Лекция
02.04.2019

DBA2 Theory. p4.

Асинхронная репликация MariaDB

Ильшат Каразбаев
руководитель группы DBA
АО ТК Центр

Немного обо мне

Вместе со своей командой администрирую:

СУБД MySQL, Mariadb, galeracluster, Postgres

Главный по базам в ТК Центр

Повестка дня:

1. Вводная
2. Как работает репликация
3. Настройка репликации
4. RBR репликация
5. SBR репликация
6. Топологии репликации
7. Планирование ресурсов
8. Мониторинг
9. Администрирование
10. Проблемы и решения
11. Литература

Вводная

На первой лекции мы ознакомились с разновидностями репликации.

В курсе DBA2 Theory будет, как минимум, три занятия посвящено асинхронной репликации:

- Репликация в классическом виде по позиции бинарного лога
- Репликация по GTID
- Multisource репликация

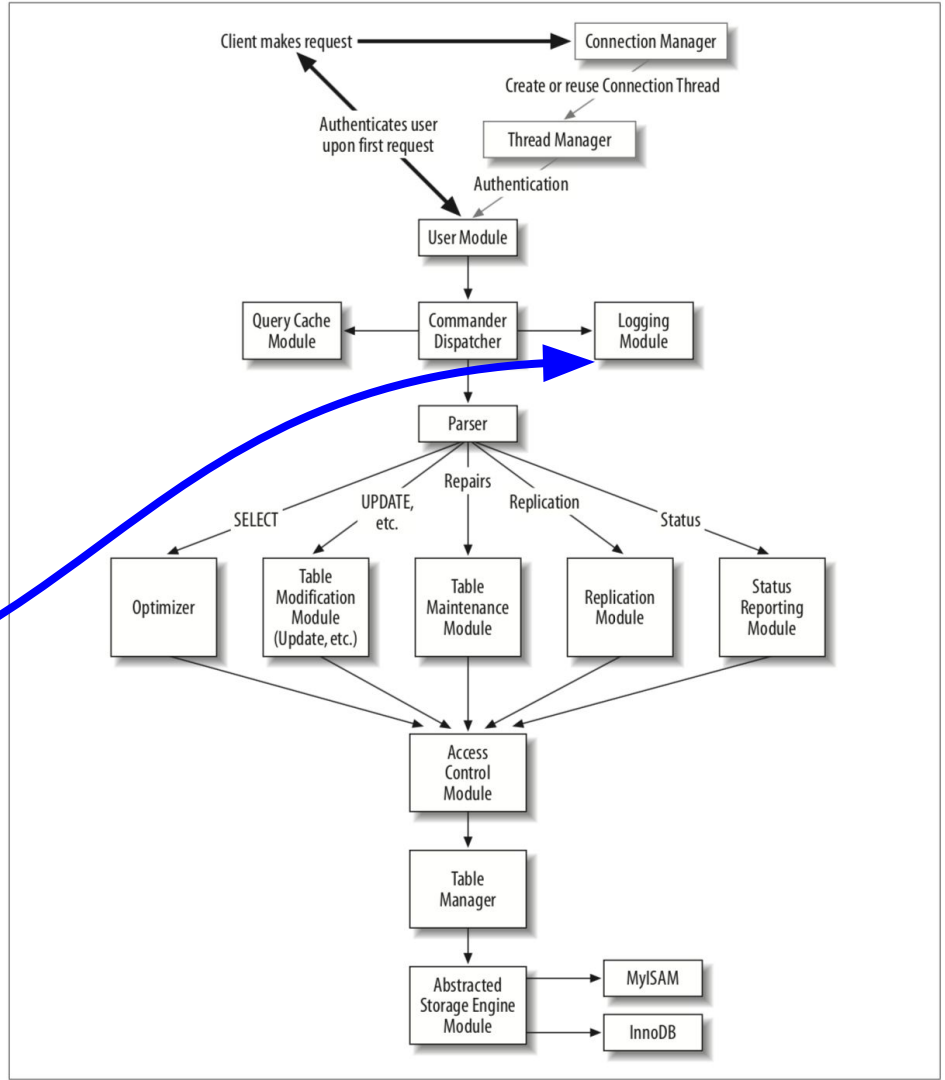
Вводная

Для чего нужна репликация

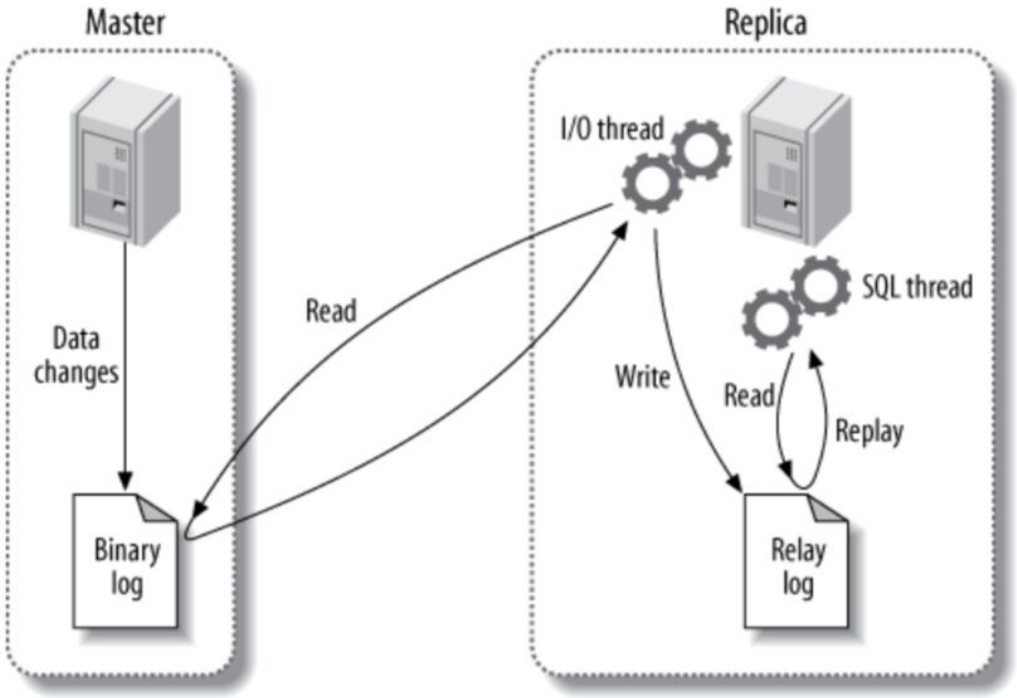
- География, доставить данные в удаленный дц
- Распределение нагрузки (чтение)
- Резервные копии
- Высокая доступность и фейловер
- Апгрейд версий СУБД
- Миграция данных

Место в архитектуре

Место в архитектуре

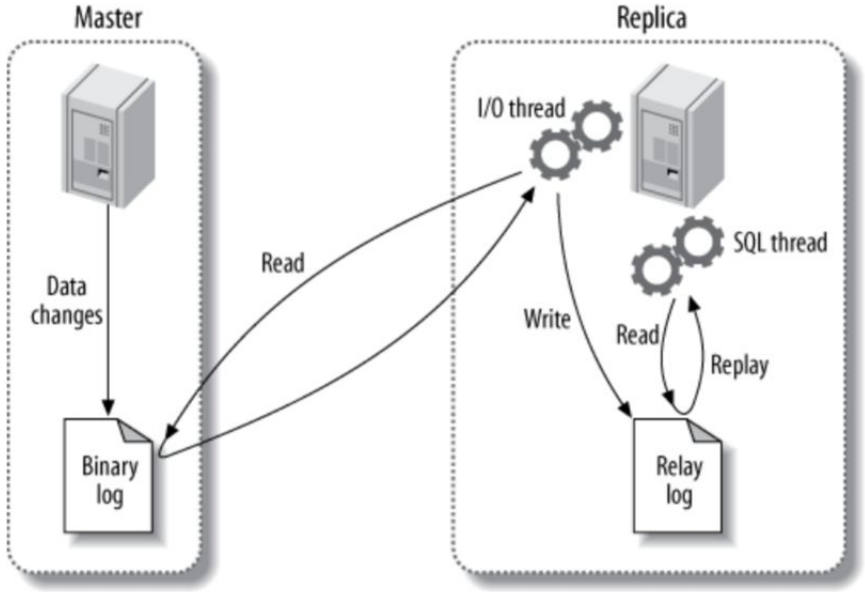


Как работает репликация



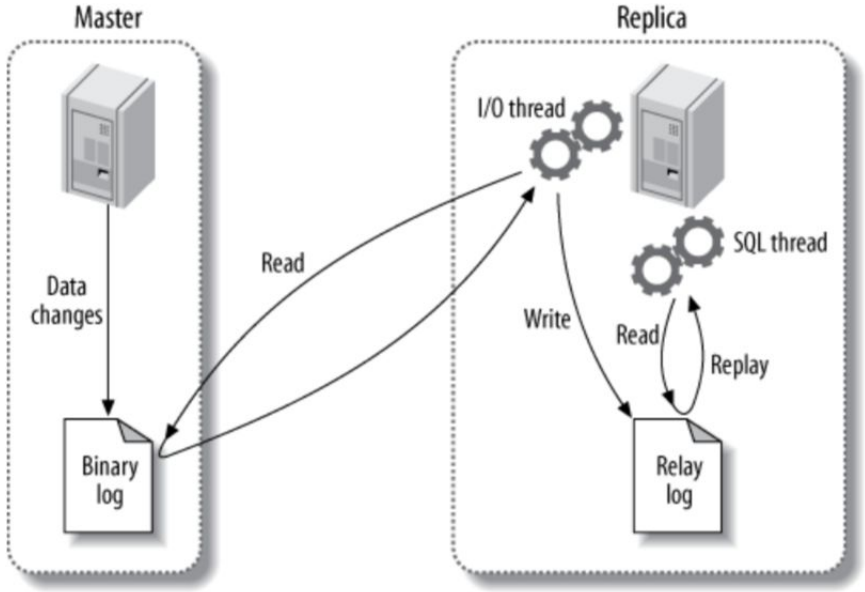
Как работает

1. Мастер пишет binary logs events
2. Реплика копирует бинлоги с мастера в свой relay log
3. Реплика проигрывает события из relay log и применяет их

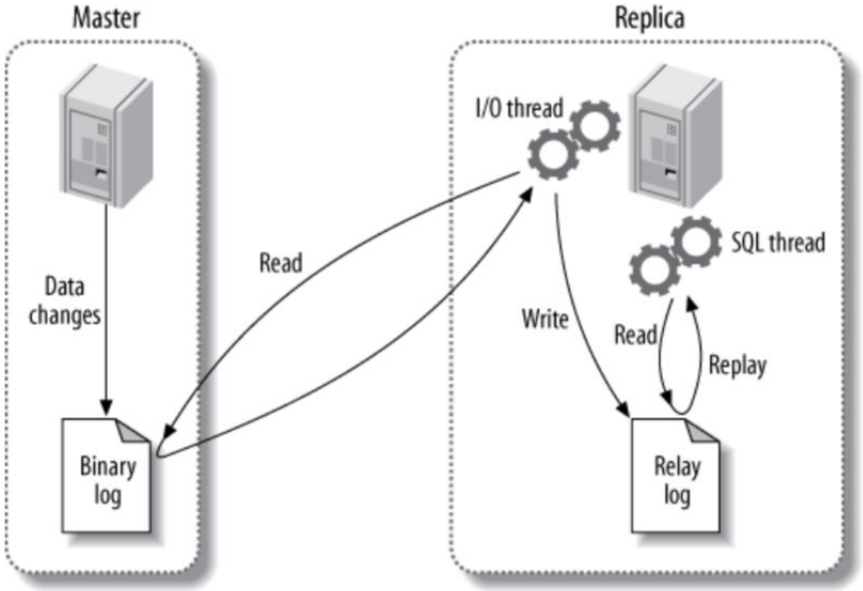


Как работает

1. Мастер пишет binary logs events перед коммитом и коммитит
2. Реплика копирует бинлоги с мастера в свой relay log
3. Реплика проигрывает события из relay log и применяет их

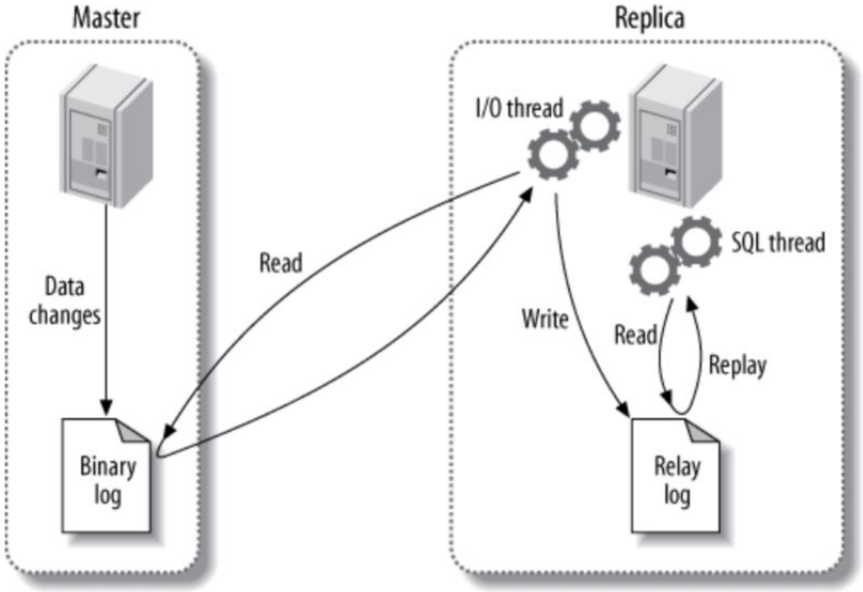


Как работает



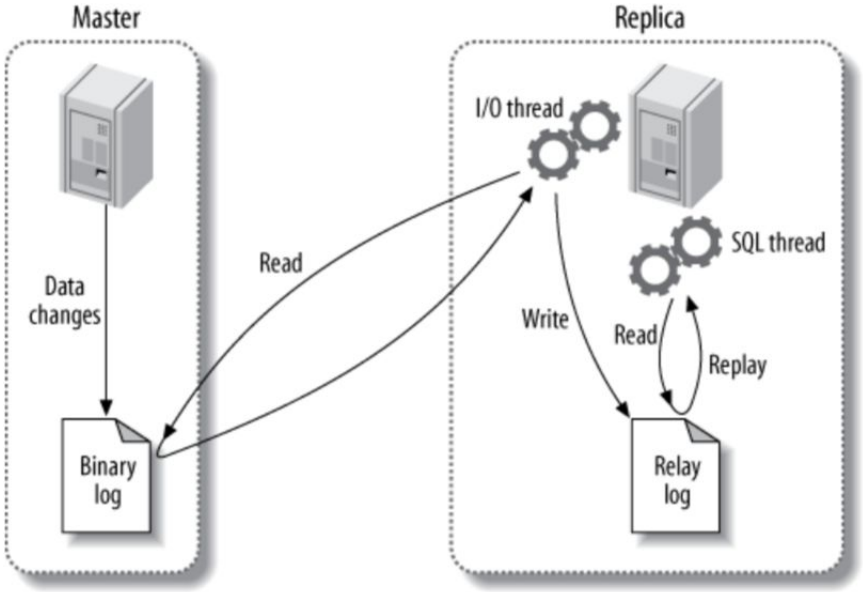
1. Мастер пишет binary logs events перед коммитом и коммитит
2. Реплика копирует бинлоги с мастера в свой relay log на жесткий диск. Для этого она стартует I/O Slave Thread, который открывает коннект к мастеру и стартует процесс binlog dump
3. Реплика проигрывает события из relay log и применяет их

Как работает



1. Мастер пишет binary logs events перед коммитом и коммитит
2. Реплика копирует бинлоги с мастера в свой relay log на жесткий диск. Для этого она стартует I/O Slave Thread, который открывает коннект к мастеру и стартует процесс binlog dump
3. Реплика проигрывает события из relay log и применяет их. Для этого стартует SQL Slave Thread

Как работает



1. Мастер пишет binary logs events перед коммитом и коммитит
2. Реплика копирует бинлоги с мастера в свой relay log на жесткий диск. Для этого она стартует I/O Slave Thread, который открывает коннект к мастеру и стартует процесс binlog dump
3. Реплика проигрывает события из relay log и применяет их. Для этого стартует SQL Slave Thread
4. Опционально реплика может писать свой бинарный лог

Настройка репликации

Выставим привилегии на мастере

GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.*-> TO repl@'192.168.0.%' IDENTIFIED BY 'p4ssword',;

Конфигурационные параметры

Мастер

log_bin = /var/log/mysql/mariadb-bin

server_id = 10

Реплика

log_bin = /var/log/mysql/mariadb-bin

server_id = 11

Сделать дамп и развернуть на реплике

Старт репликации

CHANGE MASTER TO MASTER_HOST='server1',

-> **MASTER_USER='repl',**

-> **MASTER_PASSWORD='p4ssword',**

-> **MASTER_LOG_FILE=’mariadb-bin.000001',**

-> **MASTER_LOG_POS=0;**

Рекомендованные настройки

На мастере и реплике

innodb_flush_logs_at_trx_commit=1
sync_binlog = 1

На реплике

relay_log=/var/log/mysql/relay-bin
skip_slave_start
read_only
sync_master_info = 1
sync_relay_log = 1
sync_relay_log_info =1

Виды репликации

Statement based replication

Row based replication

Виды репликации

Statement based replication

1. Плюсы:
- Можно иметь на реплике и на мастере разные схемы
- Небольшие бинарные логи
2. Минусы:
- Очень много проблем

Row based replication

Виды репликации

Statement based replication

Row based replication

1. Плюсы:
- Корректно работает с триггерами, хранимыми процедурами, и прочими сущностями
- Меньше нагружает реплику, так как не нужно строить план запроса и исполнять его
- Данные согласованы, если изменения не могут быть применены, репликация остановится с ошибкой
- Легче искать ошибки, которые привели к несогласованности данных
2. Минусы:
- Нет такой гибкости, как в SBR, нельзя просто так работать с разными схемами таблиц
- Чаще ломается
- При использовании каскадной репликации, если применить binlog_format=STATEMENT, то это будет работать только до первых реплик в каскаде

Файлы репликации

mysql-bin.index - список бинарных логов на диске

mysql-relay-bin.index - список relay логов на диске

master.info - информация для коннекта к мастеру

relay-log.info - координаты бинарных и relay логов

Фильтры репликации

Это решение используется для репликации части БД

При использовании стоит быть осторожным, так как DDL реплицируются независимо от настроек и могут сломать репликацию, так как не будет баз или таблиц, на которых она должна выполнится

Для решения этой проблемы можно создать пустые базы или таблицы.

В любом случае, DDL при частичной репликации может сломать реплику

Фильтры репликации

На мастере

- binlog_do_db
- binlog_ignore_db

На реплике

- replicate_do_(db|table)
- replicate_ignore_(db|table)
- replicate_wild_do_table
- replicate_wild_ignore_table

Можно конфигурировать как с помощью файлов конфигурации, так и при настройке реплики через SET

Топологии репликации

Звезда

Кольцо

Пирамида

Подробнее в High Performance MySQL и на страницах документации MariaDB

Мониторинг

Seconds_Behind_Master:

NULL - репликация не работает

0 - отлично!

>0 - реплика отстает

Мониторинг

Slave_IO_Running/Slave_SQL_Running

Yes - Отлично

No - репликация не работает

Мониторинг

Last_Errno/Last_IO_Errno

0 - отлично

Вс остальное - плохо

Мониторинг

pt-heartbeat

https://www.percona.com/doc/percona-toolkit/LATEST/pt-heartbeat.html

Аварийные ситуации

Непредвиденный выход из строя мастера

Если не выставлен innodb_flush_log_at_trx_commit=1 sync_binlog=1, на реплике и мастере могут появиться несогласованные данные (то, что применилось на мастере и успело записаться на реплики до вызова fsync а мастере, на мастере и потеряется, или, если записалось на мастере, может не доехать до реплики, если она отставала

Аварийные ситуации

Аварийный выход из строя реплики

Мог не записаться master.info тогда реплика может стать нескогрласованной с мастером

Могли повредиться бинарные или relay логи

Аварийные ситуации

Повреждены бинарные логи на мастере

Придется перенастраивать реплику с нуля (рекомендовано)

Аварийные ситуации

Поврежденные relay логи на реплике

В поздних версиях это может быть не страшно и реплика их просто пересоздаст

Аварийные ситуации

Бинарный лог рассогласован с транзакционным логом innodb

Пересоздать реплику

Аварийные ситуации

Нетранзакционные таблицы

При использовании нетранзакционных таблиц, данные могут быть среплицированы, в то время как на мастере транзакция зависла и ее убили, например. Нет роллбека.

При смешивании транзакионных и нетранзакционных таблиц может слуиться то же самое

Аварийные ситуации

Недетерменированные выражения

использование LIMIT, information_schema, REPLACE IGNORE, INSERT IGNORE, @@server_id, @@hostname могут работать некорректно в разных версиях mariadb

Аварийные ситуации

Разные storage engines на мастере и реплике

Реплика может выпасть из репликации

Аварийные ситуации

Изменение данных на реплике

При вызове стейтментов сгенерируется binlog event следующий событие бинарного лога, которое придет с матсера будет конфликтовать.

Аварийные ситуации

Совпадающие server id у реплик

Неуникальные id будут конфликтовать и одна из реплик вылетит из репликации

Аварийные ситуации

Неопределенный server id на реплике

Если не сконфигурировать server id а реплике, CHANGE MASTER TO .. отработает, а START SLAVE - нет

Аварийные ситуации

Блокирующие выборки

В InnoDB выборки могут быть блокирующими (INSERT … SELECT)

Блокировки могут ворзникнуть и на реплике

Аварийные ситуации

Отстающая репликация

Аварийные ситуации

Переполнение диска

Вопросы?

Все анонсы здесь:

- telegram чат: t.me/mariadb_course

Материалы курса:

- видео: https://www.youtube.com/channel/UCGsmu6YDpcR_kWcXzeQkWrA
- слайды лекций и примеры: git@github.com:barazbay/mariadb_course.git

Меня можно найти:

- vk, instagram: barazbay
- twitter: karazbay

Литература

1. https://mariadb.com/kb/en/library/high-availability-performance-tuning-mariadb-replication/
2. High Performance MySQL, Baron Schwartz, Peter Zaitcev, Vadim Tkachenko. 2012
3. Understanding MySQL Internals, Sasha Pachev. 2007
4. https://www.percona.com/doc/percona-toolkit/LATEST/pt-heartbeat.html