



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.9

ИССЛЕДОВАНИЕ ОПТИМИЗАЦИИ РЕНДЕРИНГА БОЛЬШИХ СПИСКОВ В REACTJS

¹Пирюшов М.С., Пирюшов А.С.

ФГАОУ ВО "НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ (ИТМО)", Санкт-Петербург, Россия (197101, город Санкт-Петербург, Кронверкский пр-кт, д. 49 литер а), e-mail: ¹maximalka111@yandex.ru

Цели. На сегодняшний день роль интернета в нашей жизни неуклонно растет, это связано с тем, что людям удобнее производить различные операции в интернете. Такая тенденция ведет к росту конкуренции среди сайтов, чтобы выигрывать в ней необходимо не только предлагать качественные товары и услуги, но и иметь удобный в использовании сайт. Удобство заключается в его производительности, однако трудно добиться хорошей производительности, если необходимо отображать большие списки. Цель работы – сравнить метрики оптимизации веб-приложения, управляя количеством элементов списка необходимых для рендеринга.

Методы. Предлагается исследовать работу веб-приложений, построенных с помощью фреймворка React, при использовании методов управления количеством элементов списка необходимых для рендеринга VirtualScroll, Lazy-loading, Partition loading. Экспериментальные результаты получены путём использования инструментов разработчика, предоставляемых Google, в частности вкладки Performance.

Результаты. Представлено описание метрик оптимизации, методов оптимизации, приведены экспериментальные результаты и их сравнение.

Выводы. Проведенные исследования позволяют говорить о том, что VirtualScroll, Lazy-loading, Partition loading действительно влияют на оптимизацию веб-приложения, которое содержит большие списки элементов. С помощью выбранных методов удалось сократить время рендеринга с 647 мс до 162 мс при общем объеме списка в 1825 элементов. Улучшились значения таких основных метрик Core Web Vitals, как FCP (с 3,48 с до 661,07 мс), LCP (с 9,21 с до 5,56 с), CLS (с 0,004 до 0), TBT (с 940 мс до 190 мс).

Ключевые слова: React, Virtual Scroll, Lazy-loading, Partition loading, Core Web Vitals, рендеринг, список.

EXPLORING OPTIMIZATION OF RENDERING LARGE LISTS IN REACTJS

¹Piryushov M.S., Piryushov A.S.

NATIONAL RESEARCH UNIVERSITY OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS (ITMO), St. Petersburg, Russia (197101, St. Petersburg, Kronverkskiy pr-kt, 49), e-mail: ¹maximalka111@yandex.ru

Objectives. Today, the role of the Internet in our lives is steadily growing, this is due to the fact that it is more convenient for people to carry out various operations on the Internet. This trend leads to increased competition among websites; in order to win, it is necessary not only to offer high-quality goods and services, but also to have an easy-to-use website. The convenience lies in its performance, but it is difficult to achieve good performance if you need to display large lists. The goal of the work is to compare optimization metrics for a web application by controlling the number of list items required for rendering.

Methods. It is proposed to study the operation of web applications built using the React framework, using methods for controlling the number of list elements required for rendering: VirtualScroll, Lazy-loading, Partition

loading. Experimental results were obtained by using the developer tools provided by Google, in particular the Performance tab.

Results. A description of optimization metrics and optimization methods is presented, experimental results and their comparison are given.

Conclusions. The conducted research suggests that VirtualScroll, Lazy-loading, Partition loading really influence the optimization of a web application that contains large lists of elements. Using the selected methods, it was possible to reduce the rendering time from 647 ms to 162 ms with a total list size of 1825 elements. The values of such main Core Web Vitals metrics as FCP (from 3.48 s to 661.07 ms), LCP (from 9.21 s to 5.56 s), CLS (from 0.004 to 0), TBT (from 940) have improved ms to 190 ms).

Keywords: React, Virtual Scroll, Lazy-loading, Partition loading, Core Web Vitals, rendering, list.

Введение

На сегодняшний день любое уважающее себя предприятие, будь то магазин строительных товаров или компания по предоставлению услуг в сфере бизнеса, все они стремятся «выложить» свои товары и услуги в интернет. Это и понятно – мы живем в век бурно развивающихся технологий и доступ в интернет имеет более 65% населения мира (около 5.3 млрд. человек), а к 2025 году это число увеличится до 6.54 млрд. Всех их нужно обслуживать, всем им нужно предлагать услуги, товары и т.д. Из-за такого стремительного роста пользователей, кажется, что должно расти и количество интернет-сервисов, которые направлены на удовлетворение их потребностей, но это не так. Согласно статистике представленной на Рисунке 1, количество сайтов в последнее время снижается [1].

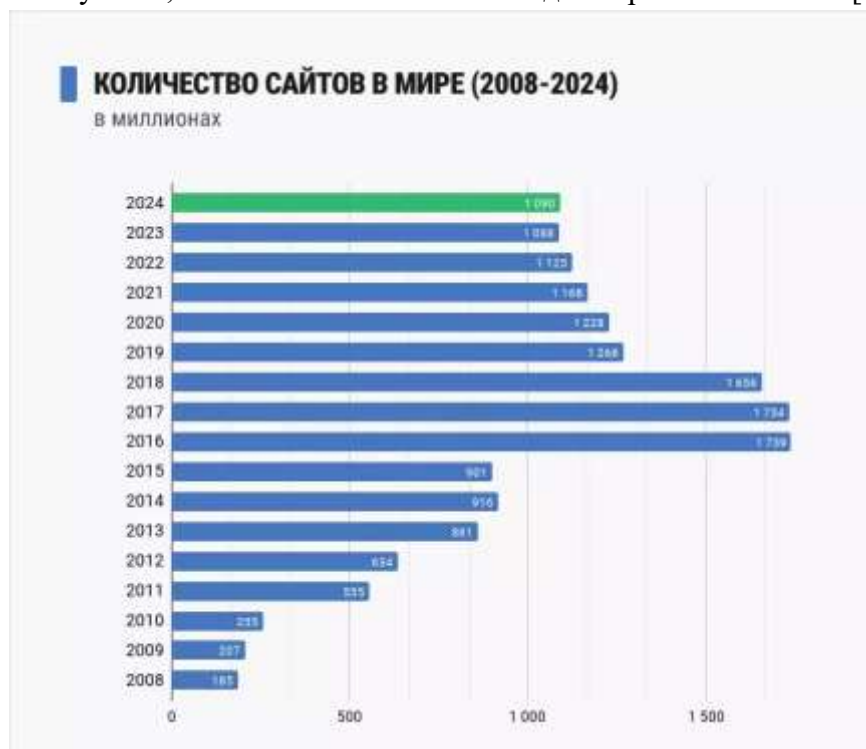


Рисунок 1.- Диаграмма «Количество сайтов в мире»

Такая тенденция связана с устареванием сайтов, нежеланием их поддерживать. Однако основным фактором является растущая конкуренция и она связана не только с качеством товаров и услуг, наполняющих сайт, но и с производительностью самого сайта.

Метрики оптимизации

В 2019 году компания Google представила набор метрик, с помощью которых определяет производительность сайта – Core Web Vitals. Этот набор состоит из следующих метрик [2]:

- Largest Contentful Paint (LCP) – метрика показывает точку во время загрузки страницы, когда были загружены основные или самые большие элементы содержимого во viewport. Разработчик должен стремиться к тому, чтобы показатель LCP обрабатывал в течение первых 2,5 секунд.
- Cumulative Layout Shift (CLS) – позволяет оценить визуальную стабильность страницы. CLS показывает, какое количество содержимого во viewport двигалось во время загрузки страницы. Эталоном является значение не превышающее 0.1.
- First Input Delay (FID) – измеряет, сколько времени потребовалось браузеру, чтобы отреагировать на первое событие пользовательского ввода. Требуемое значение: не более 100 мс.

На основе данных метрик Google поднимает в списке сайтов те из них, которые имеют лучшие показатели. Таким образом компания старается предлагать наиболее релевантный и полезный контент, слабо оптимизированные сайты являются неудобными в использовании.

Методы оптимизации

Многие ученые рассматривали методы оптимизации рендеринга контента, например, с помощью использования React-хуков, позволяющих избежать лишней перерисовки, (А. Ким, С.С. Савкин, Д.В. Логвинов) [3-4] или с помощью грамотного использования возможностей Virtual DOM (С.О. Бондаренко, К.Ю. Бетеев, Г.В. Муратова) [5-6] и (S. Aggarwal) [7], однако эти подходы не несут какого-то практического смысла, когда впервые загружается контент на страницу пользователя, особенно если его очень много, такую мысль высказали некоторые ученые (А. В. Андреев, О. В. Шиляева и А. Т. Зурабов) [8].

В данной работе рассмотрим оптимизацию рендеринга больших списков с помощью VirtualScroll, Lazy-loading и Partition loading.

Виртуальный скролл (Virtual Scroll) – это альтернатива пейджингу, при которой пользователь для получения доступа к очередным данным должен прокрутить ползунок скролла, при этом в DOM-дереве находятся элементы, которые помещаются во viewport, остальные из дерева удаляются. Данный подход позволяет ускорить работу, поскольку полный список не загружается за один раз, а только в соответствии с позицией скролла и DOM-дерево не накапливает лишние элементы [9].

Ленивая загрузка (отложенная загрузка, lazy-loading) – это способ оптимизации загрузки контента (чаще всего медиа-файлы), который не является критически важным для отображения на странице. Наиболее простым способом реализации данного метода является указание в атрибут data-src URL-адрес изображения HTML-тега img. Таким образом JavaScript отслеживает появление элемента в области видимости, перемещает из data-src в src, а затем уже происходит запрос на получение изображения. С помощью данного метода возможно избежать загрузки изображений, до которых пользователь не дошел [10].

Порционная загрузка (partition loading) – данные разбиваются на части определенного размера, назначается триггер, при срабатывании которого происходит получение очередной порции.

Описание и проведение эксперимента

Для проведения экспериментов было разработано небольшое приложение, которое состоит из списка карточек фильмов. Каждая карточка содержит название и постер к фильму.

Клиентская часть приложения была реализована с использованием библиотеки ReactJs, языком программирования был выбран TypeScript. В качестве менеджера состояний использовался Redux с расширением Redux Toolkit. Для выполнения запросов на сервер была выбрана библиотека axios. А также использовалась библиотека готовых компонентов AntDesign. Список с фильмами из приложения представлен на Рисунке 2.

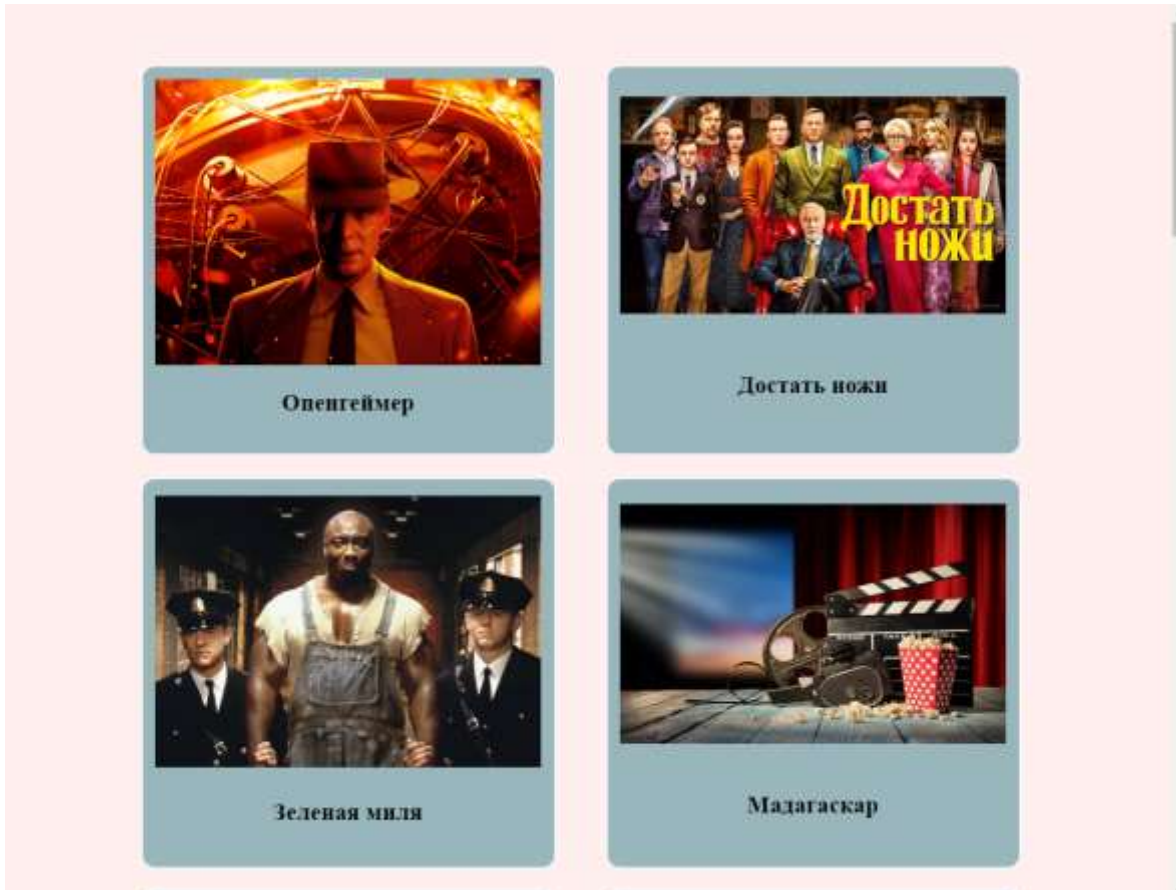


Рисунок 2. - Список карточек фильмов

Для проведения замеров был использован Google Chrome Devtools, а именно вкладка Performance. Замеры производились на оригинальном приложении, а затем после применения методов оптимизации. В таблице 1 представлено время в миллисекундах, затраченное на загрузку, выполнение скриптов, рендеринг, отрисовку и т.д.

Таблица 1. - Сравнение данных из вкладки Performance

Версия	Performance				
	Loading	Scripting	Rendering	Painting	Общее время
Оригинальная	3	898	647	509	14680
Оптимизированная	4	499	162	171	9668

В Таблице 2 отражены метрики производительности для оригинальной и оптимизированной версий.

Таблица 2. - Сравнение метрик из вкладки Performance

Метрика	FCP (мс)	LCP (мс)	CLS	TBT (мс)
Версия				

Оригинальная	3480	9210	0,004	940
Оптимизированная	661,07	5560	0	190

Заключение

По результатам проведенного эксперимента, представленных в таблицах 1,2, Virtual Scroll, Lazy-loading, Partition loading влияют на время рендеринга веб-приложения, содержащего список элементов, уменьшая его с 647 мс до 162 мс. Также данные методы позволяют улучшить метрики, входящие в набор Core Web Vitals: FCP (с 3,48 с до 661,07 мс), LCP (с 9,21 с до 5,56 с), CLS (с 0,004 до 0), ТВТ (с 940 мс до 190 мс).

Кроме этого, после проведенного эксперимента можно сделать вывод, что использование виртуального скролла для исключения хранения в DOM-дереве элементов, которые находятся вне области видимости пользователя, а также порционной загрузки контента и ленивой загрузки изображений позволяет уменьшить время рендеринга приложения в 4 раза. Однако очевидно, что выигрыш прямо пропорционально зависит от количества контента, который необходимо загрузить пользователю. Таким образом, для приложений-каталогов представленные методы оптимизации позволят улучшить ощущаемую производительность пользователем, а для приложений, где не предполагается наличие большого объема загружаемых данных может не иметь никакого смысла, а только затруднить процесс разработки.

Список литературы

1. 100+ ИНТЕРНЕТ-СТАТИСТИКИ И ТЕНДЕНЦИЙ. [Электронный ресурс] // websiterating: [сайт]. URL: <https://www.websiterating.com/ru/blog/research/internet-statistics-facts>.
2. Основные сведения о показателях Core Web Vitals и результатах поиска Google. [Электронный ресурс] // developers.google.com: [сайт]. URL: <https://developers.google.com/search/docs/appearance/core-web-vitals?hl=ru>.
3. Virtual Scrolling in React: Implementation from scratch and using react-window. [Электронный ресурс] // Tech With Vedansh: [сайт]. URL: <https://vedanshmehra.hashnode.dev/virtual-scrolling-in-react-implementation-from-scratch-and-using-react-window>.
4. Ким А. Анализ и оптимизация ре-рендеринга компонентов *Научно-технические инновации и веб-технологии*. 2022. № 1. С. 4-10. URL: https://www.elibrary.ru/download/elibrary_48658820_11841256.
5. Савкин С. С. Оптимизационные возможности JavaScript-библиотеки REACT 18 / С. С. Савкин, Д. В. Логвинов *Вызовы глобализации и развитие цифрового общества в условиях новой реальности : сборник материалов IV Международной научно-практической конференции*, Москва, 19 декабря 2022 года. – Москва: Алеф, 2022. – С. 126-129. URL: <https://elibrary.ru/item.asp?id=50138369>.
6. Бондаренко С.О. Современные интерактивные веб приложения – построение пользовательского интерфейса с React *Вестник науки и образования*. 2018. № 5 (41). С. 46-48 URL: <https://elibrary.ru/upbkif?ysclid=ln614gvozy964658019>.

7. Бетеев К.Ю., Муратова Г.В. Концепция Virtual DOM в библиотеке React.js // *Инженерный вестник Дона*. 2022. № 3 (87). С. 170-180. URL: https://www.elibrary.ru/download/elibrary_48413155_74717385.pdf.
8. Aggarwal S. Modern Web-Development Using ReactJS *International Journal of Recent Research Aspects*. 2018. №5(1). С. 133-137. URL: <https://idoc.tips/modern-web-development-using-reactjs-pdf-free.html>.
9. Андреев А. В. Исследование процесса оптимизации функциональных веб-интерфейсов / А. В. Андреев, О. В. Шиляева, А. Т. Зурабов *Актуальные проблемы науки и техники*. 2022: Материалы Всероссийской (национальной) научно-практической конференции, Ростов-на-Дону, 16–18 марта 2022 года / Отв. редактор Н.А. Шевченко. – Ростов-на-Дону: Донской государственный технический университет, 2022. – С. 312-313. URL: <https://elibrary.ru/item.asp?id=49165838>.
10. Virtual Scrolling in React: Implementation from scratch and using react-window. [Электронный ресурс] // Tech With Vedansh: [сайт]. URL: <https://vedanshmehra.hashnode.dev/virtual-scrolling-in-react-implementation-from-scratch-and-using-react-window>.
11. Абдураманов, З. Ш. Оптимизация веб-страницы с использованием "ленивой подгрузки изображений" / З. Ш. Абдураманов, А. Г. Ибраимов, Г. С. Сейдаметов *Информационно-компьютерные технологии в экономике, образовании и социальной сфере*. – 2021. – № 1(31). – С. 147-154. – EDN ZKCCMW. URL: <https://www.elibrary.ru/item.asp?id=45677816>.

References

1. 100+ INTERNET STATISTICS AND TRENDS. [Electronic resource] // websiterating: [site]. Available from URL: <https://www.websiterating.com/ru/blog/research/internet-statistics-facts>.
2. Understanding Core Web Vitals metrics and Google search results. [Electronic resource] // developers.google.com: [website]. Available from URL: <https://developers.google.com/search/docs/appearance/core-web-vitals?hl=ru>.
3. Virtual Scrolling in React: Implementation from scratch and using react-window. [Electronic resource] // Tech With Vedansh: [website]. Available from URL: <https://vedanshmehra.hashnode.dev/virtual-scrolling-in-react-implementation-from-scratch-and-using-react-window>.
4. Kim A. Analysis and optimization of rerendering of components. In: *Scientific and Technical Innovations and Web Technologies*. 2022. No. 1. P. 4-10. Available from URL: https://www.elibrary.ru/download/elibrary_48658820_11841256.
5. Savkin S. S. Optimization capabilities of the JavaScript library REACT 18 / S. S. Savkin, D. V. Logvinov In: *Challenges of globalization and the development of digital society in the new reality: collection of materials of the IV International Scientific and Practical Conference*, Moscow, December 19, 2022. – Moscow: Aleph, 2022. – P. 126-129. Available from URL: <https://elibrary.ru/item.asp?id=50138369>.
6. Bondarenko S.O. Modern interactive web applications - building a user interface with React In: *Bulletin of Science and Education*. 2018. No. 5 (41). pp. 46-48 Available from URL: <https://elibrary.ru/upbkif?ysclid=ln614gvozy964658019>.

7. Beteev K.Yu., Muratova G.V. The concept of Virtual DOM in the React.js library In: *Engineering Bulletin of Don*. 2022. No. 3 (87). pp. 170-180. Available from URL: https://www.elibrary.ru/download/elibrary_48413155_74717385.pdf.
 8. Aggarwal S. Modern Web-Development Using ReactJS *International Journal of Recent Research Aspects*. 2018. №5(1). С. 133-137. Available from URL: <https://idoc.tips/modern-web-development-using-reactjs-pdf-free.html>.
 9. Andreev A. V. Study of the optimization process of functional web interfaces / A. V. Andreev, O. V. Shilyaeva, A. T. Zurabov In: *Current problems of science and technology. 2022: Materials of the All-Russian (national) scientific and practical conference, Rostov-on-Don, March 16–18, 2022* / Rep. editor N.A. Shevchenko. – Rostov-on-Don: Don State Technical University, 2022. – P. 312-313. Available from URL: <https://elibrary.ru/item.asp?id=49165838>.
 10. Virtual Scrolling in React: Implementation from scratch and using react-window. [Electronic resource] // Tech With Vedansh: [website]. Available from URL: <https://vedanshmehra.hashnode.dev/virtual-scrolling-in-react-implementation-from-scratch-and-using-react-window>.
 11. Abduramanov, Z. Sh. Optimization of a web page using “lazy loading of images” / Z. Sh. Abduramanov, A. G. Ibraimov, G. S. Seydametov In: *Information and computer technologies in economics, education and social sphere* . – 2021. – No. 1(31). – pp. 147-154. – EDN ZKCCMW. Available from URL: <https://www.elibrary.ru/item.asp?id=45677816>.
-