Yandex Cloud

Как CDN Яндекса работает с трафиком видеоплатформы

Евгений Зайцев,
ведущий технический менеджер проектов

Для чего нужны CDN

- У нас есть много (сотни гигабит / терабиты) данных
- Их надо раздавать в пользователей
- Экономить ресурсы и каналы ДЦ
- Меньше нагружать стыки операторов связи с внешним миром

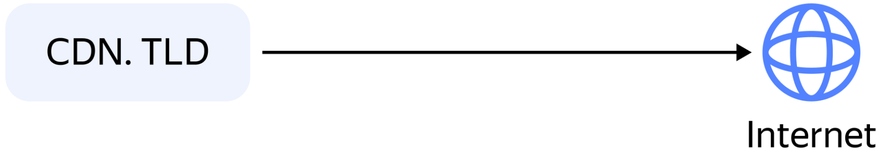
internationalpanorama.com/deal.php?pageId=S_AMERICA_LUX&lang=es

1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Один сервер

Ставим один сервер в ДЦ и раздаем данные с него

Назначаем ему CDN.TLD → IPv4 1.2.3.4



Проблемы

- Нет отказоустойчивости

  Сервер поломался

- Мало места на дисках под кеши
- Уперлись в сетевую карту

Несколько серверов

Добавляем несколько серверов в ДЦ

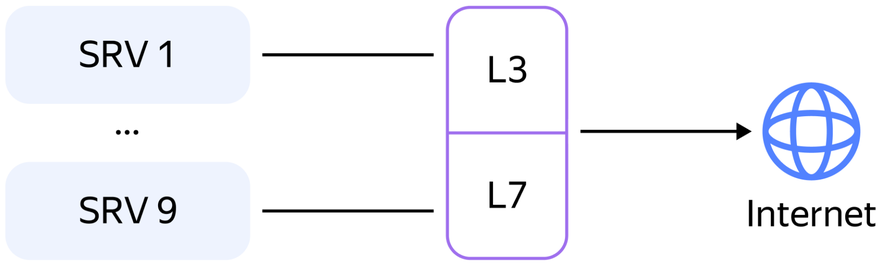
- Не зависим от одного сервера
- Много сети и места под кеш
- Нужен L3-балансировщик

  Отвечать с одного IP-адреса CDN.TLD

- Теперь зависим от одного балансировщика

  Сеть, всего одна железка

- На всех серверах одинаковый кеш
- А еще может поломаться ДЦ



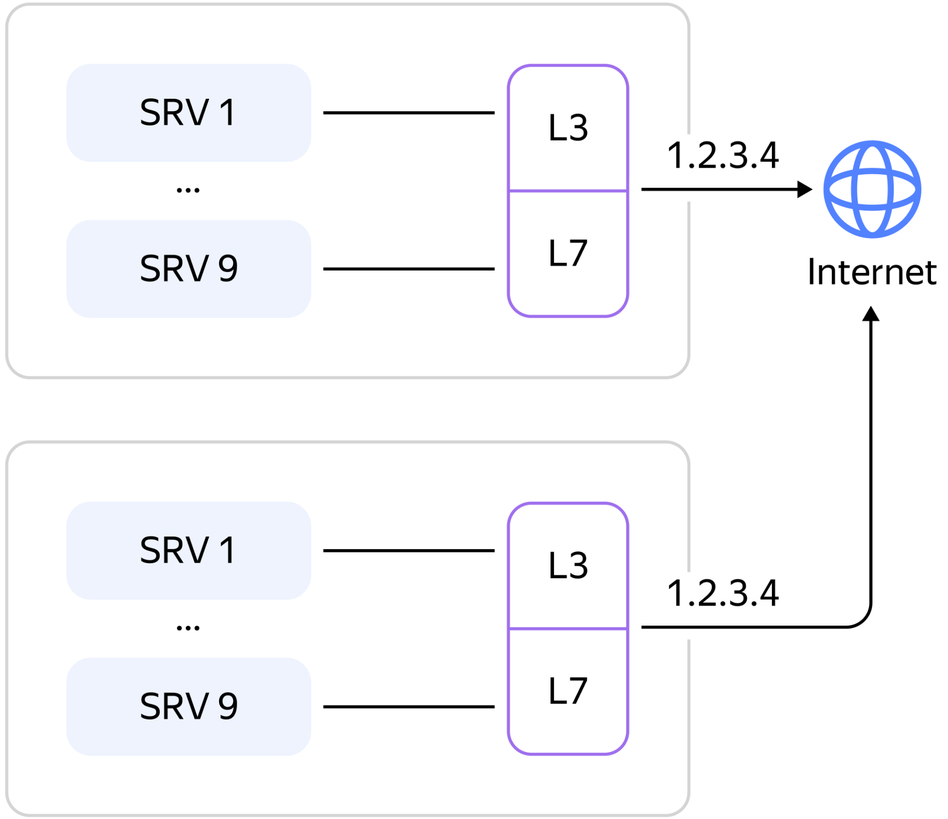
Несколько дата-центров

Ставим аналогичную конструкцию в несколько ДЦ

Наружу светим одним-несколькими IP-адресами с L3-балансировщиков

BGP-Anycast (запрос может прийти в любой ДЦ)

- Нет зависимости от одного ДЦ, все красиво
- Новые ДЦ строить дорого и долго
- По-прежнему упираемся в L3-балансировщик, через который проходит трафик
- BGP Anycast для IPv4 требует целой /24 сети (256 адресов)

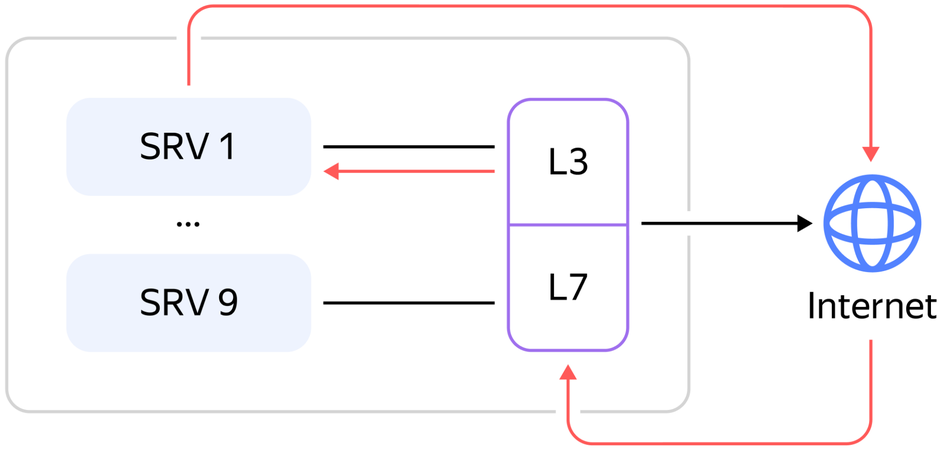


...обратный трафик мимо L3

Запрос через L3 балансировщик отправляется на кеширующий сервер,

Ответ (тяжелый контент, трафик) идет напрямую, мимо L3

- Не нагружаем сеть на L3
- По-прежнему не управляем трафиком

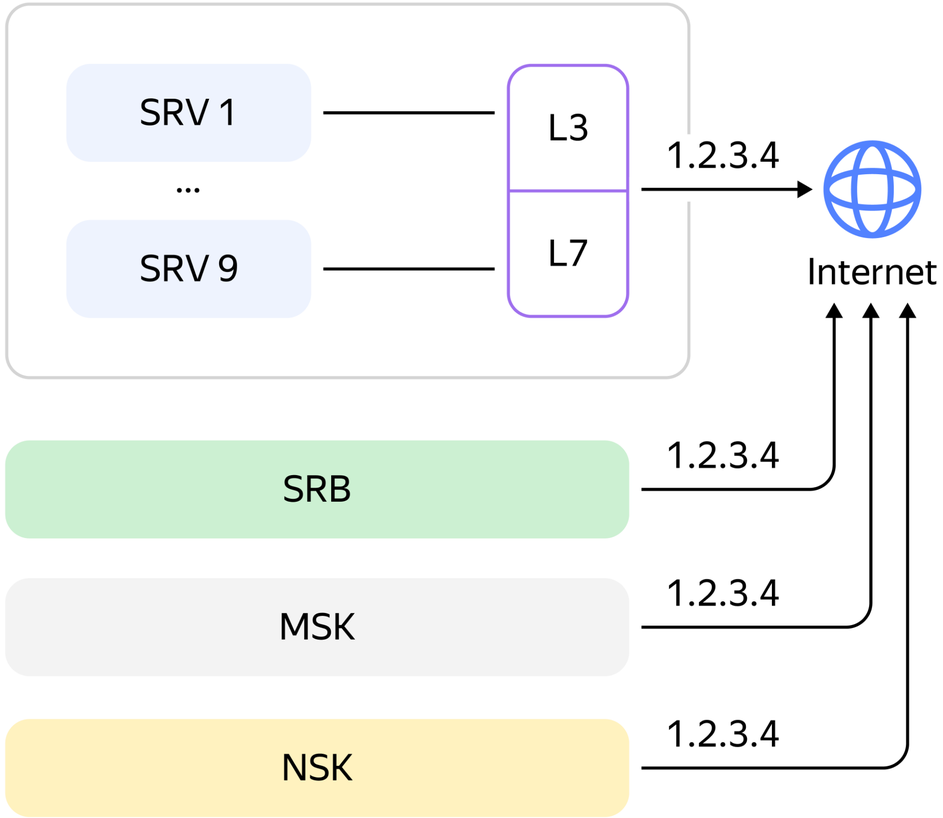


Вынос CDN-кешей из ДЦ

Ставим CDN-кеши прямо к операторам или на IX-ы

Растягиваем наш BGP-Anycast (один-несколько IP) на такие выносы

- Пользователи ходят к ближайшему L3 CDN.LTD (к кеш своего оператора)
- Все ещё требуется целая /24 IPv4 сеть (256 адресов) для BGP-Anycast-а
- Перегруженный CDN-узел (CPU, сеть на машинах, сеть в сторону оператора) можно только выключить
- Трафик с выключенного CDN-узла перераспределится по соседям «как-то»
- Уже установленные сессии пользователей — порвуться (таймауты/stalled-ы)



Что дальше

Есть несколько вариантов дальнейшей эволюции CDN

★

Внедрение GeoDNS

- Каждый узел у оператора имеет свой IP
- GeoDNS возвращает ближайший пользователю узел CDN.TLD
- Нужна EDNS-client-subnet в запросах
  Не все кеширующие сервера присылают такие запросы

★

Отправка в плеер нескольких URL-ов вида {1,2,3,…}.CDN.TLD

- Требуется свой кастомный плеер

★

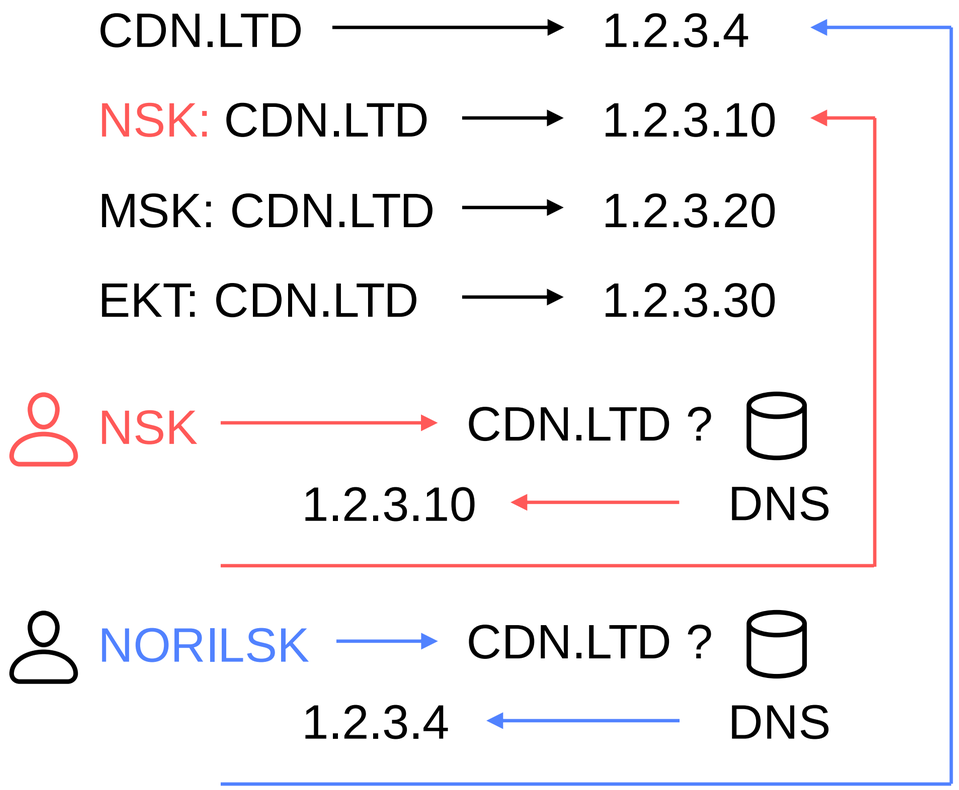
Другие варианты

GeoDNS

Вариант дальнейшей эволюции CDN: внедрение GeoDNS

- Каждый узел у оператора имеет свой IP
- GeoDNS возвращает ближайший пользователю узел CDN.TLD
- Нужна EDNS-client-subnet в запросах

  Не все кеширующие сервера присылают такие запросы

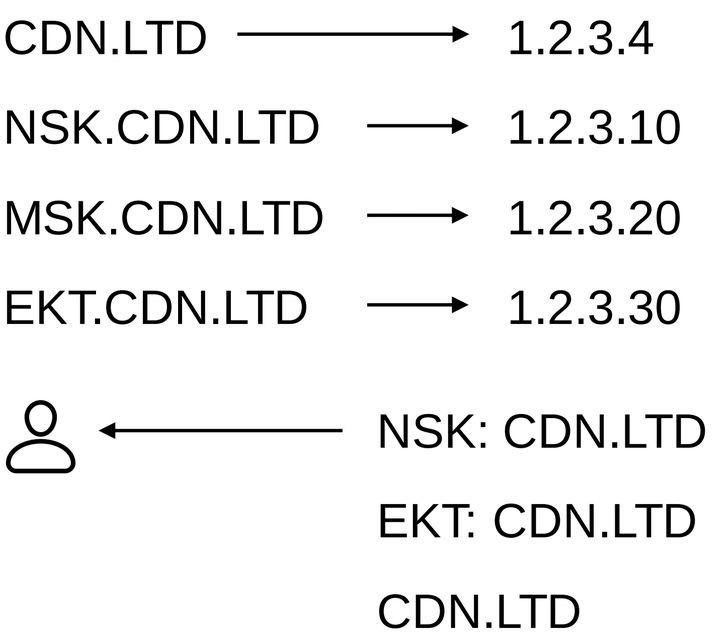


Несколько URL

Для видео

Альтернативный вариант: отдать несколько доменов CDN-а в плеер

- Отправка в плеер нескольких URL-ов вида {1,2,3,…}.CDN.TLD
- Требуется свой кастомный плеер



Более сложные варианты: внедрение своего протокола

1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Что требуется от CDN-платформы

Линейное увеличение объёма кеша в CDN-локации при добавлении машин

Отсутствие узких мест типа L3 для раздачи терабит данных

Гибкое управление трафиком для каждой локации / стыка

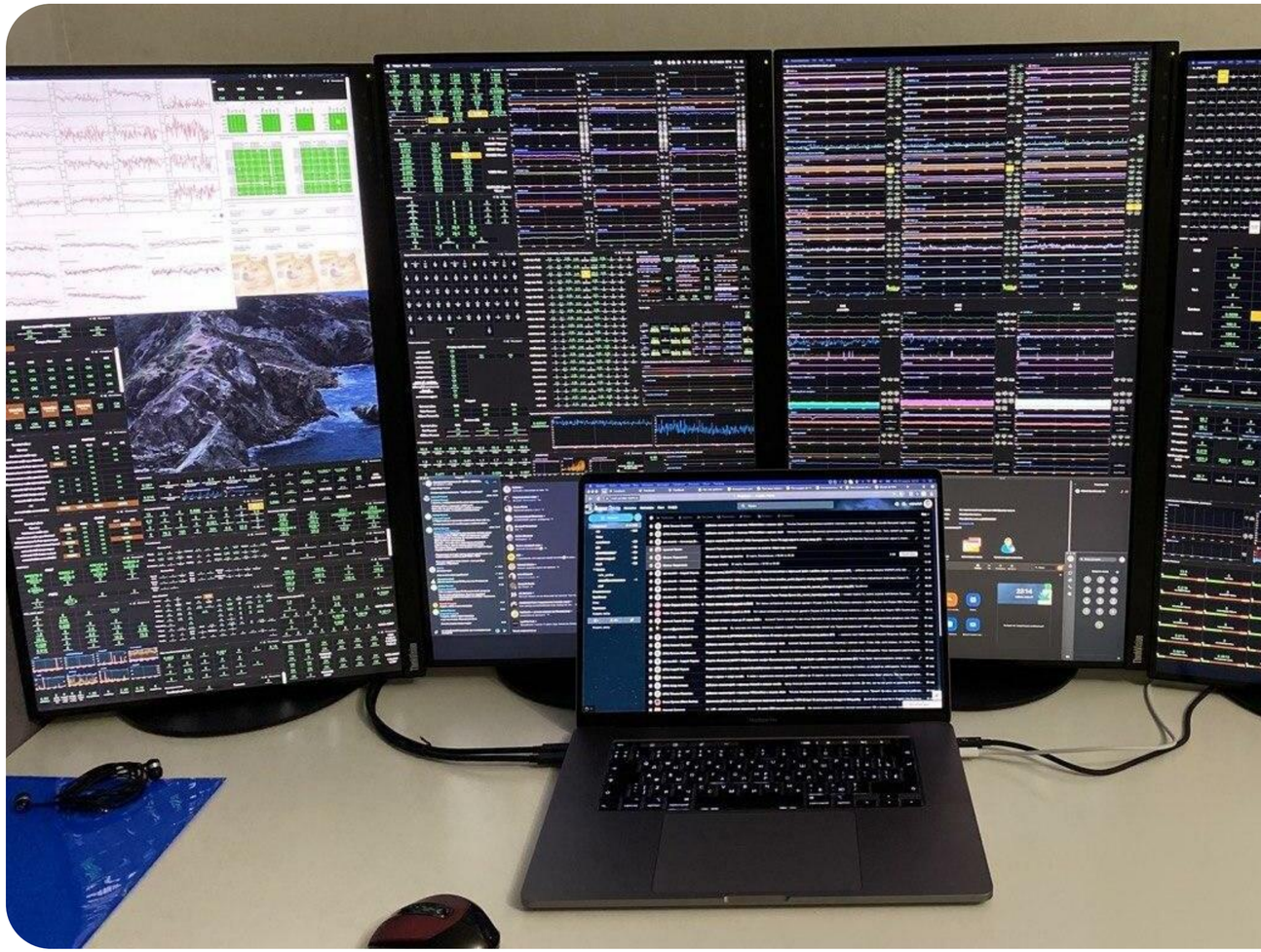
Возможность отправить трафик в любой стык явным образом

Плавный увод % пользователей с перегруженных локаций / машин

Не нагружать CDN непопулярным контентом

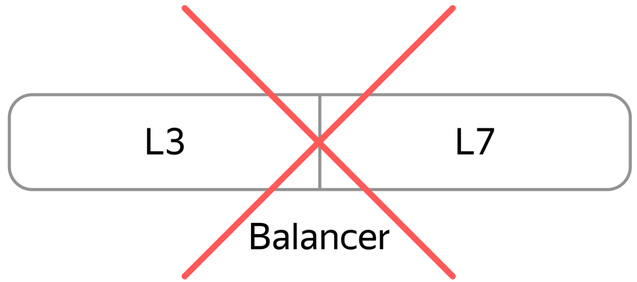
Необходимость риалтайм-приборов

- Риалтайм-метрики ресурсов (сеть / диски / CPU / каналы связи) на каждой CDN-машине и стыке с оператором
- Метрики качества сети / раздачи с каждой CDN-локации



Основное

- Никаких L3-балансировщиков над кешами
- Каждый кеш со своим уникальным IPv4/IPv6-адрес и хостнеймом
- Отсутствие метаданных распределения контента по кешам
- Уметь отправлять трафик в заданный стык с оператором
- Минимизировать запись на диски / подкачку для CDN-кешей



Совместимость

- Работает из коробки для стандартов DASH/HLS
- Совместимо со стандартными html5-видеоплеерами
  Native, shaka,…

mpeg-DASH

HLS — HTTP Live Streaming

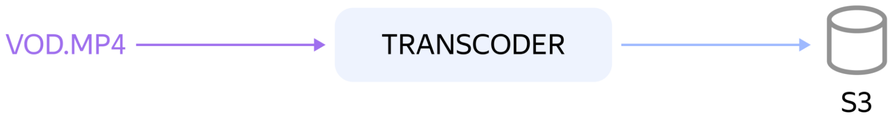
1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

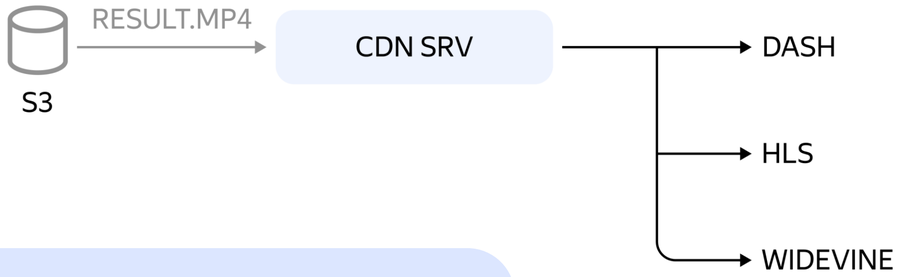
Управление трафиком через плейлисты

- Отдельный компонент-генератор плейлистов (plg) для vod/live видео
- Генератор плейлистов занимается так же балансировкой нагрузки на CDN
- Для ссылок на чанки используется абсолютная ссылка на заданный cdn-хост
- В ссылках на чанки указывается ID линка, в который слать ответный трафик

Детали в докладе
"Такой CDN будущего вы хотите"
от @indigo
clck.ru/32QjuQ

Храним только один формат контента





- После транскодирования храним результат в сторадже и кешах в едином формате
- Dash/hls/mss стримы (пекеджинг) делаем на лету, при отдаче чанка пользователю
- Опциональное шифрование (drm) — тоже делается на лету

В сторадже требуется в шесть раз меньше места

Больше разнообразного контента влезает в кеши

Генерим меньше трафика подкачки из ДЦ в кеши

Consistent hashing роликов в каждой cdn-локации

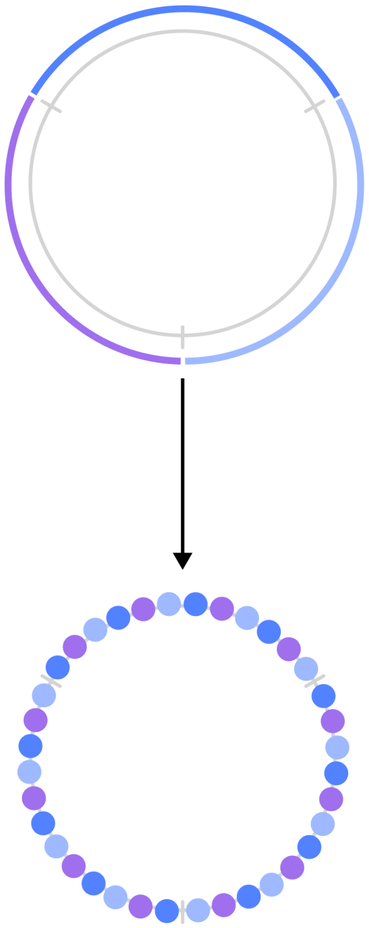
Используем consistent hashing по ключам-роликам

- Каждый vod-ролик/live-стрим отдается с одного кеша локации (или двух)
- Разные кеши содержат разные данные

Увеличиваем кешхит (меньше трафика подкачки)

Общий объем кеша локации линейно растет с добавлением машин

Потеря одной/нескольких машин не приводит к перестроению всего hash-ring-а

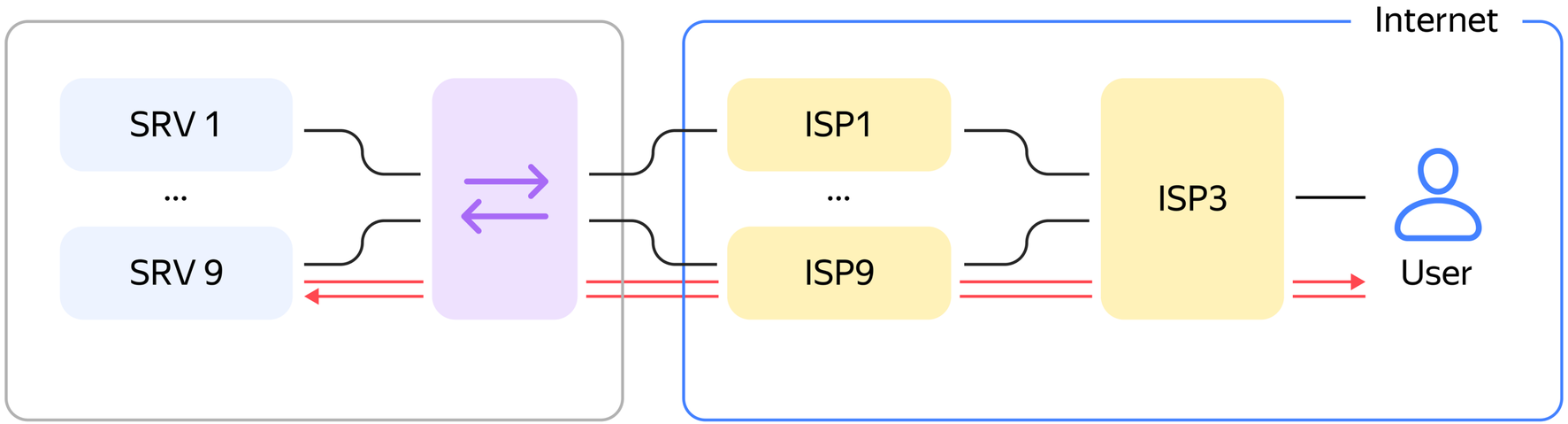


LRU на кешах

- В кеши заранее ничего не доставляется
- Балансировщик знает, какой ролик должен лежать на каком кеше
- Просто отправляем пользователя на заданный кеш
- Если ролик еще не в кеше, он подкачается из датацентра (один раз)

1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Отсутствие L3-балансеров над кешами



- Не нужно думать про отказоустойчивость L3
- Нет узких мест с pps-ом и трафиком на вход и выход
- Нет bgp-эникаста

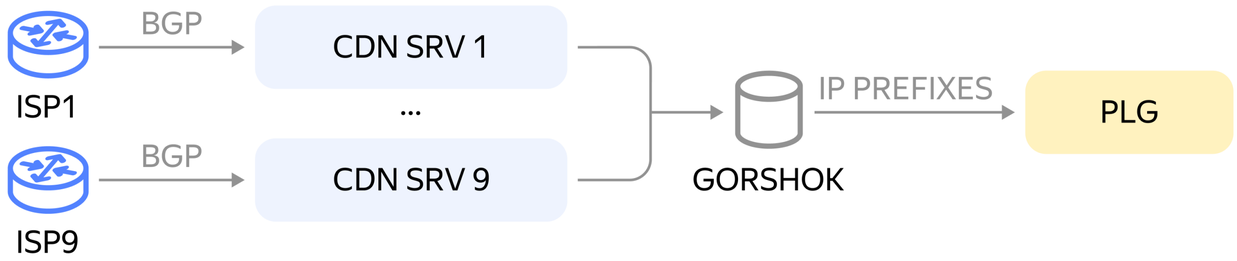
- Можем работать на IP-адресах оператора
- Явно отправляем заданный % пользователей в ту или иную cdn-локацию
- Добавляя хосты в локацию, увеличиваем пропускную способность сети на выход

Простукиватель кешей

- Отдельный компонент (ponger) в риалтайме простукивает живость кешей
- Собирает метрики загруженности ресурсов (CPU / диски / доступное место / сеть) на каждом кеше
- Быстро (секунды) детектить мертвую машину и выкидывает её
- Все эти данные отправляются в PLG-балансировщик

«Трое из Простоквашино» © Киностудия «Союзмультфильм», 1978

BGP все-таки нужен



- BGP-Anycast-а нет, но надо знать, какие сети может обслужить каждая CDN-площадка
- Техническая BGP-сессия, на каждой площадке, получающая префиксы сетей оператора

- Отдельный компонент (gorshok), который «варит» список IP-префиксов (сетей) с весами, для каждого линка и каждой CDN-площадки
- Балансировщик PLG, который теперь знает, с каких CDN-площадок и с каких линков можно обслуживать заданного пользователя

1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Единый балансировщик

Балансировщик PLG, который знает все про CDN

Нагрузка на линки/площадки/машины/префиксы сети за каждым линком/метрики

Данные в подготовленном виде ему поставляют другие компоненты

Ponger, gorshok

Задача PLG

На запрос за плейлистом, отправить пользователя в оптимальный кеш и линк с оператором

Дополнительные метрики, влияющие на балансировку

BGP

BGP-метрики для каждого префикса

As-path, prepand

Сетевые

Серверная: TCPInfo и BBR-метрики с каждой TCP-сессии раздачи данных

RTT, скорость, ретрансмиты

Клиентская: Perf-метрики

Perfomance log с браузера и с плеера

Веса префиксов и линков

- Выставляем статичный вес всем линкам, которые видит наш балансировщик PLG
- Учитываем BGP-метрики as-path и prepand при вычислении веса пары (сетевой префикс : линк)
- Группируем метрики сети (TCPInfo, BBR, pref) по сетевым префиксам

На выходе имеем пары
(сетевой префикс: линк) → вес

Простой алгоритм балансировки

Вводные

В каждой CDN-локации расположено один-несколько кешей

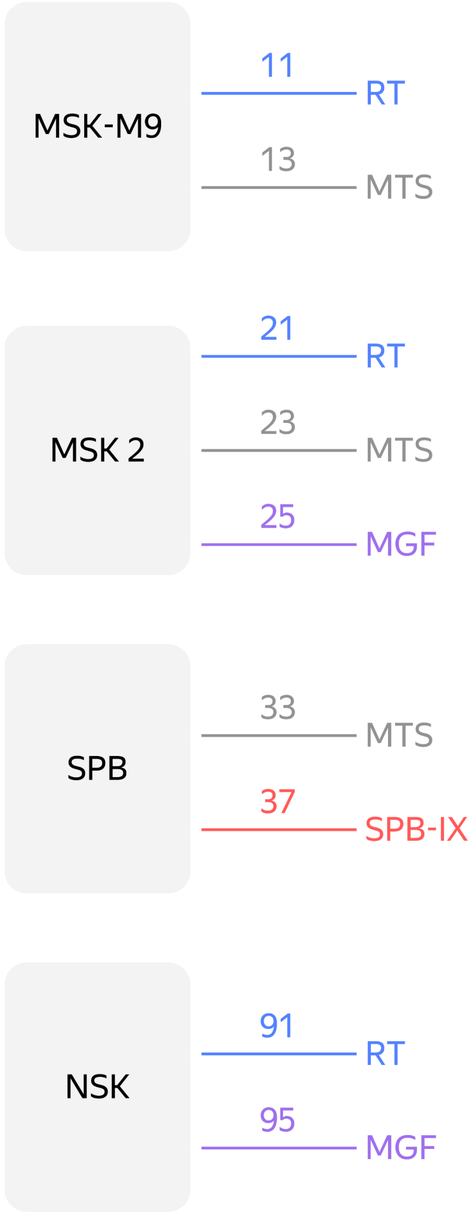
К каждой локации подключено один-несколько линков

За каждым операторским стыком (линком) есть список сетевых префиксов

Алгоритм

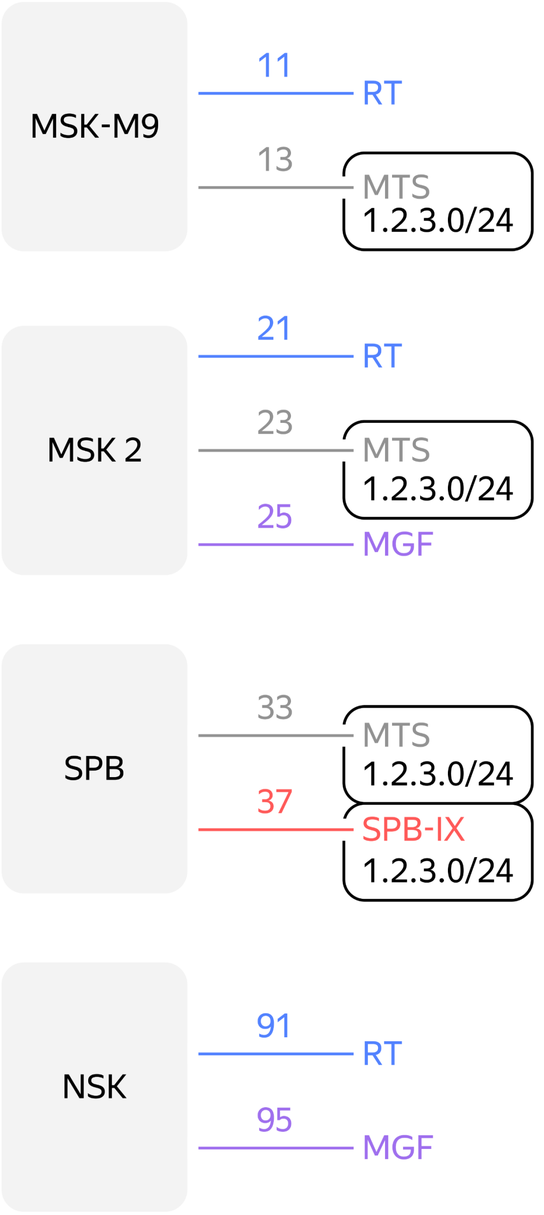
Берем IP-адрес пользователя

Идем снизу вверх: Сетевой префикс → линк → локация → CDN-кеш



Не всё так просто

- Один и тот же сетевой префикс доступен с разных линков и операторов
- Префикс может быть виден через публичные IX-ы
- Точка зрения BGP (best path) может не совпадать с мнением нашего балансировщика



Отвечать с одного адреса в разные стыки

- Есть «толстые» локации с десятками кешей и десятками операторов
- С каждым оператором организован один или несколько стыков
- Не все операторы готовы предоставлять десятки IPv4-адресов для кешей
- Научились отвечать с собственных IP-адресов в такие «бедные» стыки без собственных IP
- На кешах не нужно поднимать новый интерфейс и IP-адрес для каждого такого стыка

Магия в докладе "Дрессированный медийный трафик Яндекса" от aschurov@
clck.ru/32QhLN

Алгоритм балансировки.
Детали

- Находим все префиксы, куда входит IP пользователя
- Оставляем префиксы с одинаковым весом
- Находим все стыки, где присутствуют эти префиксы
- Оставляем только работающие (не перегруженные, включенные) стыки
- Получаем список из одного-нескольких стыков

- Вес стыка (линка) влияет на вероятность выбора оного для конкретного запроса пользователя

  Суммарный вес всех выбранных линков — 100
- По выбранному линку определяем CDN-локацию
- Внутри CDN-локации по хешу от ключа(ролика) определяем кеш
- Получаем hostname + ID линка, куда отправить пользователя и куда слать ответ

1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

PID-регулятор на линках

Всё сложнее, нагрузка на операторские линки меняется динамично

Задача — не допускать перегрузок стыков с операторами и по максимуму утилизировать стыки

- Смотрим не только текущую загрузку каждого стыка, но и историю роста/падения этой загрузки
- Алгоритм PID-регулятора в PLG позволяет замедлять рост нагрузки на стыки

  При резком всплеске
- Минимизирует «волны»

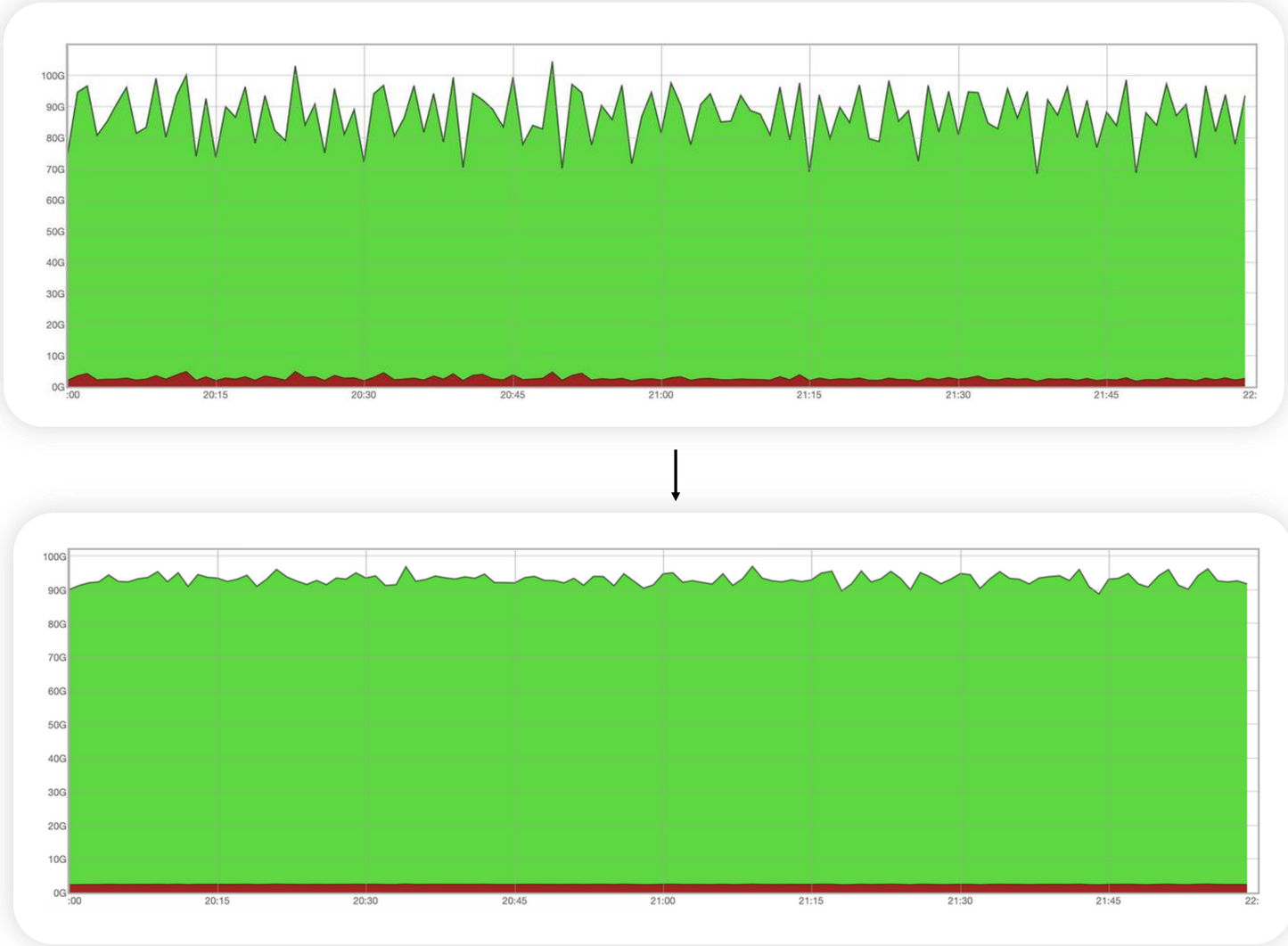
  Перетекание трафика между разными стыками
- Ориентирован на «target»

  Поддерживает загрузку линка в районе X% от полосы

Детали в докладе
"Такой CDN будущего вы хотите"
от indigo@
clck.ru/32QjuQ

PID-регулятор на хостах

- Отдельные CDN-кеши могут нагружаться неравномерно
- Нельзя допускать перегрузки исходящей сети и CPU на кешах
- Смотрим историю нагрузки и снижаем процент новых пользователей, отправляемых на этот кеш
- Следим за таргетами на нагрузку сети и CPU



1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Популярность контента

Малопопулярный (холодный) контент

Редко спрашивают

Вредно гонять через кеши

Популярный (горячий) контент

Смотрят заметно больше, по сравнению с другим контентом

Надо всегда раздавать из кешей

thumbs.gfycat.com/DishonestDetailedAnteater-poster.jpg

geographyofrussia.com/russkaya-mednaya-kompaniya/

Горячий, теплый, холодный контент

- Вводим понятие горячего контента
- Определяем популярность по логам раздачи и «вес» популярности
- Вес = объем розданных гбайт за окно времени / размер контента
- Строим top популярных vod-ов в таблице вида:
  Вес, content_ID, размер контента, общий размер в кеше
- Если контент находится в top-е, считаем его теплым и ищем CDN-локацию для него
- Контент должен поместится в общий кеш CDN-локации, не вытеснив контент с большим весом
- Если нашли хотя бы одну CDN локацию, куда контент помещается — он горячий и отправляется в CDN
- Никуда не поместился — теплый и раздаем из дата-центра как холодный трафик

ID	SIZE	CACHE SIZE	WEIGHT
AAA	10	10	5 000
BBB	20	10 + 20 = 30	440
CCC	44	30 + 44 = 74	420
…	…	…	…
YYY	555	99 999	5
…	…	…	…
ZZZ	456	1 000 000	3

Типы популярного контента

Мегапопулярный

По количеству сгенерированного трафика с одной машины

Какая то новинка, которую все бросились смотреть одновременно

Или мегапопулярная трансляция (спорт?)

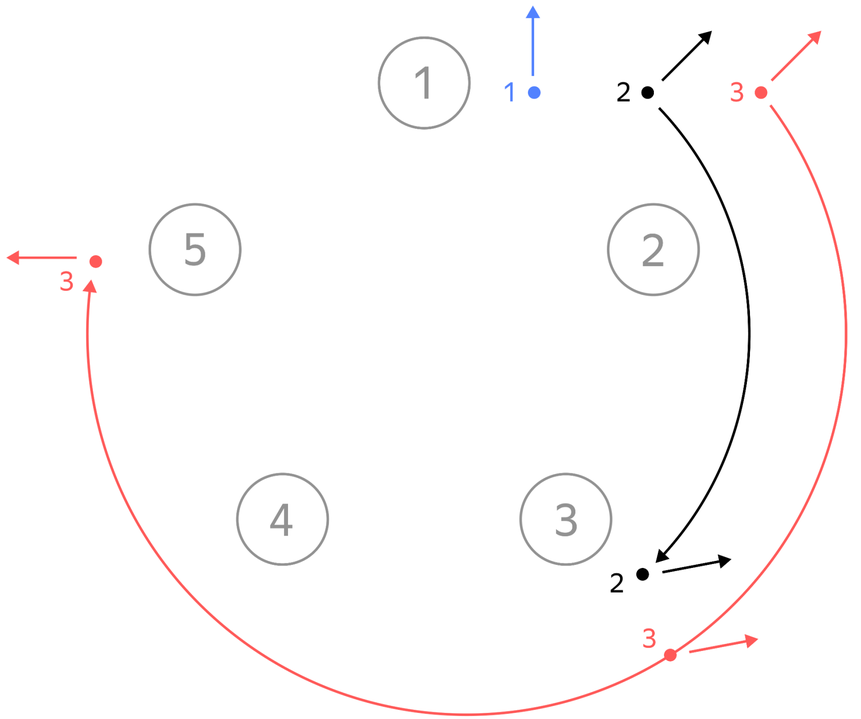
Горячий

Определяем длительную популярность по логам раздачи

Регулярно смотрят на длинном периоде времени

Мегапопулярный контент

- Один ключ (одна единица контента) обслуживается с одного кеша
- Если единица сгенерила N Гбит/сек трафика с кеша за 5-минут, она мегапопулярна
- К обслуживанию такой единицы добавляется еще один кеш в той же локации
- Как только сгенерили 2N Гбит/сек — добавляем еще один кеш
- Уменьшаем кол-во кешей в обратном порядке
- Consistent hashing всегда применяется при выборе следующего кеша



1. Эволюция CDN
2. Варианты от Яндекса
3. Как храним и раздаем
4. Нюансы отсутствия L3-балансеров
5. Балансировка пользователей по CDN
6. Следим за перегрузкой стыков
7. Популярность контента
8. Бонус-трек

Управляемая деградация

Если потребуется снизить потребление трафика клиентами

Вдруг CDN закончится

Soft-деградация — говорим плееру играть низкое разрешение по-умолчанию

Не работает на iOS и телевизорах

Hard-деградация — убираем из плейлистов высокие разрешения

nadersteel.com/wp-content/uploads/pipeline-2-1024x1024.jpg

Минутка рекламы

- Активно сотрудничаем с операторами и ставим кеши
- По списку AS оператора можем вычислить объем горячего трафика в него
- При наличии 10+Гбит/сек горячего трафика — отправляем кеши

- Контакты: noc@yandex.net

supermicro.com/en/products/system/wio/1u/sys-110p-wtr

Буду рад продолжить общение и... мы нанимаем! :)

Евгений Зайцев
Ведущий технический менеджер проектов
Yandex Cloud
eightn@yandex-team.ru