

# Сайт тормозит. Что делать?



ИСПОЛЬЗОВАТЬ СПРАЙТЫ

БАНДЛИТЬ КОД

РАЗБИВАТЬ НА ЧАНКИ

PREFETCH \ PRELOAD

ОТДАВАТЬ ТОЛЬКО НУЖНЫЙ CSS

SSR

SERVER-SIDE PUSH

ВОПРОС  
НА СОБЕСЕДОВАНИИ

ИСПОЛЬЗОВАТЬ СПРАЙТЫ

БАНДЛИТЬ КОД

РАЗБИВАТЬ НА ЧАНКИ

PREFETCH \ PRELOAD

ОТДАВАТЬ ТОЛЬКО НУЖНЫЙ CSS

SSR

SERVER-SIDE PUSH

ВОПРОС  
НА СОБЕСЕДОВАНИИ





СКОЛЬКО ДОЛЖЕН ГРУЗИТЬСЯ САЙТ?

БЫСТРО? ОЧЕНЬ БЫСТРО?

0.1 СЕКУНДЫ

НАЧНЕМ С ЦИФР

1 СЕКУНДА

НАЧНЕМ С ЦИФР



10 СЕКУНД

НАЧНЕМ С ЦИФР

1 МИНУТА

НАЧНЕМ С ЦИФР

10 МИНУТ

НАЧНЕМ С ЦИФР

СЕТЬ

ПРОЦЕССОР

ОТ ЧЕГО ЗАВИСИТ ВРЕМЯ  
ЗАГРУЗКИ?

# MOBILE PHONES

128 KBIT/S

384 KBIT/S

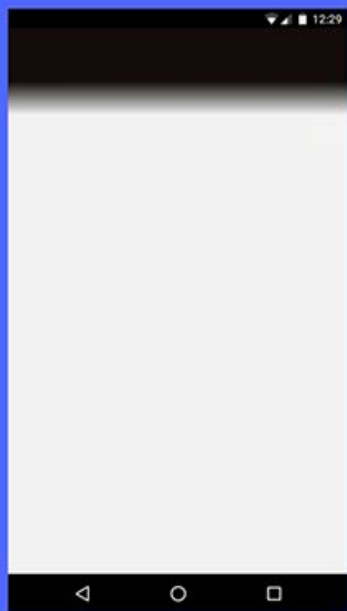
MOBILE PHONES

128 KBIT/S

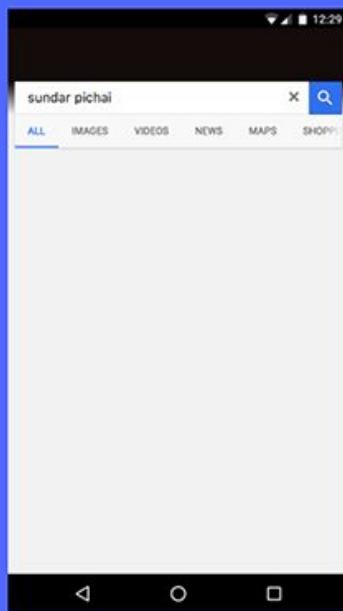
384 KBIT/S



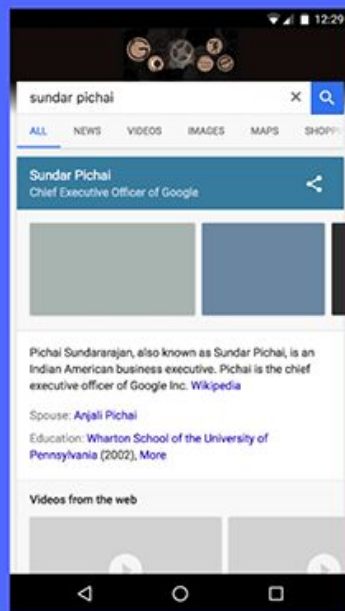
First Paint  
(FP)



First Contentful  
Paint (FCP)

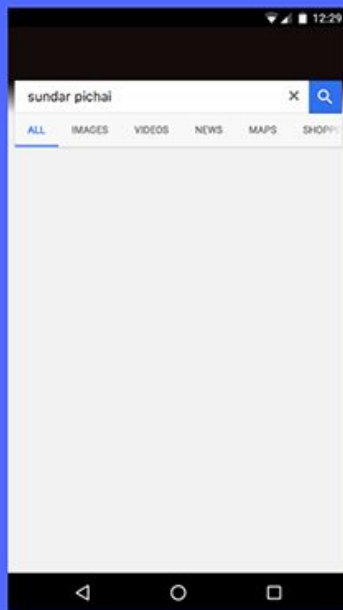
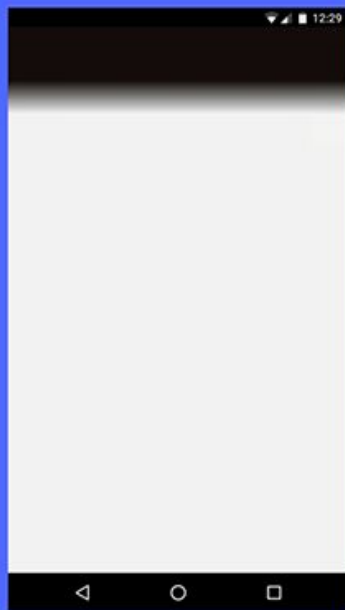


First Meaningful  
Paint (FMP)

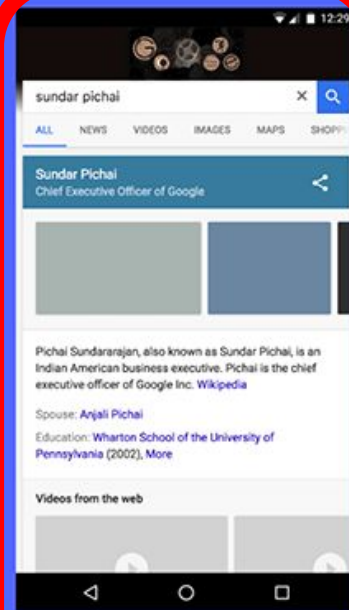
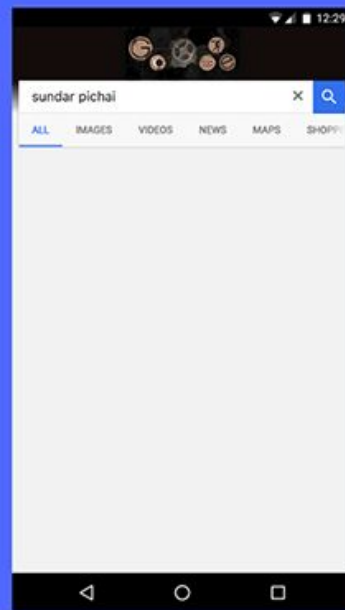




First Paint  
(FP)



First Contentful  
Paint (FCP)



First Meaningful  
Paint (FMP)



SSR

СЕТЬ

КОЛИЧЕСТВО ДАННЫХ

КАК МЫ МОЖЕМ  
ОПТИМИЗИРОВАТЬ FMR?

SSR

СЕТЬ

КОЛИЧЕСТВО ДАННЫХ

КАК МЫ МОЖЕМ  
ОПТИМИЗИРОВАТЬ FMR?

# CLIENT-SIDE RENDERING

- 1) СЕРВЕР ОТПРАВЛЯЕТ  
СТРАНИЦУ-ЗАГЛУШКУ
- 2) СЕРВЕР ОТПРАВЛЯЕТ JS
- 3) КЛИЕНТ ИДЕТ ЗА  
ДААННЫМИ
- 4) КЛИЕНТ ИСПОЛНЯЕТ JS

# CLIENT-SIDE RENDERING

HTML-ЗАГЛУШКУ

2) СЕРВЕР ОТПРАВЛЯЕТ JS

3) КЛИЕНТ ИДЕТ ЗА  
ДААННЫМИ

4) КЛИЕНТ ИСПОЛНЯЕТ JS

5) БРАУЗЕР РЕНДЕРИТ КАДР

# CLIENT-SIDE RENDERING

HTML-ЗАГЛУШКУ

2) СЕРВЕР ОТПРАВЛЯЕТ JS

3) КЛИЕНТ ИДЕТ ЗА  
ДААННЫМИ

4) КЛИЕНТ ИСПОЛНЯЕТ JS

5) БРАУЗЕР РЕНДЕРИТ КАДР

---

# SERVER-SIDE RENDERING

- 1) СЕРВЕР ОТПРАВЛЯЕТ  
СТРАНИЦУ
- 2) КЛИЕНТ РЕНДЕРИТ  
СТРАНИЦУ
- 3) КЛИЕНТ ИДЕТ ЗА JS
- 4) КЛИЕНТ ИСПОЛНЯЕТ JS

# SERVER-SIDE RENDERING

- 1) СЕРВЕР ОТПРАВЛЯЕТ  
СТРАНИЦУ
- 2) КЛИЕНТ РЕНДЕРИТ  
СТРАНИЦУ
- 3) КЛИЕНТ ИДЕТ ЗА JS
- 4) КЛИЕНТ ИСПОЛНЯЕТ JS

# SERVER-SIDE RENDERING

ВЫИГРЫШ: ОТСУТСТВИЕ JS  
БАНДЛА И ВРЕМЕНИ  
ИСПОЛНЕНИЯ JS



# SERVER-SIDE RENDERING

ВЫИГРЫШ  $\neq$  TTI - FMP

FMP / SSR

КАК СДЕЛАТЬ?

NEXT.JS

GATSBY

REACT-STATIC

СВОЕ РЕШЕНИЕ

# СВОЕ РЕШЕНИЕ

NODE.JS — API-GATEWAY

NODE.JS — ONLY

RENDERING

FMP / SSR

РОУТИНГ И ПОХОД ЗА ДАННЫМИ

ПОДГОТОВКА ДАННЫХ

КЛИЕНТСКИЙ КОД БУДЕТ ИЗМЕНЕН

NODE.JS — API-GATEWAY

FMP / SSR

/RENDER URL

PUT ЗАПРОС НА РЕНДЕР-СЕРВЕР ДЛЯ  
ПОЛУЧЕНИЯ ДАННЫХ

ПРАКТИЧЕСКИ НЕИЗМЕНЕННЫЙ КЛИЕНТ

NODE.JS — ONLY  
RENDERING

SSR

СЕТЬ

КОЛИЧЕСТВО ДАННЫХ

КАК МЫ МОЖЕМ  
ОПТИМИЗИРОВАТЬ FMR?

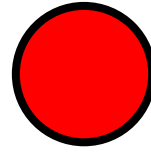
# СЕТЬ

- 1) HTTP/2
- 2) СЖАТИЕ
- 3) CDN

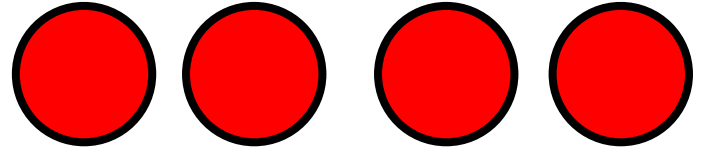
# HTTP/2

- 1) МУЛЬТИПЛЕКСИНГ
- 2) СЖАТИЕ ЗАГОЛОВКОВ
- 3) SERVER-PUSH

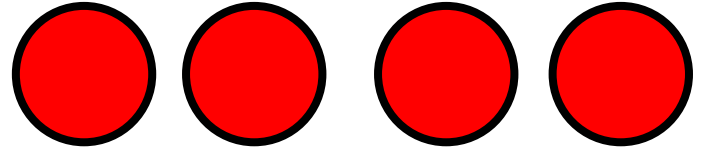




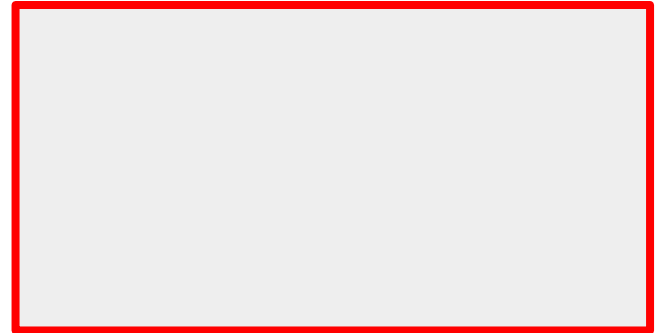
# МУЛЬТИПЛЕКСИНГ



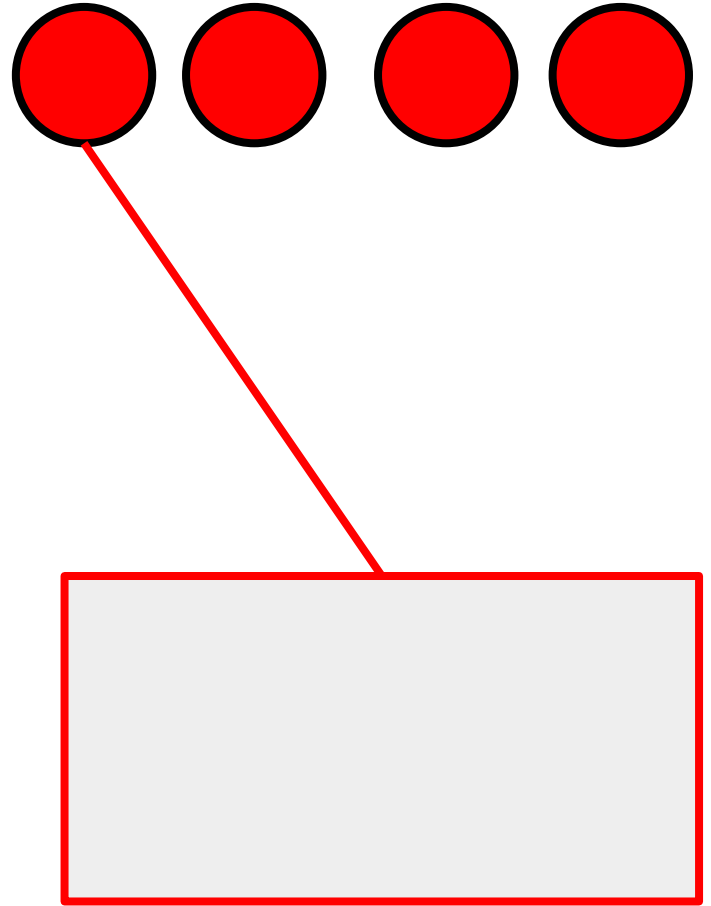
# МУЛЬТИПЛЕКСИНГ



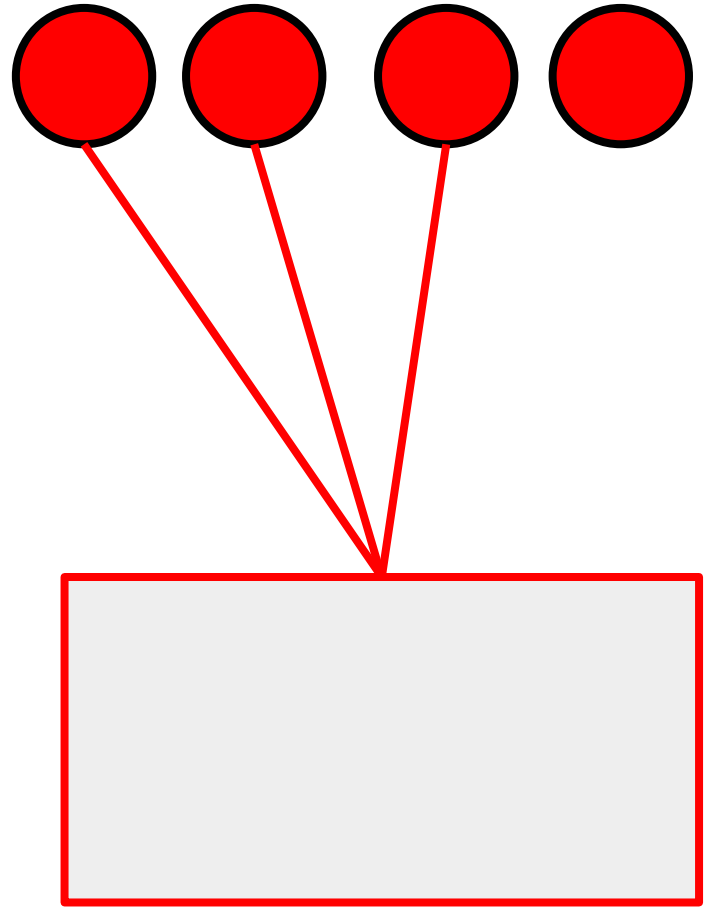
# МУЛЬТИПЛЕКСИНГ



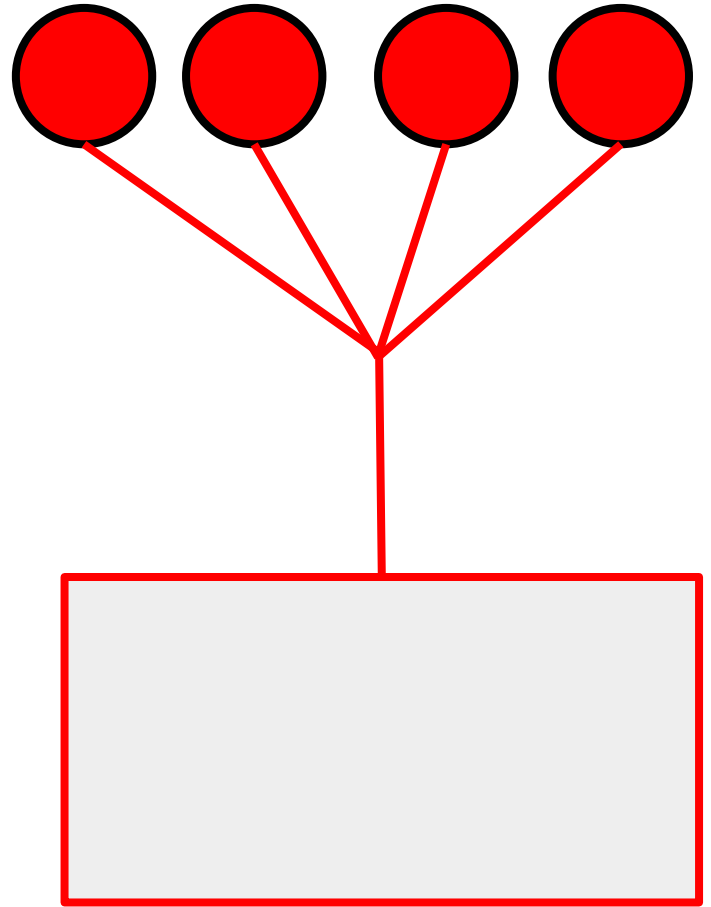
# МУЛЬТИПЛЕКСИНГ



# МУЛЬТИПЛЕКСИНГ



# МУЛЬТИПЛЕКСИНГ



HPACK

СЖАТИЕ ЗАГОЛОВКОВ

## НРАСК

- 1) СТАТИЧЕСКИЙ СЛОВАРЬ
- 2) ДИНАМИЧЕСКИЙ
- 3) HUFFMAN ENCODING

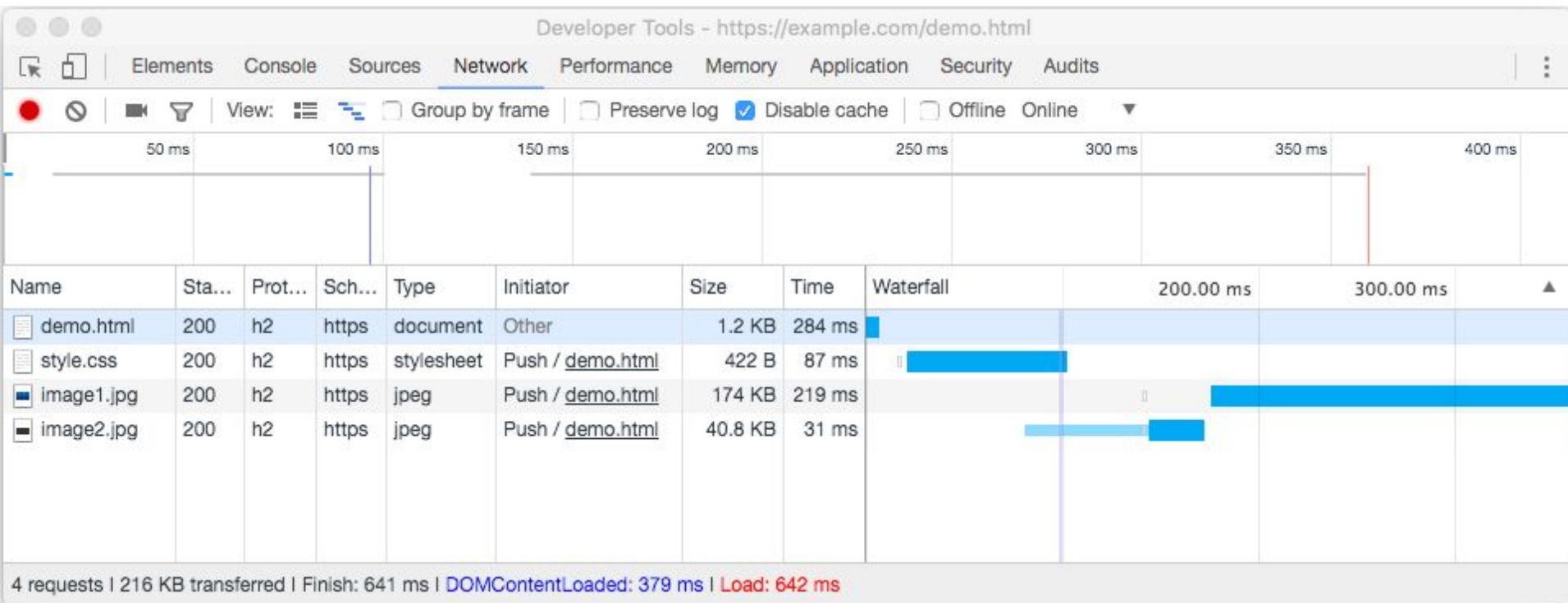


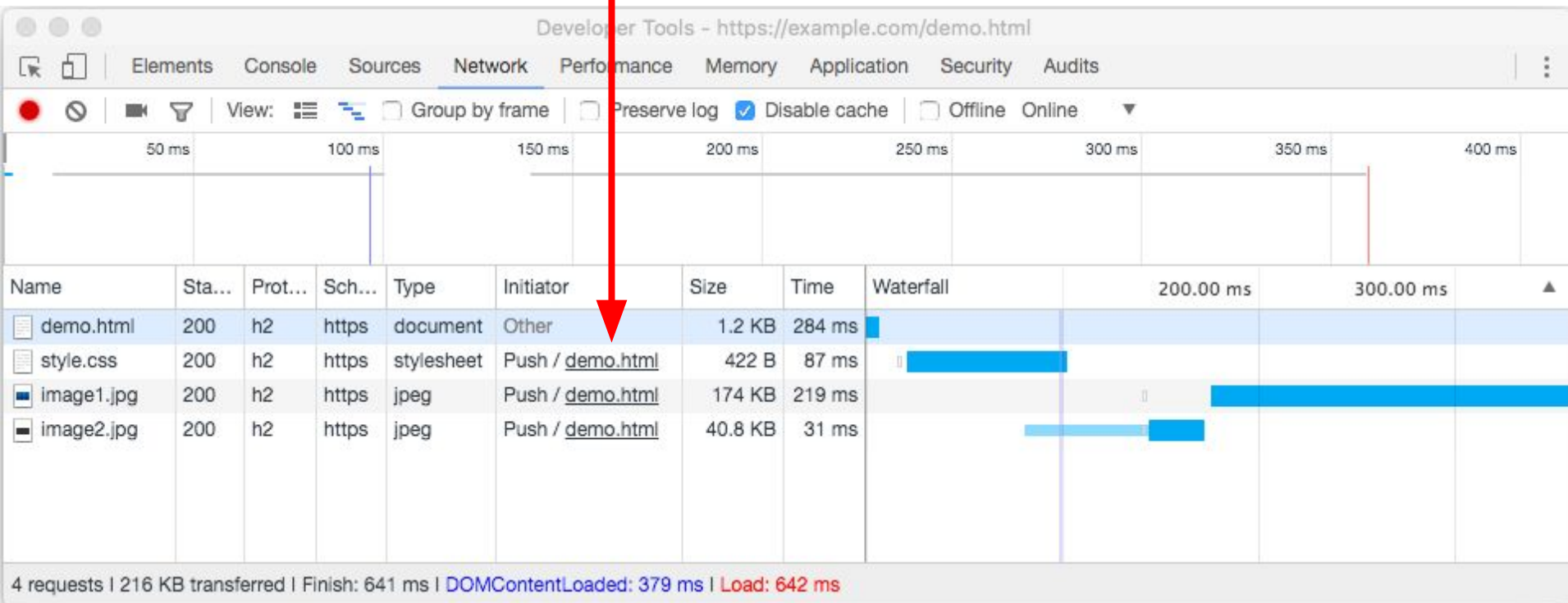
# HTTP/2

- 1) МУЛЬТИПЛЕКСИНГ
- 2) СЖАТИЕ ЗАГОЛОВКОВ
- 3) SERVER-PUSH

## SERVER-PUSH

```
location = /index.html {  
    http2_push /style.css;  
    http2_push /image1.jpg;  
    http2_push /bundle.js;  
}
```





СЕТЬ

1) HTTP/2

2) СЖАТИЕ

3) CDN

GZIP VS BRO

СЖАТИЕ

FMP / СЕТЬ / СЖАТИЕ

GZIP

BRO

СТЕПЕНЬ СЖАТИЯ

FMP / СЕТЬ / СЖАТИЕ

GZIP

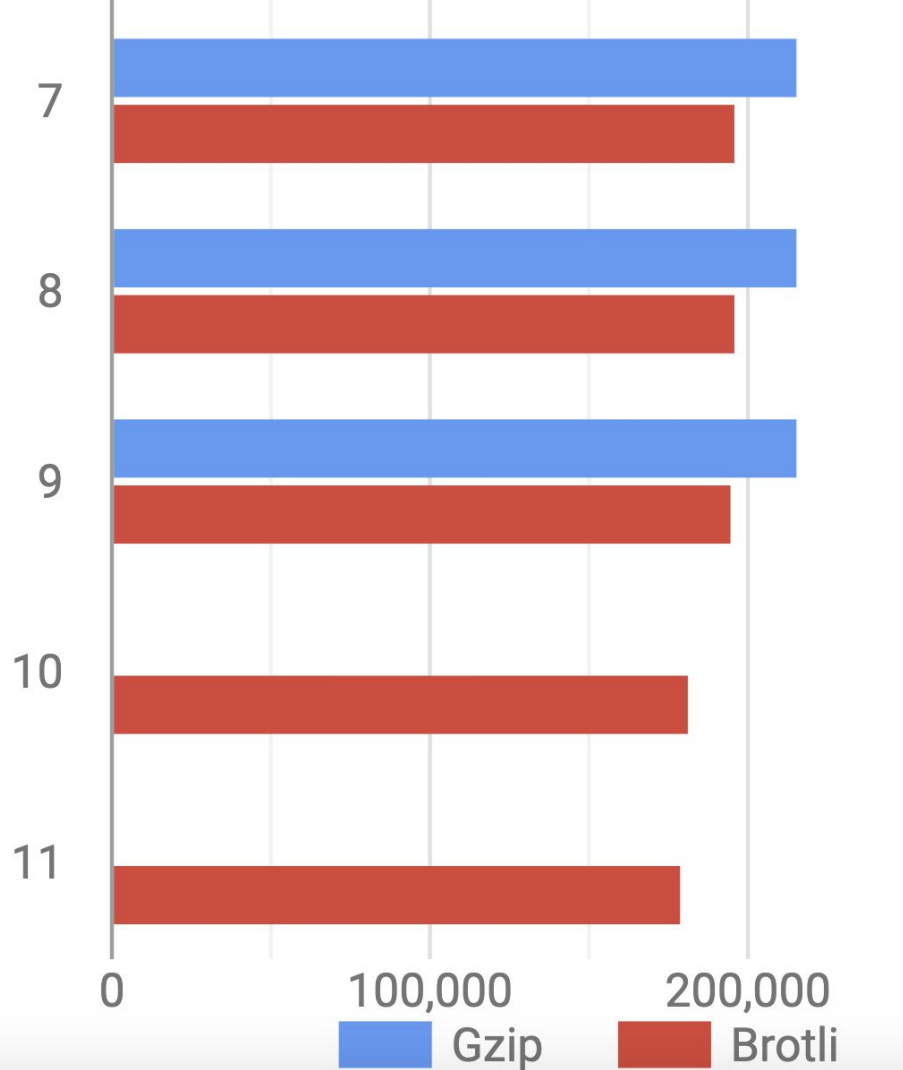
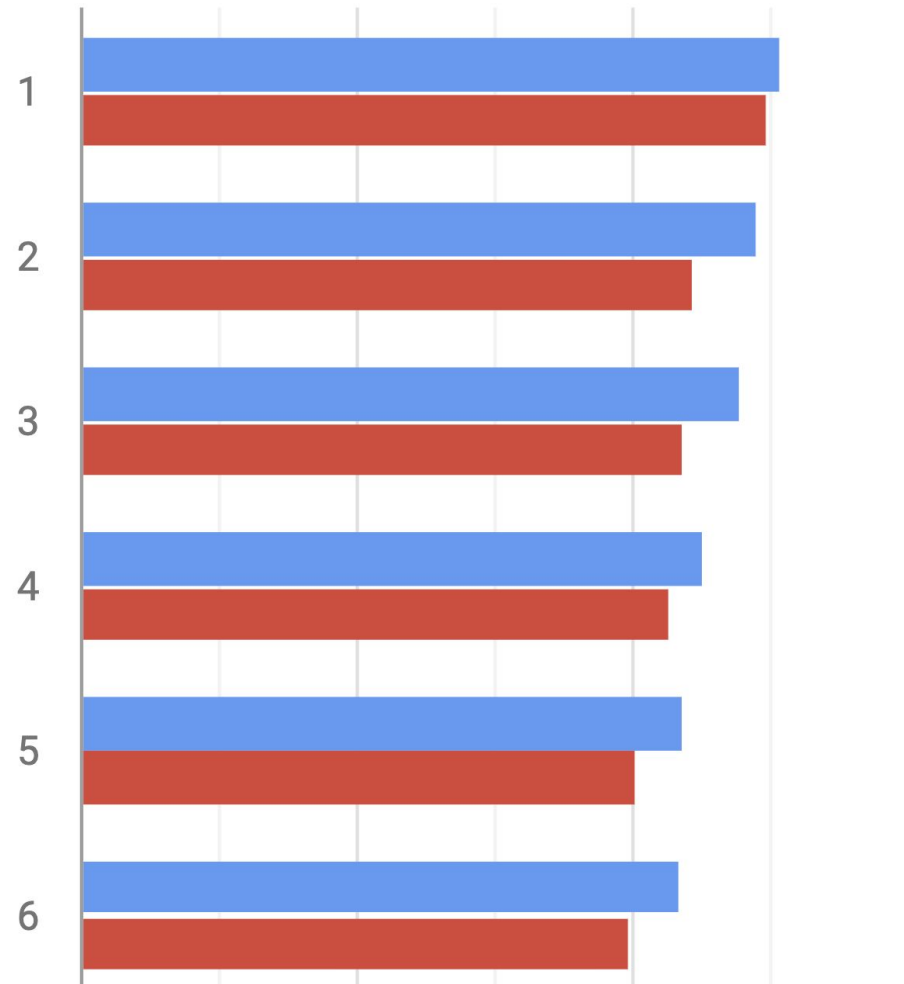
BRO

СТЕПЕНЬ СЖАТИЯ

<https://tools.paulcalvano.com/compression.php>



## Bytes Per Compression Level



FMP / СЕТЬ / СЖАТИЕ

GZIP

BRO

СТЕПЕНЬ СЖАТИЯ

: (

НА 12-15% ВЫШЕ НА 5-М УРОВНЕ

СКОРОСТЬ СЖАТИЯ

ВЫШЕ

: (

FTP / СЕТЬ / СЖАТИЕ

GZIP

BRO

СТЕПЕНЬ СЖАТИЯ

:(

НА 11-14% ВЫШЕ НА 5-М УРОВНЕ

СКОРОСТЬ СЖАТИЯ

ВЫШЕ

СЖИМАЙТЕ ЕДИНОРАЗОВО

СКОРОСТЬ ПАРСИНГА

НЕ ИЗМЕНИЛАСЬ

НЕ ИЗМЕНИЛАСЬ

НАСКОЛЬКО ЭТО ВЫГОДНО? НА ПРИМЕРЕ  
I.NH.RU/VENDORS.JS

$$657 * 1.12 - 657 == 78.84 \text{ КБАЙТ}$$

FTP / СЕТЬ / СЖАТИЕ

GZIP

BRO

СТЕПЕНЬ СЖАТИЯ

:(

НА 11-14% ВЫШЕ НА 5-М УРОВНЕ

СКОРОСТЬ СЖАТИЯ

ВЫШЕ

СЖИМАЙТЕ ЕДИНОРАЗОВО

СКОРОСТЬ ПАРСИНГА

НЕ ИЗМЕНИЛАСЬ

НЕ ИЗМЕНИЛАСЬ

ЭТО ТВОЙ BRO :)

СЕТЬ

1) HTTP/2

2) СЖАТИЕ

3) CDN

SSR

СЕТЬ

КОЛИЧЕСТВО ДАННЫХ

КАК МЫ МОЖЕМ  
ОПТИМИЗИРОВАТЬ FMR?

FMP / КОЛИЧЕСТВО ДАННЫХ

КОЛИЧЕСТВО ДАННЫХ

- 1) РАЗНЫЕ ЧАНКИ
- 2) PREFETCH/PRELOAD
- 3) CRITICAL-CSS И  
СТРАНИЧНЫЙ CSS



КОЛИЧЕСТВО ДАННЫХ

- 1) РАЗНЫЕ ЧАНКИ
- 2) PREFETCH/PRELOAD
- 3) CRITICAL-CSS И  
СТРАНИЧНЫЙ CSS

НЕТ ИЗОЛЯЦИИ СТИЛЕЙ  
ИЗ КОРОБКИ, ИЗ-ЗА ЧЕГО  
СТАНОВИТСЯ СЛОЖНЕЕ  
ПОДЕЛИТЬ НА ЧАНКИ

В КАКОМ СЛУЧАЕ РАЗДЕЛЯТЬ  
СЛОЖНО?

FMP / КОЛИЧЕСТВО ДАННЫХ

CSS, ИСПОЛЬЗУЕМЫЙ НА  
СТРАНИЦЕ

[https://github.com/theKashey/  
used-styles](https://github.com/theKashey/used-styles)

## CSS, ИСПОЛЬЗУЕМЫЙ НА СТРАНИЦЕ

1. ВЫБРАТЬ ВСЕ CLASS ИЗ CSS ФАЙЛОВ
2. ВЫБРАТЬ ВСЕ КЛАССЫ ИЗ HTML
3. ОПРЕДЕЛИТЬ ИСПОЛЬЗУЕМЫЕ СТИЛИ
4. ОПРЕДЕЛИТЬ СТИЛИ, КОТОРЫЕ НУЖНО ПОСТАВИТЬ
5. ВЫДАЕМ ТОЛЬКО НУЖНЫЕ СТИЛИ :)

МЕНЬШЕ ИСПОЛНЕНИЯ КОДА

МЕНЬШЕ LAYOUT

CACHING

HOT RUN

КАК МЫ МОЖЕМ  
ОПТИМИЗИРОВАТЬ TTI?

TTI / CACHING

ВРОДЕ БЫ ВСЕ ПРОСТО, НО!

CACHING

TTI / CACHING

ЛУЧШЕ НЕ ПИСАТЬ КОД

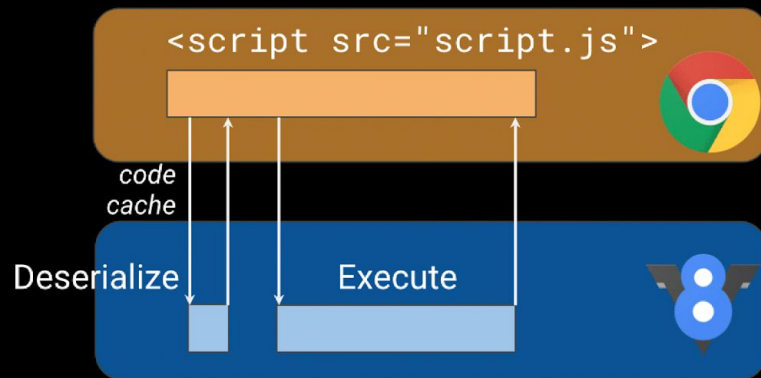
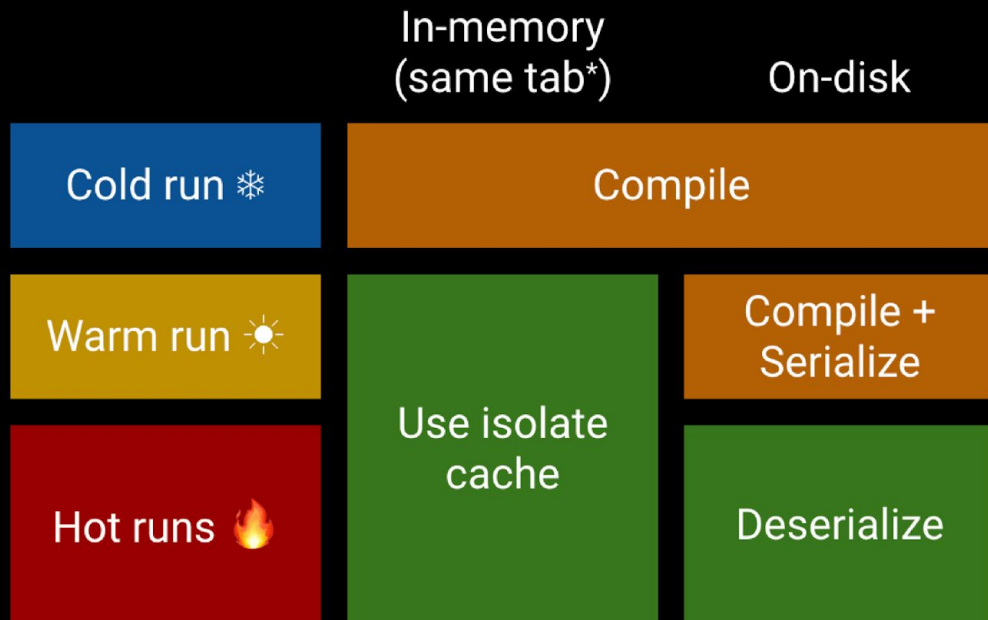
“МАГИЧЕСКИЙ” ICB

JSON.PARSE БЫСТРЕЕ, ЧЕМ JAVASCRIPT

CHROME://TRACING

CACHING

# CODE CACHING IN V8



*\* actually "same V8 isolate", which means same renderer process and same thread*

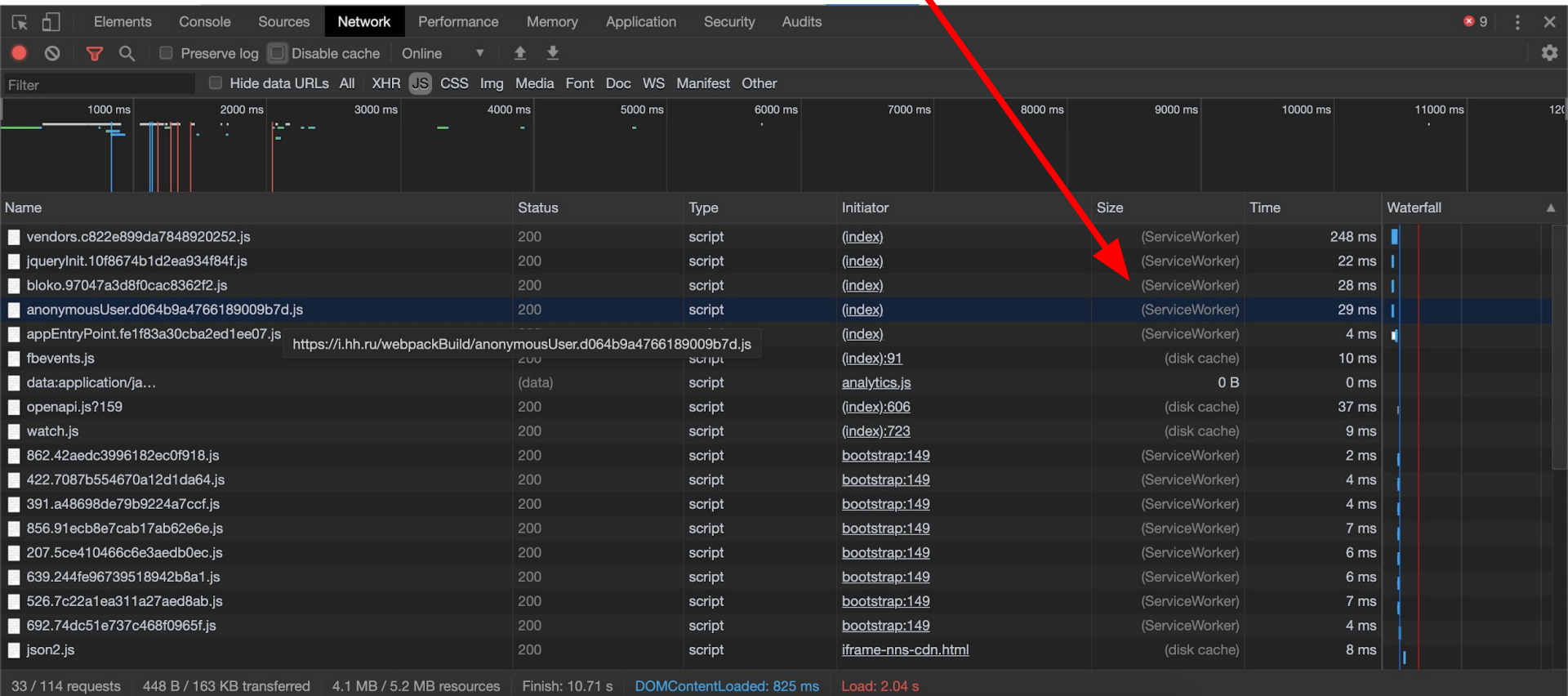


TTI / HOT RUN

3-й запуск в течение последних 72  
часов

SERVICE WORKER

КАК ДОБРАТЬСЯ ДО HOT RUN



- 1) SSR
- 2) РАЗДЕЛЯЙТЕ СТРАНИЧНЫЙ CSS
- 3) ИСПОЛЬЗУЙТЕ HTTP2/0 И BROTLI
- 4) КЕШИРУЙТЕ КОД
- 5) ОБРАЩАЙТЕ ВНИМАНИЕ НА PROFILER  
И CHROME://TRACING

ИТОГО

СПАСИБО  
ЗА ВНИМАНИЕ!

МОСТОВОЙ НИКИТА (@XNIMORZ)  
NIK.MOSTOVOY@GMAIL.COM

