

**ТЕХНОЛОГИИ ОПТИМИЗАЦИИ РАБОТЫ САЙТА
НА ПРИМЕРЕ АНАЛИТИЧЕСКОЙ СИСТЕМЫ
ПУБЛИКАЦИОННОЙ АКТИВНОСТИ
ПЕНЗЕНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА**

Р. А. Торопкин, Я. В. Зиновьев, Н. С. Рассказов, М. А. Митрохин

**SITE OPTIMIZATION TECHNOLOGIES ON THE EXAMPLE
OF THE ANALYTICAL SYSTEM OF PUBLISHING ACTIVITY
OF PENZA STATE UNIVERSITY**

R. A. Toropkin, Ya. V. Zinov'ev, N. S. Rasskazov, M. A. Mitrokhin

Аннотация. *Предмет и цель работы.* Рассматриваются существующие методы оптимизации пользовательского интерфейса в web-приложениях, способствующие минимизации времени, затраченного на отображение страниц. *Методы.* Акцентируется внимание на сравнении MPA и SPA технологий и обосновывается целесообразность применения инструментов для сборки и компоновки проекта. Описываются возможности оптимизации загрузки медиафайлов и компонентов с помощью серверного рендеринга и отложенной загрузки. Предлагаются способы повышения эффективности загрузки медиаконтента. *Результаты и выводы.* В результате исследования было установлено, что существующие технологии оптимизации загрузки сайтов направлены в основном на работу с мультимедийным контентом. Практическое применение описанных методов и технологий в ходе разработки системы аналитической публикационной активности Пензенского государственного университета показало, что в общем случае время загрузки страниц сокращается на 40 % по сравнению с неоптимизированным вариантом.

Ключевые слова: web-приложение, оптимизация, отложенная загрузка, JavaScript, Single Page Applications, фреймворк, рендеринг, Chart.js, кэширование, Webpack.

Abstract. *Subject and goals.* This article discusses existing methods for optimizing the user interface in web applications that help minimize the time spent on displaying pages. *Methods.* The article focuses on the comparison of MPA and SPA technologies and justifies the feasibility of using tools for project Assembly and layout. It describes how to optimize the loading of media files and components using server rendering and deferred loading. Ways to optimize the loading of media content are suggested. *Results and conclusions.* As a result of the research, it was found that the existing technologies for optimizing site loading are mainly aimed at working with media content. The practical application of the described methods and technologies during the development of the PSU analytical publication activity system has shown that page loading time is reduced by about 40 % compared to the non-optimized version.

Keywords: web application, optimization, deferred loading, JavaScript, Single Page Applications, framework, rendering, Chart.js, caching, Webpack.

Введение

Оптимизация производительности web-сайтов является приоритетом при их администрировании в современных условиях борьбы за трафик. Согласно отчету о цифровой сфере Digital 2020 число активных пользователей

интернета достигло отметки 4,5 млрд человек [1]. Растет и количество web-сайтов, а вместе с тем развивается и их структура. Из простых ресурсов, содержащих статический контент, web-приложения эволюционировали до сложно сконструированных платформ, решающих большой спектр задач в области визуализации данных сети Internet.

Избыток информации приводит к тому, что пользователь становится более прихотливым в процессе поиска информации и к современным сайтам предъявляются все более высокие требования, в том числе и по скорости загрузки. Исследование компании Radware показало, что 47 % пользователей ожидают, что страница загрузится не более чем за 2 с, а если время загрузки составляет более 3 с, то 40 % клиентов сразу покинут сайт [2]. Поэтому задача оптимизации скорости загрузки сайта является одной из наиболее приоритетных.

На рынке существует множество способов, позволяющих уменьшить время загрузки страниц, но все они решают задачи:

- уменьшения объема загружаемых данных;
- сохранения части информации на устройстве пользователя или кэширования;
- «визуальной» оптимизации web-страницы.

В ходе решения данных задач рассмотрим наиболее эффективные варианты клиентской оптимизации и их применение на примере аналитической системы публикационной активности ПГУ.

Материалы и методы

Современные web-приложения используют все больше вычислений на стороне клиента, что вместе со сторонними библиотеками значительно увеличивает объем передаваемой на пользовательское устройство информации. Данную проблему помогают решить минификация кода и применение различных инструментов для сборки, таких как gulp, Webpack. Так называемые «сборщики» обладают огромным функционалом по оптимизации web-приложения, который позволяет:

- указывать среду, это означает, что при компиляции в финальную сборку не будут включаться различные артефакты тестирования и разработки;
- минимизировать исходные JavaScript-файлы, а также различные версии исходной карты (source map), размер которой также возможно уменьшить до двух раз;
- использовать различные плагины для оптимизации сборки, дающие возможность формировать .gz файлы, импортировать только отдельные модули из библиотек или дедуплицировать похожие файлы. Наиболее популярные из них: compression-webpack-plugin, dedupe-plugin, uglifyjs-plugin, ignore-plugin;
- анализировать размеры используемых зависимостей;
- создавать дерево зависимостей и определять необходимые ресурсы для каждой страницы. Данный подход к оптимизации доступен для React и Vue и позволяет значительно ускорить загрузку каждой страницы.

Серьезным шагом в развитии web-технологий стало использование Single Page Applications (SPA), т.е. разработка таких сайтов, в ходе взаимодействия с которыми происходит загрузка только необходимой информации, а не всей страницы целиком. Классические многостраничные web-ресурсы (Multi Page Application (MPA)) отправляют запрос при каждом незначительном изменении на странице, что приводит к ее принудительному обновлению. Single Page Applications устроены немного иначе, они используют динамическое обновление DOM-дерева, что позволяет исключить перезагрузку web-страницы.

Сравнение технологий многостраничных и одностраничных web-приложений приведено на рис. 1.

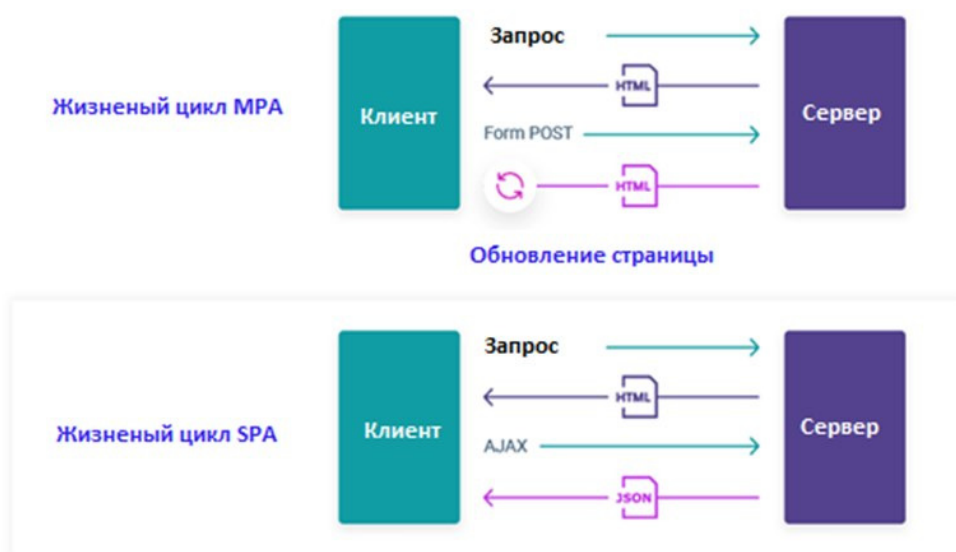


Рис. 1. Сравнение Multi Page Application и Single Page Applications

Наиболее популярны SPA-решения с использованием таких JavaScript библиотек и фреймворков, как React.js, Vue.js, Angular.js. Например, во Vue.js и React технология SPA организована следующим образом: происходит копирование DOM-дерева, его обработка и сравнение с исходным вариантом, далее фреймворк заменяет части только тех элементов, которые отличаются от первоначальной структуры DOM-дерева.

В последнее время набирает популярность «отложенная загрузка» – оптимизация загрузки медиафайлов и компонентов, не критичных для отображения web-страницы и взаимодействия с интерфейсом. Чаще всего «ленивая загрузка» применяется для элементов, расположенных за линией видимости пользователя и отображаемых только после прокрутки. Отложенная загрузка реализована во многих современных JavaScript фреймворках, таких как Vue.js, Angular и React, где возможна настройка динамической загрузки маршрутов и компонентов. Оценить, насколько эффективным окажется применение отложенной загрузки изображений возможно для конкретного сайта, проверив его на ресурсе Google PageSpeed Insights (рис. 2).

Оптимизация — Эти рекомендации могут помочь вам ускорить загрузку страницы. Они не влияют на показатель производительности [напрямую](#).

Возможности	Приблизительная экономия
▲ Отложите загрузку скрытых изображений	8,16 s
▲ Используйте современные форматы изображений	5,36 s
▲ Сократите время ответа сервера (время до получения первого байта)	2,75 s
▲ Настройте эффективную кодировку изображений	1,31 s
■ Удалите неиспользуемый код CSS	0,15 s

Рис. 2. Рекомендации Google PageSpeed Insights

Также рекомендуется использование экономичных форматов файлов, позволяющих сократить объем передаваемого трафика. Например, формат WebP дает возможность экономить до 34 % при конвертировании картинки из JPG и 45 % для PNG [3]. Поддержка данного формата уже присутствует у 79,2 % пользователей интернета. Результаты анализа представлены на рис. 3.



Рис. 3. Результат анализа наличия поддержки браузерами формата WebP

Еще одним способом, позволяющим уменьшить объем загружаемых файлов, а также снизить нагрузку на устройство при рендеринге страницы, является использование серверного рендеринга, при этом вся нагрузка по отображению страницы ложится на сервер, а ресурсы клиентского устройства освобождаются для других процессов [4]. Скорость загрузки при серверном рендеринге увеличивается до двух раз, так как отсутствует обработка кода JavaScript процессором, а загружаются только текст и ссылки. Схема отображений страниц представлена на рис. 4.

Одним из прогрессивных способов кэширования является использование «сервис-воркеров». В данном случае сайт разделяется на две части: со статическим и динамическим контентом, при этом при повторной загрузке, скачивается только часть с уникальным контентом и объединяется со статическим [5]. Сравнение скорости загрузки по метрике First Contentful Paint, характеризующей скорость загрузки на сайте первых элементов, показало снижение нагрузки на сетевой трафик на 47,6 % и повышение скорости отображения первых элементов на 40 % (рис. 5).

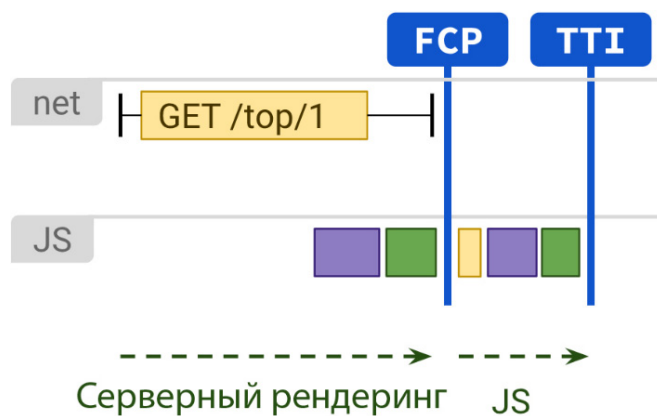


Рис. 4. Загрузка страницы при серверном рендеринге

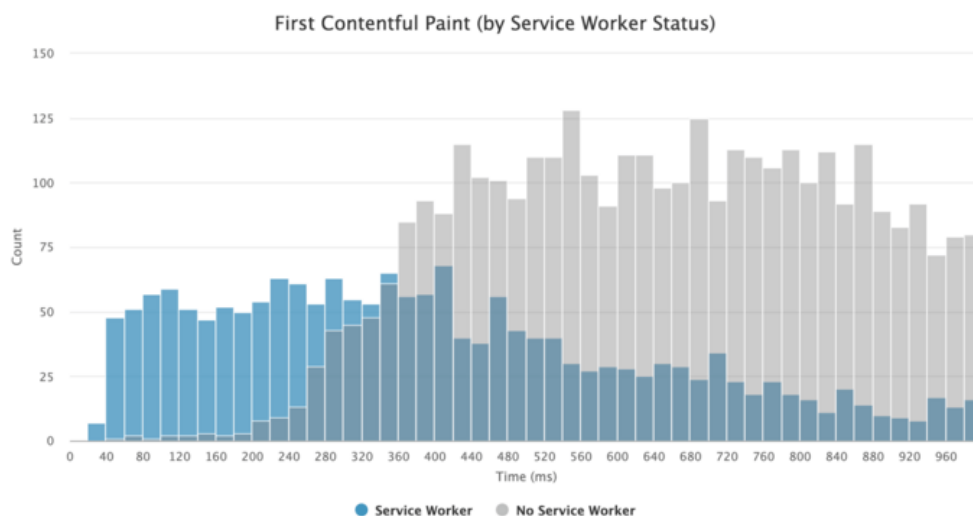


Рис. 5. Сравнение скорости рендера элементов на сайте

Результаты

Описанные технологии были применены в аналитической системе публикационной активности ПГУ. Современные аналитические системы содержат множество числовых данных, оптимальным представлением для которых является использование графиков и диаграмм. С технической точки зрения визуализация информации посредством диаграмм довольно трудозатратна и требует большого количества вычислительных ресурсов. Данную проблему решает JavaScript библиотека – Chart.js. Использование формата svg в совокупности с частичным аппаратным ускорением позволяет добиться значительного прироста производительности. В библиотеке Chart.js применена технология асинхронного рендеринга, что позволяет не загружать ядро JavaScript продолжительное время. Таким образом, данный модуль позволил внедрить «живые» графики на сайт анализа публикационной активности ПГУ¹ без сильного ущерба производительности. Более того, это дало воз-

¹ При поддержке конкурса «Ректорские гранты – 2020». Номер договора: ХП-107/20.

возможность сократить срок разработки, так на реализацию собственного решения с помощью встроенной библиотеки canvas ушло бы много ресурсов, что является критическим фактором при создании web-приложений.

При разработке высоконагруженных систем немаловажным параметром остается качество сборки web-приложения. Проектируя web-ресурс, разработчик взаимодействует с множеством различных библиотек и модулей, с десятками, а порой и сотнями файлов .js и .css. Появляется очевидная проблема, состоящая в сложности управления порядком загрузки различных файлов и модулей для поддержания работоспособности приложения. Для решения данной проблемы в системе анализа публикационной активности ПГУ был применен статический модульный сборщик для приложений «Webpack». Данный инструмент позволяет выполнять сборку js-файлов в оптимальном, правильном порядке. Стоит отметить, что Webpack v4+ по умолчанию поддерживает минимизацию кода, что в общем случае дало возможность сократить размер файлов сайта от 30 до 40 %, исходя из анализа, представленного на рис. 3. В итоге удалось увеличить время загрузки web-сайта на 40 %.

За счет использования технологии SPA, применяемой в системе публикационной активности ПГУ, удалось значительно повысить производительность конечного web-сайта. По статистике среднее значение глубины просмотра для информационного портала составляет 3–4 страницы [6]. В результате применения Single Page Applications пользователь в среднем получает доступ к информации на 9–12 с раньше по сравнению с Multi Page Application (при средней скорости загрузки страницы в 2 с [7] при использовании МРА). Данная технология позволила в кратчайшие сроки добавить новый модуль выгрузки данных. Данный подход дал возможность:

- избавиться от создания новых страниц;
- не дублировать html/css код;
- сохранить единообразие контента;
- использовать существующий API.

Заключение

Таким образом, построение эффективных высоконагруженных систем требует комплекса мероприятий, направленных на повышение отклика, а также скорости загрузки и отображения конечного web-ресурса. Применение рассмотренных методов оптимизации web-интерфейса позволяет сократить время загрузки страницы на 40 % по сравнению с неоптимизированным вариантом. Минимизация времени отклика способствует более комфортной работе с web-приложением, что наряду с оптимизацией web-интерфейса позволяет сформировать положительный опыт эксплуатации пользователям.

Библиографический список

1. Глобальная статистика интернета на 2020 год: отчет Digital о состоянии цифровых технологий в мире и в России. – URL: <https://www.web-canape.ru/business/internet-2020-globalnaya-statistika-i-trendy> (дата обращения: 24.08.2020).
2. Тамми, Э. Скорость как конкурентное преимущество / Э. Тамми. – URL: https://www.methodlab.ru/articles/time_is_money2 (дата обращения: 24.08.2020).
3. How to Make Your Website Load Faster With WebP Images. – URL: <https://techstacker.com/load-website-faster-with-webp-images/mhzdwjfko9eb4fbep/> (дата обращения: 22.08.2020).

4. Start Performance Budgeting. – URL: <https://medium.com/@addyosmani/start-performance-budgeting-dabde04cf6a3> (дата обращения: 22.08.2020).
5. Как ускорить загрузку сайта и улучшить поведенческие факторы. – URL: <https://ru.megaindex.com/blog/service-workers-site-speed> (дата обращения: 24.08.2020).
6. Глубина просмотра сайта и методы оптимизации. – URL: <http://blog.getgoodrank.ru/glubina-prosmotra-sajta-i-metody-optimizacii/> (дата обращения: 26.08.2020).
7. Оптимальная скорость загрузки сайта. – URL: <https://steptosleep.ru/%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D1%81%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1%82%D1%8C-%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8-%D1%81%D0%B0%D0%B9%D1%82%D0%B0/> (дата обращения: 27.08.2020).

References

1. *Global'naya statistika interneta na 2020 god: otchet Digital o sostoyanii tsifrovyykh tekhnologiy v mire i v Rossii* [Global Internet Statistics for 2020: Digital report on the state of digital technologies in the world and in Russia]. Available at: <https://www.web-canape.ru/business/internet-2020-globalnaya-statistika-i-trendy> (accessed 24.08.2020). [In Russian]
2. Tammi E. *Skorost' kak konkurentnoe preimushchestvo* [Speed as a competitive advantage]. Available at: https://www.methodlab.ru/articles/time_is_money2 (accessed 24.08.2020). [In Russian]
3. *How to Make Your Website Load Faster With WebP Images*. Available at: <https://techstacker.com/load-website-faster-with-webp-images/mhzdwjfko9eb4fbep/> (accessed 22.08.2020).
4. *Start Performance Budgeting*. Available at: <https://medium.com/@addyosmani/start-performance-budgeting-dabde04cf6a3> (accessed 22.08.2020).
5. *Kak uskorit' zagruzku sayta i uluchshit' povedencheskie factory* [How to speed up site loading and improve behavioral factors]. Available at: <https://ru.megaindex.com/blog/service-workers-site-speed> (accessed 24.08.2020). [In Russian]
6. *Glubina prosmotra sayta i metody optimizatsii* [Depth of view of the site and methods of optimization]. Available at: <http://blog.getgoodrank.ru/glubina-prosmotra-sajta-i-metody-optimizacii/> (accessed 26.08.2020). [In Russian]
7. *Optimal'naya skorost' zagruzki sayta* [Optimal site loading speed]. Available at: <https://steptosleep.ru/%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F-%D1%81%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1%82%D1%8C-%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8-%D1%81%D0%B0%D0%B9%D1%82%D0%B0/> (accessed 27.08.2020). [In Russian]

Торопкин Роман Александрович
студент,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: good.romka@yandex.ru

Toropkin Roman Aleksandrovich
student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Зиновьев Ярослав Вячеславович
студент,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: yaroslav005@bk.ru

Zinov'yev Yaroslav Vyacheslavovich
student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Рассказов Никита Сергеевич
студент,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: nikita_serdobsk12@mail.ru

Rasskazov Nikita Sergeyevich
student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Митрохин Максим Александрович
доктор технических наук, профессор,
заведующий кафедрой вычислительной
техники,
Пензенский государственный
университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: vt@pnzgu.ru

Mitrokhin Maksim Aleksandrovich
doctor of technical sciences, professor,
head of sub-department of computer
science,
Penza State University
(40 Krasnaya street, Penza, Russia)

Образец цитирования:

Торопкин, Р. А. Технологии оптимизации работы сайта на примере аналитической системы публикационной активности Пензенского государственного университета / Р. А. Торопкин, Я. В. Зиновьев, Н. С. Рассказов, М. А. Митрохин // Модели, системы, сети в экономике, технике, природе и обществе. – 2020. – № 4 (36). – С. 71–78. – DOI 10.21685/2227-8486-2020-4-7.