



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

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



УДК 004.424

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ФРЕЙМВОРКОВ REACT, NEXT И ASTRO

<sup>1</sup>Мелихова К.А., <sup>2</sup>Томилин Е.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: <sup>1</sup>kcniamelihova@gmail.com, <sup>2</sup>tomilin229@gmail.com

В данной статье было проведено сравнение производительности веб-фреймворков: React, Next и Astro. Для каждого из них были разработаны три однотипных тестовых веб-приложения, каждое из которых представляло собой сайт с несколькими страницами. С использованием библиотеки Puppeteer проведено 10 циклов тестирования, имитирующих реальное взаимодействие с приложением, для измерения параметров производительности, таких как время первой отрисовки, время рендеринга, время загрузки страницы, потребление памяти. Результаты анализа показали, что фреймворк Next демонстрирует лучшую производительность по сравнению с React и Astro.

Ключевые слова: React, Next, Astro, Puppeteer, веб-приложение, фреймворк, производительность приложения, эффективность приложения.

## ANALYSIS OF REACT, NEXT, AND ASTRO FRAMEWORKS PERFORMANCE IMPLEMENTATION

<sup>1</sup>Melikhova K. A., <sup>2</sup>Tomilin E. S.

NATIONAL RESEARCH UNIVERSITY OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS (ITMO), St. Petersburg, Russia (197101, St. Petersburg, Kronverkskiy pr-kt, 49, lit. a), e-mail: <sup>1</sup>kcniamelihova@gmail.com, <sup>2</sup>tomilin229@gmail.com

This study presents a performance comparison of web frameworks: React, Next, and Astro. Three identical test web applications were developed for each framework, each representing a website with multiple pages. Using the Puppeteer library, 10 testing cycles were conducted, simulating real interaction with the application, to measure performance parameters such as first paint time, rendering time, page load time, and memory consumption. The analysis results revealed that the Next framework demonstrates superior performance compared to React and Astro.

Keywords: React, Next, Astro, Puppeteer, web application, framework, application performance, application efficiency.

### Введение

На данный момент в области веб-разработки широко используются различные фреймворки, предназначенные для оптимизации процесса создания веб-приложений. Большинство из них ориентированы на язык программирования JavaScript, который в настоящее время считается стандартом для разработки веб-приложений. По данным индекса ТЮВЕ, который является показателем популярности языка программирования, данный язык на момент марта 2024 года находится на 7 месте. Существует множество фреймворков, каждый из которых нацелен на достижение конкретных целей или улучшение определенных

аспектов разработки. Некоторые из них фокусируются на создании и управлении пользовательским интерфейсом, в то время как другие стремятся улучшить взаимодействие с сервером. В данном исследовании рассмотрены следующие фреймворки: React, Next и Astro.

React — это js-фреймворк, разработанный для создания пользовательских интерфейсов. В основе React лежит концепция CSR (Client-Side Rendering), при которой процесс отображения веб-страницы выполняется прямо в браузере клиента. Основным принципом работы React заключается в использовании виртуального DOM, который отслеживает изменения состояния и передает их в поддерева узлов. Механизм согласования React сравнивает предыдущую модель пользовательского интерфейса с новой, выявляет изменения и применяет соответствующие обновления только в реальном DOM, тем самым уменьшая количество операций с DOM [1].

Фреймворки Next.js и Astro предназначены для обеспечения рендеринга веб-страниц на сервере перед тем, как они будут отправлены клиенту (браузеру). Фреймворки такого типа предоставляют улучшенную SEO-оптимизацию, ускоряют отдачу контента, обеспечивают лучшую производительность на медленных устройствах и облегчают поддержку и масштабирование приложений [2].

### **Цель исследования**

Целью данного исследования является проведение анализа производительности трех веб-фреймворков – React, Next и Astro. Основными задачами в рамках исследования являются измерение и сравнение скорости загрузки страниц, а также общей производительности при разработке и эксплуатации веб-приложений на основе данных фреймворков. Результаты данного исследования предоставляют разработчикам информацию, необходимую для выбора наиболее подходящего фреймворка в соответствии с требованиями и спецификациями их проекта, а также для оптимизации производительности разрабатываемых веб-приложений.

### **Материал и методы исследования**

Для достижения этой цели разработаны три однотипных тестовых приложения, по одному для каждого из фреймворков. Каждое из этих приложений представляет собой веб-сайт, состоящий из трех основных страниц: домашней, страницы с постами и страницы с фотографиями. Кроме того, были созданы страницы для отдельных постов и фотографий, которые предоставляют информацию о конкретных элементах.

Методология в данной работе представлять собой эксперимент. Для оценки производительности проведено 10 циклов тестирования с использованием библиотеки Puppeteer, обеспечивающей возможность автоматизированного взаимодействия с веб-браузером. Каждый цикл тестирования включает в себя определенный набор действий, имитирующих реальное взаимодействие с приложением, такие как загрузка приложения и взаимодействие с интерфейсом. В процессе выполнения тестов были измерены различные параметры производительности, включая время загрузки страницы, отзыв интерфейса при взаимодействии с пользователем и использование ресурсов системы.

Полученные в результате эксперимента данные о производительности включали в себя время первой отрисовки, рендеринга, загрузки страницы и потребление памяти. Полученные результаты были представлены в виде графиков и анализировались с целью сравнения производительности различных фреймворков.

### Результаты исследования и их обсуждение

React — это библиотека на языке JavaScript, созданная с целью облегчить проектирование интерактивных веб-приложений. React упрощает задачу построения переиспользуемых компонентов пользовательского интерфейса, что способствует простому управлению и обновлению сложных пользовательских интерфейсов [3].

Концепции, использующиеся в библиотеке React:

1. Компонентный подход и JSX (JavaScript XML): в React JSX используется для описания структуры компонента. Данный подход предлагает синтаксис, схожий с HTML, облегчая восприятие элементов интерфейса и делая код доступным для понимания.

2. Виртуальный DOM: React использует виртуальный DOM для оптимизации процесса рендеринга. Виртуальный дом – облегченная копия реального DOM дерева. Изменения, производимые в интерфейсе, вначале реализуются в виртуальном DOM. Затем React определяет различия между актуальным и виртуальным DOM, обновляя лишь нужные элементы интерфейса, что уменьшает время рендеринга [4].

3. State и props: Эти концепции важны для управления данными внутри пользовательского компонента. Локальное состояние компонента (state) может изменяться с течением времени, что провоцирует перерисовку этой части интерфейса. Props применяются для передачи данных от родительского компонента к дочернему, что обеспечивает однонаправленный поток данных [5].

4. Экосистема и сообщество: в настоящее время React является одной из самых используемых библиотек JavaScript, имея обширную экосистему, включающую в себя много дополнительных инструментов.

5. Client Side Rendering (CSR): при рендеринге на стороне клиента браузер изначально загружает минимальную HTML-страницу и необходимый для страницы JavaScript. Затем JavaScript используется для обновления DOM и рендеринга страницы. Основной проблемой такого подхода является возможная задержка при первичной загрузке приложения. Заметная пауза до полной загрузки страницы происходит из-за того, что страница не отобразится в полном объеме до тех пор, пока не загрузится, не будет проанализирован и не выполнится весь JavaScript код [6] (Рисунок 1).

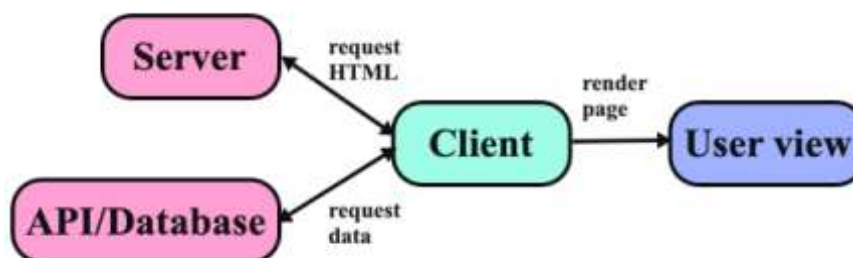


Рисунок 1 – Client-Side Rendering

Next — это платформа для создания веб-приложений, основанная на React. Она включает в себя набор инструментов, спроектированных для облегчения разработки современных и эффективных веб-приложений [7, 8].

Главные особенности Next:

1. Серверный рендеринг (SSR): при помощи Next можно создавать полноценные HTML-страницы на сервере, которые моментально отображаются на стороне клиента, что помогает уменьшить количество запросов к серверу и улучшить производительность приложения, особенно важно для SEO (Рисунок 2) [8].

2. Оптимизация для поисковых систем (SEO): благодаря SSR, Next помогает приложениям занимать более высокие позиции в поисковых системах, позволяя лучше индексировать контент.

3. Редактирование тега <head>: Next предоставляет возможность редактирования тега, что недоступно в стандартном React. Это также способствует улучшению SEO-рейтинга приложения.

4. Простота развертывания: Компания Vercel, разработчик Next.js, упрощает процесс развертывания React-приложений [9].

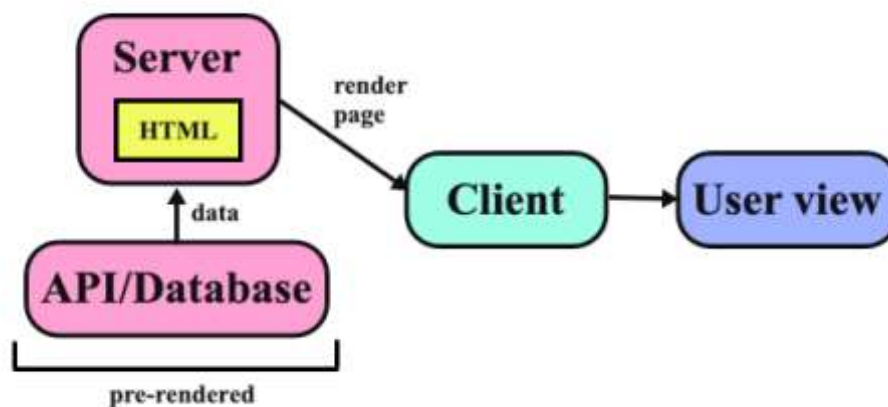


Рисунок 2 – Серверный рендеринг (SSR)

Фреймворк Astro предоставляет возможность создавать сайты без использования JavaScript при сборке. Данный фреймворк совместим с React, Vue, Angular и другие библиотеки. Главное достоинство Astro заключается в его способности по умолчанию генерировать HTML и CSS, требуя ручной разметки только для динамических элементов. Такой подход называется "архитектурой островов", где HTML и CSS играют роль воды, а JavaScript - островков.

Astro стал популярным из-за нескольких причин:

1. Высокий уровень удобства в процессе создания благодаря интеграции с Vite, обеспечивающий высокую скорость сборки.

2. Легкость интеграции и сочетания различных библиотек облегчает процесс создания веб-приложений.

3. Автономность проектов от сопровождения, благодаря использованию HTML, CSS и CDN.

4. Легкость в применении: в сравнении с React, Astro предлагает разработку проектов, требующих меньше зависимостей.

Выбор подходящего фреймворка в веб-разработке существенно влияет на производительность и эффективность приложения. Для оценки производительности веб-приложений проводятся измерения различных параметров, таких как время первой отрисовки, рендеринга, загрузки страницы, потребление памяти и другие, чтобы

соответствовать ожиданиям пользователей относительно скорости и отзывчивости приложения.

Для проведения анализа производительности фреймворков были разработаны три одинаковых тестовых веб-приложения, по одному для каждой технологии. В каждом из этих приложений есть три страницы: главная, страница с постами и страница с изображениями. Кроме основных страниц, есть отдельные страницы для постов и изображений, где можно получить информацию о каждом элементе.

В исследовании используется экспериментальный подход для оценки производительности трех фреймворков: React, Next и Astro. Эксперимент представляет собой систематически проведенное исследование, научно поставленный опыт, реализуемый в строго контролируемых условиях. Эти условия позволяют не только отслеживать процесс, но и обеспечивают возможность его повторения. В лабораторных условиях эксперимент можно провести с высокой степенью эффективности, симулируя сценарии, подобные тем, что встречаются в реальном мире. Исходя из этого, для данного исследования был выбран экспериментальный метод.

В качестве технологии тестирования был выбран Puppeteer. Данный инструмент предназначен для имитации действий пользователя на сайте, что позволяет выполнять в автоматическом режиме операции, такие как заполнение формы, клики мыши, копирование текста и т. д. С помощью Puppeteer можно добиться надежной оценки критериев производительности веб-приложения. Были выделены следующие критерия:

1. Время первой отрисовки страницы сайта;
2. Время рендеринга (время отображения всех элементов на странице);
3. Время загрузки страницы (полная загрузка страницы вместе со всеми ее компонентами – стили, скрипты);
4. Потребление памяти (объем оперативной памяти, используемый во время работы приложения);
5. Среднее время выполнения сценария на сайте (список действий со стороны пользователя).

Полученные данные по критериям позволят составить рекомендации по выбору той или иной технологии в зависимости от контекста использования. Результаты эксперимента представлены на Рисунках 3 и 4.

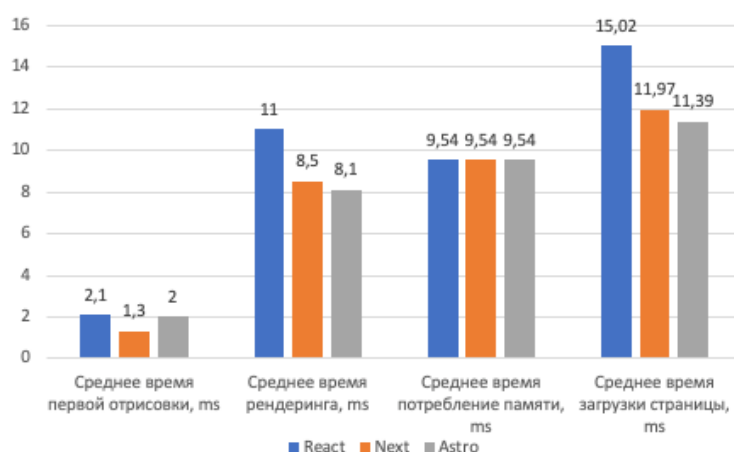


Рисунок 3. – Основные метрики производительности приложений (React/Next/Astro)

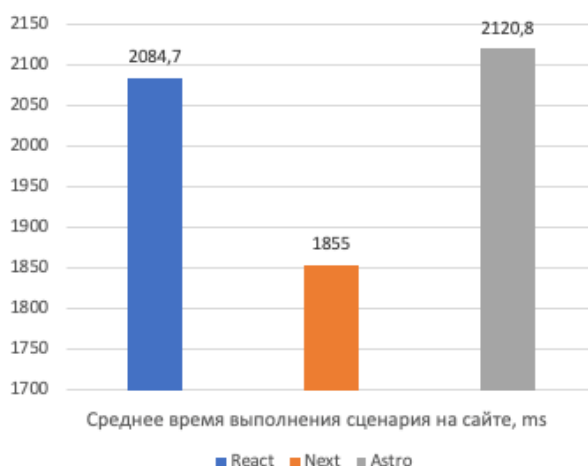


Рисунок 4. – Среднее время выполнения сценария на сайте (React/Next/Astro)

По полученным данным можно сказать, что фреймворк Next продемонстрировал лучшие показатели среди трех технологий. Next лучше React и Astro по среднему времени первой отрисовки и по времени выполнения сценария на сайте. Однако, по времени рендеринга страницы и загрузке страницы React значительно проигрывает Next и Astro, при этом последние имеют схожие показатели.

### Заключение

На основе проведенного исследования были получены данные по ключевым показателям производительности фреймворков и сформированы рекомендации применения того или иного решения в зависимости от контекста использования. Фреймворк Next показал лучшие результаты по большинству параметров производительности в сравнении с React и Astro. Данное решение обладает высокой скоростью отрисовки страницы и имеет оптимальную производительность. Фреймворки React и Astro показывают схожие результаты, однако Astro выигрывает React по времени рендеринга и загрузке страницы. С учетом описанных выше особенностей, формируется выбор фреймворка, на котором будет разработан проект. Если для будущего веб-приложения необходима высокая производительность, то лучшим выбором будет Next. В зависимости от других факторов, React и Astro также могут быть подходящими вариантами.

### Список литературы

1. Потовиченко М. А., Шатилов Ю. Ю. Разработка клиентской части одностраничного web-приложения с использованием библиотек React // Научное обозрение. Технические науки. – 2020. – №1. – С. 39–43.
2. Евдокимов А. О. Анализ и сравнение клиентского рендеринга (CSR) и серверного рендеринга (SSR) в современных веб-приложениях // Студенческая наука: созидая будущее. – 2023. – С. 143–148.
3. Давыдова М. П. Интерактивное web-приложение с использованием библиотеки React JS // Студент: наука, профессия, жизнь. – 2023. – С. 203–206.

- Сукиасян В. М., Придиус Е. С. Современные принципы и подходы к Frontend архитектуре веб-приложений // Наука, техника и образование. – 2019. – № 10 (63). – С. 54-57.
- Бетеев К. Ю., Муратова Г. В. Концепция Virtual DOM в библиотеке React.JS // Инженерный вестник Дона. – 2022. – № 3 (87). – С. 170–180.
- Арянов В.А. Особенности разработки пользовательских интерфейсов на React JS // Материалы XXV Научно-практической конференции молодых ученых, аспирантов и студентов национального исследовательского Мордовского государственного университета. – 2022. – С. 271–275.
- Скороходов И. С., Тихомирова А. Н. Исследование и сравнение современных реализаций Flux-архитектур разработки веб-приложений // Наука, техника и образование. – 2016. – № 6 (24). – С. 47–54.
- Кожантаев Т.Р. Виды рендеринга в веб-программировании // Известия Ошского технологического университета. – 2022. – № 2. – С. 65–69.
- Swiatkowski A., Scibior K. Comparative analysis of React, Next and Gatsby programming frameworks for creating SPA applications // Journal of Computer Sciences Institute. – 2022. – № 24. – pp. 224–227.

## References

- Potovichenko M. A., Shatilov Yu. Yu. Development of the client part of a single-page web application using React libraries // Scientific Review. Technical sciences. 2020. – № 1. – pp. 39-43.
  - Evdokimov A. O. Analysis and comparison of client rendering (CSR) and server rendering (SSR) in modern web applications // Student Science: creating the future. – 2023. – pp. 143-148.
  - Davydova M. P. Interactive web application using the React JS library // Student: science, profession, life. – 2023. – pp. 203-206.
  - Sukiasyan V. M., Pridius E. S. Modern principles and approaches to Frontend architecture of web applications // Science, technology and education. – 2019. – № 10 (63). – pp. 54-57.
  - Beteev K. Yu., Muratova G. V. The concept of Virtual DOM in the library React.JS // Engineering Bulletin of the Don. – 2022. – № 3 (87). – pp. 170-180.
  - Aryanov V.A. Features of the development of user interfaces on React JS // Materials of the XXV Scientific and Practical Conference of Young scientists, postgraduates and students of the National Research Mordovian State University. – 2022. – pp. 271-275.
  - Skorokhodov I. S., Tikhomirova A. N. Research and comparison of modern implementations of Flux architectures for web application development // Science, technology and education. – 2016. – № 6 (24). – pp. 47-54.
  - Kozhentaev T.R. Types of rendering in web programming // Izvestiya Osh Technological University. – 2022. – № 2. – pp. 65-69.
  - Swiatkowski A., Scibior K. Comparative analysis of React, Next and Gatsby programming frameworks for creating SPA applications // Journal of Computer Sciences Institute. – 2022. – № 24. – pp. 224–227.
-