

## РАСПРЕДЕЛЕННАЯ СУБД APACHE CASSANDRA

Яна ЗАБОЛОТНЫЙ

Департамент Программной Инженерии и Автоматики, группа TI-195, Факультет Вычислительной Техники, Информатики и Микроэлектроники, Технический Университет Молдовы, Кишинев, Республика Молдова

Автор корреспонденции: Яна ЗАБОЛОТНЫЙ, e-mail: [zabolotnii.iana@isa.utm.md](mailto:zabolotnii.iana@isa.utm.md)

Научный руководитель: Дориан САРАНЧУК, DISA, FCIM, UTM

**Аннотация:** Данная работа посвящена СУБД Apache Cassandra. Она описывает историю этой популярной базы данных, рассказывает о ее преимуществах, недостатках и принципе работы. Статья даёт ответы на множество вопросов, которыми задаются без исключения все пользователи этой базы данных.

**Ключевые слова:** Apache Cassandra, архитектура, СУБД, база данных

### Введение

Apache Cassandra - это распределенная и децентрализованная СУБД, используемая для управления огромными объемами данных, разбросанных по всему миру. Cassandra обеспечивает высокодоступный сервис без единой точки отказа. Проект был разработан в 2008 году компанией Facebook на языке Java и передан в фонд Apache Software Foundation в 2009 году. На рисунке 1 изображен логотип Apache Cassandra.

Эта СУБД классифицируется как гибридное решение NoSQL, потому что она сочетает в себе модель сохранения данных, основанную на типе столбцов, с концепцией «ключ-значение» (key-value). Apache Cassandra используется многими крупными компаниями, такими как Cloudkick, Reddit, Digg, Huawei, Netflix, Apple, Instagram, GitHub, Twitter, Spotify [1].



Рисунок 1 – Логотип Apache Cassandra.

### 1. Модель данных Apache Cassandra

Модель данных Apache Cassandra (рис.2) состоит из таких элементов:

- Столбец (column) – представляет собой ячейку с данными, состоящую из 3 частей: имени (column name), метки времени (timestamp) и значения (value). Метка времени связана с каждым значением и является 64 - битным числом, которое задает пользователь.
- Семейство столбцов (column family) – набор именованных строк.
- Строка (row) – набор именованных столбцов.
- Пространство ключей (keyspace) – группа, состоящая из семейств столбцов.

Пространство ключей логически группирует семейства столбцов [1].

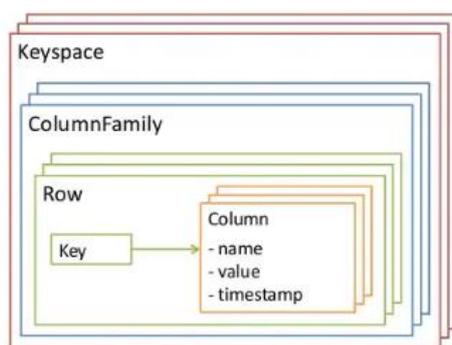


Рисунок 2 – Модель данных Apache Cassandra.

## 2. Архитектура Apache Cassandra

Apache Cassandra состоит из нескольких узлов, по которым размещаются данные. Данная СУБД не поддерживает концепцию master/slave, когда один из серверов является сервером управления для других кластерных компонентов, что отличает ее от других решений, используемых для больших данных.

Для того, чтобы распределить элементы данных между узлами Cassandra использует поочередное хеширование, применяя алгоритм хеширования для вычисления хеш-значений ключей каждого элемента данных. Спектр всех вероятных значений хеш-функции, то есть пространство ключей, распределяется между узлами кластера таким образом, что каждому элементу данных должен присваиваться узел, который отвечает за сохранение и регулирование этого элемента данных [1].

Таким образом, тот факт, что в Cassandra отсутствует центральный узел лишает ее недостатка, который присущ многим системам master / slave, в которых при сбое основного сервера кластер полностью выходит из строя. В кластере Cassandra все узлы эквивалентны друг другу, и, если один из них выйдет из строя, некоторые из оставшихся возьмут на себя его функции. Благодаря такой функции Apache Cassandra прекрасно подходит для распределенных систем с большим доступом, расположенных в разных центрах обработки данных. Но даже при всех достоинствах этой гибкой адаптивной архитектуры она определяет особенности операций чтения и записи, а также возлагает ограничения на использование СУБД в реальных проектах. На рисунке 3 изображена архитектура кластера Apache Cassandra[2].

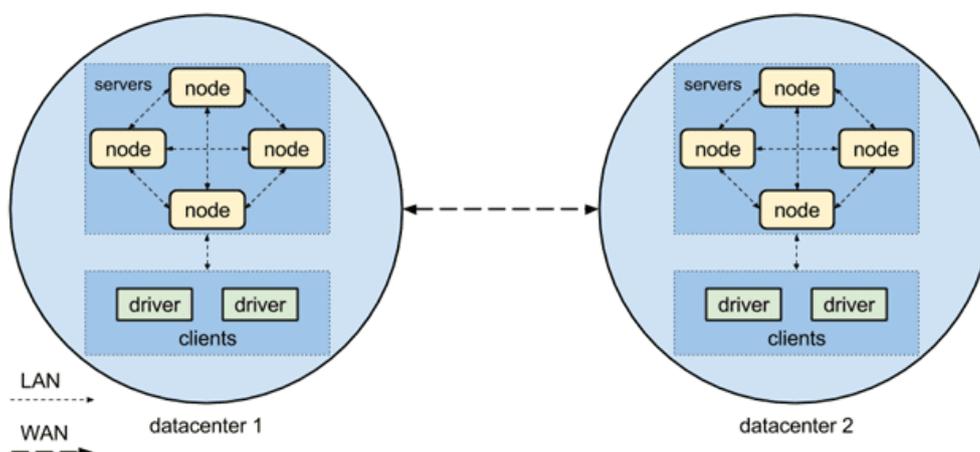


Рисунок 3 – Кластерная архитектура Apache Cassandra.

### 3. Особенности и преимущества Apache Cassandra

Apache Cassandra стала популярной из-за своих технической исключительности. Вот некоторые особенности Apache Cassandra:

- Простое распределение данных - гарантирует гибкость для распределения данных там, где это нужно, путем репликации данных в нескольких центрах обработки данных [2].
- Гибкое хранение данных - поддерживает все возможные форматы данных, в том числе структурированные, полуструктурированные и неструктурированные. Cassandra вносит динамические изменения в структуры данных [2].
- Эластичная масштабируемость - обладает высокой масштабируемостью и позволяет добавлять дополнительное оборудование для поддержки большого количества клиентов и данных [3].
- Быстрая запись - была разработана для работы на недорогом типовом оборудовании. Выполняет невообразимо быструю запись и может сохранять сотни терабайт данных без потерь [3].
- Always on Architecture - Cassandra не имеет единой точки отказа и постоянно доступна для критически важных бизнес-приложений, которые не могут позволить себе выйти из строя.
- Быстрая линейно-масштабируемая производительность - линейно масштабируемая, поэтому она повышает пропускную способность, когда увеличивается численность узлов в кластере. Cassandra поддерживает быстрое время отклика.
- Поддержка транзакций и репликации (рис.4) - поддерживает такие свойства транзакций, как дробность, согласованность, изоляция и долговечность.

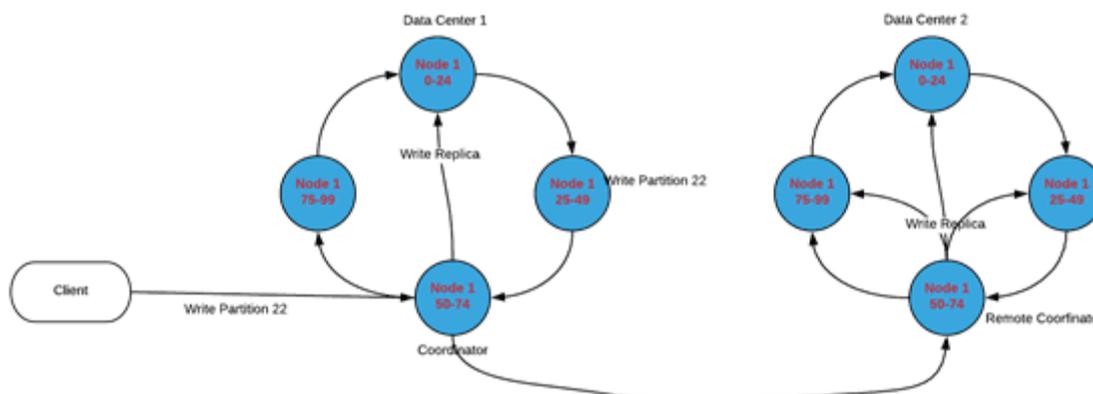


Рисунок 4 – Пример репликации данных в Apache Cassandra

### 4. Недостатки Apache Cassandra

Ключевые недостатки Cassandra, как и ее преимущества, связаны с особенностями архитектуры этой распределенной СУБД. В частности, наиболее значительными недостатками данной системы, основными с точки зрения применения в проектах больших данных, являются такие:

- Особенности внутреннего языка запросов – язык CQL существенно отличается от SQL, несмотря на небольшую схожесть. В основном, операции INSERT и UPDATE - это одно и то же. Кроме того, Cassandra не использует операции объединения. Объединение двух семейств столбцов потребует программного извлечения и объединения данных, что является дорогостоящим и трудозатратным для больших наборов данных[4].
- Высокие дополнительные расходы при работе с большими объемами данных - из-за подхода к проектированию данных, на основе SELECT, требуется очень сильная денормализация, что подразумевает фактическое создание отдельной таблицы для каждого запроса [4].

- Требования к уникальности ключа – каждый ключ обязательно должен быть уникальным. Данные перезаписываются, если один и тот же ключ используется дважды. Хотя это решается путем добавления случайного значения или метки времени к ключу, а также благодаря составному ключу (рис.3).

- Сложность с поисковой операцией, которая не предусмотрена в архитектуре Cassandra, и поисковые системы построены поверх нее путем сортировки данных. Cassandra поддерживает вторичные индексы с довольно узким функционалом, который создает система[4].

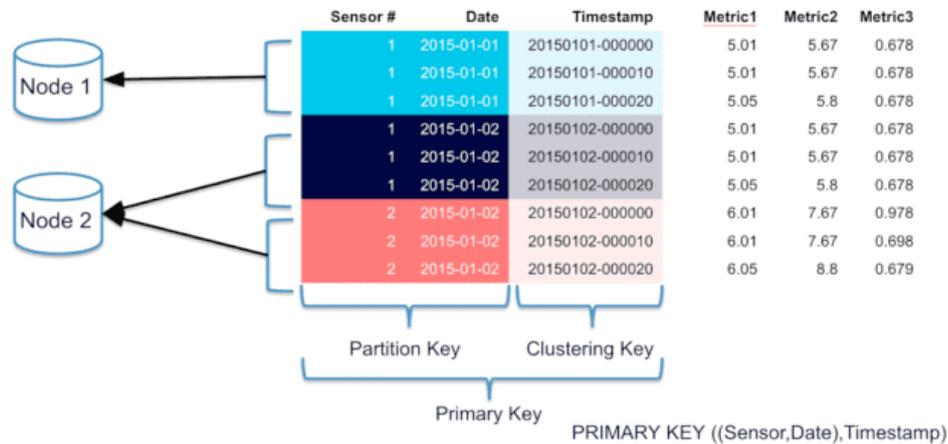


Рисунок 4 – Пример формирования ключей в Apache Cassandra

### Заключение

Apache Cassandra - это СУБД, которой доверяют тысячи компаний благодаря своей масштабируемости и высокой доступности без вреда для производительности. Линейная масштабируемость и проверенная отказоустойчивость на стандартном оборудовании или облачной инфраструктуре делают ее идеальной платформой для критически важных данных. Как и у любой другой СУБД, у Apache Cassandra есть свои преимущества и недостатки, именно поэтому так важно изучить каждую из них.

### Библиография

1. Wikipedia. Apache Cassandra. – [Электронный ресурс]. – [дата обращения 02.03.2023], Доступно: [https://ru.wikipedia.org/wiki/Apache\\_Cassandra](https://ru.wikipedia.org/wiki/Apache_Cassandra)
2. Что такое Apache Cassandra: архитектура, история и принципы. – [Электронный ресурс]. – [дата обращения 02.03.2023], Доступно: <https://www.bigdataschool.ru/wiki/cassandra>
3. Плюсы и минусы Apache Cassandra. – [Электронный ресурс]. – [дата обращения 02.03.2023], Доступно: <https://www.bigdataschool.ru/blog/cassandra-key-features.html>
4. Apache Cassandra. СУБД с распределенной системой хранения данных. – [Электронный ресурс]. – [дата обращения 02.03.2023], Доступно: <https://blog.bissquit.com/dbms/apache-cassandra-subd-s-raspredelelennoj-sistemoj-hr/>