# <https://lifehacker.ru/>

\*Зеленым выделил на что повлияет.

**400 requests**

**22.6 MB transferred**

**DomContentLoaded: 1.43**

**Load: 11.52**

Далее оптимизации буду перечислять в порядке критичности (по моему мнению)

1. **Картинки**

Размер: 21.6 MB

Запросов: 135

А) Уменьшить число картинок (сделать меньше ленту, очень длинная, достаточно одного двух экранов) – уменьшим количество запросов, ожидание ответа, время декодирования

Б) Вставлены очень жирные gif и png файлы (лучше не вставлять, а если очень нужно оптимизируем) – ожидание ответа, декодирование

(например <https://lifehacker.ru/wp-content/uploads/2016/07/moooney_1469626535-630x315.gif> 3.8MB)

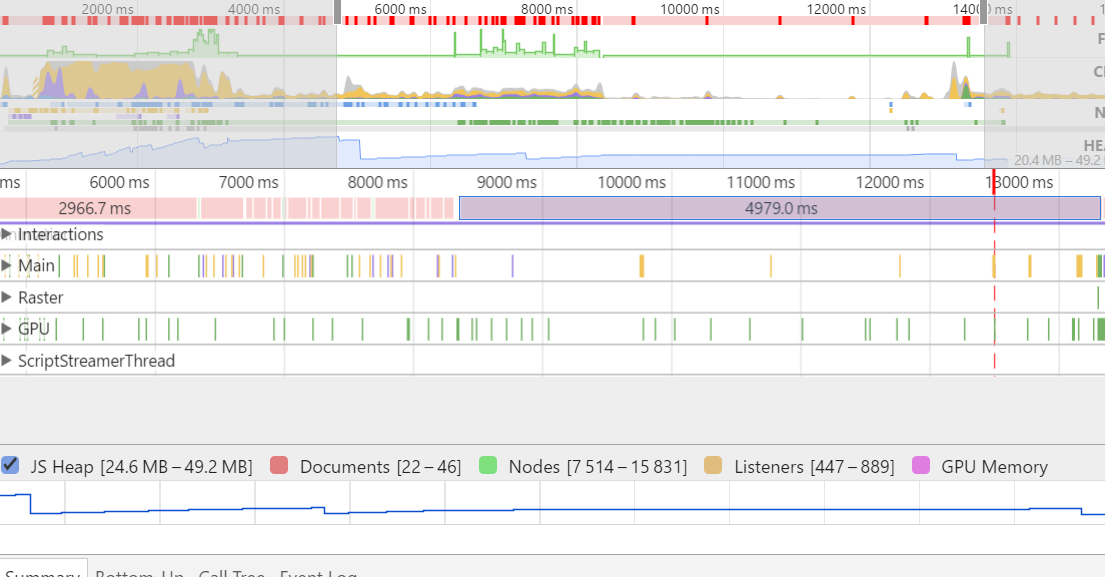
В) Адаптивности нет, использовать разные картинки для разных размеров экранов, а еще лучше резать на сервере в зависимости от размера экрана клиентского устройства) - ожидание ответа, декодирование

Г) Сделать из svg иконок, иконочный шрифт - уменьшим количество запросов

Примеры: <https://lifehacker.ru/wp-content/themes/lifehacker/static/img/category-icons/sport.svg>

<https://lifehacker.ru/wp-content/themes/lifehacker/static/img/category-icons/food.svg>

д) Слишком большие картинки очень медленно рисуются, видно, как проседает fps на графике



д) запросы непонятного вида (может это маяк, старый вид опросы сервера), в общем надо разобраться и может их убрать - уменьшим количество запросов

пример: <https://sync.teads.tv/um?eid=45&uid=0a4f554edfecb137df67fc59035abebf>

*PS: сайт адаптивный, но на телефоне он у меня так и не заработал, первый раз картинки не загружались второй раз был, просто белый экран*

1. **CSS**

Размер 75.8 KB

Запросов 20

А) Необходимо объединить в 1 файл - уменьшим количество запросов

Б) Минифицировать (часть не минифицирована) - ожидание ответа

В) Все они являются блокирующими (стоит media=all) возможно стоит разделить на два, например, файла, и часть реже используемую загружать не сразу (media=none/all) - ожидание ответа

Г) Удалить не используемые правила (3924 rules (89%) of CSS not used by the current page.) - ожидание ответа

1. **JS**

Размер: 766 KB

Запросов: 151

А) Необходимо объединить - уменьшим количество запросов

Б) Минифицировать - ожидание ответа

В) Все они блокирующие поэтому стоит перенести из шапки вниз (ну собственно после объединения) - рендер

Г) Я думаю стоит поставить этому бандлу async ибо нет причин делать его здесь блокирующим, сайт просто с контентом - рендер

1. **Шрифты**

Здесь есть 4 файла с веб шрифтами, часть вверху и являются блокирующими, я бы все же сначала использовал обычный шрифт, потом догружал и при повторном входе использовал (но это уже дело вкуса думаю) - ожидание ответа

1. **XHR и другие**

Запросов: около 80

Например https://lifehacker.ru/pocket/?url=https%3A%2F%2Flifehacker.ru%2F2016%2F07%2F14%2Fpokemon-go-tips-and-tricks%2F

Если часть не нужна - уменьшим количество запросов

1. **Минифицировать html страницу** (здесь явно можно уложиться в 14 KB, сейчас 19) - ожидание ответа
2. **Кэшировать на более длинное время файла**( касается бандлов css, js) - уменьшим количество запросов

Например: <https://lifehacker.ru/wp-content/plugins/lh-social-slider/assets/css/style.css?ver=1.10>

1. **Cache-Control:**

max-age=86400

1. **Date:**

Thu, 04 Aug 2016 07:26:07 GMT

1. **Expires:**

Fri, 05 Aug 2016 07:26:07 GMT

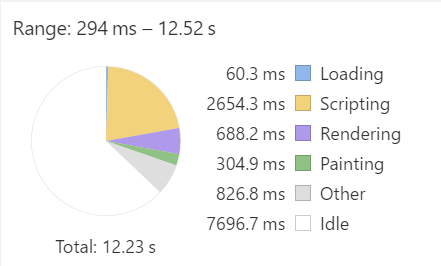
1. **Last-Modified:**

Fri, 29 Jul 2016 15:07:10 GMT

Здесь видно, что последний раз изменяли 23 Июля, но кэш работает 1 день

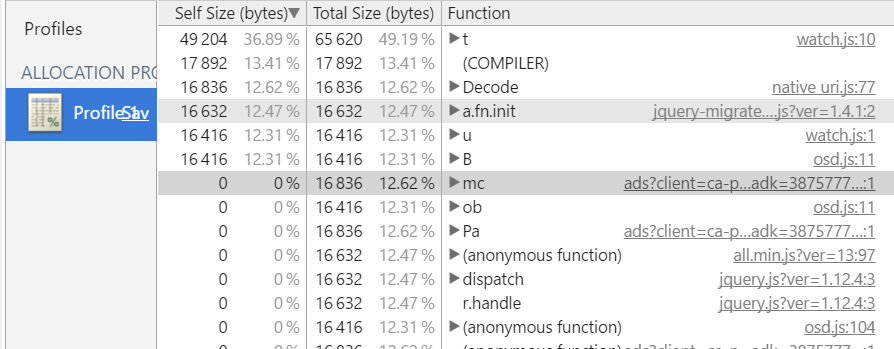
Можно ставить больше, как вариант что меняется часто выносить в отдельный файл, а остальное кэшировать на меньшее время

1. **Оптимизации написанных стилей и скриптов** – декодирование, исполнение, вычиление

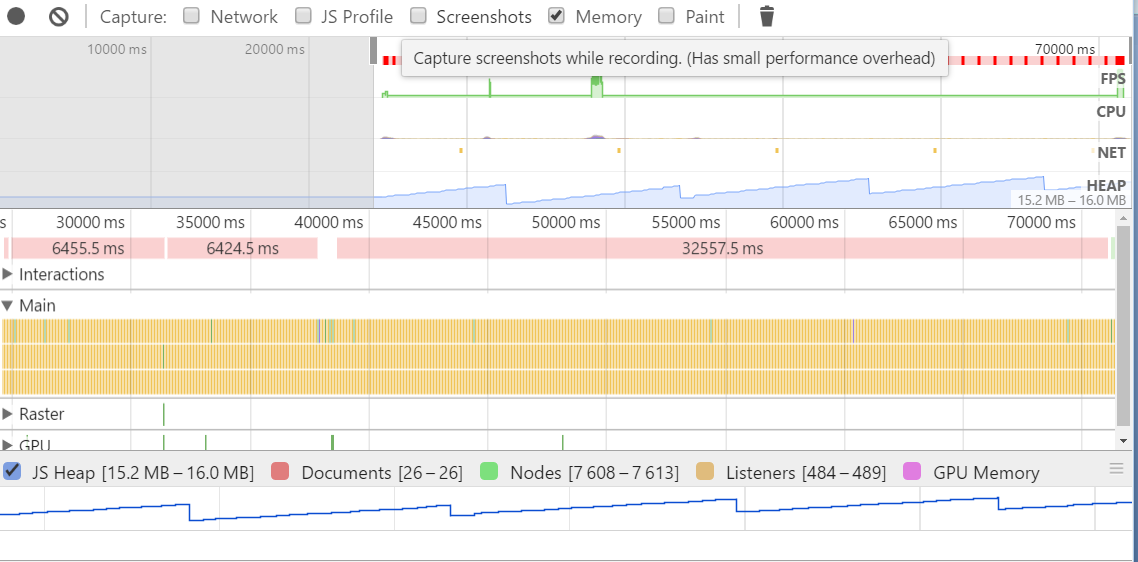


Это более трудозатратно но видно что выполнения скриптов, рендеринг и отрисовка занимают много времени, нужно разбираться во внутреннем устройстве и что то менять

А) Скрипт watch.js очень много ресурсов ест



Б) Судя по этому графику есть утечки памяти



После каждой сборки увеличивается на 100 KB количество памяти

# <http://www.akado.ru/>

**85 requests**

**2.7 MB transferred**

**DomContentLoaded: 2.53**

**Load: 3.72**

1. **Включить gzip** (сразу станет лучше) сжимается здесь тока html страница – ожидание загрузки
2. **Картинки**

Размер: 1.3

Запросов: 47

А) Использовать векторную графику вместо png для каждой иконки, либо объединить в один битмап и управлять переключением в css – количество запросов

1. **JS**

Размер: 997 KB

Запросов: 20

А) Объединить – количество запросов

Б) Минифицировать - ожидание загрузки

В) Я бы вынес инлайн скрипты в файл - количество запросов

Г) Перенести вниз - рендеринг

Д) Использовать async - рендеринг

1. **CSS**

Размер: 265

Запросов: 8

А) Объединить - количество запросов

Б) Минифицировать - ожидание загрузки

В) Удалить лишние правила (1125 rules (72%) of CSS not used by the current page) – декодирование, ожидание загрузки

1. **HTML**

А) Минифицировать - ожидание загрузки

1. **Битые ссылки**

Удалить - количество запросов

GET http://www.akado.ru/js/adfox/adfox.asyn.code.ver3.js

(index):35 GET http://www.akado.ru/js/adfox/adfox.asyn.code.scroll.js

(index):34 GET http://www.akado.ru/js/adfox/adfox.asyn.code.ver3.js

(index):35 GET http://www.akado.ru/js/adfox/adfox.asyn.code.scroll.js

(index):556 GET <http://www.googletagmanager.com/gtm.js?id=GTM-W8D9KM>

**7) Оптимизировать выполнения скриптов и рендеринг**

