IT NonStop

DataArt

IT NonStop

DataArt

gRPC в продакшне

Минкин Андрей (Mad Devs)

Кто я?

7 лет — сисадмин

5 лет — Python

3+ — года Go

3+ — лет тимлид

О чем доклад?

Зачем нам вообще gRPC

О чем доклад?

Зачем нам вообще gRPC

Как мы докатились до такой жизни

О чем доклад?

Зачем нам вообще gRPC

Как мы докатились до такой жизни

Опыт эксплуатации

О чем доклад?

Зачем нам вообще gRPC

Как мы докатились до такой жизни

Опыт эксплуатации

Проблемы в продакшне и их решения

О чем доклад?

Зачем нам вообще gRPC

Как мы докатились до такой жизни

Опыт эксплуатации

Проблемы в продакшне и их решения

Выводы

С чего все начиналось?

Водительское приложение для службы такси

С чего все начиналось?

Водительское приложение для службы такси

Разработка, перешедшая от фрилансера

С чего все начиналось?

Водительское приложение для службы такси

Разработка, перешедшая от фрилансера

Огромный технический долг

С чего все начиналось?

Водительское приложение для службы такси

Разработка, перешедшая от фрилансера

Огромный технический долг

Невозможность поддержки

С чего все начиналось?

Водительское приложение для службы такси

Разработка, перешедшая от фрилансера

Огромный технический долг

Невозможность поддержки

Пройденная точка невозврата

С чего все начиналось?

Водительское приложение для службы такси

Разработка, перешедшая от фрилансера

Огромный технический долг

Невозможность поддержки

Пройденная точка невозврата

Множество хотелок у заказчика

Как работало приложение?

Историческая самописная сериализация данных

Свой протокол и куча костылей вокруг него

Poll based клиент-серверное приложение на TCP сокетах

Текущий стек

Python/Go

Percona/Redis/Elasticsearch

Minio/nginx

Docker

Что хочет заказчик?

Очереди заказов/отправка сервером заказов с разными фильтрами

Низкий time-to-market

Управление приложением со стороны сервера

Что хотели технари?

Design/API first подход

Рассылки с сервера

Единый протокол

Остаться в рамках текущего стека технологий

Вычистить исторический хлам

Смена протокола

Избавиться от текущих костылей(TCP+UDP)

Избавиться от костыльной сериализации

Убить последний питон на серверах (почти)

Что рассматривали?

MQTT

UDP+велосипеды(QUIC, uTP)

gRPC

REST

Websockets

Чем не подошли MQTT/QUIC/UDP?

Отсутствие design first подхода

Чем не подошли MQTT/QUIC/UDP

Отсутствие design first подхода

Нужно реализовывать свой тулинг для реализации первого пункта

Чем не подошли MQTT/QUIC/UDP

Отсутствие design first подхода

Нужно реализовывать свой тулинг для реализации первого пункта

Нужен свой брокер и реализовывать протокол

Чем не подошли MQTT/QUIC/UDP

Отсутствие design first подхода

Нужно реализовывать свой тулинг для реализации первого пункта

Нужен свой брокер и реализовывать протокол

UDP может резаться провайдерами

Чем не подошли MQTT/QUIC/UDP

Отсутствие design first подхода

Нужно реализовывать свой тулинг для реализации первого пункта

Нужен свой брокер и реализовывать протокол

UDP может резаться провайдерами

Pattern adapter+разный транспорт. Но зачем?

Чем не подошли websockets

Нестабильны на плохом интернете

Полностью асинхронны

Будет работать только в связке с REST

Чем не подошел REST

Нам ближе RPC, чем работа с ресурсами

Можно забить на документацию

Нужно специально менять процессы разработки

Пилить свои тесты/инструменты, чтобы все работало по канонам

Что такое gRPC

Protocol Buffers

Design First

HTTP/2

Двунаправленный стриминг

Мультиплексинг

HTTPS

Множество языков


C++


GO


node
JS

Почему мы выбрали gRPC?

Design First

Генерация кода

Относительно легковесно по трафику

HTTP/2 streaming+fallback

Документация и код

https://www.grpc.io

https://github.com/grpc

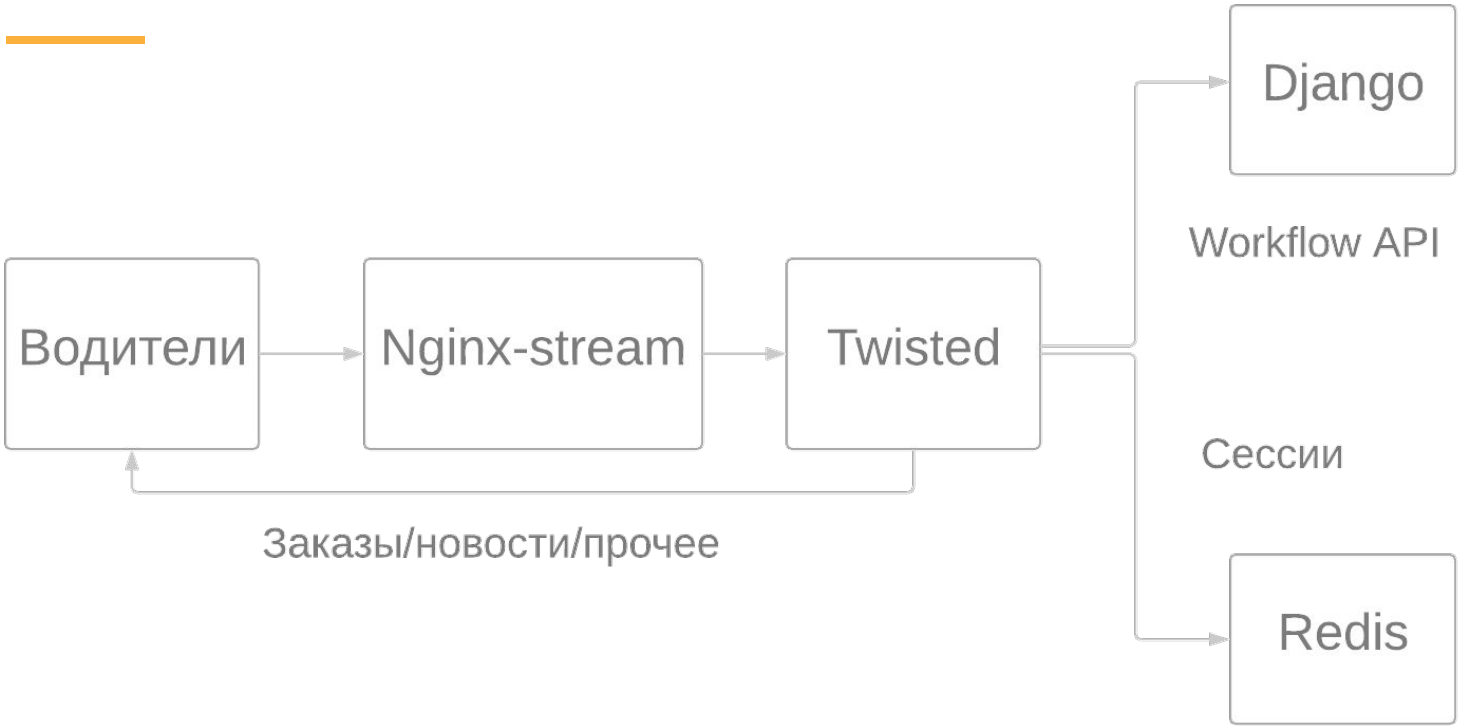
https://github.com/grpc-ecosystem

Помощь и поддержка

https://gitter.im/grpc/grpc

https://groups.google.com/forum/#!forum/grpc-io

Архитектура до



Первая реализация

HTTP API Client

Свой внутренний storage для хранения коннектов

Pub/sub на уровне приложения

Внедряем

Docker

12 factor app

Как балансировать?

Как балансировать gRPC

Со стороны клиента

Со стороны сервера

Nginx-stream + SSL в качестве балансера

Инструментирование и мониторинг

Prometheus

https://github.com/grpc-ecosystem/go-grpc-prometheus

sensu

Приложение не подключается к серверу

У нас пре-релиз, а коннекта нет

Клиент на Go - подключается без проблем

Обложили код логами

Читаем спеку http/2, доку и ченжлоги nginx/android

Не совместимые протоколы

Android:

+ALPN

-NPN

NGINX:

-ALPN

+NPN

NGHTTPx решает нашу проблему

В ЗБТ приложение теряет связь

Рандомно отваливается стрим

Приложение не переподключается

Почему?

Краши сервера

Нет явного статуса стрима, но есть статус канала

Оператор связи рубит пустые http соединения(o’ rly?)

Трассировка и отладка

http://charithe.github.io/tracing-grpc-services-in-go.html

net/http/pprof

Вторая реализация сервера

Избавились от самописного storage

pub/sub с уровня приложения на каждого клиента

Масштабирование

Рефакторинг дал более удобный механизм для масштабирования

Нет состояния - нет проблем

Работаем с падающими стримами

Circuit breaker для упавших стримов

+keepalive

Keepalive на мобильных операторах

Обычно рубят NAT сесии через 1-5 минут

Выставили 20 секунд для приложений

waitForReady

https://github.com/grpc/grpc/blob/master/doc/wait-for-ready.md

Ждет READY состояния

Падет на всех других (SHUTDOWN, Deadline exceeded…)

Плохая связь

Fallback на unary

Circuit breaker

Observer на клиенте со счетчиками

- Ошибка или таймаут

Перезапуск через N секунд в случае достижения maxRetries

Как балансировать на сервере

Nginx socket -нет ALPN

Встроенные механизмы - клиент наружу

Proxy_pass в NGINX - все плохо с поддержкой HTTP/2

NGHTTPX - ликает

HAPROXY

NGHTTPX vs HAProxy

Есть опыт с HAProxy

Поддержка H2

Тесты ОК

Полет нормальный

А еще?

Nginx 1.13.10+

Træfik

Istio

envoy

Выводы

gRPC можно использовать вместо REST и это удобно

gRPC применим и в мобильных приложениях (с некоторыми оговорками)

Балансировать на сервере удобно через HAPRoxy

waitForReady может как помочь, так и усложнить жизнь

Нет состояния - нет проблем

Контакты

@maddevsio

https://maddevs.io

@Gen1us2k