

Настройка репликации

1. Настроить асинхронную реплику и провести стресстестирование user/get/{id} и user/search

Настройка асинхронной репликации:

- Поднять контейнер с postgres базой
- `docker exec -it db_container sh`
- В `postgresql.conf` добавить
 - `ssl = off`
 - `wal_level = replica`
 - `max_wal_senders = 4`
- В `pg_hba.conf` добавить
 - `host replication replicator 172.16.0.0/16 md5`
- Перезапустить мастер
- На мастере сделать
 - `mkdir /pgslave`
 - `pg_basebackup -h db_container -D /pgslave -U replicator -v -P --wal-method=stream`
 - `docker cp db_container:/pgslave postgresql/db_data_async`
- Поднять контейнер с репликой
- `docker exec -it db_async_container sh`
- Внутри реплики
 - `touch postgresql/db_async_data/standby.signal`
- В `postgresql.conf` добавить
 - `primary_conninfo = 'host=db_container port=5432 user=replicator password=replicator_password'`
 - `application_name=db_async_container'`
- Запросы `user/get/{id}` и `user/search/` переведены на чтение из асинхронной реплики

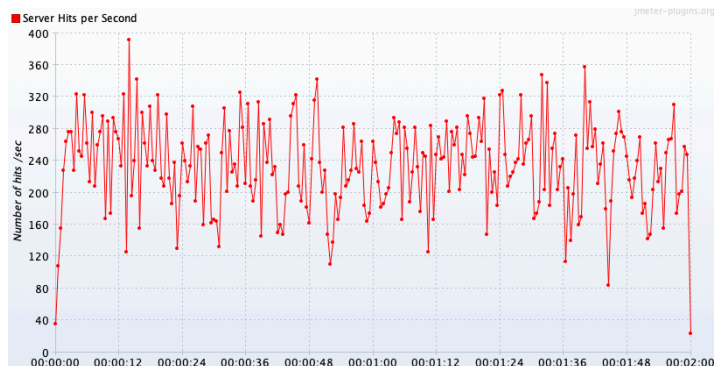
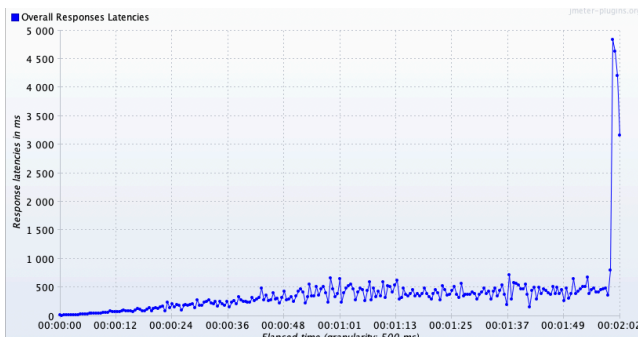
Number of Threads: 100

Ramp-up Period: 60s

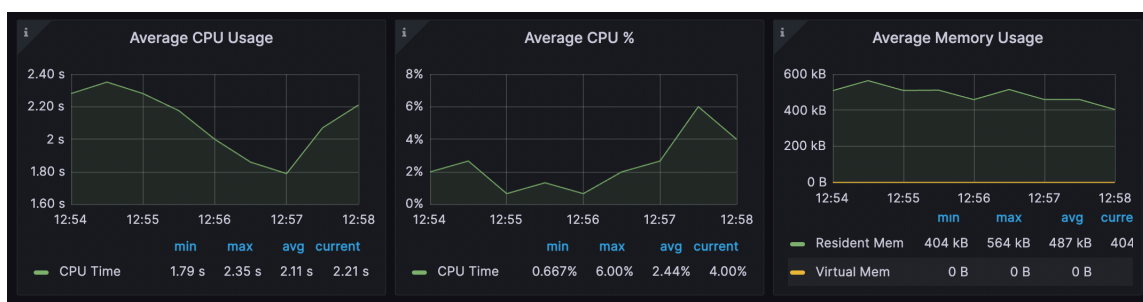
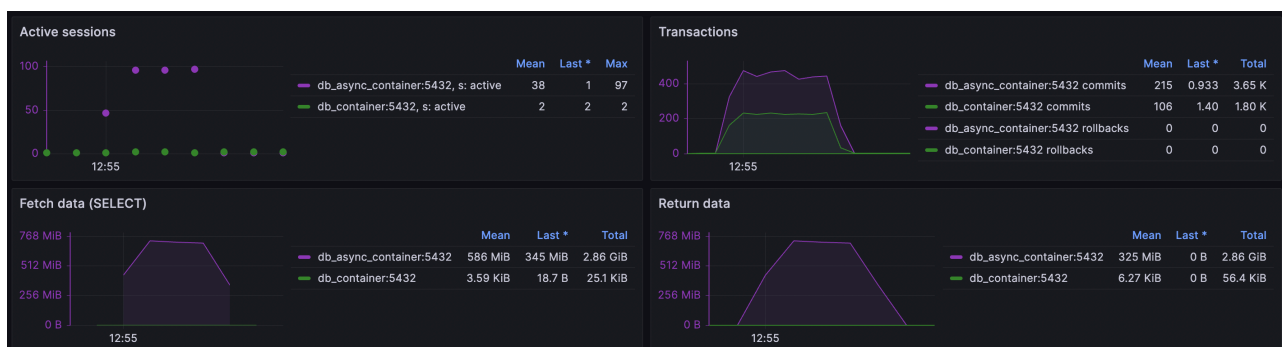
Duration: 120s

Requests: user/get/{id}+ user/search/

Service metrics



Postgres metrics



2. Настроить 2 слейва 1 мастер, настроить кворумную синхронную репликацию

Настройка двух синхронных реплик (db_sync_1_container, db_sync_2_container):

- поднимаю 2 контейнера с postgres и настраиваю синхронную реплику
`docker cp db_container:/pgslave postgresql/db_data_sync_1`
`primary_conninfo = 'host=db_container port=5432 user=replicator password=replicator_password application_name=db_sync_1_container'`
`touch postgresql/db_sync_1_data/standby.signal`
- Настраиваем кворумную репликацию. На мастере
`synchronous_commit = on`
`synchronous_standby_names = 'ANY 1 (db_sync_1_container, db_sync_2_container)'`
- Запускаем нагрузочные тест с запросами на запись в мастер
`/user/register`
- Убиваем контейнер с мастером
`docker compose kill db`
- Заканчиваем нагрузочное тестирование
- Промоутиим sync реплику до мастера
`select * from pg_promote();`
`synchronous_commit = on`
`synchronous_standby_names = 'ANY 1 (db_container, db_async_container)'`
- И переключаем на него вторую реплику (*host=)
`primary_conninfo = 'host=db_sync_container port=5432 user=replicator password=replicator_password application_name=db_async_container'`
- Проверяю репликацию для нового мастера

	application_name	sync_state
1	db_sync_1	quorum
2	db_sync_2	quorum
3	db_async	async

	application_name	sync_state
1	db_async	async
2	db_sync_2	quorum

- Проверяем есть ли потери транзакций:
Проверяю insert на sync_1 реплику, которая стала мастером. Запись успешно записалась, и дошла до всех реплик

Начальное количество записей в таблице: 0

Записей на момент остановки мастера:

Число успешных запросов: 373

Записей в sync_1: 373

Записей в sync_2: 373

(Записей в async: 373)