

## ИССЛЕДОВАНИЕ ОПТИМИЗАЦИИ РЕНДЕРИНГА КОМПОНЕНТОВ НА ПРИМЕРЕ SVELTE

Летон Г. (Университет ИТМО, г. Санкт-Петербург),

Глазько П.Е. (Университет ИТМО, г. Санкт-Петербург),

**Научный руководитель – доцент, кандидат педагогических наук Государев И.Б.**

(Университет ИТМО, г. Санкт-Петербург)

В современном мире высокая производительность веб-приложения важна и для его пользователей, и для компании. Разработчики используют в работе фреймворки, которые в результате показывают различные параметры эффективности. Svelte представляет отличный от других технологий подход к созданию пользовательских интерфейсов, поэтому было решено проанализировать принципы повышения производительности и провести сравнение эффективности с помощью метрик.

**Введение.** Эффективность рендеринга является проблемой для современной веб-разработки. Скорость рендеринга влияет на удовлетворенность пользователей при использовании сайта. Современный пользователь не готов долго ждать получения информации или возможности взаимодействия со страницей. Данный факт не только снижает заинтересованность пользователей, но и приносит компаниям крупные убытки. В том числе с каждым годом растет число интернет-магазинов и сайтов с высоким содержанием интерактивных элементов, что увеличивает вероятность возникновения проблемы производительности, так как такие приложения содержат изображения, видео, анимации, замедляющие рендеринг.

**Основная часть.** Большинство компаний при разработке пользовательских интерфейсов включает в список используемых технологий фреймворки. Они позволяют разработчикам увеличить скорость работы, создавать масштабируемые приложения, структурировать код. Каждый фреймворк имеет свои подходы к оптимизации и показатели эффективности рендеринга. Svelte относительно молодой фреймворк и был представлен в качестве принципиально нового подхода к разработке пользовательских интерфейсов.

Svelte представляет собой компонентный фреймворк, в котором компиляция компонентов происходит при сборке. Одним из теоретических подходов к оптимизации рендеринга является уменьшение конечного размера веб-приложения, его отдельных компонентов и дополнительных файлов. Svelte в принципе требует меньшее количество кода в сравнении с другими фреймворками. Конечный размер файлов, которые отправляются клиенту, является в сравнении также малым.

Операции с Document Object Model (DOM) имеют значительный вес в производительности веб-приложений. Для увеличения эффективности рендеринга основным подходом будет уменьшение количества взаимодействий с DOM, а также их упрощение. Фреймворки используют различные подходы к обновлению модели DOM. Одним из наиболее известных алгоритмов является Virtual DOM. Данный подход использует React. Модель виртуального DOM хранится в памяти, является легковесной копией реального DOM и синхронизируется с ней. Для обновления элементов реального DOM используется алгоритм диффинга (diffing). При изменении некоторых частей интерфейса производится сравнение моделей до и после изменений, затем определяются те элементы, которые должны быть обновлены. В итоге точно производится повторный рендеринг. Svelte предлагает иной подход, аргументируя это тем, что Virtual DOM требует дополнительных вычислений, что влияет на оптимизацию. Svelte - компилятор, который преобразует приложение в код JavaScript, не используя при этом лишний код для запуска в браузере. Анализируя код, строится дерево зависимостей, и затем в JavaScript коде обновляются только необходимые части DOM.

Для оценки производительности приложений используются различные инструменты веб-аудита. Самый доступный - Lighthouse, который включен в Chrome и предоставляет

возможность оценки в том числе метрик эффективности рендеринга. Существует большое число типов метрик, при этом не все могут объективно отражать производительность приложений. Таким образом Google и W3C запустили инициативу Web Vitals, где наиболее важными показателями эффективности являются Core Web Vitals: Largest Contentful Paint (LCP) - скорость загрузки основного контента, First Input Delay (FID) - время ожидания до первого взаимодействия с контентом, Cumulative Layout Shift (CLS) - измеряет визуальную стабильность.

Было решено провести эксперимент, который заключается в измерении метрик, отражающих эффективность рендеринга для приложения на React и приложения на Svelte. React был выбран в качестве технологии для сравнения, поскольку он использует Virtual DOM, о котором было упомянуто. Приложение содержит страницу, содержащую анимированные элементы. Анимации часто используются в современных пользовательских интерфейсах, так как помогают улучшить восприятие информации, но при этом могут снизить производительность приложения. Для эксперимента были выбраны метрики LCP, FID, а также взяты следующие метрики для исследования:

1. First Contentful Paint – время, за которое отобразится какое-то содержимое страницы.
2. Time to Interactive - время, за которое страница станет полностью интерактивной.
3. Total Blocking Time – количество времени между First Contentful Paint и Time to Interactive, когда страница практически не реагирует на ввод.
4. DOM Content Loaded – время, за которое HTML загружен, не дожидаясь стилей или изображений.
5. Load end – время, за которое страница загружается полностью, включая стили и изображения.
6. Byte Weight – общий размер всех ресурсов, запрошенных страницей.

**Выводы.** В докладе были проанализированы подходы к оптимизации рендеринга в Svelte. При проведении эксперимента были получены результаты, отражающие большую эффективность рендеринга приложения на Svelte. Например, Svelte показал значение метрики First Contentful Paint лучше на 0.3 секунды, чем приложение на React, а Time to Interactive на 0.2 секунды. Данная работа может быть полезна разработчикам при выборе технологий для создания веб-приложений с точки зрения производительности.

Летон Г. (автор)

Подпись

Глазко П.Е. (автор)

Подпись

Государев И.Б. (научный руководитель)

Подпись