CouchDB, автоматически согласовываются с другими, подобно системе управления версиями. Любые конфликты между версиями документа сохраняются, как предыдущие редакции этого документа. Это и есть главная отличительная особенность данной СУБД от конкурентов.

Эта согласованная модель полезна для баз данных, которые не всегда или последовательно подключены (например, для периодически подключаемых мобильных приложений), или в случаях, когда вам не нужна последняя и самая большая версия данных в конкретном узле. Но возможная сиюминутная несогласованность является одним из самых больших предостережений при выборе CouchDB. Если вам нужна немедленная согласованность, то CouchDB не отвечает этим требованиям [5].

MongoDB является наиболее широко используемой документо-ориентированной системой управления базами данных и наиболее известной среди сообщества разработчиков. Данная СУБД воплощает в себе большинство ключевых требований, предъявляемых документо-ориентированным базам данных и системам NoSQL в целом: иерархическое хранение, масштабируемая архитектура и дизайн без общего доступа [6].

В MongoDB реализованы индексы, в том числе по вложенным документам и массивам, а также геопространственные индексы. Реализованы составные и уникальные индексы. Также есть атомарные операции, compare-and-swap, запись без подтверждения, курсоры и даже MapReduce.

Размеры коллекции в MongoDB могут быть ограничены каким либо информационным объемом или количеством документов. Если коллекция очень сильно укрупняется, старые документы автоматически удаляются. Это очень полезно, если у вас стоит задача хранить какие-либо временные данные [3, с. 59].

В документах MongoDB реализована возможность хранения двоичных файлов — изображения, музыкальные файлы и так далее. Однако для файлов объемом более 1 Мб разработчики рекомендуют использовать GridFS. GridFS — это соглашение о хранении бинарных данных любого объема, поддерживаемое всеми официальными драйверами.

Версия MongoDB с открытым исходным кодом уже включает в себя подавляющее большинство функций, необходимых для создания базового развертывания. Коммерческие лицензии добавляют ключевые функции для коммерческой разработки, включая резервное копирование, расширения автоматизации, мониторинг, средства исследования данных.

В статье мы описали основные отличия SQL баз данных от NoSQL на примере двух лидеров рынка MySQL и MongoDB. Выполнено сравнение трех главных конкурентов на рынке документо-ориентированных СУБД. В результате проведенного исследования было установлено, что на сегодняшний день лидером среди документо-ориентированных СУБД является MongoDB. Которая позволяет оптимальным образом хранить и обрабатывать большие объемы слабоструктурированных данных.

Список литературы

1. Ф. М. Садаладж П.Дж., NoSQL: Новая методология разработки нереляционных баз данных, Вильямс, 2013.
2. Д. К. Дж., Введение в системы баз данных, Вильямс, 2005.
3. Б. К., MongoDB в действии, ДМК Пресс, 2012.
4. G. Harrison, «Next Generation Databases: NoSQL, NewSQL, and Big Data,» 2015.
5. Serdar Yegulalp, «NoSQL standouts: The best document databases,» 24 Январь 2018. [В Интернете]. Available: <https://www.infoworld.com/article/3201884/nosql/nosql-standouts-the-best-document-databases.html>. [Дата обращения: 1 декабрь 2018].
6. G. L. Noel Yuhanna, «Big Data NoSQL,» 2016.

{social}