

УДК 004.738.5

ВЛИЯНИЕ ПОЛЬЗОВАТЕЛЬСКИХ СКРИПТОВ НА ПРОИЗВОДИТЕЛЬНОСТЬ ВЕБ-ПРИЛОЖЕНИЙ

Омаров Э.Г.

Студент 3 курса направления подготовки

09.03.02 Информационные системы и технологии

Прикладные информационные системы и технологии

СПбГУТ им. проф. М. А. Бонч-Бруевича

Санкт-Петербург, Россия

Аннотация

Статья посвящена анализу влияния пользовательских скриптов на производительность веб-приложений. Рассматриваются механизмы, посредством которых скрипты взаимодействуют с браузером и объектной моделью документа, выявляются основные проблемы, возникающие при интенсивной работе клиентского кода. Особое внимание уделяется вопросам оптимизации, правильного использования ресурсов и современных подходов к обеспечению высокой отзывчивости пользовательских интерфейсов. Обоснована необходимость системного подхода к разработке скриптов для минимизации их негативного воздействия на время загрузки, плавность интерфейса и потребление памяти.

Ключевые слова: пользовательские скрипты, производительность, JavaScript, DOM, оптимизация, веб-приложения.

THE IMPACT OF USER SCRIPTS ON WEB APPLICATION PERFORMANCE

Omarov E.G.

Student of the course of preparation

*09.03.02 Information systems and technologies
including information systems and technologies
SPBSUT named after Prof. M. A. Bonch-Bruевич
St. Petersburg, Russia*

Abstract

The article is devoted to the analysis of the impact of user scripts on the performance of web applications. The mechanisms by which scripts interact with the browser and the document object model are considered, and the main problems that arise during intensive work of the client code are identified. Special attention is paid to issues of optimization, proper use of resources and modern approaches to ensuring high responsiveness of user interfaces. The need for a systematic approach to script development is justified in order to minimize their negative impact on loading time, interface smoothness and memory consumption.

Keywords: user scripts, performance, JavaScript, DOM, optimization, web applications.

Развитие веб-технологий за последние два десятилетия привело к тому, что современные приложения в браузере практически не уступают по сложности традиционному настольному программному обеспечению. Пользовательские скрипты стали неотъемлемой частью веб-программирования, обеспечивая динамическую генерацию контента, интерактивность и индивидуализацию поведения сайтов [1]. Однако увеличение объема и сложности клиентского кода нередко сопровождается негативным эффектом — снижением производительности, что проявляется в увеличении времени загрузки страниц, замедлении отклика интерфейсов и росте потребления ресурсов устройства пользователя [2].

Понимание того, каким образом пользовательские скрипты воздействуют на производительность веб-приложений, является важным условием создания эффективных и удобных для конечного пользователя решений. В этой связи особую актуальность приобретают вопросы выявления ключевых факторов деградации производительности и выработки методов минимизации их влияния.

Пользовательские скрипты оказывают комплексное влияние на разные стадии жизненного цикла веб-страницы. На этапе начальной загрузки браузер может задерживать отображение контента, ожидая загрузки и выполнения скриптов, особенно если они не помечены атрибутами `async` или `defer` [1]. Даже после полной загрузки страницы активность скриптов может вызывать частые перерисовки интерфейса (`reflow` и `repaint`), что приводит к зависаниям и снижению плавности анимации.

Одной из наиболее частых причин ухудшения производительности является неэффективное взаимодействие с DOM. Многократное чтение и изменение элементов документа инициирует перерасчет стилей и повторную отрисовку страницы, что особенно критично на устройствах с ограниченными вычислительными ресурсами [3]. Кроме того, неправильно организованная обработка событий, таких как `scroll` или `resize`, без использования методов оптимизации, приводит к перегрузке главного потока исполнения браузера.

Не следует забывать и о влиянии тяжёлых вычислений, выполняемых непосредственно в основном потоке браузера. Например, сложные алгоритмы обработки данных без использования веб-воркеров могут временно "замораживать" интерфейс, лишая пользователя возможности взаимодействовать с приложением.

Для анализа влияния скриптов на производительность используются встроенные в современные браузеры инструменты профилирования и аудита, такие как Chrome DevTools и Lighthouse. Посредством этих инструментов Дневник науки | www.dnevnika.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

можно получить информацию о времени выполнения скриптов, частоте перерисовок и потреблении памяти [4].

Более глубокой диагностики можно достичь, применяя программные средства измерения, такие как API Performance, который позволяет отслеживать временные метки выполнения операций, и функции requestIdleCallback и requestAnimationFrame, которые помогают организовать выполнение тяжёлых задач в периоды минимальной загрузки основного потока [5].

Проведённые исследования демонстрируют, что без оптимизации большое количество мелких изменений DOM за короткое время значительно увеличивает нагрузку на движок браузера. Так, простой скрипт, добавляющий тысячу элементов в тело документа в реальном времени, может приводить к увеличению времени отклика интерфейса в два-три раза по сравнению с оптимизированной версией, использующей промежуточный контейнер DocumentFragment.

Рациональная организация работы пользовательских скриптов требует применения целого ряда оптимизационных практик. В первую очередь необходимо минимизировать количество операций прямого изменения структуры DOM. Эффективной стратегией является создание и модификация элементов в памяти, с последующим разовым добавлением их в документ.

Также рекомендуется ограничивать количество глобальных обработчиков событий и применять техники «дресселирования» (throttling) и устранения «дребезга» (debouncing) для часто срабатывающих событий. Это позволяет значительно сократить количество обращений к тяжёлым обработчикам в ответ на пользовательские действия.

Организация асинхронной загрузки скриптов посредством атрибутов async и defer позволяет существенно уменьшить время блокировки построения DOM при первичной загрузке страницы. Для задач, требующих больших вычислительных ресурсов, целесообразно использовать веб-
Дневник науки | www.dnevnika.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

воркеры, которые позволяют перенести выполнение тяжёлых вычислений в отдельный поток, не влияя на отзывчивость пользовательского интерфейса.

Наконец, важным элементом оптимизации является минификация и бандлинг кода, а также эффективное кэширование скриптов, что особенно важно для мобильных пользователей с ограниченной пропускной способностью каналов связи.

Таким образом, пользовательские скрипты, являясь мощным инструментом расширения функциональности веб-приложений, могут одновременно стать источником серьёзных проблем производительности. Безответственное или неоптимальное использование скриптов приводит к увеличению времени загрузки страниц, ухудшению отзывчивости интерфейсов и росту потребления ресурсов пользовательских устройств.

Эффективное управление пользовательскими скриптами требует системного подхода, включающего правильную архитектуру клиентского кода, применение современных техник оптимизации и регулярный мониторинг производительности. Следование данным принципам позволяет создавать веб-приложения, которые обеспечивают пользователям быстрый, плавный и приятный опыт взаимодействия.

Библиографический список

1. Оптимизация веб-приложений: на что стоит обратить внимание [Электронный ресурс] — URL: <https://habr.com/ru/companies/mws/articles/768266/> (дата обращения: 25.04.2025).
2. Гридин В.Н., Анисимов В.И., Васильев С.А. Методы повышения производительности современных веб-приложений // Информационные системы и технологии. 2020. N 1. С. 194–198.

3. Ильин А.Ю., Плотников С.Б. Использование предиктивного анализа пользовательского поведения для повышения скорости реактивной загрузки клиентской части веб-приложения // International Journal of Humanities and Natural Sciences. 2025. N 3. С. 140–142.
4. Рахмани Д., М.Д. Баранов, Кузьмин Д.А. Разработка метода оптимизации веб-приложений с целью повышения производительности // International Journal of Humanities and Natural Sciences. 2025. N 3. С. 237–240.
5. Прогрессивный рендеринг для повышения производительности веб-приложений [Электронный ресурс] — URL: <https://webformyself.com/progressivnyj-rendering-dlya-povysheniya-proizvoditelnosti-veb-prilozhenij/> (дата обращения: 26.04.2025).

Оригинальность 77%