**Доклад по RethinkDB**

**Выполнили работу:**

Комиссаров Денис

Полякова Татьяна

Шкуматова Мария

23КНТ-5

**1. Описание RethinkDB**

RethinkDB — это открытая распределенная NoSQL-база данных, разработанная для хранения JSON-документов. Она была создана в 2009 году компанией RethinkDB Inc. с целью предоставить разработчикам удобный инструмент для работы с реальным временем и изменяющимися данными. Основная идея RethinkDB заключалась в том, чтобы сделать работу с базами данных более интуитивной и удобной для разработчиков, особенно в контексте современных веб-приложений.

RethinkDB поддерживает горизонтальное масштабирование, репликацию и шардирование данных, что делает её подходящей для высоконагруженных приложений. В 2016 году проект был закрыт, но позже его исходный код был открыт, и сейчас он поддерживается сообществом.

**2. Основные особенности RethinkDB**

Отличительной особенностью RethinkDB является первоклассная поддержка потоков изменений в режиме реального времени. Запрос на изменение возвращает курсор, который позволяет блокировать или не блокировать запросы, чтобы отслеживать потенциально бесконечный поток изменений в режиме реального времени.

**Тип NoSQL:**  
RethinkDB относится к документо-ориентированным NoSQL-базам данных, так как данные хранятся в виде JSON-документов, что позволяет гибко структурировать информацию без необходимости строгой схемы, и выполнять запросы на уровне документов, а не строк таблицы (как в SQL).

**Механизмы и принципы работы:**

* **Реальное время:** RethinkDB предоставляет встроенную поддержку реального времени через механизм подписок (changefeeds). Это позволяет приложениям получать уведомления об изменениях в данных без необходимости постоянного опроса базы.
* **Горизонтальное масштабирование:** RethinkDB поддерживает шардирование и репликацию данных, что позволяет распределять нагрузку между несколькими серверами.
* **Язык запросов ReQL:** RethinkDB использует собственный язык запросов ReQL, который интегрируется с JavaScript, Python, Ruby и другими языками программирования. ReQL позволяет строить сложные запросы с использованием цепочек методов.

**Преимущества:**

* Удобство работы с JSON-документами.
* Встроенная поддержка реального времени.
* Гибкость и масштабируемость.
* Простота интеграции с современными языками программирования.

**Недостатки:**

* Меньшая зрелость по сравнению с классическими SQL-базами данных.
* Ограниченная поддержка транзакций (по сравнению с SQL).
* Меньшее количество инструментов и библиотек по сравнению с MongoDB или PostgreSQL.

**Сравнение с SQL:**

* В SQL данные хранятся в таблицах с фиксированной структурой, в то время как RethinkDB позволяет хранить данные в виде JSON-документов, что делает её более гибкой.
* SQL поддерживает сложные транзакции и ACID-гарантии, в то время как RethinkDB фокусируется на масштабируемости и реальном времени.
* RethinkDB поддерживает вторичные индексы, но нет JOIN-ов, как в SQL.

**Сравнение с другими NoSQL-решениями:**

| Критерий | RethinkDB | MongoDB | Cassandra | Redis |
| --- | --- | --- | --- | --- |
| Тип | Документо-ориентированная | Документо-ориентированная | Столбцовая | Ключ-значение |
| Хранение данных | JSON | BSON | Таблицы (колонки) | Ключ-значение |
| Поддержка real-time | Есть | Нет | Нет | Нет |
| Язык запросов | ReQL | MQL | CQL | Нет |
| Транзакции | Нет полной ACID-поддержки | ACID с версии 4.0 | Нет полной ACID-поддержки | Нет |

**3. Основные функциональные возможности**

**1. Хранение JSON-документов:**  
RethinkDB позволяет хранить данные в виде JSON-документов, что делает её удобной для работы с неструктурированными или полуструктурированными данными.

**2. Язык запросов ReQL:**  
ReQL позволяет строить сложные запросы, используя цепочки методов. Например:

*r.table('users').filter({ age: 30 }).run(connection);*

Этот запрос вернет всех пользователей с возрастом 30 лет.

**3. Поддержка реального времени:**  
RethinkDB предоставляет механизм changefeeds, который позволяет приложениям получать уведомления об изменениях в данных. Например:

*r.table('users').changes().run(connection, function(err, cursor) {*

*cursor.each(console.log);*

*});*

Этот код будет выводить все изменения в таблице users в реальном времени.

**4. Горизонтальное масштабирование:**  
RethinkDB поддерживает шардирование и репликацию данных, что позволяет распределять нагрузку между несколькими серверами.

**Сравнение с SQL:**

* В SQL запросы пишутся на языке SQL, который требует знания синтаксиса и структуры таблиц.
* В RethinkDB запросы строятся с использованием ReQL, который интегрируется с языками программирования и позволяет использовать цепочки методов.

**4. Возможность интеграции со сторонними приложениями**

**Драйверы и утилиты:**  
RethinkDB предоставляет официальные драйверы для множества языков программирования, включая JavaScript, Python, Ruby, Java и другие. Это позволяет легко интегрировать RethinkDB с существующими приложениями.

Пример подключения к RethinkDB с использованием драйвера для Node.js:

*const r = require('rethinkdb');*

*const connection = await r.connect({ host: 'localhost', port: 28015 });*

**Сравнение с SQL:**

* В SQL для доступа к данным используются драйверы, такие как JDBC или ODBC, которые поддерживаются большинством баз данных.
* В RethinkDB используются собственные драйверы, которые предоставляют более тесную интеграцию с языками программирования.

**Встроенные решения:**  
RethinkDB может быть интегрирована с современными фреймворками и платформами, такими как Express.js, Django или Flask. Для этого требуется установка соответствующего драйвера и настройка подключения к базе данных.

**Требования для реализации доступа:**

* Установка RethinkDB на сервер.
* Установка драйвера для выбранного языка программирования.
* Настройка подключения к базе данных через конфигурационные файлы или переменные окружения.

**Заключение**

RethinkDB — это мощное NoSQL-решение, которое идеально подходит для приложений, требующих работы с реальным временем и гибкого хранения данных. Её документо-ориентированная модель и встроенная поддержка реального времени делают её отличным выбором для современных веб-приложений. Однако, как и у любого NoSQL-решения, у RethinkDB есть свои ограничения, особенно в сравнении с классическими SQL-базами данных. Тем не менее, для задач, где важны масштабируемость и гибкость, RethinkDB остается одним из лучших вариантов.