ОНЛАЙН-ОБРАЗОВАНИЕ

OTUS

Репликация. Часть 2.

Иван Ремень

Как меня слышно и видно?

> Напишите в чат

+ если все хорошо

- если есть проблемы со звуком или с видео

!проверить запись!

План занятия

- Как происходит коммит транзакции.
- Проблемы асинхронной репликации.
- Проблемы мастер-мастер репликации.
- Ответы на вопросы.

Как происходит коммит транзакции

- Prepare the transaction in the storage engine (InnoDB),
- Write the transaction to the binary logs,
- Complete the transaction in the storage engine,
- Return an acknowledgment to the client.

Отставание репликации

Отставание репликации

В нормальной ситуации отставание достигает секунды.

Плохие запросы (и по мастеру, и по реплике) могут вызвать отставание репликации.

Рекомендация:

- Убивайте медленные запросы.
- Держите отдельную реплику для медленных запросов.
- Думайте о кросс-СУБД репликации.
- Старайтесь избегать паттерна запись-чтение.

Чтение своих записей.

Варианты:

- Свои данные читаем с мастера, чужие - со слейва.
- Читаем с мастера n секунд после записи.

Монотонное чтение.

Ожидается, что пользователь не будет видеть пропадающие комментарии.

Вариант - привязать пользователя к реплике.

Согласованное префиксное чтение.

Характерно для шардированных баз данных.

Важно для сохранения причинно-следственных связей.

GTID

Global transaction identifier

Имеет формат `server-id:transaction-id`:

```
3E11FA47-71CA-11E1-9E33-C80AA9429562:2
```

Позволяет убедиться, что транзакция принадлежит только одному серверу.

Позволяет убедиться, что транзакция применена только один раз в системе

master-master репликация

Самое важное - не масштабирует запись. Для масштабирования записи нужен шардинг!

Применение в 1 ЦОДе - сомнительная идея.

Варианты применения:

- Географическая распределенность.
- hot-standby реплика (VIP).
- offline клиенты. Релизовать сложно. CouchDB была сделана специально для этого случая.

Цена master-master:

- Усложнение логики.
- Проблемы с конфликтами.

master-master репликация

Для географически распределенных ЦОД будут следующие преимущества:

- Производительность.
- Устойчивость к уходу ЦОДа.
- Устойчивость к проблемам сети.

Конфликт в m/m репликации

Решение конфликтов.

- Избегание конфликтов.
- Last wins.
- Ранг реплик. Выйгрывает запись от старшей реплики.
- Слияние.
- Решение конфликтов на клиенте.
- Conflict-free replicated data types (CRDT).
- Mergeable persistent data structures.

Безмастерная репликация

Безмастерная репликация

Такая репликация есть в:

- DynamoDB
- cassandra
- scylla
- riak
- voldemort

Формула для расчета кворума: `w + r > number of replicas`

Безмастерная репликация - поддержание консистентности

- Обновление при чтении
- Противодействие энтропии

Безмастерная репликация - проблемы

- Нестрогий кворум. Возможно чтение старых данных при `w + r < n`
- Нет отката транзакций.
- Конфликт записей и потерянные обновления.
- Проблемы с линеаризуемостью.

Вывод - гарантий нет.

Важный вывод - как всегда в интернете бывает 2 исхода - успех и неизвестность.

Нестрогий кворум

Что лучше для системы при отсутствии кворума:

- Вернуть ошибку?
- Применить запись без квормума?
- Вернуть устаревшие данные?

Можно писать в узлы, не входящие в `n`. Это называется нестрогий кворум.

Безмастерная репликация - конфликты

Решение конфликтов - LWW

Алгоритм Last write wins.

Выигрывает последняя запись.

Нормально не может работать из-за физической невозможность синхронизации часов.

Единственный нормальный способ - не обновлять ключи.

Решение конфликтов - "происходит до"

Две операции конкуренты тогда и только тогда, когда они независимы.

- Сервер хранит номера версий для всех ключей, увеличивая номер версии всякий раз при выполнении записи значения для этого ключа, и сохраняет новый номер версии вместе с записанным значением.
- При чтении ключа клиентом сервер возвращает все неперезаписанные значения, а также последний номер версии. Клиент должен прочитать ключ перед операцией записи.
- Клиент, записывая значение для ключа, должен включить номер версии из предыдущей операции чтения, а также объединить все полученные при предыдущей операции чтения значения. (Полученный в результате операции записи ответ может быть таким же, как и для чтения, с возвратом всех текущих значений, что позволяет соединять несколько операций записи последовательно, подобно примеру с корзиной заказов.)
- Сервер, получив информацию об операции записи с конкретным номером версии, может перезаписать все значения с этим или более низким номером версии (так как знает, что они все слиты воедино в новом значении), но должен сохранить все значения с более высоким номером версии (поскольку эти значения конкурентны данной входящей операции записи).

Решение конфликтов - tombstones

Решение конфликтов - обязанность клиентов.

Удаляемую запись нельзя просто убрать из списка. Необходимо сделать явную отметку об удалении.

Вопросы?

Результаты занятия

- Как происходит коммит транзакции.
- Проблемы асинхронной репликации.
- Проблемы мастер-мастер репликации.
- Ответы на вопросы.

Опрос

Заполните пожалуйста опрос

https://otus.ru/polls/3938/

Спасибо за внимание!