Live Streaming CDN (WebRTC/HLS/RTMP) глазами DevOps

Александр Строгонов
Streaming DevOps инженер
Mayflower

О себе

10+ лет в IT

Прошёл путь от эникейщика до разработчика на Python и DevOps инженера.

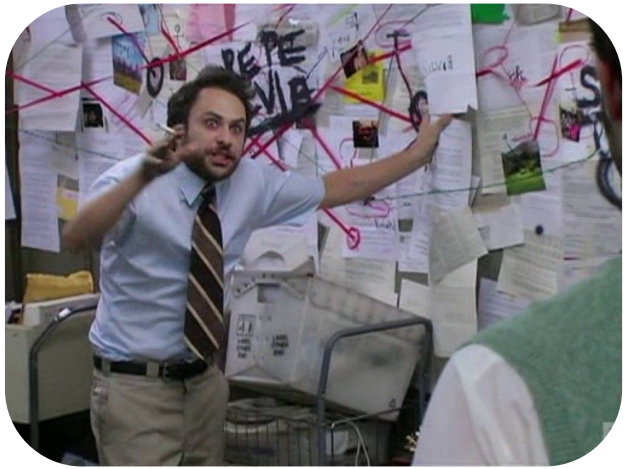
3+ года в стриминге

Занимаюсь SRE, инфраструктурой, архитектурой и дежурствами

О чём

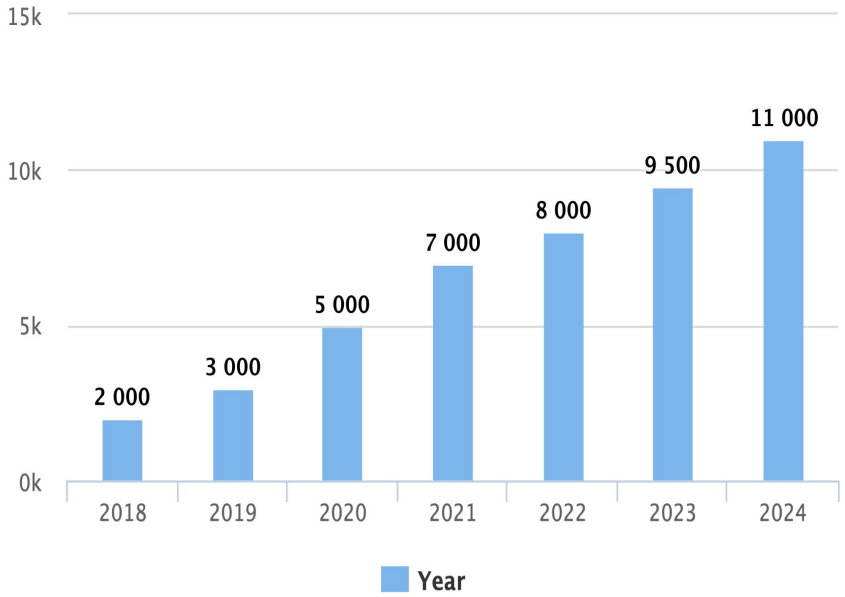
- Зачем мы начали строить свой CDN
- Построение CDN (итерация первая)
- In-house решение (итерация вторая)
- Box solution (итерация третья)
- Итоги

Зачем мы начали строить свой CDN



Live Streaming Platform

- 10K+ стримов
- 1 млн пользователей в пике
- 12 Пбайт/день — HLS трафик
- 1.5 Пбайт/день — WebRTC трафик



SaaS vs On-premise

В поиске идеального стриминг решения

В поиске идеального стриминг решения: SaaS

Сколько стоит WebRTC трафик в расчёте на 1000 стримов длительностью 8 часов при 2000 вьюверов?

$ 700K в среднем

Platform	**Cost per month, $**
AWS	878.559
Wowza	599.000
Cloudflare	1.000.000
Gcore	844.800

В поиске идеального стриминг решения: On-Premise

- Стоимость лицензии
- Снижение затрат на масштабирование
- Оптимизация использования сетевых ресурсов
- Контроль над инфраструктурой и данными

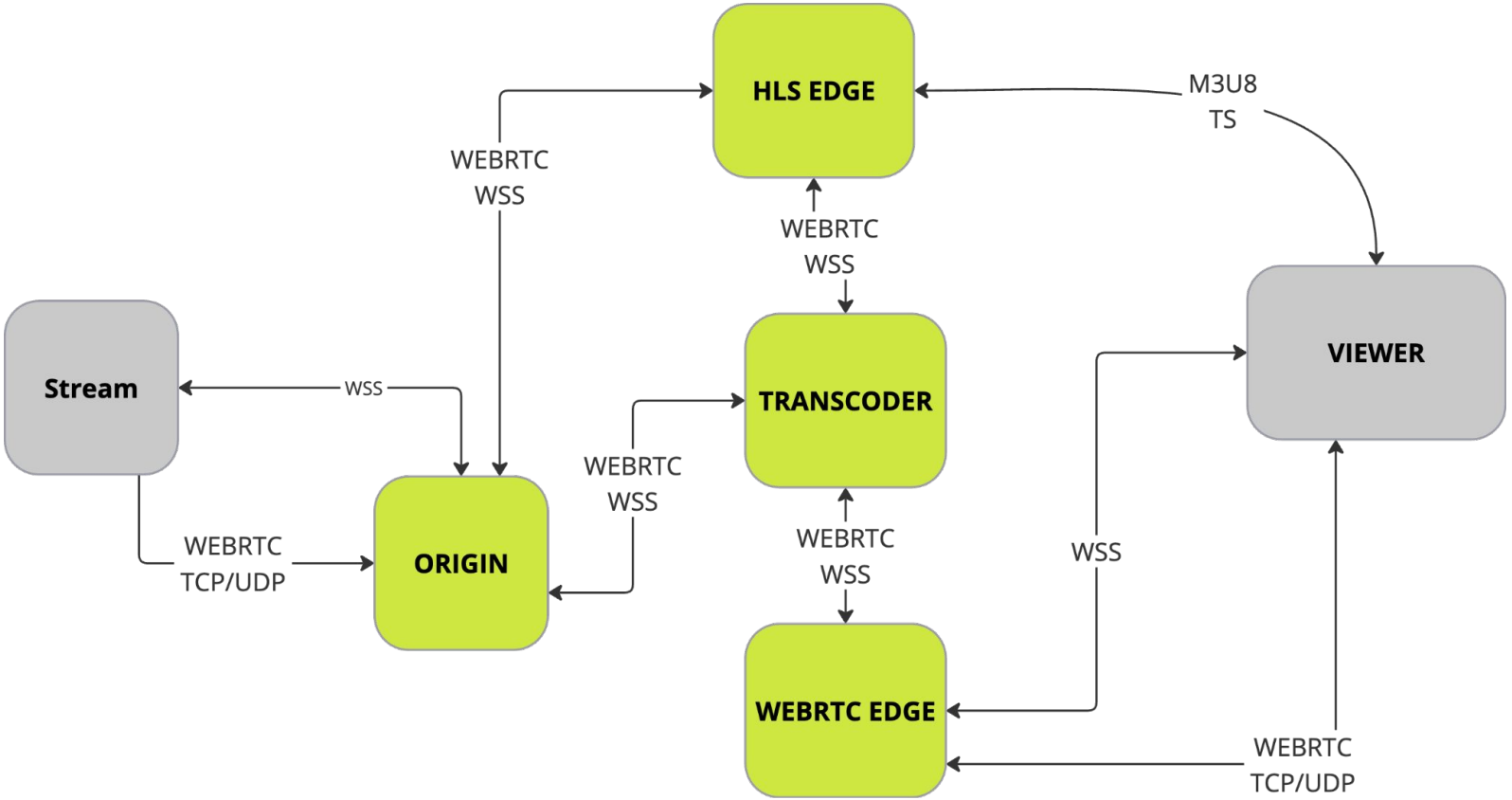
Platform	**Cost per month, $**
Wowza	200.000
Flashphoner	20.833
Nimble Streamer	5.000

В поиске идеального стриминг решения

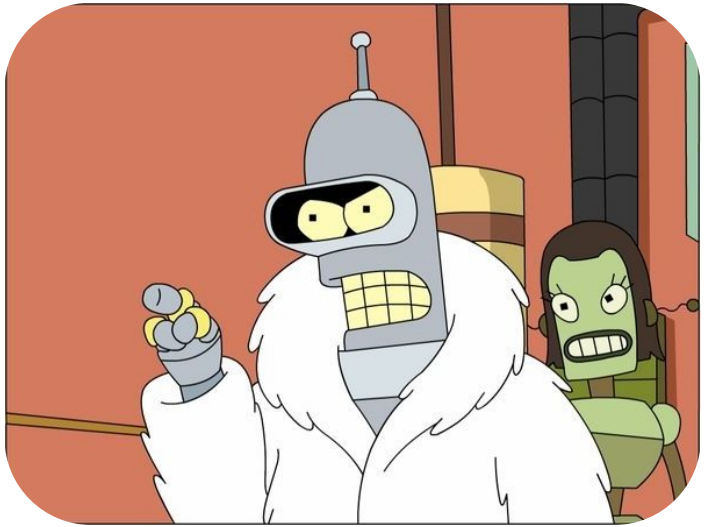
- Написан на java
- Коробочное решение
- Бета-тест

Flashphoner

Архитектура



Строим свой CDN



Инфраструктура

- Dell R840/4xGold/128GB;
- CentOS 7
- Ansible
- GitLab CI/CD
- Victoria Metrics
- OpenObserve

Почему Bare metal

Аренда

- servers.com **$15.816/год**
  2 x Intel Xeon Gold 5220

- OVH **24.000$/год**
  2 x Intel Xeon Gold 6226

+ трафик

Покупка

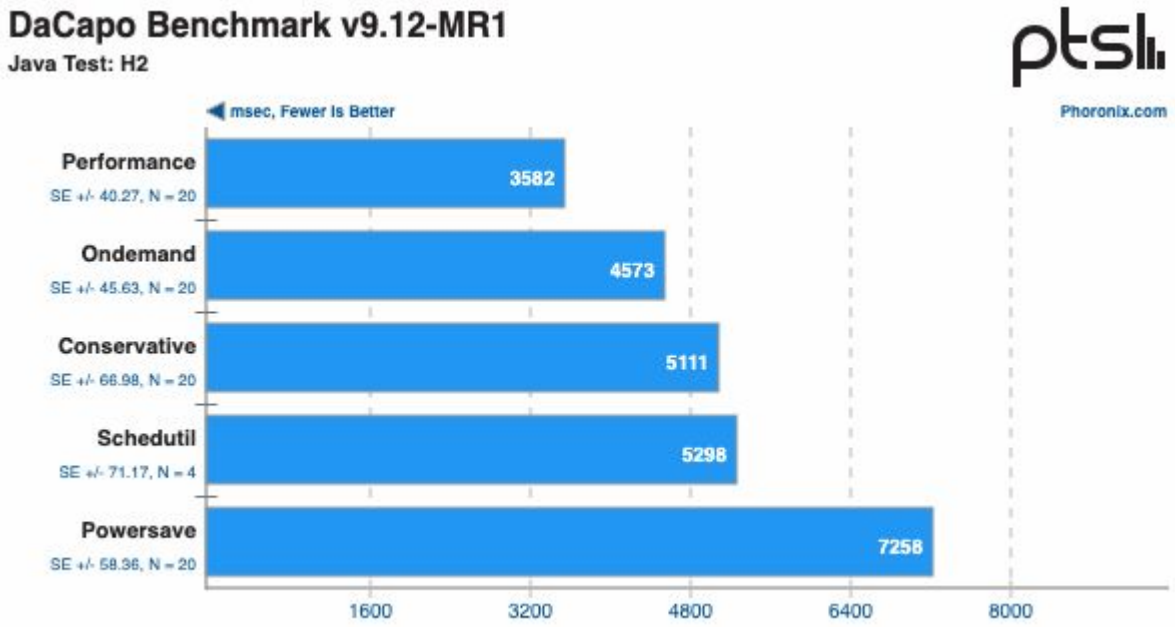
Dell — **$20.000**
4 x Intel Xeon Gold 6252

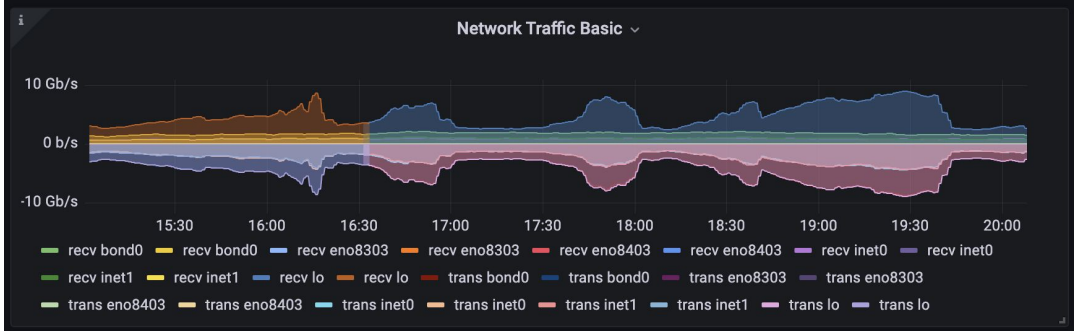
Почему Bare metal

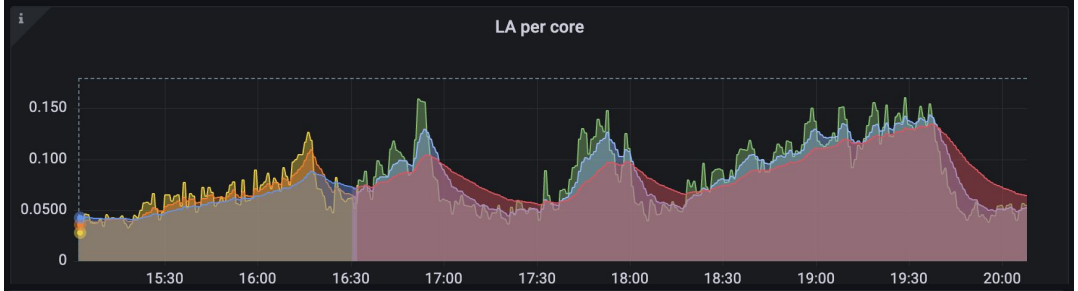
- 🍪 Полное управление сервером
- 🍪 BIOS (C/P-state, SMT, numa nodes)
- 🍪 Кастомизация железа (SFP, GPU, etc)
- 🍪 Нет накладных расходов на виртуализацию

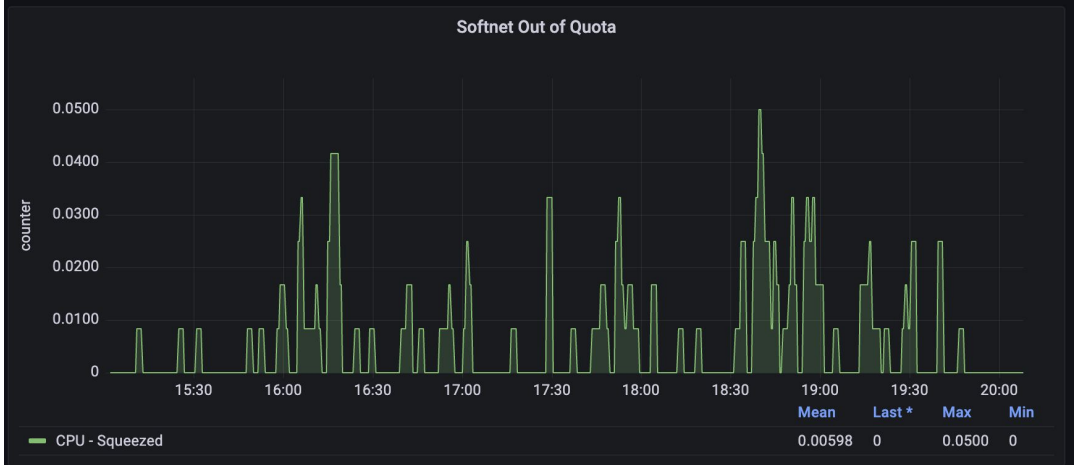
Оптимизация: PowerPolicy

- cpu_governor powersave/ondemand -> performance
- x86_energy_perf_policy -> performance




Network Traffic Basic
10 Gb/s
0 b/s
-10 Gb/s
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00
19:30
20:00
recv bond0
recv bond0
recv eno8303
recv eno8303
recv eno8403
recv eno8403
recv inet0
recv inet0
recv inet1
recv inet1
recv lo
recv lo
trans bond0
trans bond0
trans eno8303
trans eno8303
trans eno8403
trans eno8403
trans inet0
trans inet0
trans inet1
trans inet1
trans lo
trans lo


LA per core
0.150
0.100
0.0500
0
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00
19:30
20:00

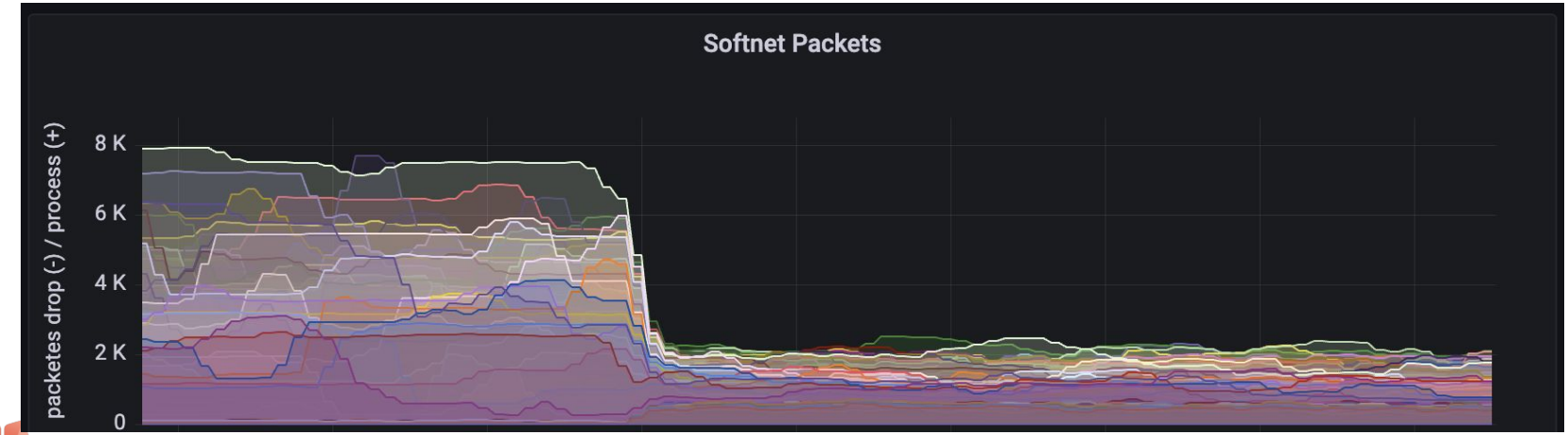

Softnet Out of Quota
counter
0.0500
0.0400
0.0300
0.0200
0.0100
0
15:30
16:00
16:30
17:00
17:30
18:00
18:30
19:00
19:30
20:00
Mean
Last *
Max
Min
CPU - Squeezed
0.00598
0
0.0500
0

Оптимизации: ethtool

- При сильном росте трафика может случится bottleneck на сетевых интерфейсах, что приведёт к dropped frames
- Максимально увеличиваем RX/TX ring buffers
- ethtool -G {eth} tx {MAX_BUFFER} rx {MAX_BUFFER}

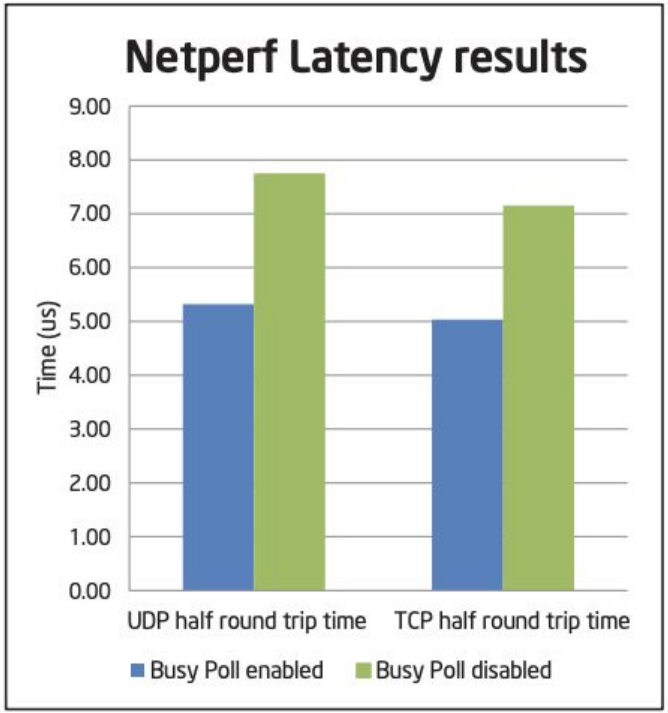
Оптимизации: sysctl busy polling

- Выключаем adaptive interrupt throttle rate (ITR), чтобы уменьшить нагрузку на CPU.
- Ставим rx-usecs и tx-usecs на 80 микросекунд — это ограничивает количество прерываний к 12,500 в секунду на очередь.
- ethtool -C <ethX> adaptive-rx off adaptive-tx off rx-usecs 80 tx-usecs 80



Оптимизации: sysctl busy polling

- Busy polling помогает сократить задержку, позволяя сокетам опрашивать очередь eth rx и отключает сетевые прерывания.
- Возрастает нагрузка на CPU
- sysctl.net.core.busy_poll = 50
- sysctl.net.core.busy_read = 50



Оптимизации: остальные крутилки

- Увеличение буферов для tcp/udp
- Уменьшения таймаутов для syn, retry, fin
- Java heap, Java GC, etc настроены согласно документации.

Не дали ощутимых изменений:

- tcp_congestion_control
- tcp_(timestamps|no_metrics_save|slow_start_after_idle)

Производительность

При LA 0.8 на ядро мы получили:

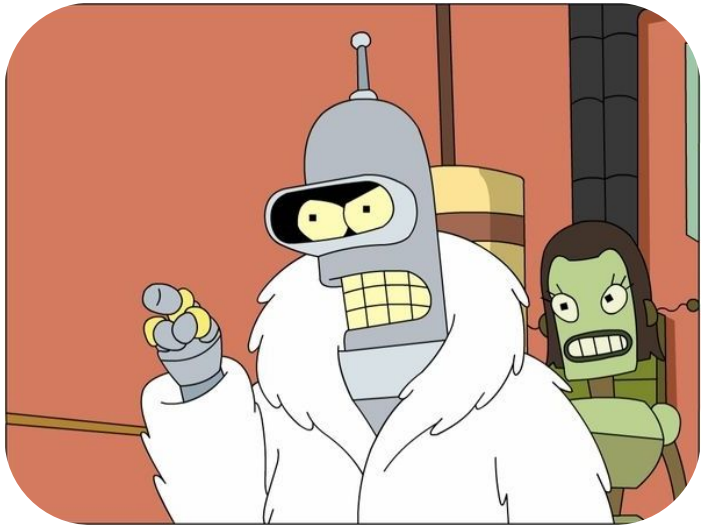
- 45 стримов на software encoder 720p30;
- 15 стримов на software encoder 1080p30.

Source	Transcoded	Transcoded	Transcoded	Transcoded
1080p30	720p	480p	240p	160p
720p30	480p	240p	160p	

Проблемы

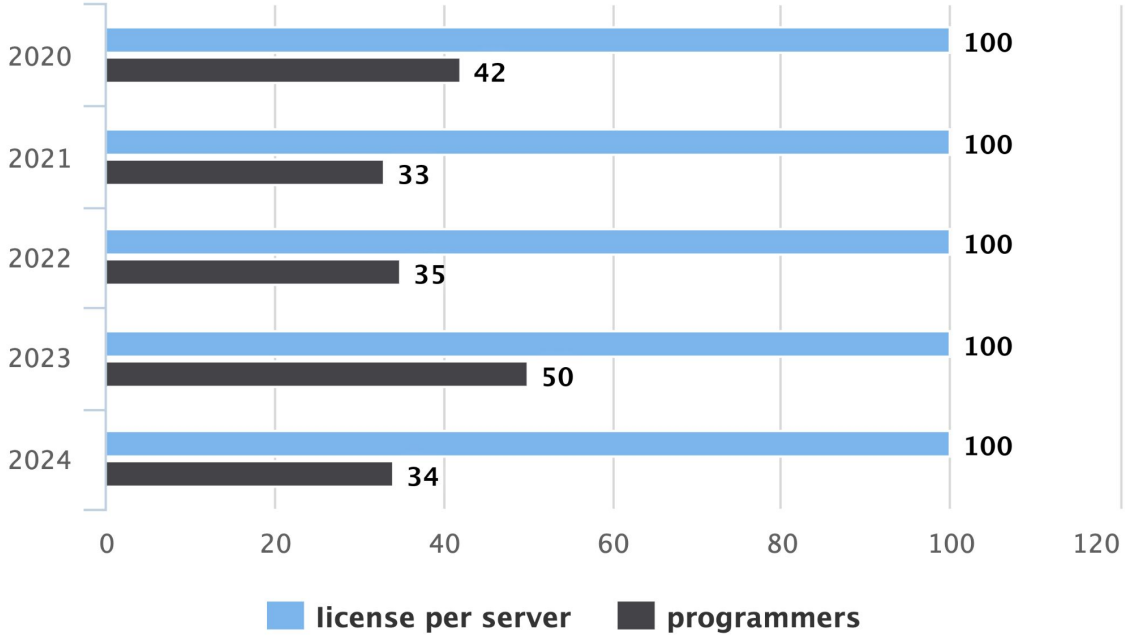
- Бета-тест
- Фиксы и новые фичи доставляются в продукт всё медленнее
- Проблема с Java (GC, stall’ы)
- Приходится горизонтально масштабироваться
- Стоимость

In-house решение



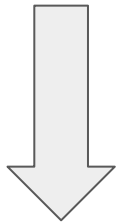
Причины перехода к своей разработке

Во-первых, это ~~красиво~~ дешевле



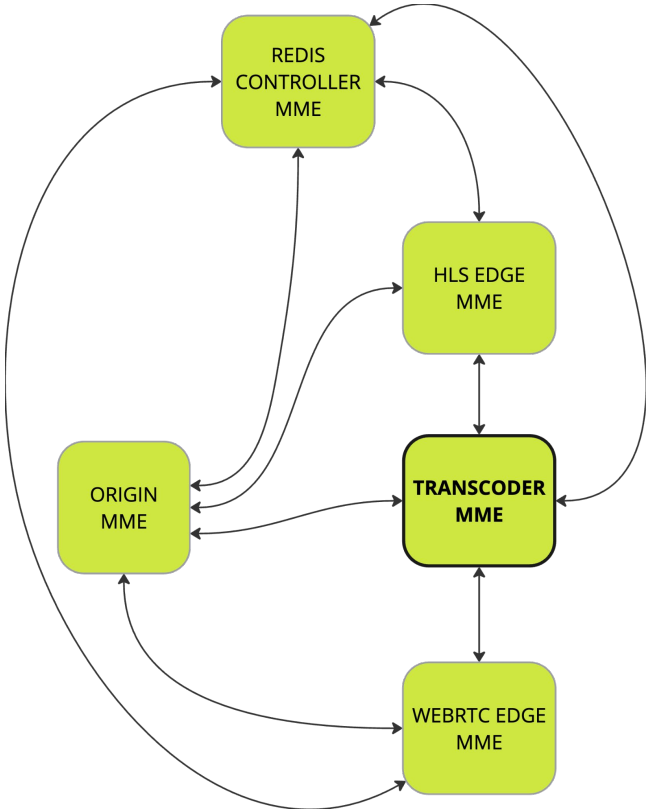
Причины перехода к своей разработке

+ Быстрее реализуются фичи и устраняются баги
+ Тесная интеграция команды стриминга и платформы



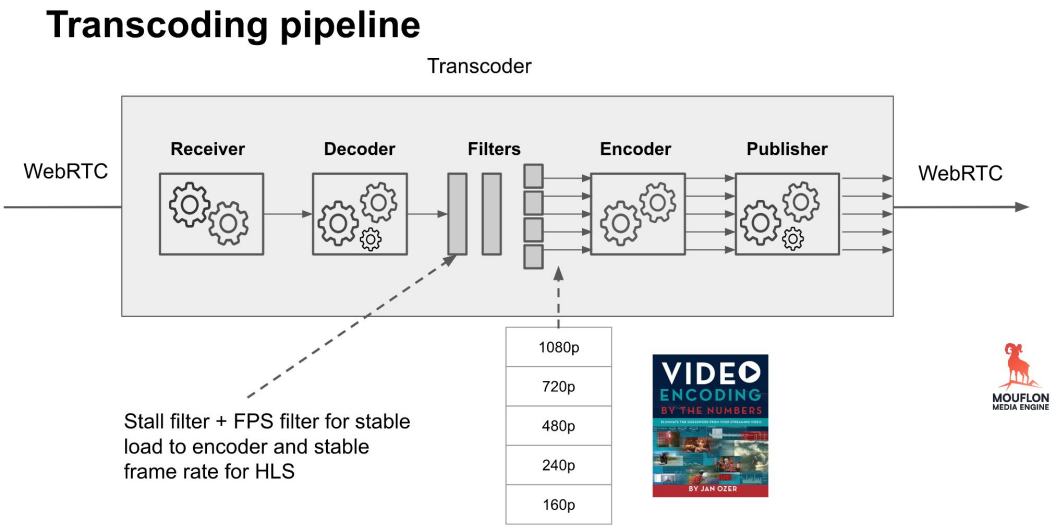
Появляется команда разработчиков на C++

Транскодинг



Транскодинг

- Самый высоконагруженный компонент
- Схема общения и API между компонентами такая же, для совместимости
- Написан отдельный сервис controller, занимающийся балансировкой стримов



Транскодинг

- 15 стримов 1080p30 Flashphoner
- LA 0.8

Type	**Card stream density**	**Server stream density (1080p30)**	**Number of servers**
Software	0	15	666.6(6)

Транскодинг

- 20 стримов 1080p30 in-house
- LA 0.8

Type	**Card stream density**	**Server stream density (1080p30)**	**Number of servers**
Software	0	15	666.6(6)
Software	0	20	500

Транскодинг

- 16 стримов 1080p30 in-house (nvidia)
- Dell r940xa на 6 pci-e слотов
- 150 стримов на Dell r940xa

Type	**Card stream density**	**Server stream density (1080p30)**	**Number of servers**
Software	0	15	666.6(6)
Software	0	20	500
NVIDIA T4 (6 pci-e)	16	96	105

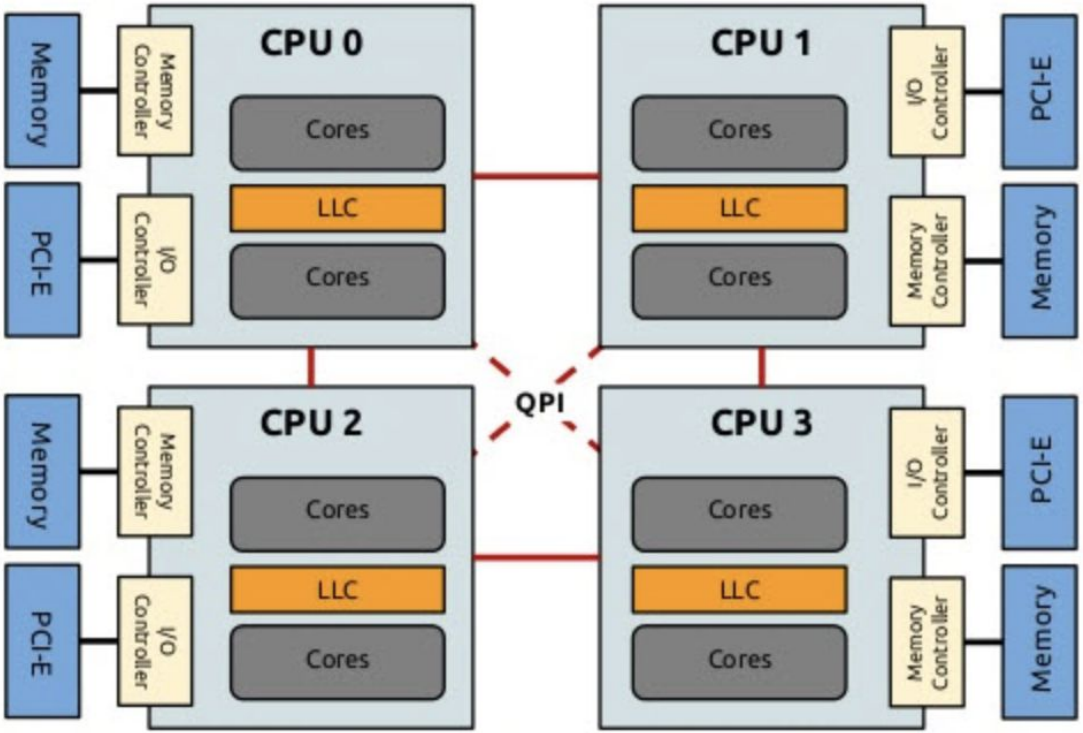
Транскодинг

- 25 стримов 1080p30 in-house (asic)
- 150 стримов 6 pci-e на Dell r940xa
- LA 0.8 -> 0.3
- стоимость одного стрима упала с $1000 до $197

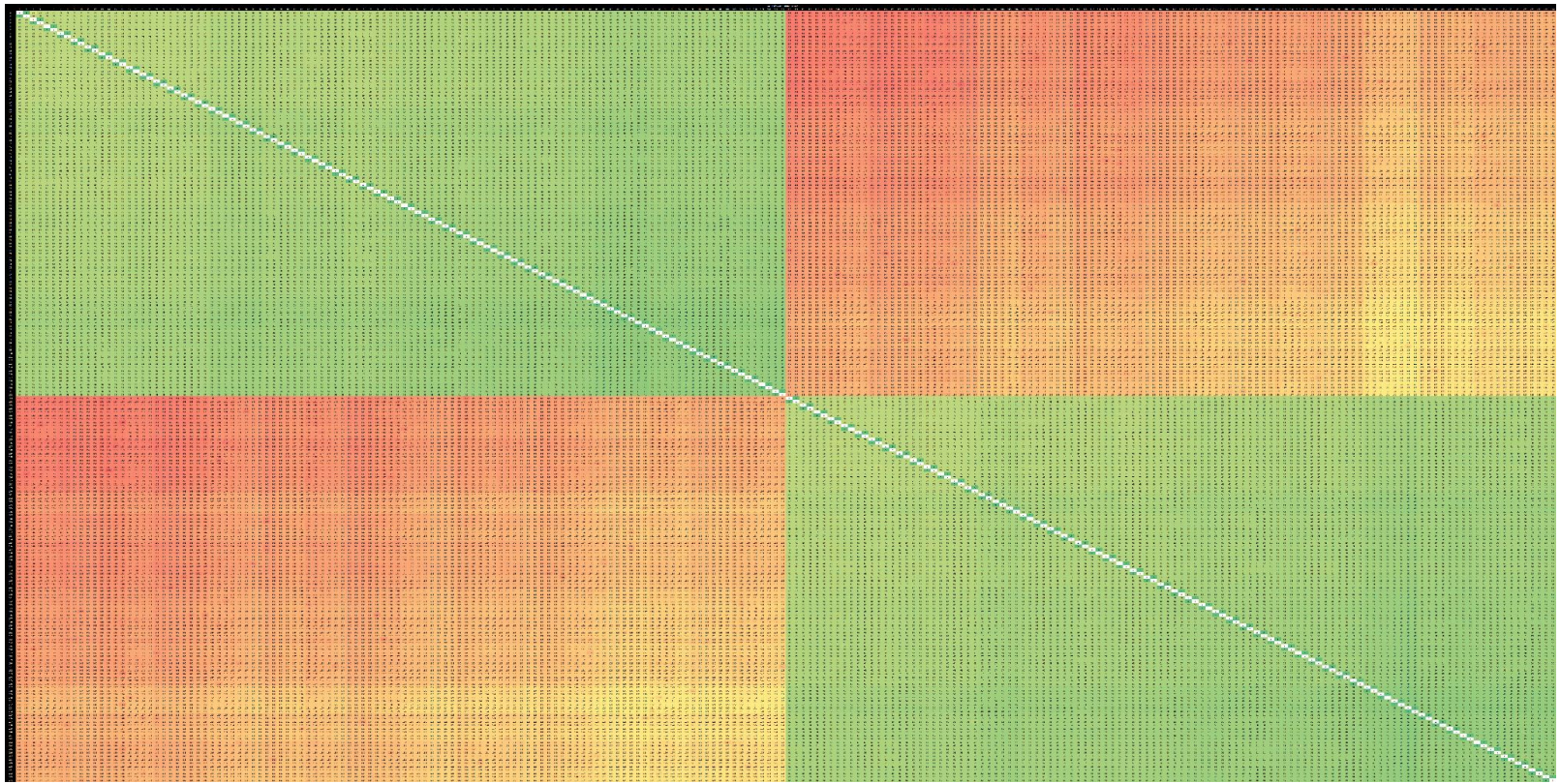
Type	**Card stream density**	**Server stream density (1080p30)**	**Number of servers**
Software	0	15	666.6(6)
Software	0	20	500
NVIDIA T4 (6 pci-e)	16	96	105
ASIC (6 pci-e)	25	150	67

Транскодинг: Numa

- numactl
- taskset



Транскодинг: Numa

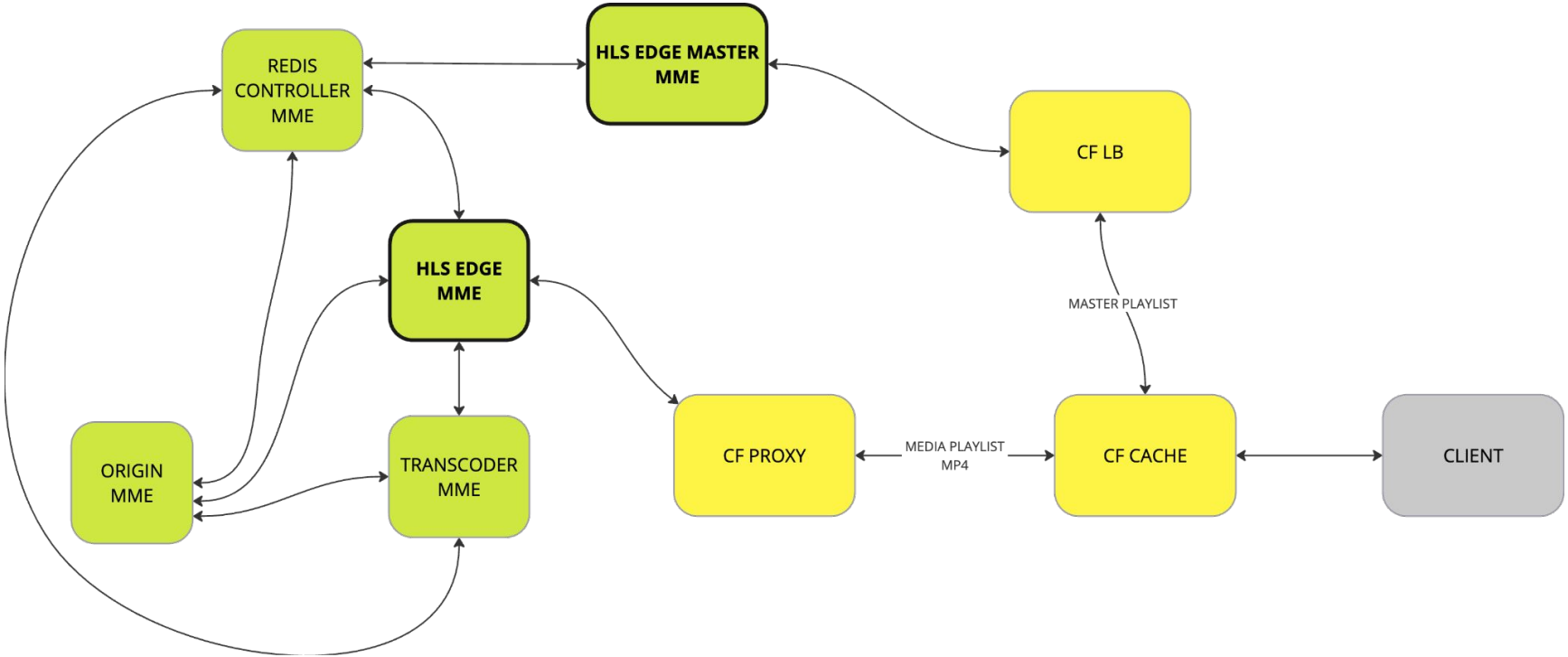


Транскодинг

- существуют шасси с большим кол-вом pci-e слотов

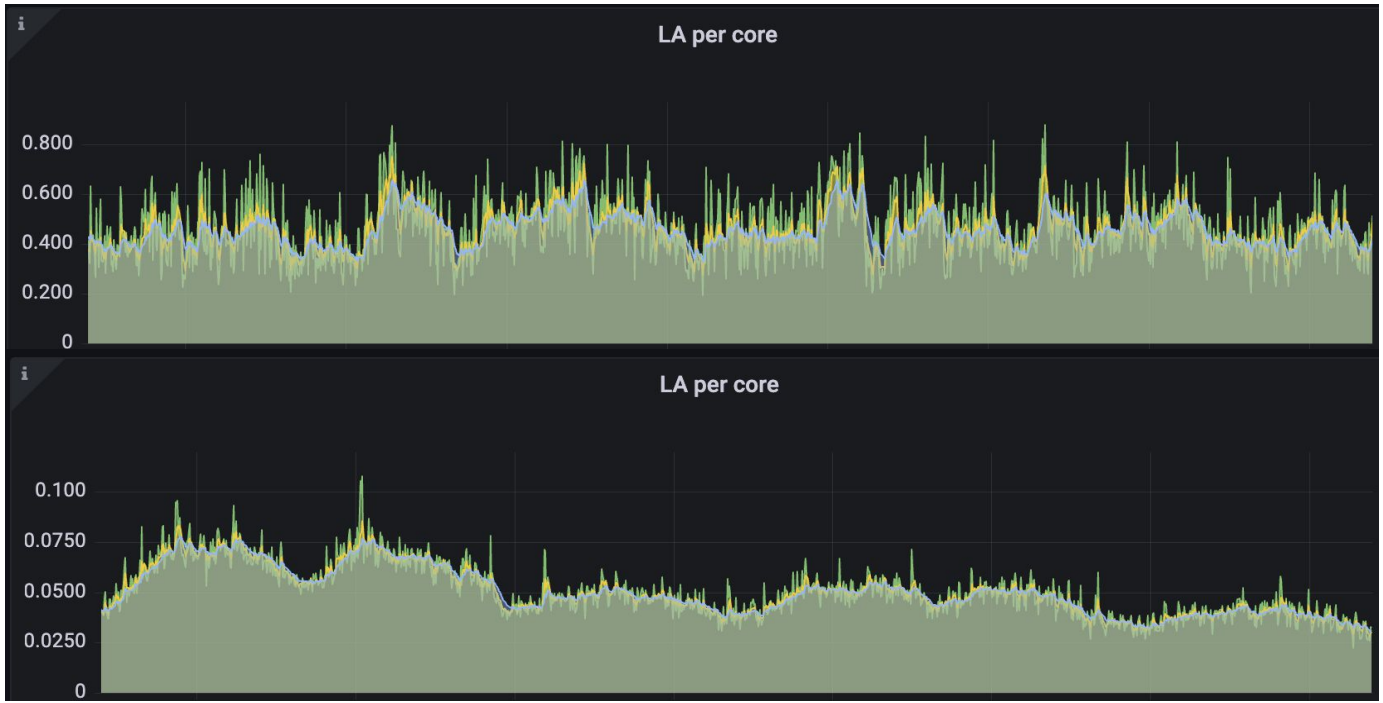
Type	**Card stream density**	**Server stream density (1080p30)**	**Number of servers**
Software	0	15	666.6(6)
Software	0	20	500
NVIDIA T4 (6 pci-e)	16	96	105
ASIC (6 pci-e)	25	150	67
ASIC (16 pci-e)	25	400	25

HLS-EDGE


REDIS CONTROLLER MME
HLS EDGE MASTER MME
CF LB
HLS EDGE MME
MASTER PLAYLIST
ORIGIN MME
TRANSCODER MME
CF PROXY
MEDIA PLAYLIST
MP4
CF CACHE
CLIENT

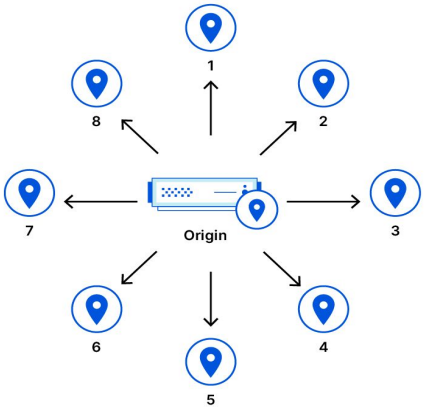
HLS-EDGE

Уменьшение LA 0.7 -> 0.08

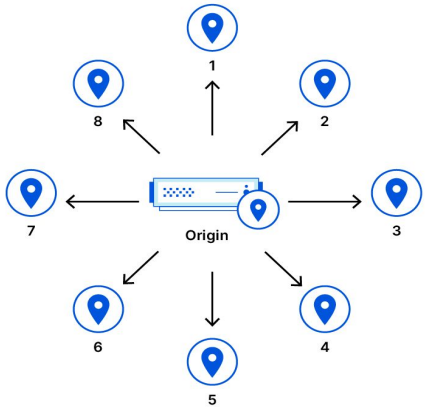


HLS-EDGE: Tiered Cache (Origin Shield)

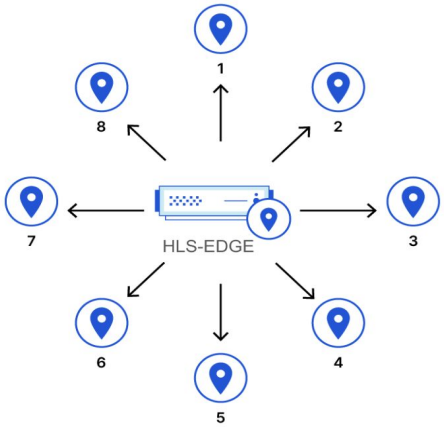
- Каждый Edge ExtCDN отправляет запрос на наши сервера
- Нагрузка умножается на количество ExtCDN провайдеров



ExtCDN №1



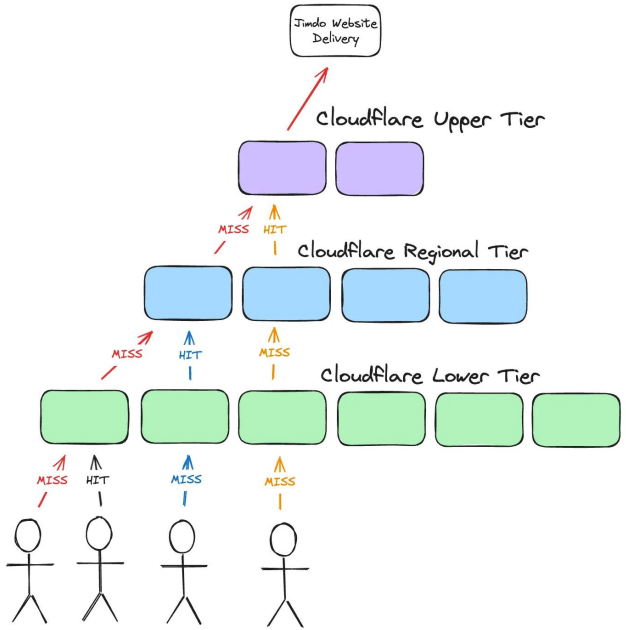
ExtCDN №2



ExtCDN №X

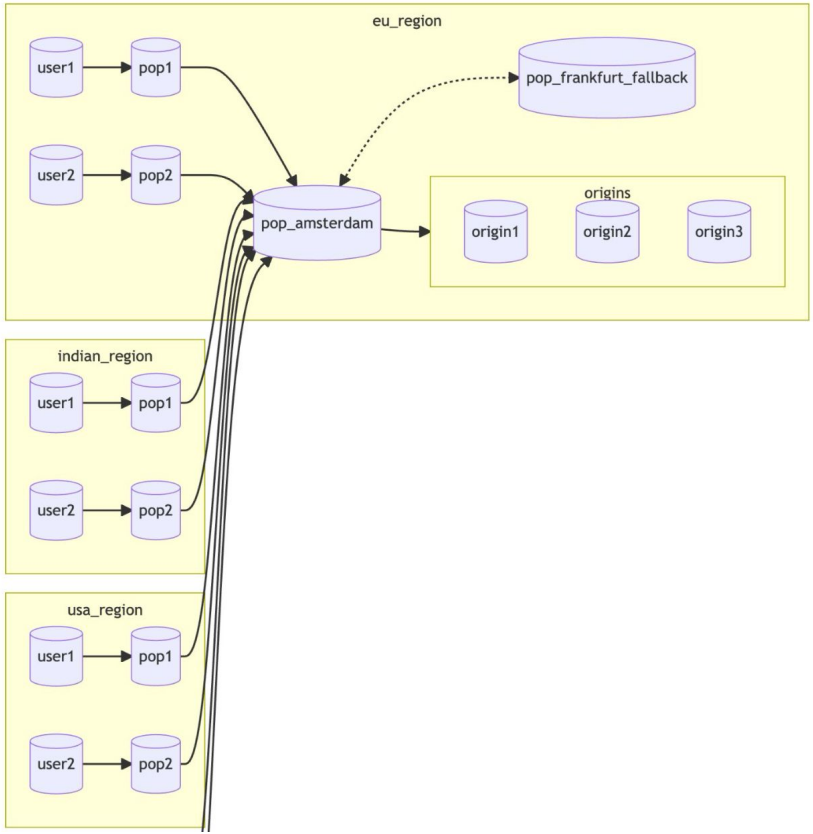
HLS-EDGE: Tiered Cache (Origin Shield)

Разные виды уровней кеша ExtCDN



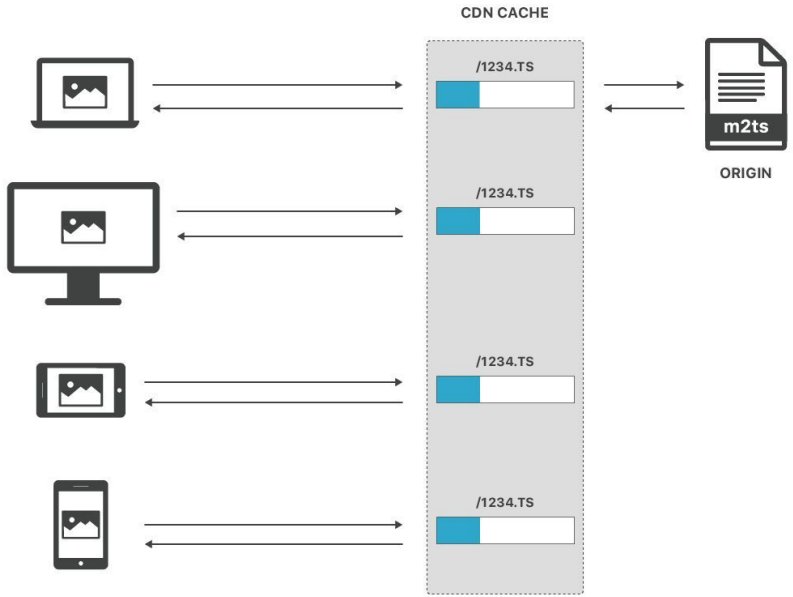
HLS-EDGE: Tiered Cache (Origin Shield)

- Два уровня кеша ExtCDN
- Мастер ExtCDN Edge (Origin PoP)
- Региональный ExtCDN Edge



HLS-EDGE: ExtCDN Concurrent Cache

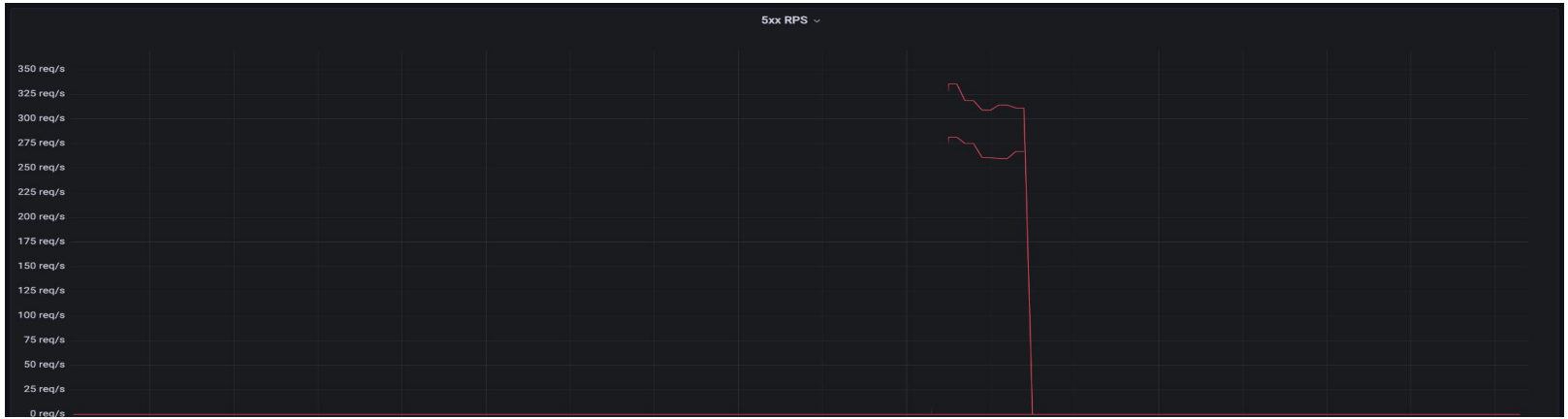
ExtCDN при отсутствии файла в cache выполняется только один запрос



HLS-EDGE: ExtCDN Concurrent Cache

Не работает:

- превышен таймаут.
- код ответа 5xx.
- «вымылся» кеш из PoP.
- добавили ещё одного ExtCDN провайдера.



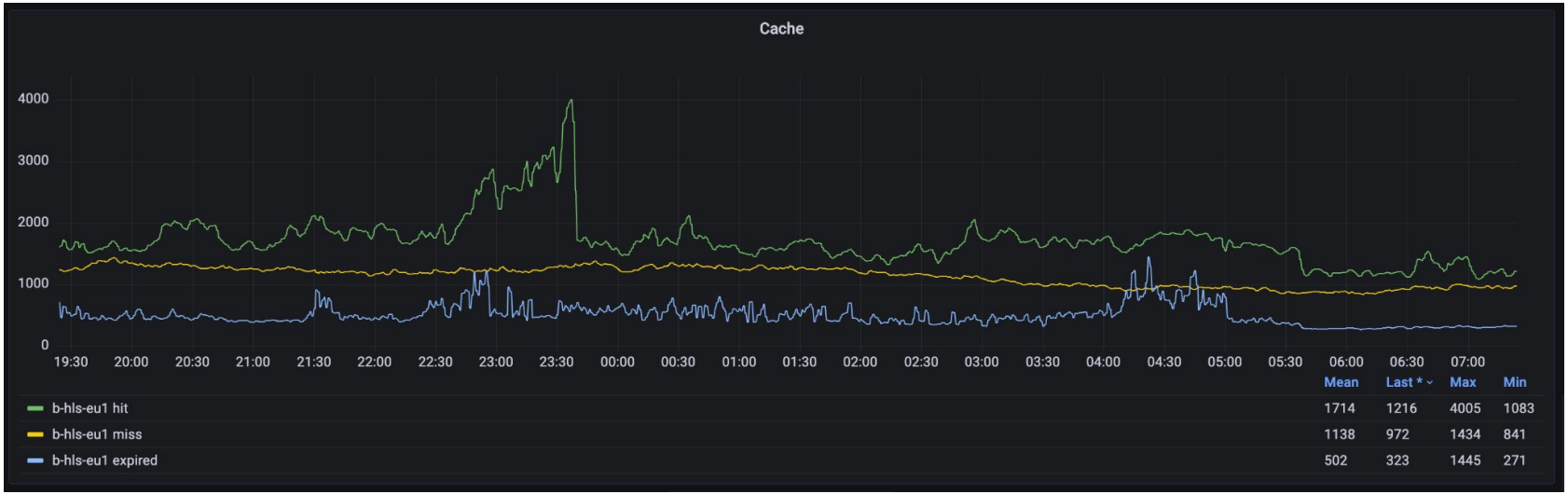
HLS-EDGE: Nginx Cache

- sub_filter domain.local $map_host_tld

```
#EXTM3U
#EXT-X-VERSION:6
#EXT-X-TARGETDURATION:2
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MEDIA-SEQUENCE:320
#EXT-X-DISCONTINUITY-SEQUENCE:0
#EXT-X-MAP:URI="https://                    .local/hls/108784200/108784200_init_FeyPpN6IoTk2vXxE.mp4"
#EXT-X-PROGRAM-DATE-TIME:2024-02-16T21:26:25.743+0000
#EXTINF:2.003
https://                   .local/hls/108784200/108784200_320_SVxjn0q5999N04aa_1708118779.mp4
#EXT-X-PROGRAM-DATE-TIME:2024-02-16T21:26:21.783+0000
#EXTINF:2.002
https://                   .local/hls/108784200/108784200_321_QpJwOJ7oD1aIGWnx_1708118781.mp4
#EXT-X-PROGRAM-DATE-TIME:2024-02-16T21:26:23.763+0000
#EXTINF:2.002
https://                   .local/hls/108784200/108784200_322_yOkLmfbH7gOgALhH_1708118783.mp4
```

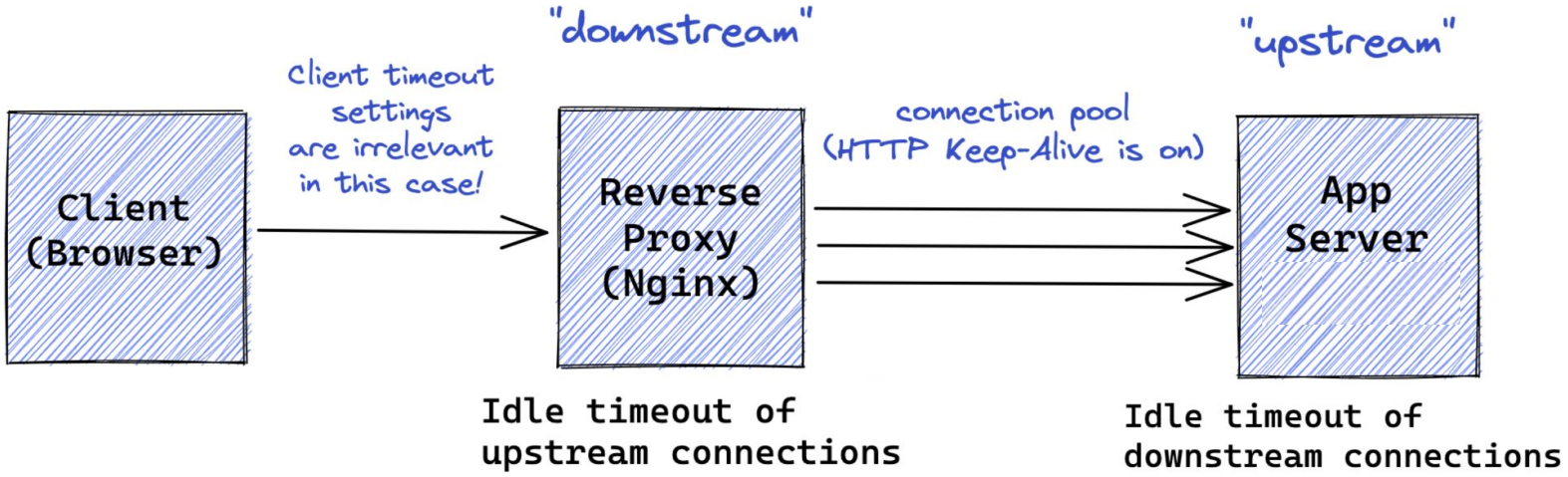
HLS-EDGE: Nginx Cache

- proxy_cache



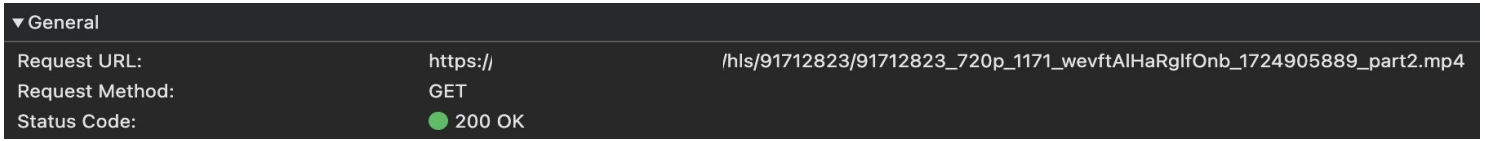
HLS-EDGE: Nginx

- nginx try_files стоит намного дешевле, чем proxy_pass
- сохранение чанков на tmpfs



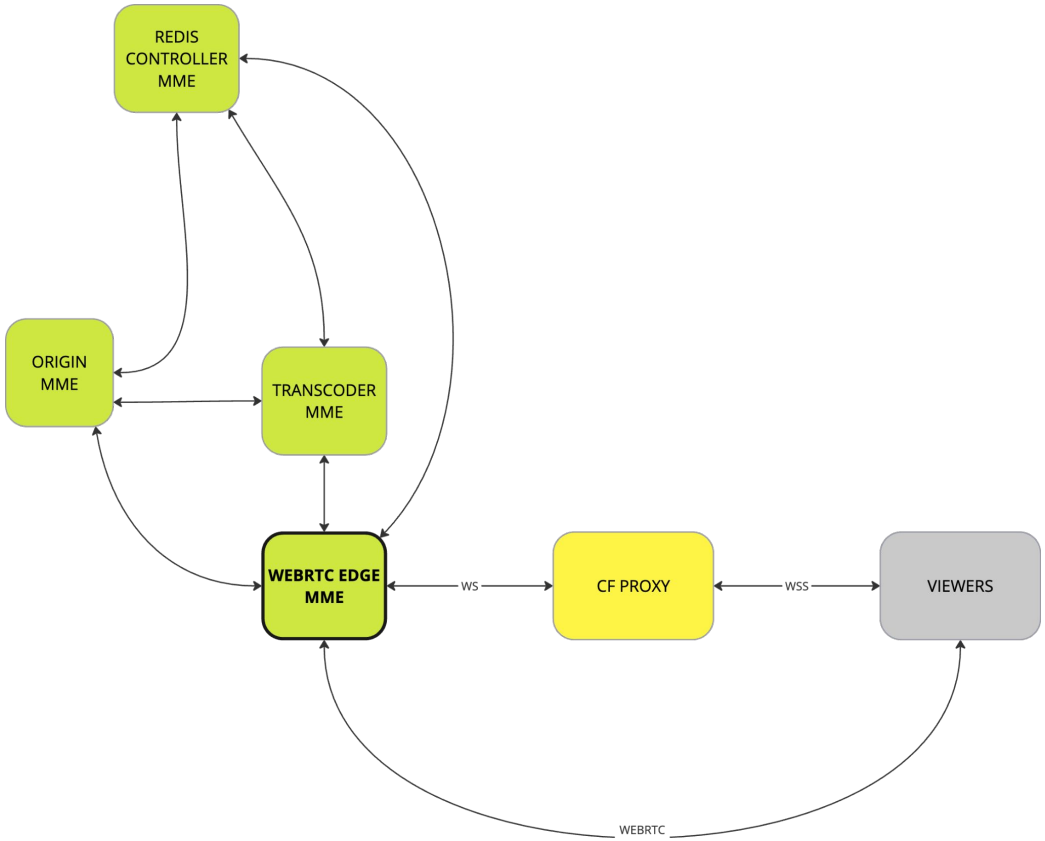
HLS-EDGE: Nginx

- nginx njs (ограничение отдачи чанков по timestamp)



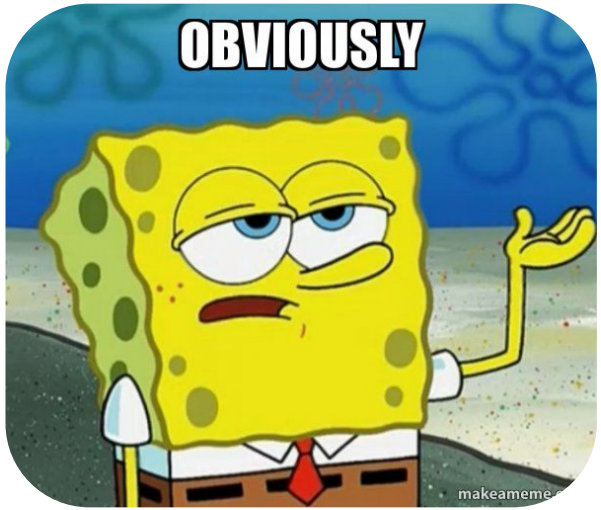

```
const lines = /_(\d{10})(?:_(?:part\d+))?\.mp4/.exec(r.uri);
if (!lines || !lines[1]) {
    return 0;
}
const max_ts_diff = 30; // 30s
const date_now = Math.floor(Date.now() / 1000);
const date_diff = date_now - parseInt(lines[1]);

if (date_diff > max_ts_diff || date_diff < -5) {
    timestamp = 0;
}
```

WebRTC-EDGE


REDIS
CONTROLLER
MME
ORIGIN
MME
TRANSCODER
MME
WEBRTC EDGE
MME
WS
CF PROXY
WSS
VIEWERS
WEBRTC

WebRTC-EDGE

Уменьшение LA 0.6 -> 0.2

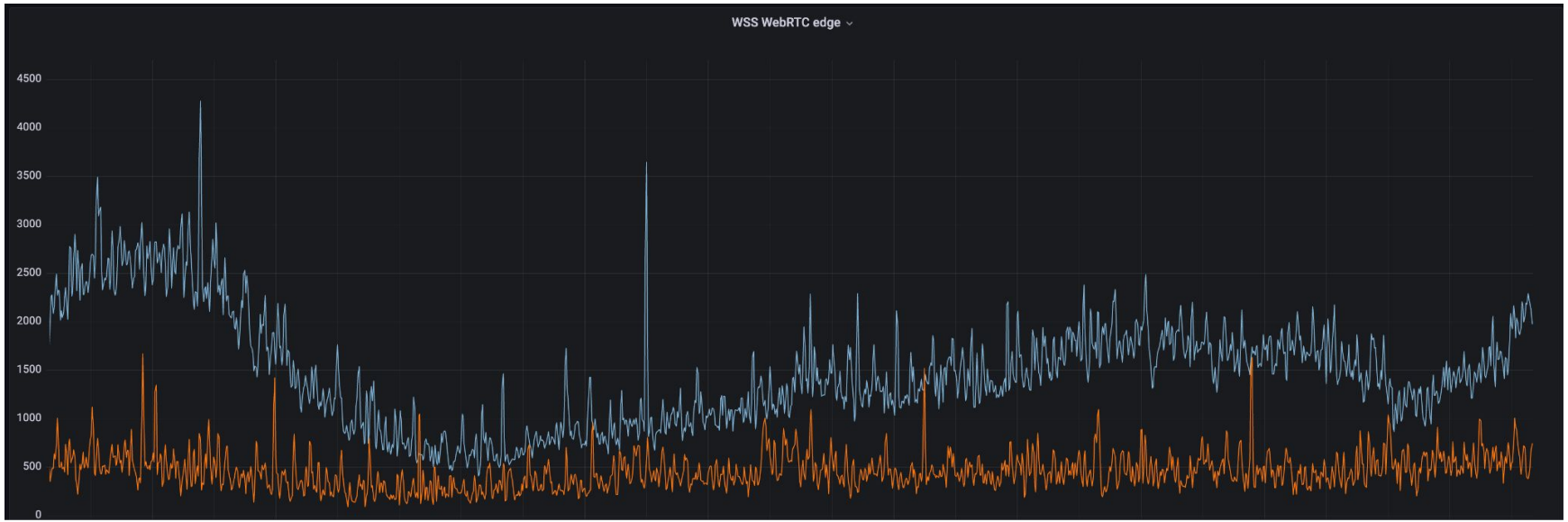


WebRTC-EDGE

- Сигналинг через WSS
- Некачественная связь вызывает реконнект у пользователя

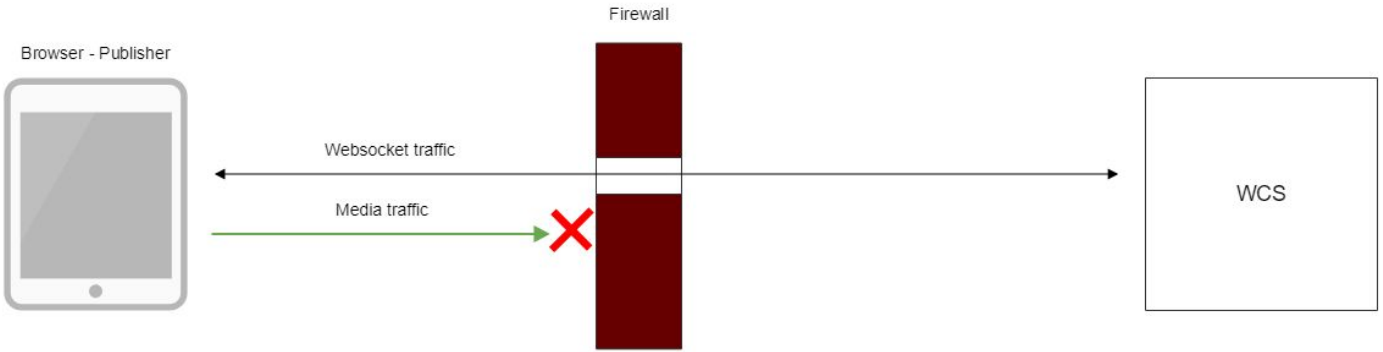
WebRTC-EDGE: Proxied WSS

- WSS с помощью проксирования средствами ExtCDN



WebRTC-EDGE

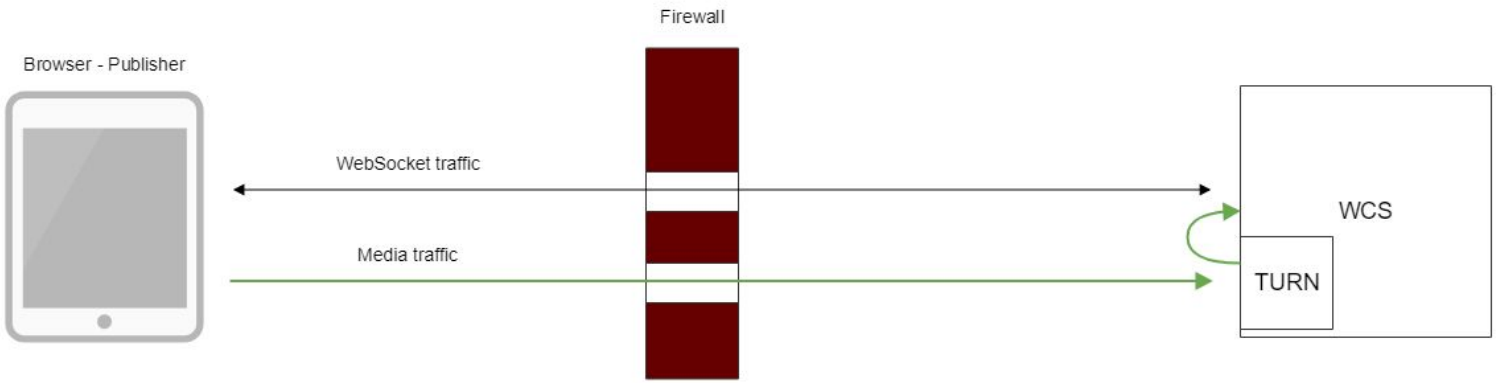
У пользователя открыты только порты 80,443



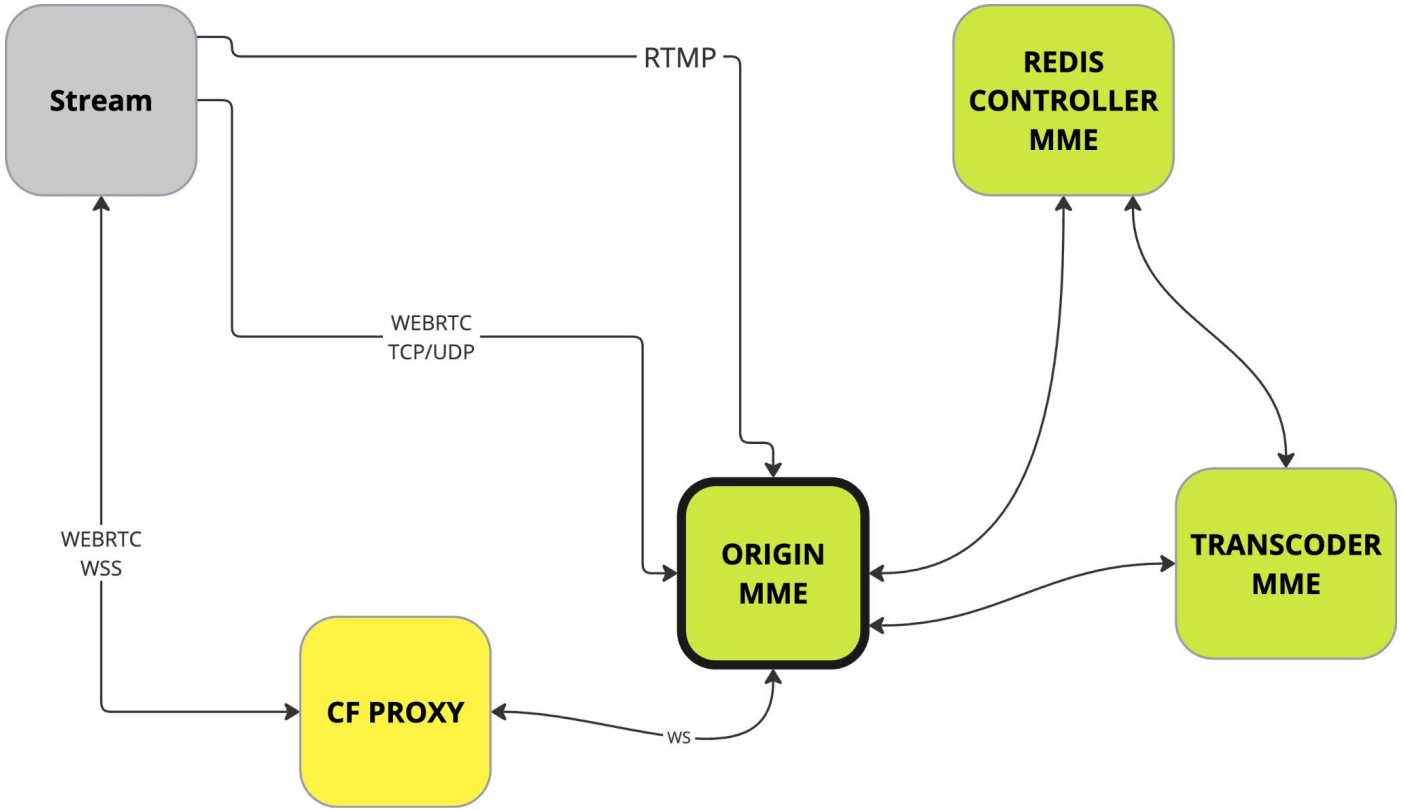
WebRTC-EDGE: TURN

- turn listen wan_ip:80 udp
- turn listen relay_addr 127.0.0.x:80 tcp
- nginx tcp stream wan_ip:80 tcp

["turn:turn.domain.local:80?transport=udp",
"turn:turn.domain.local:80?transport=tcp"]

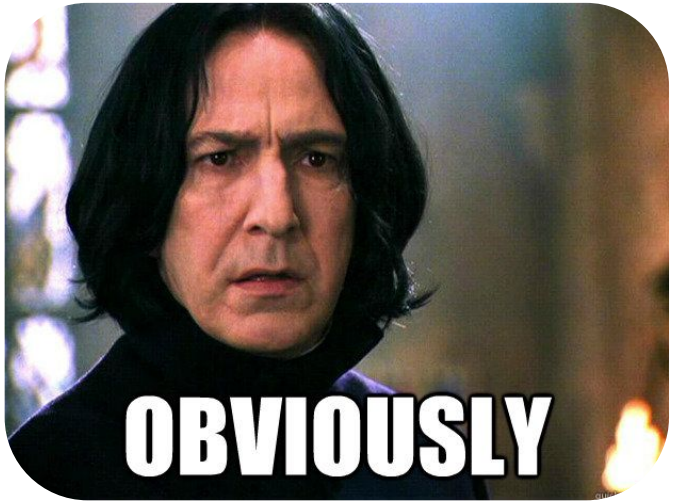


ORIGIN


Stream
RTMP
REDIS CONTROLLER MME
WEBRTC TCP/UDP
WEBRTC WSS
ORIGIN MME
TRANSCODER MME
CF PROXY
WS

ORIGIN

Уменьшение LA 0.6 -> 0.2

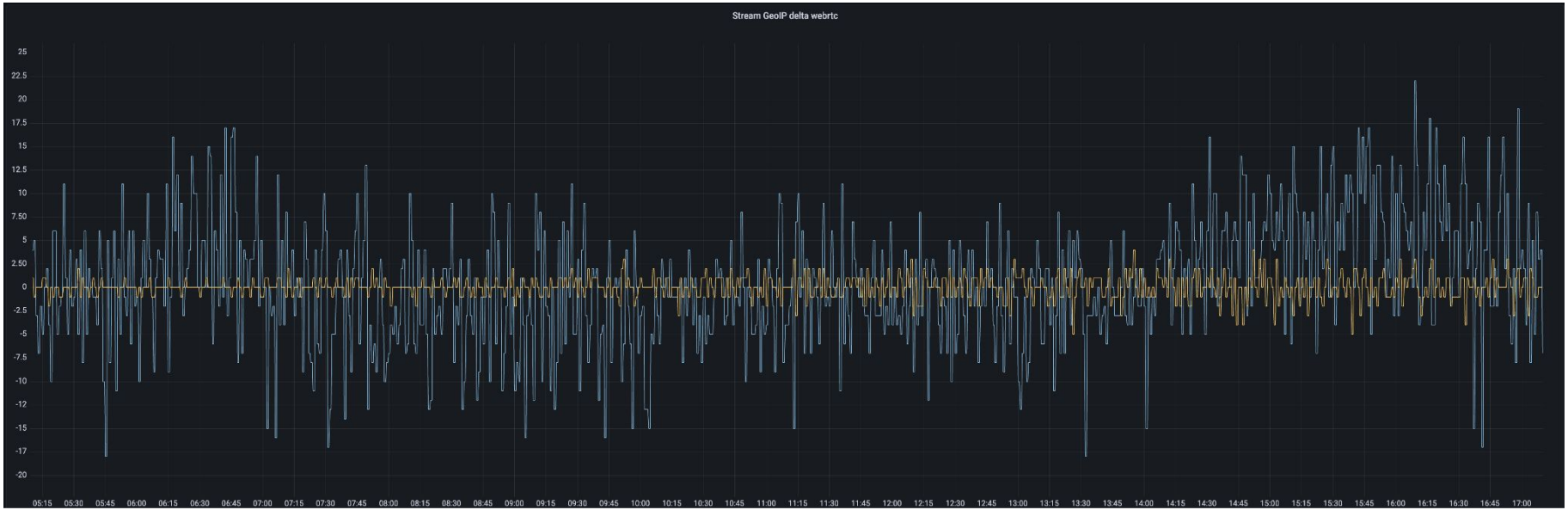


ORIGIN

- сигналинг через WSS для WebRTC
- некачественная связь вызывает реконнект у стримера

ORIGIN: Proxied WSS

WSS с помощью проксирования средствами ExtCDN

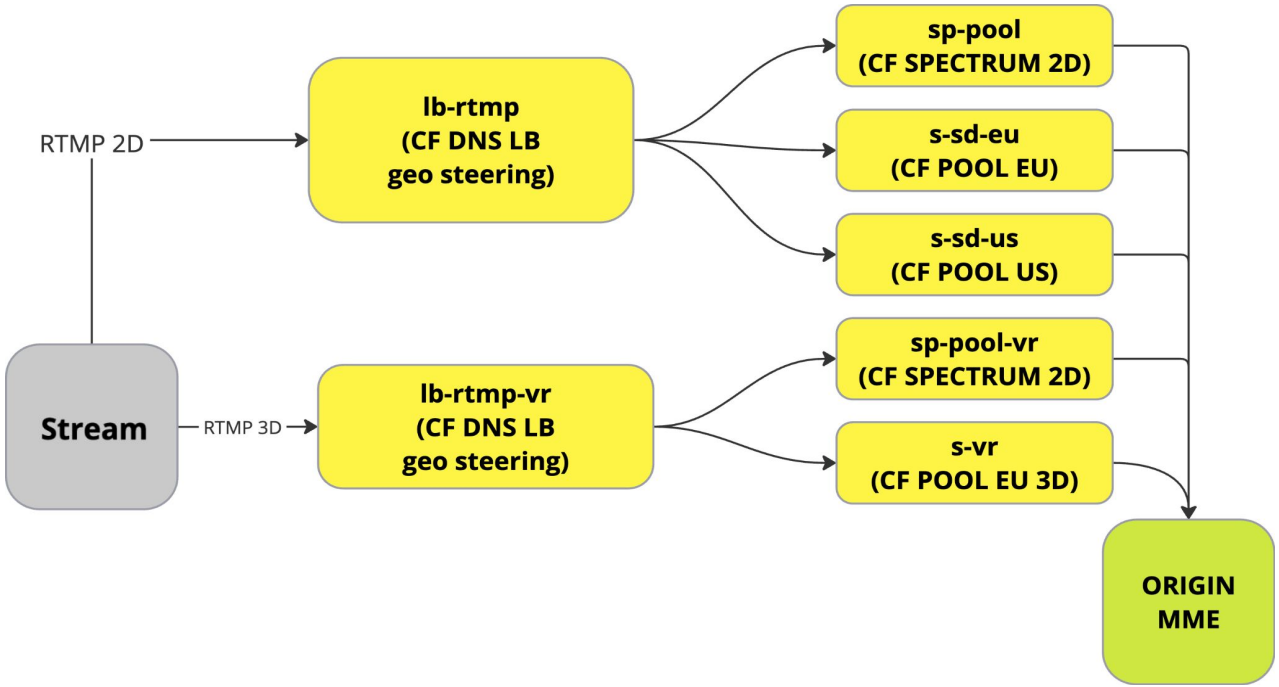


ORIGIN

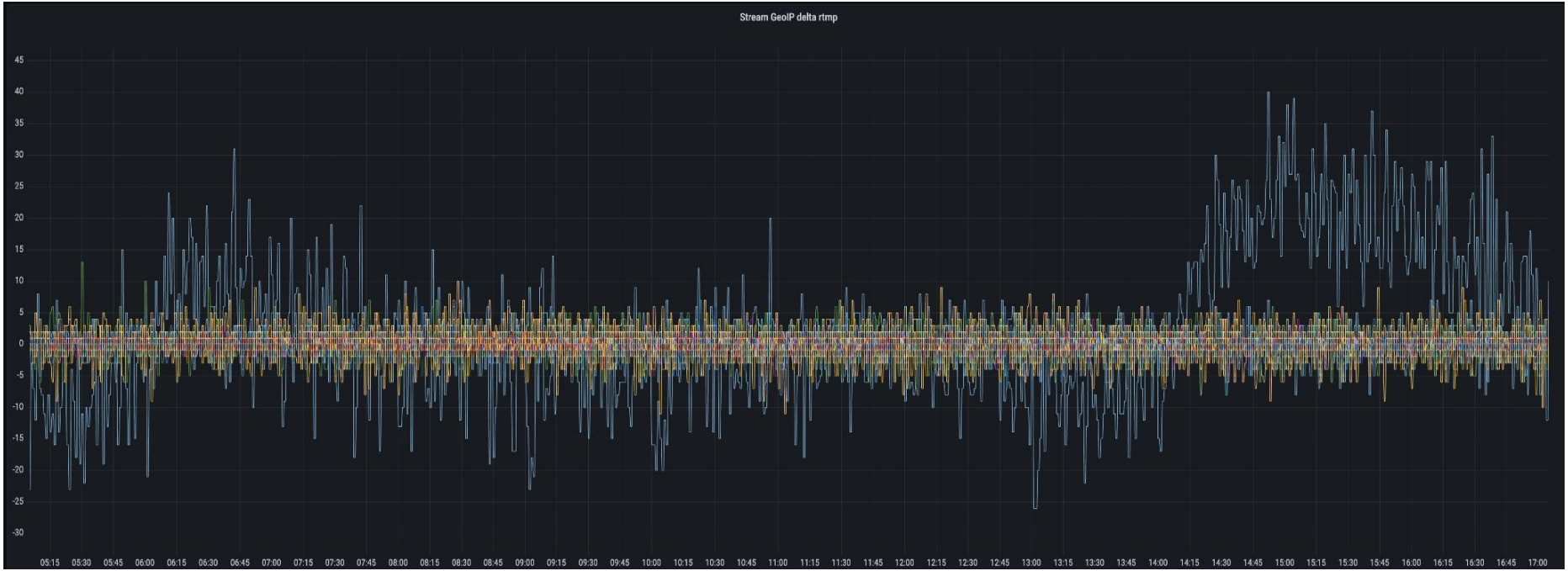
- такая же проблема у стримеров по RTMP
- хочется балансировать нагрузку по регионам

ORIGIN: LB L4 ExtCDN

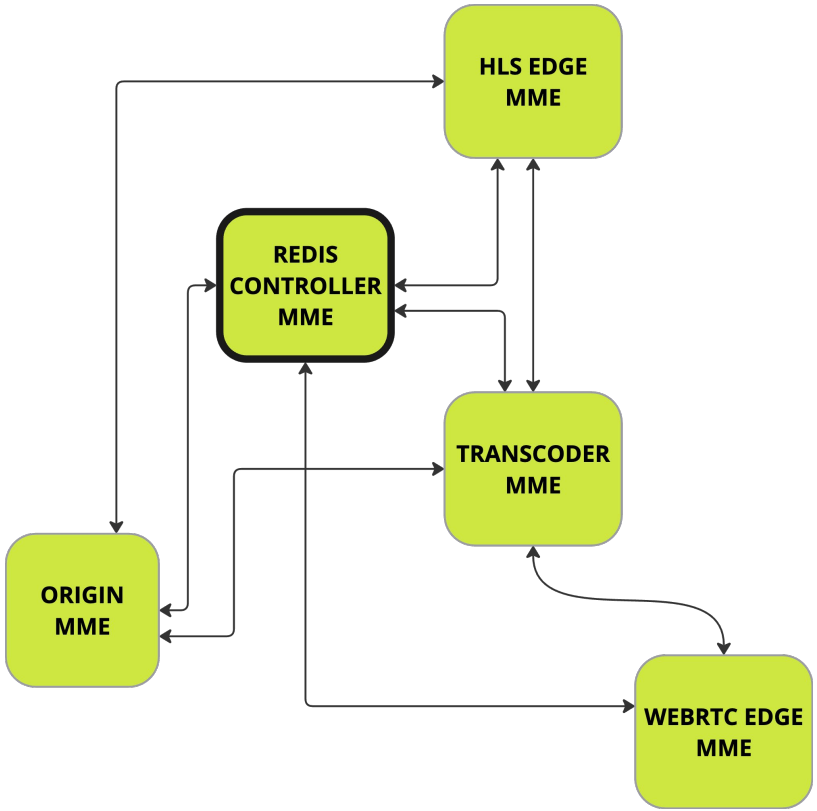
Load Balancing L4 на RTMP подачу стрима



ORIGIN: LB L4 ExtCDN



Controller

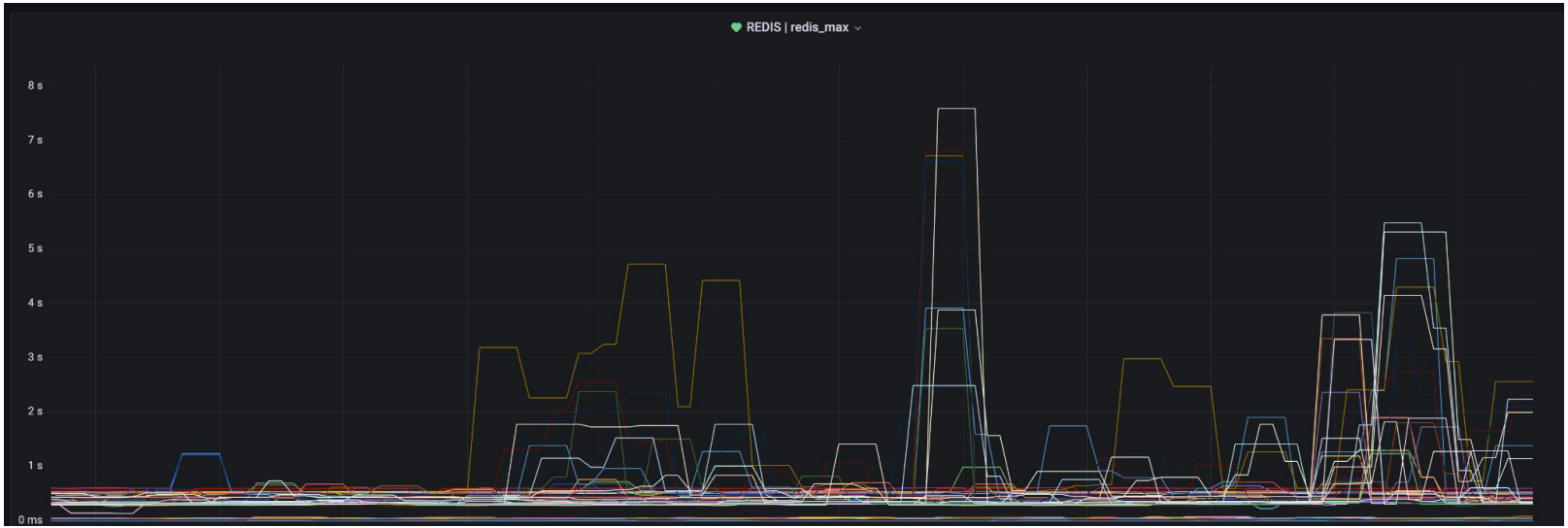

HLS EDGE MME
REDIS CONTROLLER MME
TRANSCODER MME
ORIGIN MME
WEBRTC EDGE MME

Controller

- Управляющий сервер с СУБД Redis
- Позволяет вводить/выводить сервера
- Управляет нагрузкой у компонентов
- Все компоненты работают через redis (pub/sub)

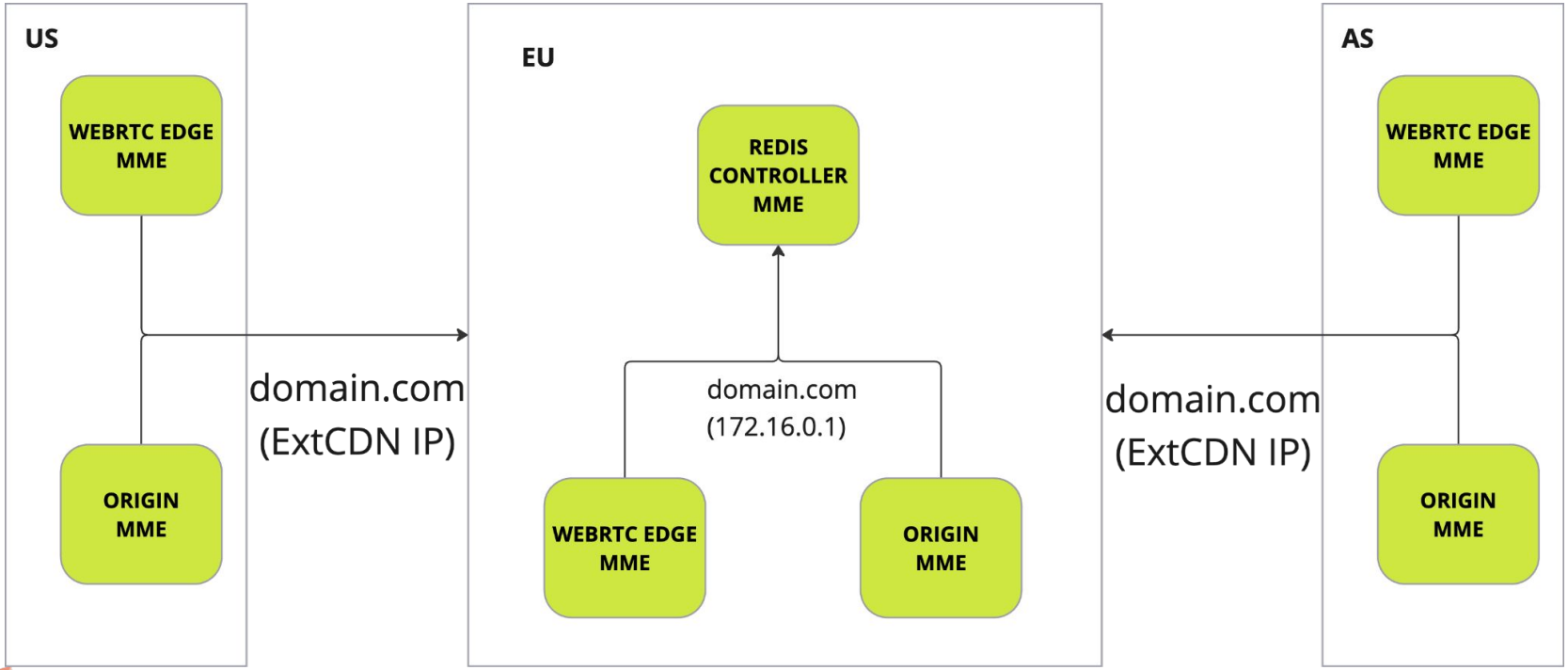
Controller

- между регионами плохая связность
- задержка запроса больше 5 секунд к редису критична

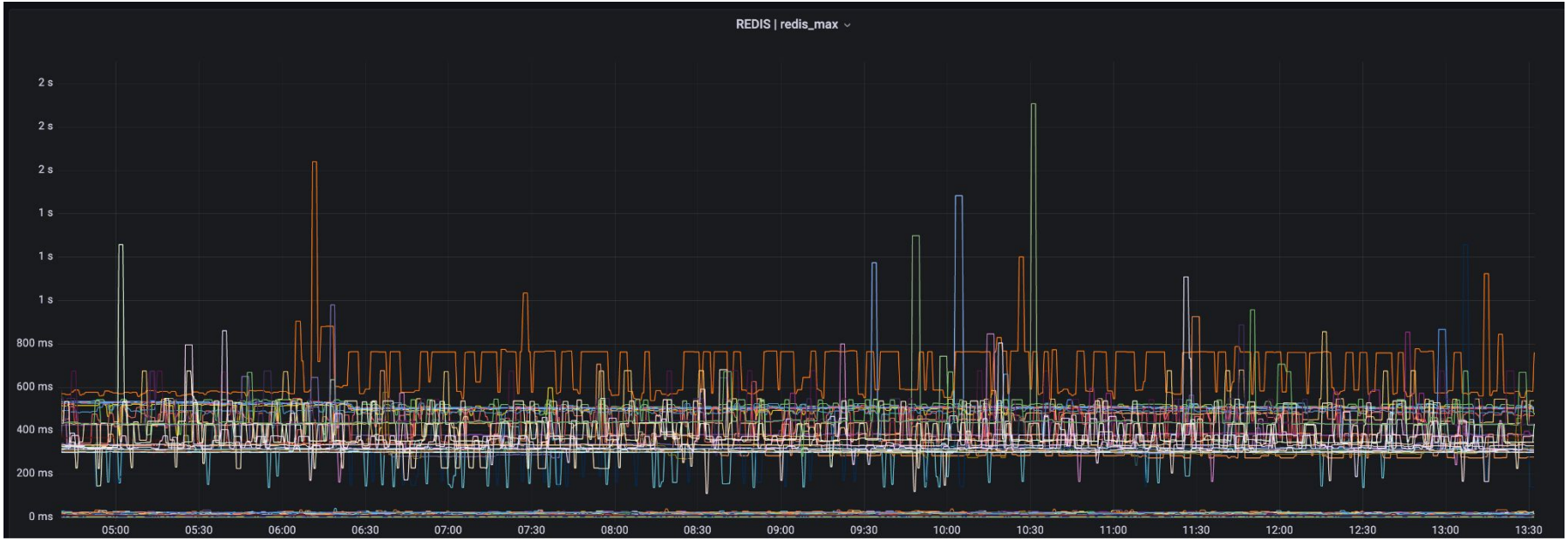


Controller: LB L4 ExtCDN

LB L4 единый GeoIP DNS steering



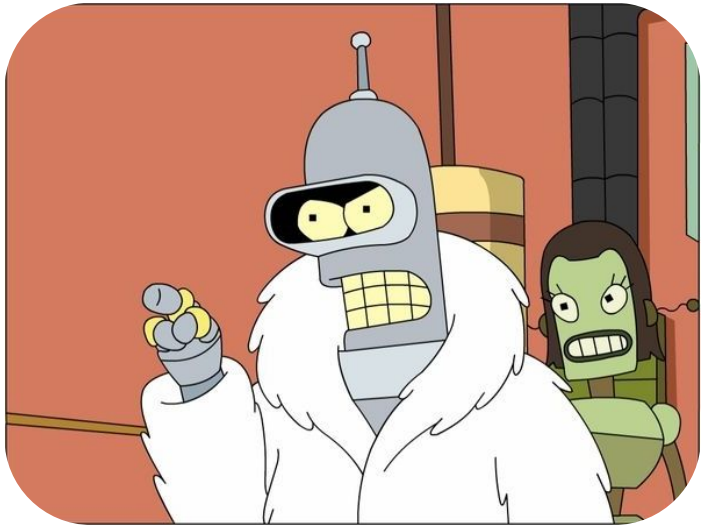
Controller: LB L4 ExtCDN



Результат

- ✓ Больше не платим за лицензии
- ✓ Освободили в среднем x3 ресурсов
- ✓ Рост трафика обеспечивается заменой сетевых карт на более емкие

Box solution



Box Solution

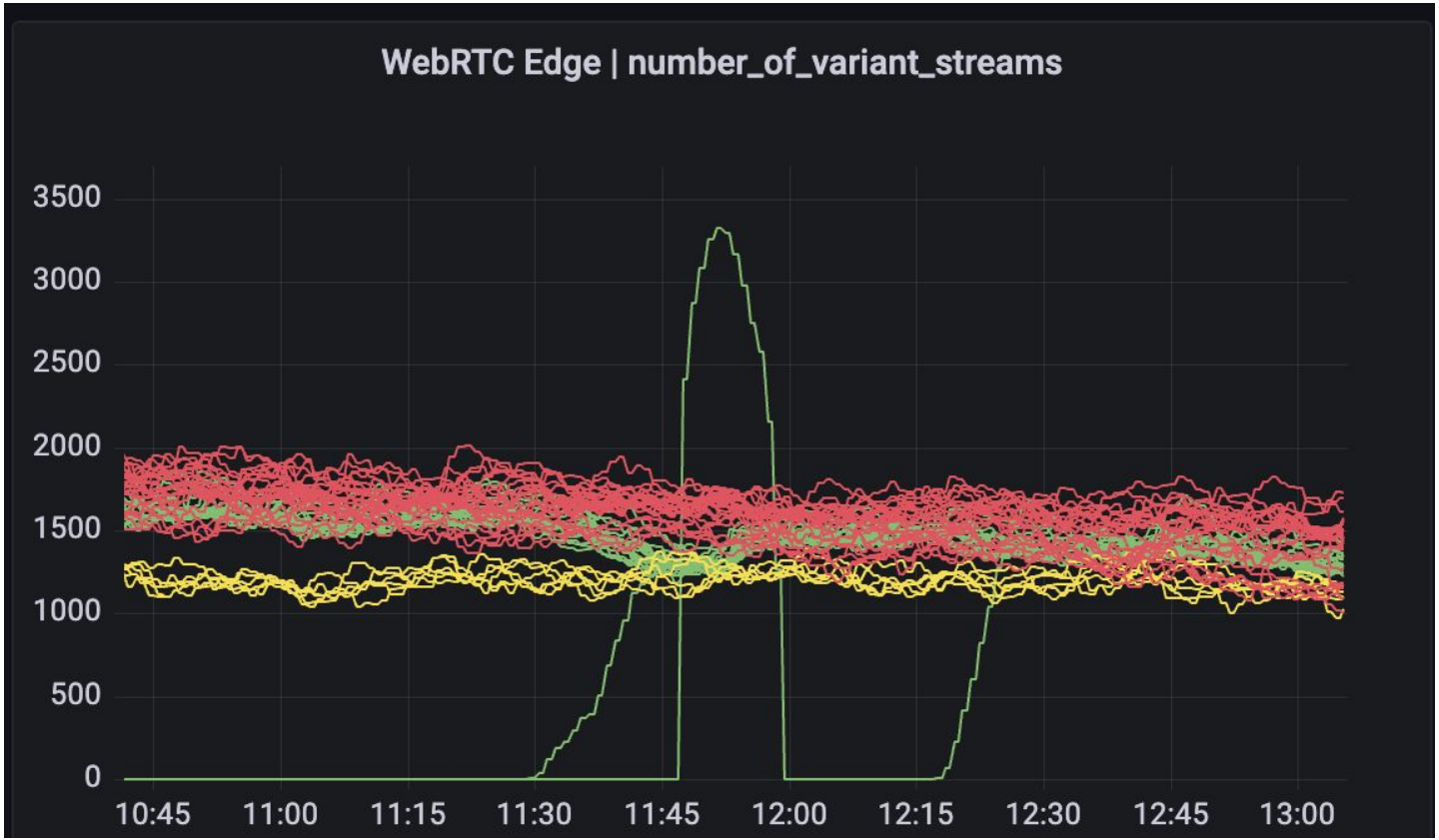
- Просмотр/подача WebRTC не имеет единой точки входа (DNS).
- Балансировкой занимается Live Streaming Platform.
- Нужно делать лишние действия для maintenance серверов.

Box Solution

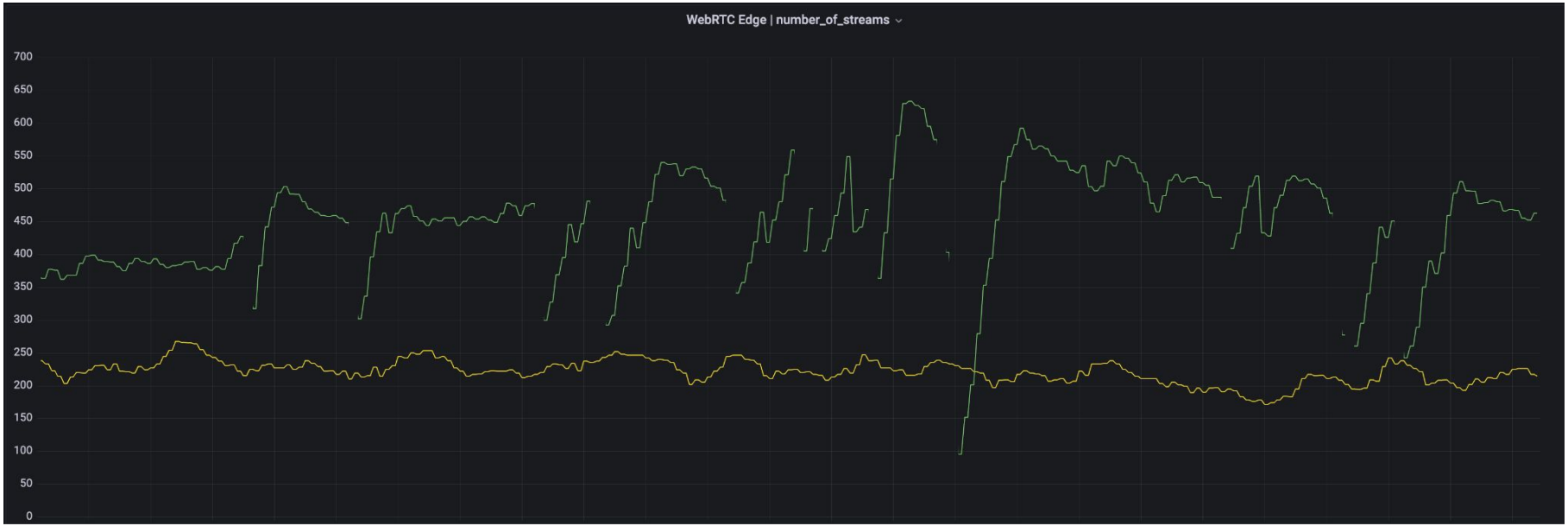
```
[
    'servers' => [
        'us1' => 200,
        'us2' => 200,
        'us3' => 200,
        'us4' => 200,
        'us5' => 200,
        'us6' => 200,
        'us8' => 50,
        'us9' => 50,
        'us15' => 200,
        'us16' => 200,
        'us25' => 200,
    ],
    'type' => 'flashphoner',
    'origin' => 'as',
    'continents' => [],
    'countries' => ['au'],
],
```

```
[
    'servers' => [
        'eu20' => 250,
        'eu21' => 250,
        'eu4' => 1,
        'eu5' => 1,
        'eu6' => 1,
        'eu7' => 1,
    ],
    'type' => 'flashphoner',
    'origin' => 'as',
    'continents' => ['eu', 'af', 'as'
    'countries' => ['nz', 'hk', 'sg',
],
```

Box Solution

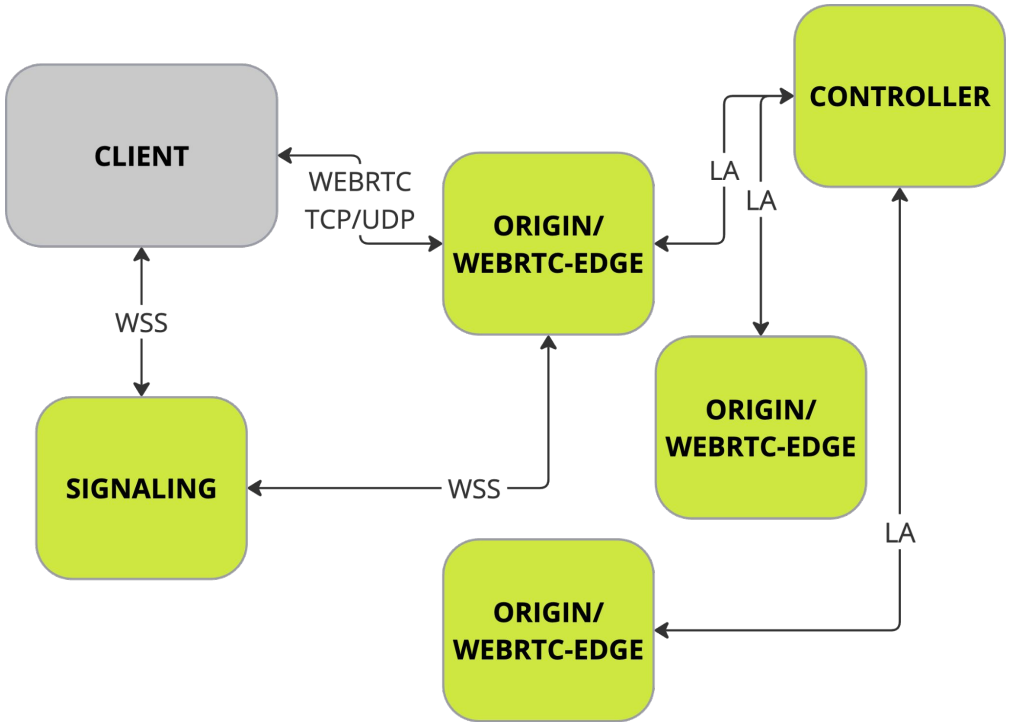


Box Solution



Signaling Server

- Выступает как wss proxy interceptor
- Получает от контроллера информацию о нагрузке WebRTC EDGE/ORIGIN

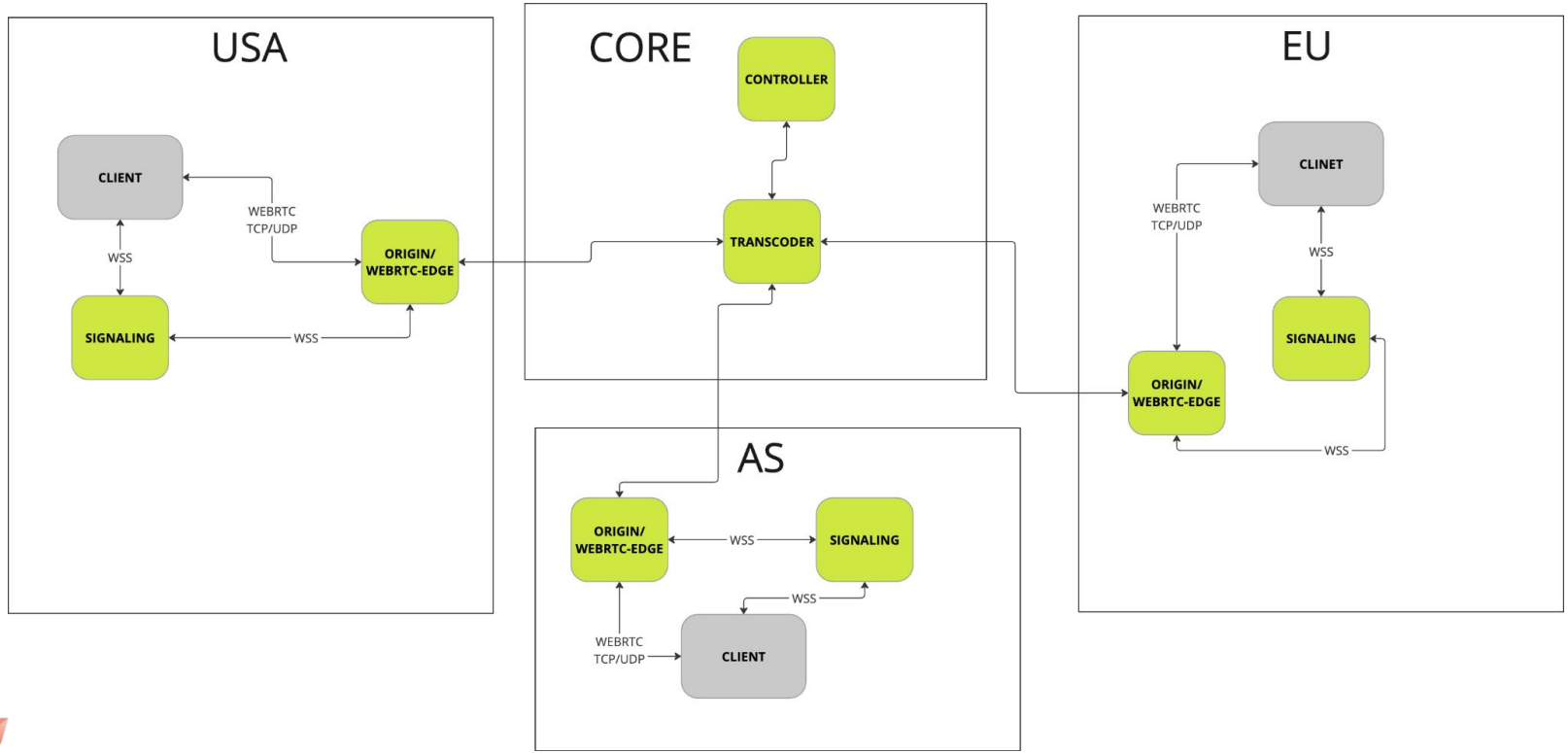


Signaling Server

- Единое DNS имя, через GeoIP steering можем направлять в разные signaling server
- Через nginx с помощью GeoIP мы можем направлять клиентов в нужную группу WebRTC EDGE/ORIGIN
- Засетаплены в AWS EC2 (c4.xlarge)

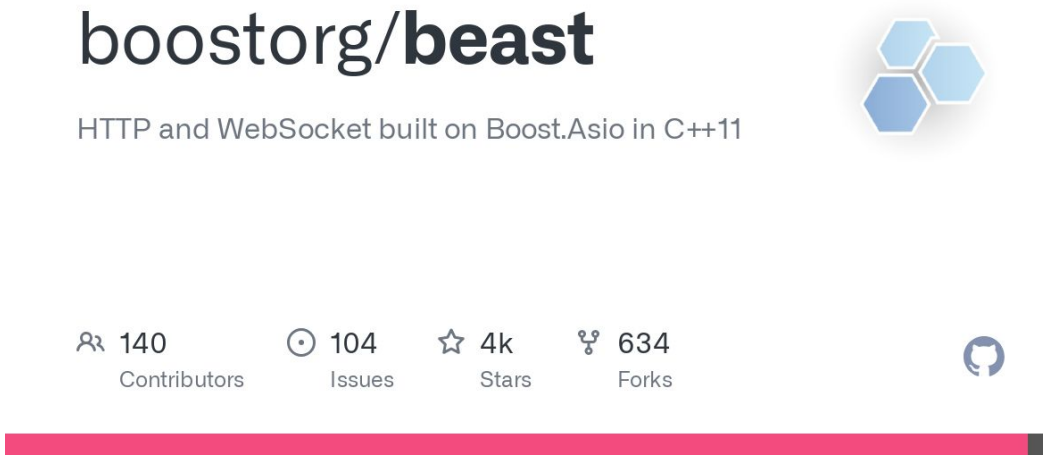
```
NGINX_MAP: |
  map $geoip2_country_code:$geoip2_region $map_group_by_country {
      "~*^hk:"      webrtc_edge_as;          # Hong kong
      "~*^in:"      webrtc_edge_eu;          # India
      "~*^jp:"      webrtc_edge_jp;          # Japan
      "~*^kp:"      webrtc_edge_us_va;       # North Korea
```

Signaling Server



Signaling Server

Проблема: при 2.5к клиентов начали получать 502.
В логах nginx "upstream_response_time": "7.001", "status": 502.
Грешили на C++ Beast.



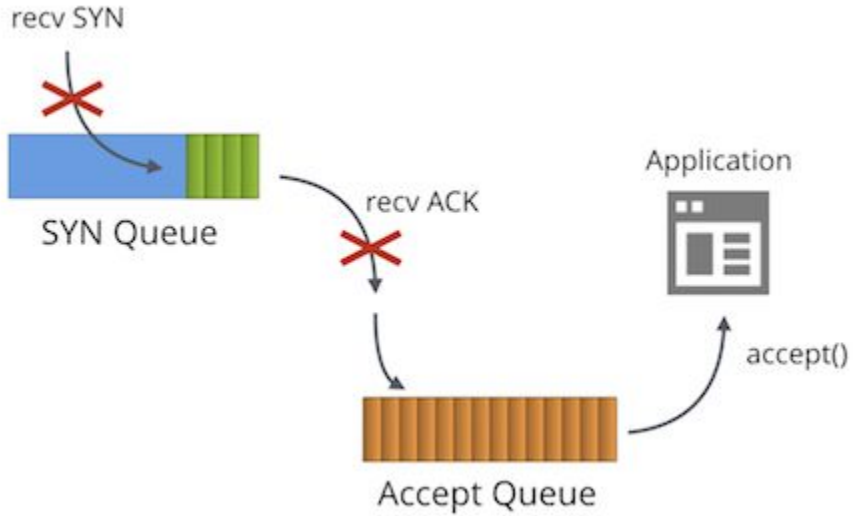
Signaling Server

Проблема:

- Всегда странный таймаут в 7 секунд.
- tcpdump показывает 3 retry tcp handshake.
- Таймаут растёт ногами из TCP хэндшейка и его SYN запросов, который он ретрает через 1, 2, 4 секунды.

Signaling Server

- Проблема с syn/accept queue.
- nginx default backlog = 511.
- app listen(int sockfd, int backlog = 128).

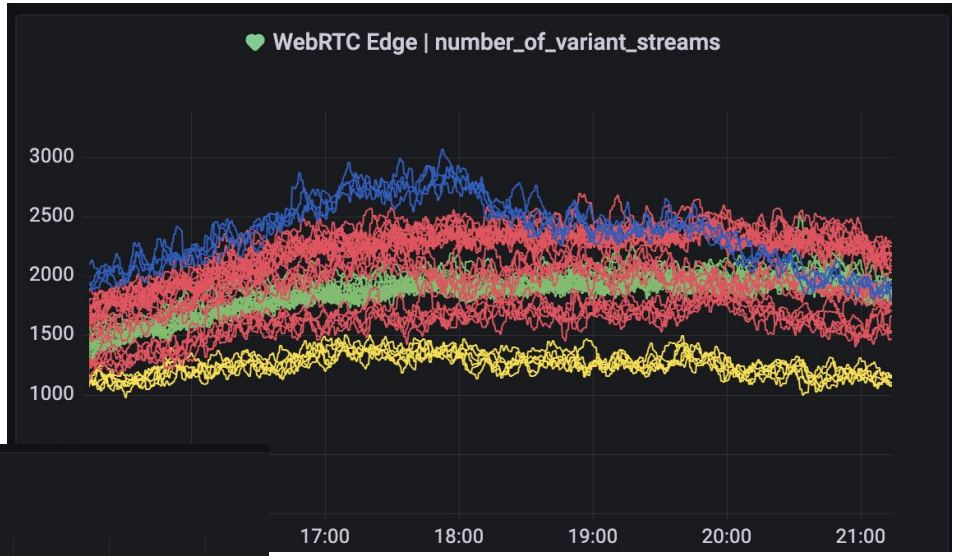


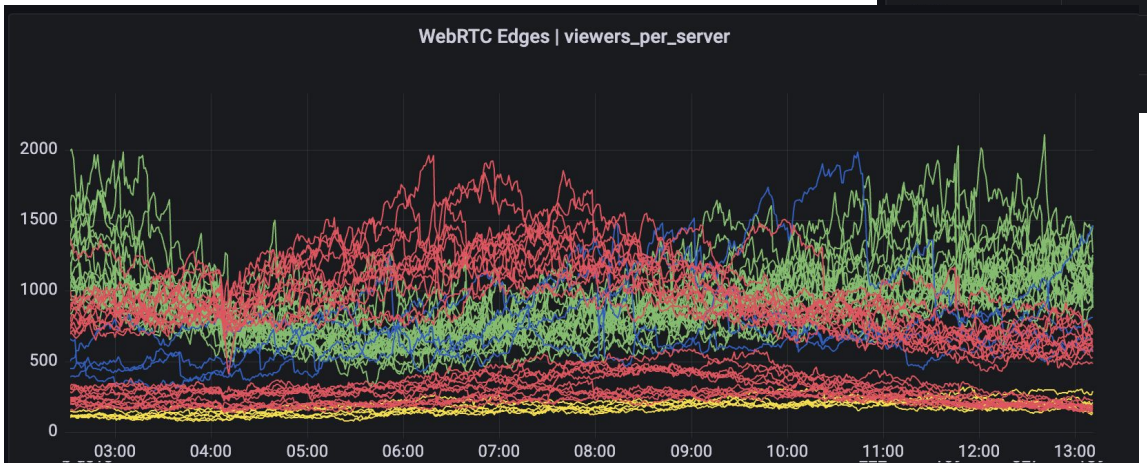
Signaling Server

Решение:

- sysctl net.core.somaxconn = 16384.
- nginx listen backlog = net.core.somaxconn.
- app listen backlog = net.core.somaxconn.
- Переход на unix domain socket.

Signaling Server





Итоги

Итоги

- ✓ Появилась отдельная команда стриминга с большой экспертизой
- ✓ Инфраструктура стала предсказуемой
- ✓ Требования бизнеса выполняются быстрее и качественнее
- ✓ Выросло качество доставки контента

Александр Строгонов
TG @zulgabis

Вопросы