



ДинамоDB против MongoDB: выберите один и скажите “нет” SQL

Описание

Базы данных являются ключевым компонентом большинства приложений и веб-сайтов. Они хранят важные данные, такие как контент и данные о пользователях, а также могут фиксировать события, происходящие в приложении. Существует множество различных технологий баз данных, и выбранная вами технология может повлиять на все – от архитектуры приложения до процессов разработки. Здесь мы подробно рассматриваем ДинамоDB и MongoDB, два популярных предложения, которые не являются традиционными реляционными базами данных. Вы сами можете решить, подходит ли такой подход “NoSQL” для вашего приложения и какая из этих двух платформ баз данных лучше всего справится с поставленной задачей.

Понимание технологии: ДинамоDB против MongoDB

За последнее десятилетие популярность баз данных NoSQL возросла в связи с потребностью в более гибкой и масштабируемой технологии. Рост социальных сетей и пользовательского контента отчасти стимулировал этот спрос. В течение многих лет большинство приложений использовали традиционные базы данных SQL, в которых каждая строка данных описывает отдельную сущность и ее связи с другими сущностями. Отвечая на запрос, реляционная база данных объединяет данные из разных строк и возвращает их клиентскому приложению. Такая архитектура оптимизирована для сокращения объемов хранения и исключения дублирования повторяющихся данных, поскольку каждый элемент набора результатов обычно хранится в базе данных только один раз.

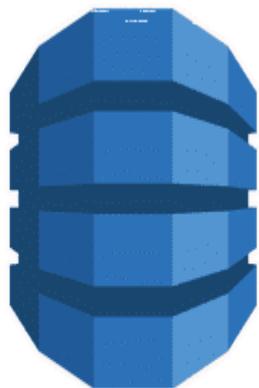
В отличие от этого, базы данных NoSQL обычно хранят базовую сущность и связанные с ней данные в одном документе. Это означает, что некоторые данные записываются несколько раз, что увеличивает требования к хранению. Но по мере снижения стоимости хранения данных масштабируемость решений NoSQL стала превалировать для многих случаев использования. Их базовая архитектура делает базы данных NoSQL исключительно подходящими для работы в кластерах с данными, распределенными по нескольким узлам.

Что такое DynamoDB?

DynamoDB – это бессерверная база данных NoSQL, разработанная компанией Amazon для поддержки своего знаменитого интернет-магазина и ставшая доступной для разработчиков в 2012 году на своей платформе облачного хостинга Amazon Web Services (AWS). Как и в случае с магазином Amazon, DynamoDB отвечает на запросы без вычислительных затрат, необходимых для выполнения иногда сложных соединений в реляционной базе данных. Оптимизированная для масштабирования, DynamoDB представляет собой хранилище элементов “ключ-значение”, где все данные, относящиеся к определенному ключу, хранятся вместе с ним.

Никаких объединений не требуется! Устранив это требование, инженеры оптимизировали DynamoDB для быстрого возврата данных. Amazon делает DynamoDB доступной через AWS как полностью управляемую услугу. Именно поэтому она стала популярной среди разработчиков, которые могут использовать DynamoDB с минимальными накладными расходами на базу данных. Кроме того, сервис легко масштабируется по мере увеличения числа пользователей. А если

компания использует другие инструменты AWS, DynamoDB может легко подключаться к ним.



DynamoDB



Как работает DynamoDB?

DynamoDB – это NoSQL-хранилище ключевых элементов, и поскольку это управляемый сервис, его базовая архитектура и инфраструктура абстрагированы от разработчика. Под капотом данные разбиваются на разделы с помощью первичного ключа, причем каждый раздел содержит три узла, на которых хранятся копии данных. Такое разделение позволяет легко масштабировать систему, поскольку данные распределены по нескольким серверам. Кроме того, репликация данных обеспечивает избыточность в случае отказа конкретного узла.

Один из трех узлов назначается ведущим узлом, на который сначала производятся все записи, а затем они реплицируются на два других узла. Запись сначала на ведущий узел позволяет DynamoDB поддерживать согласованность записей. Для оптимальной согласованности чтения, все чтения DynamoDB также должны использовать ведущий узел.

Такой подход в приложениях с высокой нагрузкой на чтение сопряжен с расходами на производительность, поэтому DynamoDB предоставляет пользователям возможность выбрать согласованность “в конечном итоге”. Это позволяет распределить чтение по всем узлам. Однако этот вариант может привести к снижению точности, если данные не успели распространиться по всем узлам до

операции чтения.

Преимущества и недостатки DynamoDB

Основные преимущества DynamoDB заключаются в том, что это бессерверная, масштабируемая и быстрая СУБД. Вы можете настроить ее за несколько минут с минимальными затратами на инфраструктуру. AWS занимается безопасностью, обновлениями, управлением серверами и обеспечением высокой доступности. Более того, DynamoDB может масштабироваться бесконечно (ограничена только физическим размером AWS) по модели “оплата за использование”. Наконец, DynamoDB обеспечивает быстрое чтение и запись, а возможности масштабирования гарантируют постоянную высокую производительность по мере роста хранилища данных.

Недостатком является минимальные возможности запросов. Поиск по ключу полезен, но если вам нужны более сложные запросы – например, агрегирование – вам придется экспортировать данные во внешнюю базу данных для анализа. Кроме того, DynamoDB доступна только через AWS. Если вы уже интегрированы в экосистему AWS, это не будет проблемой. Если же ваша компания хочет развернуть систему на месте, ей придется искать другое решение.

Что такое MongoDB?

MongoDB – это хранилище документов NoSQL, первоначально выпущенное в 2009 году в качестве базы данных с открытым исходным кодом и возможностью приобретения коммерческих пакетов поддержки. Она хранит документы JSON в виде двоичных объектов JSON (BSON), которые могут хранить сложные структуры данных, такие как вложенные свойства и массивы. MongoDB работает на платформах Windows, macOS и Linux и поддерживает множество языков программирования. Вы также найдете MongoDB на популярных облачных платформах Amazon, Google и Microsoft.

MongoDB популярна в первую очередь благодаря своей гибкости. Она обладает всеми преимуществами базы данных NoSQL и при этом удовлетворяет более сложным требованиям, таким как транзакции ACID для обеспечения целостности данных, запросы на агрегацию и объединение наборов данных. Кроме того, благодаря встроенной поддержке JavaScript, эта база данных становится все более популярной бэкенд-базой для веб-сайтов, использующих этот популярный язык

программирования.



Как работает MongoDB?

MongoDB хранит объекты BSON, содержащие одну или несколько пар ключ-значение. Значения могут быть простыми типами данных, такими как строки и временные метки, или сложными типами, такими как вложенные объекты и массивы. MongoDB хранит документы в коллекциях, подобно строкам в таблице. Обычно документы в коллекции имеют общую схему, но даже документы с одинаковыми ключами могут иметь разные схемы – ключи с одинаковыми именами не обязательно должны хранить один и тот же тип данных.

MongoDB может принудительно проверять свойства при вставке или обновлении записей, и существует требование, чтобы данные в определенных ключах были определенного типа. MongoDB также предлагает шардинг, который аналогичен разделению в DynamoDB. Шардинг распределяет данные между несколькими серверами на основе ключа шардинга. Распределение данных позволяет распределить операции чтения и записи между несколькими серверами, предотвращая перегрузку одного сервера.

Преимущества и недостатки MongoDB

В дополнение к типичным преимуществам NoSQL, MongoDB поддерживает множество платформ, что очень важно для компаний, которым нужна гибкость в выборе облачного провайдера или возможность размещения базы данных на месте. MongoDB также поддерживает сложные запросы и агрегации благодаря своему

конвейеру агрегации. Это делает ее исключительно гибкой по сравнению с другими вариантами NoSQL, поскольку она поддерживает аналитические запросы, поиск документов и объединение данных. Однако производительность может снижаться при больших масштабах. Кроме того, MongoDB Atlas предоставляет бессерверный вариант для компаний, стремящихся снять с себя бремя управления инфраструктурой.

Одним из недостатков MongoDB является то, что объем данных в ней может быть довольно большим, поскольку документы должны хранить как данные, так и ключи. И хотя ее конвейеры агрегации являются мощными инструментами, использование специального синтаксиса делает их исключительно сложными для написания и создания. Наконец, для тех, кто не использует MongoDB Atlas, существуют инфраструктурные накладные расходы, особенно для производственных систем, использующих репликацию и шардинг для обеспечения производительности системы. Это включает в себя мониторинг, управление перекосом данных и выполнение обновлений, а также управление безопасностью.

Итог: DynamoDB против MongoDB

Хотя DynamoDB и MongoDB являются базами данных NoSQL, они существенно отличаются друг от друга. Ваш сценарий использования будет в значительной степени определять, какую из них использовать. Во-первых, DynamoDB имеет меньше доступных вариантов типов данных на уровне схемы. Например, она не поддерживает даты, что означает, что разработчики должны поддерживать логику в своих приложениях для их анализа и обработки. MongoDB поддерживает гораздо больше типов данных, включая даты.

Во-вторых, DynamoDB позволяет выполнять запросы с помощью поиска по ключевым значениям и сканирования таблиц. Однако производительность сканирования таблиц недостаточна. Для более сложных или специальных запросов лучшим вариантом будет экспорт в другую базу данных. В отличие от этого, MongoDB поддерживает поиск ключей поверх аналитических запросов и объединение данных. Она предлагает более гибкую поддержку индексов, поскольку вы можете добавлять вторичные индексы к любому полю. Хотя DynamoDB поддерживает вторичные индексы, они ограничены концептуальной сложностью и количеством, которое вы можете создать.

С точки зрения безопасности DynamoDB выигрывает от платформы AWS.

Аутентификация в DynamoDB проходит через контроль доступа пользователей AWS. Это делает защиту базы данных DynamoDB довольно простой. В случае с MongoDB вы сами отвечаете за все ограничения безопасности, включая создание учетных записей пользователей и управление доступом к сети. Это означает гораздо более высокие накладные расходы. Однако MongoDB Atlas превращает MongoDB в управляемое решение, устраняя эти проблемы.

Как управляемое решение, DynamoDB имеет встроенные функции резервного копирования и восстановления, включая возможность многорегиональных реплик и простого восстановления из резервных копий. По умолчанию MongoDB не управляет этими процессами, требуя ручной настройки. MongoDB Atlas несколько облегчает этот процесс, но требует дополнительной настройки.

С первого взгляда: DynamoDB против MongoDB

Характеристика	DynamoDB	MongoDB
Типы данных	Большинство стандартных типов данных, за исключением дат	Любой тип данных в спецификации BSON, включая даты
Запрос	Поиск ключей при ограниченном многодокументном запросе	Поиск ключей, запросы, объединения и агрегирование
Безопасность	Встраивание в AWS с минимальными накладными расходами	Самоуправляемые, если не используется MongoDB Atlas
Масштабируемость	Встраивание в AWS с минимальными накладными расходами	Самоуправляемые, если не используется MongoDB Atlas
Резервное копирование и восстановление	Встраивание в AWS с минимальными накладными расходами	Самоуправляемые, если не используется MongoDB Atlas

В целом, MongoDB является более гибким вариантом – ее можно развернуть в любом месте, и она предоставляет более богатый набор инструментов для хранения и запроса данных. Тем не менее, если речь идет об экономичной базе

данных с тесной интеграцией с AWS и сниженными накладными расходами на инфраструктуру и безопасность, то DynamoDB является наиболее вероятным выбором.

Заключение

MongoDB и DynamoDB – это высокопроизводительные предложения NoSQL, чьи относительные преимущества делают их подходящими для совершенно разных целей. Для простого и легкого решения DynamoDB является сильным выбором. Но если вам требуются более сложные схемы, агрегационные запросы и лучшая поддержка индексов, то победителем окажется MongoDB. Помните, что ценой гибкости является увеличение накладных расходов на инфраструктуру. Но если вам нужна гибкость MongoDB и вы хотите снизить накладные расходы, MongoDB Atlas часто представляет собой доступный компромисс.

Дата Создания

07.07.2023