МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

РЕФЕРАТ ПО ТЕМЕ
«ТЕХНОЛОГИИ УДАЛЕННОГО ДОСТУПА К СИСТЕМАМ БАЗ ДАННЫХ, ТИРАЖИРОВАНИЕ И СИНХРОНИЗАЦИЯ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ БАЗ ДАННЫХ»

студента 4 курса 431 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Серебрякова Алексея Владимировича

Преподаватель
профессор, д.ф.-м.н. _____ А.С. Гераськин
подпись, дата

Саратов 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Распределенная база данных (РСБД) – это база данных, которая распределена между несколькими узлами или компьютерами, которые находятся в разных физических местах, но связаны между собой сетью. Каждый узел содержит часть данных, которые доступны другим узлам через сеть.

Отличие распределенной базы данных от централизованной заключается в том, что данные в РСБД распределены между несколькими узлами, а не хранятся в единственном центральном месте. Это позволяет распределять нагрузку на несколько узлов, обеспечивать отказоустойчивость, улучшать производительность и уменьшать время ответа при запросе данных. Ещё одним отличием РСБД от централизованной базы данных является наличие дополнительных технологий, таких как механизмы синхронизации и транзакций, которые позволяют гарантировать целостность данных в распределенной среде. РСБД также может иметь более сложную архитектуру, которая включает в себя несколько узлов и промежуточных серверов для решения задач маршрутизации запросов, синхронизации данных и обеспечения безопасности.

Распределенные базы данных часто используются в крупных предприятиях и организациях, где необходимо хранить большое количество данных, обеспечивать доступ к ним из разных мест и обеспечивать высокую доступность и производительность.

1. Виды технологий удаленного доступа к системам баз данных

Технологии удаленного доступа к системам баз данных позволяют пользователям получать доступ и работать с базами данных, которые физически расположены на удаленных серверах. Они обеспечивают возможность удаленного управления базами данных без необходимости находиться непосредственно рядом с ними.

Такие технологии обычно включают следующие компоненты:

- Клиентский интерфейс: это программное обеспечение или приложение, которое позволяет пользователю взаимодействовать с базой данных. Клиентский интерфейс может быть представлен в виде графического интерфейса пользователя (GUI), командной строки или веб-интерфейса.
- Протоколы передачи данных: они определяют способ обмена данными между клиентским интерфейсом и удаленной базой данных. Протоколы могут быть различными, включая TCP/IP, HTTP, HTTPS, JDBC, ODBC и другие.
- Сетевая инфраструктура: это инфраструктура, которая обеспечивает соединение между клиентским интерфейсом и сервером базы данных. Это может быть локальная сеть (LAN), широкая облачная сеть (WAN) или Интернет.
- Безопасность и аутентификация: при удаленном доступе к базам данных важно обеспечить безопасность и аутентификацию пользователей. Технологии удаленного доступа предоставляют механизмы для защиты данных, шифрования соединения и проверки подлинности пользователей.
- Операционная система и сервер базы данных: Удаленный доступ к базе данных требует наличия сервера базы данных, который работает на определенной операционной системе. Технологии удаленного доступа должны быть совместимы с операционной системой и сервером базы данных, чтобы обеспечить эффективную работу.

Существует несколько технологий удаленного доступа к системам баз данных, вот некоторые из них:

ODBC (Open Database Connectivity) – это стандартный интерфейс, который позволяет приложениям взаимодействовать с различными типами баз данных через SQL. ODBC обеспечивает унифицированное API, которое позволяет приложениям работать с базами данных, используя стандартный набор команд.

JDBC (Java Database Connectivity) – это интерфейс, который позволяет Java-приложениям взаимодействовать с базами данных. JDBC предоставляет Java-разработчикам набор классов и методов для подключения к базам данных и выполнения запросов.

OLE DB (Object Linking and Embedding Database) – это технология, которая предоставляет доступ к различным типам баз данных, используя набор объектов COM (Component Object Model). OLE DB позволяет приложениям использовать различные источники данных, такие как SQL Server, Oracle, Access и др.

ADO (ActiveX Data Objects) – это технология, которая предоставляет универсальный набор объектов и методов для доступа к различным источникам данных, включая базы данных и текстовые файлы. ADO позволяет разработчикам использовать язык SQL для выполнения запросов к базам данных.

Web-службы – это технология, которая позволяет удаленным приложениям взаимодействовать с базами данных через Интернет. Web-службы используют стандартные протоколы, такие как SOAP и HTTP, для передачи данных между приложениями и базами данных.

ORM (Object-Relational Mapping) – это технология, которая позволяет использовать объектно-ориентированный подход к работе с базами данных. ORM обеспечивает прозрачность между объектами в приложении и записями в базе данных, что упрощает и ускоряет разработку приложений.

Это не полный список технологий удаленного доступа к базам данных, но они являются наиболее распространенными и широко используемыми в индустрии.

2. Преимущества и недостатки технологий удаленного доступа к системам баз данных

Удаленный доступ к базам данных имеет ряд преимуществ и недостатков. К преимуществам удаленного доступа можно отнести:

- Гибкость и мобильность: удаленный доступ позволяет пользователям получать доступ к базам данных с любого места, где есть подключение к интернету, что обеспечивает мобильность и гибкость работы.
- Снижение затрат: удаленный доступ может значительно снизить затраты на обслуживание и поддержку баз данных, поскольку не требуется дополнительное оборудование и персонал.
- Лучшая доступность и отказоустойчивость: удаленный доступ может обеспечить более высокую доступность и отказоустойчивость, поскольку данные могут быть распределены на несколько серверов и резервных копий.
- Улучшенная безопасность: удаленный доступ может улучшить безопасность данных, поскольку они могут быть хранены на удаленных серверах с более высоким уровнем защиты и шифрования.

Недостатками технологий удаленного доступа являются:

- Ограниченная скорость и производительность: удаленный доступ может быть медленнее и иметь более высокую задержку, поскольку данные должны передаваться через сеть.
- Проблемы с доступностью сети: удаленный доступ может быть невозможен, если сеть недоступна или имеет низкую пропускную способность.
- Риски безопасности: удаленный доступ может представлять риск для безопасности, если он не защищен правильно или если пользователи не соблюдают правила безопасности.
- Ограничения в использовании некоторых функций: некоторые функции могут быть недоступны при удаленном доступе, так как они

могут требовать более высокой скорости и производительности, или просто не могут быть использованы в удаленном режиме.

Таким образом, удаленный доступ к базам данных имеет как свои преимущества, так и недостатки, и эти факторы должны быть учтены при принятии решения о выборе соответствующей технологии доступа к базе данных.

3. Тиражирование в системах с распределенной базой данных

Тиражирование (Replication) – это процесс создания копий данных и распространения их на несколько узлов или серверов в распределенной базе данных. То есть каждый узел содержит полную копию всей базы данных или ее части, которая реплицируется с другими узлами. Это позволяет ускорить доступ к данным и повысить их доступность и отказоустойчивость.

Тиражирование (Replication) решает несколько проблем в распределенных системах баз данных:

- Увеличение доступности: тиражирование позволяет увеличить доступность данных, так как клиенты могут получать доступ к ближайшей копии данных. Если один сервер выходит из строя, клиенты могут продолжать получать доступ к данным с других серверов.
- Увеличение отказоустойчивости: при использовании тиражирования, если один сервер выходит из строя, клиенты могут продолжать работать с другими серверами, где находятся копии данных.
- Увеличение скорости доступа: клиенты могут получать доступ к ближайшей копии данных, что позволяет ускорить время ответа и уменьшить нагрузку на сеть.
- Распределение нагрузки: тиражирование позволяет распределить нагрузку на несколько серверов, что позволяет увеличить производительность системы.

Однако, тиражирование также может вызвать проблемы с согласованностью данных и повлечь за собой увеличение затрат на обслуживание и хранение данных, так как каждый сервер должен содержать копию данных. Поэтому необходимо правильно выбирать стратегию тиражирования, учитывая преимущества и недостатки этой технологии, а также требования к системе баз данных.

Существует несколько технологий, которые применяются в современных распределенных системах баз данных для тиражирования, далее мы рассмотрим их более подробно.

3.1 Технология Master-slave

Технология *мастер-множество (Master-slave)* предполагает, что есть один главный сервер (мастер), который получает все запросы на запись, а остальные серверы (рабы) получают копии данных для чтения. Когда происходит запись, мастер отправляет изменения на рабов, чтобы обновить их копии данных. Данный подход позволяет увеличить доступность и отказоустойчивость, так как в случае отказа мастер-сервера, один из рабов может стать мастером и продолжать обслуживать запросы на запись. Однако, при таком подходе может возникнуть проблема задержек, если мастер не может обработать все запросы на запись. Как правило, при использовании технологии Мастер-множество (Master-slave) используются следующие подходы для обеспечения отказоустойчивости и высокой доступности:

Активный-пассивный режим: при таком подходе один сервер является активным (мастером), который получает все запросы на запись и обрабатывает их. Остальные серверы находятся в пассивном режиме (рабы) и получают копии данных для чтения. В случае отказа активного сервера, один из пассивных серверов становится активным.

Активный-активный режим: при таком подходе несколько серверов являются активными (мастерами) и могут обрабатывать запросы на запись. Однако, каждый мастер работает с определенной частью данных, чтобы избежать конфликтов при записи. В случае отказа одного из мастеров, остальные мастеры могут продолжать работу.

Шардинг: при таком подходе база данных разбивается на отдельные части (шарды), каждый из которых находится на отдельном сервере. Один сервер является мастером для определенного шарда, а остальные серверы - рабами, получающими копии данных для чтения. Этот подход позволяет более

эффективно управлять большими объемами данных и увеличить производительность системы.

Преимущества технологии Master-slave в РСБД:

- Высокая доступность: если мастер-сервер выходит из строя, рабочие сервера могут продолжать обслуживать запросы.
- Легкость в настройке и управлении: мастер-сервер отвечает только за запись, а рабочие сервера только за чтение, что позволяет легко настроить каждый сервер для выполнения определенной роли.
- Высокая производительность для операций чтения: поскольку рабочие сервера могут только читать данные, они могут быть оптимизированы для обработки запросов чтения.
- Масштабируемость: можно добавить дополнительные рабочие сервера для увеличения пропускной способности приложения.

Недостатки технологии Master-slave в РСБД:

- Возможна потеря данных: если мастер-сервер выходит из строя, то любые изменения, которые не были реплицированы на рабочие сервера, могут быть потеряны.
- Необходимость переключения на другой мастер-сервер: если текущий мастер-сервер не может выполнять свои обязанности, то система должна переключиться на другой мастер-сервер. Это может занять некоторое время и может вызвать некоторые проблемы в работе приложения.
- Ограничения на операции записи: мастер-сервер отвечает только за операции записи, что может ограничить возможности приложения для выполнения некоторых операций записи.

3.2 Технология Multi-master

Технология *Множество-множество (Multi-master)* предполагает, что есть несколько мастер-серверов, каждый из которых может принимать запросы на запись и обрабатывать их. В этом случае каждый мастер является равноправным и может изменять данные в любой момент времени, а остальные серверы должны

получить обновленные данные как можно скорее. Такой подход позволяет распределять нагрузку между несколькими серверами и увеличить производительность системы. Кроме того, он обеспечивает более высокую отказоустойчивость, поскольку отказ одного мастер-сервера не приводит к остановке всей системы. Однако при использовании технологии Multi-master могут возникать проблемы конфликтов при изменении одних и тех же данных на разных мастер-серверах. Эти конфликты могут быть решены с помощью различных подходов:

- Оптимистическая блокировка: при этом подходе каждый мастер-сервер делает изменения в своей копии данных и сохраняет версию данных, на которую были сделаны изменения. При обновлении данных, мастер-серверы сравнивают версии данных, чтобы определить, были ли они изменены другими мастер-серверами. Если обнаруживается конфликт, то система выполняет откат изменений и повторяет операцию на обновленных данных.
- Пессимистическая блокировка: при этом подходе каждый мастер-сервер блокирует данные, которые он изменяет, чтобы другие мастер-серверы не могли изменять их. Этот подход может привести к ухудшению производительности и увеличению задержек при обновлении данных.
- Резолюция конфликтов: при этом подходе система использует алгоритмы для резолюции конфликтов при изменении одних и тех же данных на разных мастер-серверах. Резолюция конфликтов может быть реализована с помощью различных методов, таких как выбор "победившей" версии данных на основе времени изменения или выбор "победившей" версии на основе определенных правил.

Преимущества технологии Multi-master в РСБД:

- Высокая производительность: в многопользовательской среде каждый узел базы данных может принимать запросы на чтение и запись одновременно, что повышает производительность системы.

- Отказоустойчивость: если один узел выходит из строя, другие узлы могут продолжать работу и обеспечивать доступ к данным.
- Распределение нагрузки: при масштабировании системы можно добавлять новые узлы и распределять нагрузку между ними.

Недостатки технологии Multi-master в РСБД:

- Сложность настройки: настройка многомастер-конфигурации может быть сложной и требует дополнительных знаний и опыта в администрировании баз данных.
- Конфликты записей: если несколько узлов пытаются изменить одни и те же данные одновременно, может возникнуть конфликт записей, который нужно разрешать.
- Распределение данных: необходимо разработать стратегию распределения данных между узлами, что может потребовать дополнительных ресурсов и времени на настройку.

3.3 Технология Multi-master replication

Репликация многих мастеров (Multi-master replication): Технология Репликации многих мастеров (Multi-master replication) в тиражировании используется для создания распределенной базы данных, в которой все узлы являются мастерами и могут изменять данные. Эта технология позволяет ускорить обработку транзакций и увеличить отказоустойчивость системы, так как любой узел может продолжить работу в случае сбоя другого узла. В такой системе каждый узел имеет свою копию данных, которую он может изменять и обновлять независимо от других узлов. Изменения синхронизируются между узлами с помощью механизма репликации, который позволяет копировать изменения из одного узла в другой. Репликация может быть синхронной или асинхронной. В синхронной репликации изменения отправляются на все узлы сразу, чтобы гарантировать, что данные на всех узлах будут одинаковыми в любой момент времени. В асинхронной репликации изменения отправляются на другие узлы после того, как они были сделаны на текущем узле. Это может привести к тому, что данные на разных узлах будут различаться на короткое

время, но такой подход может быть более эффективным в случае большой нагрузки на систему. Для устранения конфликтов при репликации изменений, используются различные стратегии разрешения конфликтов, например:

- Последний выигрывает (Last-Write-Wins, LWW): при использовании этой стратегии, если два или более узла вносят изменения в одни и те же данные в разное время, то принимается последнее внесенное изменение. Например, если узел A изменяет данные на значение X, а затем узел B изменяет их на значение Y, то значение Y будет принято в качестве окончательного результата. Эта стратегия проста в реализации, но может привести к потере данных и непредсказуемому поведению системы.
- Версионирование (Versioning): в этой стратегии каждое изменение данных получает уникальный идентификатор версии. При возникновении конфликта система сохраняет оба варианта изменений, создавая различные версии данных. При чтении данных приложение может выбрать нужную версию в соответствии с логикой приложения или согласно временной метке. Такая стратегия обеспечивает полноту данных, но требует дополнительного управления версиями и может потребовать более сложной обработки конфликтов.
- Двойное слияние (Dual-Master Merge): при этой стратегии каждый узел сохраняет изменения и разрешает конфликты путем объединения изменений из разных узлов. Если два узла вносят изменения в одни и те же данные, то система пытается автоматически объединить изменения, чтобы сохранить целостность данных. Эта стратегия требует более сложной обработки конфликтов и слияния данных, но может обеспечивать более гибкое и автоматическое разрешение конфликтов.
- Пользовательское разрешение конфликтов: вместо автоматического разрешения конфликтов, система может предоставить возможность пользователю или разработчику приложения разрешить конфликты вручную.

Преимущества технологии Multi-master replication в РСБД:

- Большая отказоустойчивость и надежность: если один из узлов выходит из строя, другие могут продолжать работу и обслуживание пользователей без простоев.
- Более высокая производительность и масштабируемость: можно добавлять новые узлы для обработки большего количества запросов и увеличения скорости ответа на запросы.
- Гибкость в работе с данными: распределенные узлы могут работать с различными типами данных и хранить их в различных форматах.

Недостатки технологии Multi-master replication в РСБД:

- Сложность конфигурации и настройки: каждый узел должен быть настроен на работу с другими узлами, а также нужно учитывать возможность конфликтов при одновременных изменениях данных на разных узлах.
- Большая вероятность возникновения конфликтов при изменении данных: при одновременных изменениях на разных узлах может возникнуть необходимость в ручной разрешении конфликтов, что может затруднить работу с данными.
- Большее потребление ресурсов: использование нескольких мастер-узлов может привести к большему потреблению ресурсов, таких как память, процессорное время и сетевой трафик.

3.4 Технология Sharding

Технология *Шардинг (Sharding)* является методом горизонтального разделения данных, при котором данные разбиваются на множество меньших фрагментов, называемых шардами. Каждый шард содержит часть данных из всей базы данных и хранится на отдельном сервере. Таким образом, база данных разбивается на несколько независимых фрагментов, которые могут храниться и обрабатываться параллельно.

Принцип работы технологии Шардинг заключается в том, что каждый запрос к базе данных обрабатывается только на тех серверах, на которых

находятся соответствующие данные. Например, если запрос содержит информацию о пользователе с идентификатором 123, то запрос будет отправлен только на тот сервер, на котором находится шард с данными об этом пользователе. Технология Шардинга позволяет улучшить производительность и масштабируемость системы. За счет параллельной обработки запросов на разных серверах можно увеличить скорость выполнения запросов и сократить время ответа. Кроме того, технология Шардинг позволяет увеличить емкость базы данных, так как ее размер не ограничен единственным сервером. Однако использование технологии Шардинг также имеет свои недостатки. Во-первых, при разделении данных на шарды возникает проблема поддержания целостности данных и согласованности транзакций. Во-вторых, необходимо учитывать неравномерную нагрузку на серверы, что может привести к деградации производительности и необходимости балансировки нагрузки. В-третьих, при использовании технологии Шардинг необходимо обеспечить безопасность и защиту данных, так как данные хранятся на нескольких серверах.

Преимущества технологии Sharding в РСБД:

- Высокая масштабируемость: позволяет увеличить объем данных, которые могут быть обработаны распределенной системой баз данных.
- Более высокая доступность: если один из узлов сети не работает, то остальные узлы продолжат работу.
- Быстрый доступ к данным: в отличие от традиционных методов масштабирования, Sharding позволяет увеличить скорость доступа к данным.
- Гибкость: Sharding может быть применен для различных видов данных и приложений.

Недостатки технологии Sharding в РСБД:

- Сложность: Sharding может быть сложной технологией для настройки и управления.

- Потеря данных: если узел сети, содержащий фрагмент данных, выходит из строя, то эти данные могут быть утеряны, если не предусмотрена репликация данных.
- Несогласованность данных: в связи с распределением данных между несколькими узлами сети может возникнуть проблема несогласованности данных. Необходимо принимать меры для поддержания целостности данных в распределенной среде.
- Сложности с управлением данными: с увеличением числа фрагментов данных может возникнуть сложность управления данными в распределенной среде. Необходимо разработать и применять соответствующие методы управления данными.

4. Синхронизация в системах с распределенной базой данных

Одной из ключевых проблем, с которой сталкиваются распределенные базы данных (РСБД), является проблема синхронизации. Эта проблема возникает из-за того, что в РСБД данные могут храниться на разных узлах, которые могут обрабатывать запросы независимо друг от друга. Если данные на одном узле изменяются, то эти изменения не могут быть автоматически переданы на другие узлы, и это может привести к несогласованности данных между узлами.

Проблема синхронизации в РСБД может привести к различным проблемам, таким как:

- Несогласованность данных: если данные на разных узлах не синхронизированы, то могут возникнуть различия в данных, что может привести к непредсказуемым результатам при обработке запросов.
- Потеря данных: если изменения на одном узле не синхронизируются с другими узлами, то это может привести к потере данных.
- Низкая производительность: при синхронизации данных между узлами возникает дополнительная нагрузка на сеть и серверы, что может привести к снижению производительности системы.

В распределенных системах баз данных (РСБД) существует несколько методов синхронизации данных, которые помогают поддерживать целостность и согласованность данных между различными узлами системы. Далее мы рассмотрим четыре основных метода синхронизации.

4.1Метод Optimistic Replication

Метод оптимистической синхронизации (Optimistic Replication) Метод оптимистической синхронизации – это один из методов синхронизации данных в распределенных системах баз данных (РСБД). Он основан на предположении, что конфликты в изменении данных будут редкими и, поэтому, можно разрешить их в момент обнаружения. Основная идея метода оптимистической синхронизации заключается в том, что каждый узел РСБД может вносить изменения в копию данных без необходимости получения разрешения на запись

с других узлов. После внесения изменений, узел пытается применить их к другим копиям данных в РСБД. Если другие копии данных не изменились, изменения принимаются без конфликта. Если же другие копии данных были изменены, то возникает конфликт, который требует разрешения. Для определения конфликтов метод оптимистической синхронизации использует метаданные, которые хранят информацию о состоянии каждой копии данных. Метаданные могут включать в себя информацию о том, когда и какая копия данных была изменена, и какие транзакции были выполнены. Разрешение конфликтов в методе оптимистической синхронизации может происходить различными способами. Например, конфликты могут решаться путем отмены изменений на одной из копий данных или путем объединения изменений в одну копию данных. Основным преимуществом метода оптимистической синхронизации является то, что он позволяет узлам вносить изменения в копии данных без ожидания разрешения с других узлов, что может ускорить работу в распределенной среде. Кроме того, метод оптимистической синхронизации может обеспечить более высокую доступность данных, так как узлы могут продолжать работать, даже если связь между ними временно прервется.

4.2 Метод Pessimistic Replication

Метод пессимистической синхронизации (Pessimistic Replication) – это метод синхронизации данных в распределенных системах баз данных (РСБД), основанный на блокировке доступа к данным во время их изменения. В отличие от метода оптимистической синхронизации, где изменения вносятся без блокировки, метод пессимистической синхронизации требует блокировки данных для предотвращения конфликтов. Основная идея метода пессимистической синхронизации заключается в том, что перед внесением изменений в данные узел должен получить блокировку на соответствующие данные. Блокировка предотвращает другие узлы от одновременного доступа к данным, пока изменения не будут завершены. Существуют различные виды блокировок, используемых в методе пессимистической синхронизации:

- Блокировка чтения (Read Lock): предотвращает другие узлы от изменения данных, пока узел не завершит чтение данных.
- Блокировка записи (Write Lock): предотвращает другие узлы как от чтения, так и от записи данных, пока узел не завершит изменение данных.
- Эксклюзивная блокировка (Exclusive Lock): предотвращает другие узлы от одновременного доступа к данным, пока узел не завершит свою операцию.

Применение метода пессимистической синхронизации требует аккуратного управления блокировками, чтобы избежать проблем с производительностью и длительными блокировками. Например, длительная блокировка записи может привести к ожиданию доступа к данным другими узлами и снижению производительности системы. Однако метод пессимистической синхронизации обеспечивает высокую степень согласованности данных, так как изменения данных блокируются и контролируются. Это позволяет избежать конфликтов при одновременных изменениях данных разными узлами. Кроме того, данный метод может быть полезен в случаях, когда требуется строгая согласованность данных, например, при выполнении транзакций.

Важно отметить, что метод пессимистической синхронизации может потребовать дополнительных ресурсов для управления блокировками и контроля доступа к данным. Это может оказывать влияние на производительность.

4.3 Метод Quorum-Based Replication

Метод кворума (Quorum-Based Replication) – это метод синхронизации данных в распределенных системах баз данных (РСБД), который основан на определении кворума, то есть минимального числа узлов, которые должны участвовать в изменении данных или подтвердить изменения данных, чтобы они стали действительными. В методе кворума каждому узлу системы присваивается голос. Голос представляет собой единицу принятия решений в системе и указывает, является ли узел активным участником или резервным. Каждое

изменение данных требует участия кворума узлов, чтобы быть успешно примененным. Примеры кворумов:

- Majority Quorum (Кворум большинства): для принятия решения по изменению данных требуется участие большинства узлов. Например, в системе с 5 узлами, кворумом может быть 3 узла. Это означает, что как минимум 3 узла должны подтвердить изменение, чтобы оно стало действительным.
- All Quorum (Кворум всех узлов): для принятия решения по изменению данных требуется участие всех узлов в системе. Это означает, что все узлы должны подтвердить изменение, чтобы оно стало действительным.
- Strict Majority Quorum (Строгий кворум большинства): для принятия решения по изменению данных требуется участие большинства узлов, но необходимо также участие определенного количества специфических узлов. Например, в системе с 5 узлами, кворумом может быть 3 узла, но также требуется участие хотя бы одного из определенных узлов.

Преимущества метода кворума включают высокую доступность данных и отказоустойчивость. Если некоторые узлы недоступны или вышли из строя, система все равно может продолжать функционировать, пока кворум узлов доступен. Кроме того, метод кворума обеспечивает согласованность данных, так как изменения должны быть подтверждены достаточным числом узлов. Недостатки метода кворумов в РСБД могут включать следующее:

- Необходимость выбора оптимального значения кворума, которое может быть сложно в распределенной среде.
- Возможность возникновения конфликтов при использовании метода кворумов.
- Недостаточная надежность в случае сбоев или отказов в системе.

4.4 Метод Multi-Version Concurrency Control

Метод многоверсионной синхронизации (Multi-Version Concurrency Control) - это метод синхронизации данных в распределенных системах баз данных (РСБД), который используется для обеспечения согласованности данных

при параллельных транзакциях. Этот метод позволяет не блокировать таблицы и не блокировать другие пользователи при выполнении транзакций. Основная идея метода MVCC заключается в том, что каждая транзакция работает с своей версией данных, а не с исходными данными, которые могут быть изменены другими транзакциями. Каждая версия данных имеет свой уникальный идентификатор, который помогает отслеживать изменения данных, выполненные каждой транзакцией. При выполнении операции чтения в MVCC используется механизм «снимков» (snapshot), который позволяет каждой транзакции работать с конкретной версией данных, которая была в момент начала выполнения транзакции. Таким образом, транзакции, которые работают параллельно, могут использовать разные версии данных, которые соответствуют состоянию данных на момент начала выполнения транзакции.

Одним из преимуществ метода MVCC является отсутствие блокировок таблиц и возможность одновременного доступа нескольких транзакций к одним и тем же данным. Это позволяет увеличить производительность системы и снизить вероятность блокировки транзакций.

Однако, метод MVCC требует дополнительных ресурсов для хранения версий данных, что может привести к увеличению занимаемого места на диске. Кроме того, при использовании этого метода может возникнуть проблема «фантомных чтений», когда одна транзакция читает данные, которые были изменены другой транзакцией во время выполнения первой транзакции.

5. Задачи, возникающие при проектировании и разработке систем с распределенной базой данных

Проектирование и разработка распределенных систем баз данных (РСБД) – это процесс создания системы, которая позволяет хранить и обрабатывать данные в распределенной среде, т.е. на нескольких компьютерах, связанных между собой сетью. РСБД используются для решения проблем масштабируемости, доступности, отказоустойчивости и совместной работы над данными.

При проектировании и разработке распределенных систем баз данных возникают следующие задачи:

- Разработка архитектуры системы: необходимо определить, как будет организована структура системы, как будут связаны различные ее компоненты и как будут распределены данные между узлами.
- Выбор технологий: необходимо выбрать технологии, которые будут использоваться для реализации системы. Например, какая технология тиражирования или синхронизации данных будет использована.
- Определение требований к надежности и отказоустойчивости: в распределенных системах баз данных необходимо учитывать возможность отказов отдельных узлов или сетевых соединений. Поэтому важно определить, какие меры будут приниматься для обеспечения отказоустойчивости системы.
- Управление данными: в распределенных системах баз данных необходимо учитывать различные проблемы управления данными, такие как консистентность, целостность и безопасность данных.
- Масштабирование системы: необходимо учитывать возможность масштабирования системы с увеличением количества узлов и объема данных.
- Обеспечение производительности: при проектировании распределенных систем баз данных необходимо учитывать производительность системы и ее возможности по обработке запросов на чтение и запись данных.

- Управление ресурсами: необходимо учитывать различные проблемы управления ресурсами, такие как управление памятью и управление сетевыми соединениями.
- Тестирование и отладка: необходимо проводить тестирование и отладку системы для обеспечения ее стабильной работы и обнаружения возможных ошибок и проблем.

6. Тенденции развития в области систем с распределенной базой данных

Развитие технологий в области распределенных систем баз данных продолжается, и на сегодняшний день можно выделить несколько тенденций:

- Использование облачных технологий для хранения и обработки данных. Облачные сервисы позволяют быстро масштабировать вычислительные ресурсы и обеспечивают высокую доступность и отказоустойчивость.
- Развитие технологий кластеризации и тиражирования. Эти технологии позволяют распределить нагрузку на несколько узлов, обеспечивая более высокую производительность и отказоустойчивость.
- Использование новых подходов к синхронизации данных. Например, методы оптимистической и пессимистической синхронизации, а также многоверсионная синхронизация позволяют уменьшить число конфликтов при одновременном изменении данных на разных узлах.
- Использование интеллектуальных алгоритмов и машинного обучения для обработки больших объемов данных. Это позволяет ускорить анализ данных и выделить из них более точные и полезные закономерности.
- Развитие технологий блокчейн, которые позволяют хранить данные в распределенном режиме и обеспечивать прозрачность и безопасность операций.
- Разработка инструментов для работы с геоданными и IoT-данными. Это позволяет обрабатывать большие объемы данных, полученных от датчиков и других устройств, и использовать их для управления процессами в реальном времени.
- Развитие технологий виртуализации и контейнеризации. Это позволяет быстро разворачивать и масштабировать приложения и базы данных на разных узлах сети.

Все эти тенденции открывают новые возможности для пользователей и разработчиков, позволяя быстро обрабатывать большие объемы данных и использовать их для принятия решений в реальном времени. Однако, при этом

возникают и новые проблемы, такие как безопасность, конфиденциальность, управление данными и ресурсами, которые также требуют развития новых технологий и подходов.

ЗАКЛЮЧЕНИЕ

Технологии удаленного доступа к системам баз данных, тиражирование и синхронизация в распределенных системах баз данных играют важную роль в современных IT-системах и позволяют обеспечить высокую доступность, отказоустойчивость и производительность при работе с большими объемами данных. Однако, при проектировании и разработке распределенных систем баз данных возникает ряд сложностей, связанных с выбором архитектуры, технологии синхронизации и масштабируемости системы. Каждая из рассмотренных в работе технологий имеет свои преимущества и недостатки, и выбор конкретного подхода зависит от конкретной задачи и требований к системе.

Таким образом, разработка и внедрение распределенных систем баз данных является актуальной задачей для современных организаций и предприятий, и требует профессионального подхода и использования передовых технологий. В то же время, необходимо учитывать, что эффективное использование распределенных систем баз данных требует поддержки со стороны квалифицированных специалистов и постоянного мониторинга работы системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Технология тиражирования данных с распределенных системах [Электронный ресурс]: – URL: https://www.osp.ru/os/1994/02/178487 (дата обращения 05.03.2023). – Яз. Рус.
2. Синхронизация данных в распределенной информационной базе [Электронный ресурс]: – URL: https://its.1c.ru/db/docprof21/content/180/hdoc (дата обращения 05.03.2023). – Яз. Рус.
3. Синхронизация реплик базы данных в распределенной системе [Электронный ресурс]: – URL: https://cyberleninka.ru/article/n/sinhronizatsiya-replik-bazy-dannyh-v-raspredelennoy-sisteme (дата обращения 05.03.2023). – Яз. Рус.
4. Средства доступа к базам данных в Internet и свободно доступная СУБД POSTGRES95 [Электронный ресурс]: – URL: https://www.osp.ru/dbms/1997/02/13031533 (дата обращения 05.03.2023). – Яз. Рус.
5. Технологии и средства доступа к удаленным бд [Электронный ресурс]: – URL: https://studref.com/382658/informatika/tehnologii_sredstva_dostupa_udalennym (дата обращения 05.03.2023). – Яз. Рус.
6. Тиражирование данных [Электронный ресурс]: – URL: https://prog.bobrodobro.ru/25930 (дата обращения 05.03.2023). – Яз. Рус.
7. Технологии реализации распределенных баз данных [Электронный ресурс]: – URL: https://all4study.ru/bd/texnologii-realizacii-raspredelennyx-baz-dannyx.html (дата обращения 05.03.2023). – Яз. Рус.
8. Технологии распределенных баз данных [Электронный ресурс]: – URL: https://ami.nstu.ru/~vms/lecture/lecture10/lecture10.htm (дата обращения 05.03.2023). – Яз. Рус.

9. Основные тенденции в развитии распределенных баз данных [Электронный ресурс]: – URL: https://www.gpntb.ru/libcom5/disk/doc/26.pdf (дата обращения 05.03.2023). – Яз. Рус.