

Гарбачук Артем Сергеевич

магистрант

Гриценко Екатерина Михайловна

канд. техн. наук, доцент, заведующая кафедрой

ФГБОУ ВО «Сибирский государственный университет

науки и технологий им. академика М.Ф. Решетнева»

г. Красноярск, Красноярский край

ПРИМЕНЕНИЕ SPA-АРХИТЕКТУРЫ ДЛЯ УЛУЧШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ САЙТА НА ПРИМЕРЕ ПРОЕКТА «ВУЛКАН-М»

Аннотация: в статье рассмотрена архитектура SPA как технология создания современных сайтов. Показаны её преимущества и влияние на конечную производительность сайта.

Ключевые слова: SPA, JavaScript, Typescript, AtScript, библиотека, фреймворк, компонентный подход, хранилище данных.

Введение

По мере роста проекта, всплывает вопрос о том, как повысить производительность сайта. На сегодняшний день существует множество способов улучшения производительности, одним из таких способов является разделение пользовательской части сайта от серверной, путем переноса части бизнес логики проекта на клиентскую сторону.

1. Серверные приложения

Самый распространённый вариант создания сайтов – серверное приложение. В таком подходе HTTP-запросы отправляются прямо на сервер, который отвечает HTML-страницей, другими словами, после получения запроса сервер ищет информацию в базе данных, встраивает ее в шаблон, а потом отправляет клиенту готовую страницу. При этом, каждая страница сайта поступает к пользователю напрямую с сервера и при переходе на другую страницу, происходит перезагрузка окна браузера.

В таком подходе проходит достаточно значительное время между отправкой запроса и получением ответа, поэтому требуется значительная переработка серверной архитектуры приложения, чтобы скорость отправки и получения стала наиболее оптимальной.

2. Одностраничное веб-приложение

SPA [1] или одностраничное веб-приложение – это приложение, размещенное на одной странице, как правило это самая главная, первая страница сайта. Приложения такого типа, появились значительно недавно, с появлением HTML5 и развитием стандартов языка JavaScript [2]. Так как единственным языком браузера является JavaScript [2], то SPA [1] – это приложения, которые написаны на языке JavaScript. Существуют так же некоторые надстройки над этим языком, а также расширения, такие как TypeScript [3], AtScript [4].

Разрабатывая одностраничное приложение, создается единственный входной файл, который скачивает пользователь, как правило это «index.html». В нем подключаются все необходимые стилевые файлы, а также самый главный файл – JavaScript [2]. По сути, при заходе на сайт, у пользователя скачивается единственный файл, в котором прописана вся логика сайта и переходы между страницами, не вызывают перезагрузку страницы в браузере и не скачивают с сервера необходимую страницу. Все сводится к тому, что пользователь получает с сервера только данные из базы, а все страницы генерируются на основании шаблонов, которые прописаны в JavaScript [2] файле.

3. Библиотеки и фреймворки

С развитием стандартов JavaScript [2], стало появляться множество библиотек [5] и фреймворков [6], которые упрощают разработку SPA [1], а также позволяются легко и быстро оперировать данными, отправлять и получать запросы от сервера. Примерами могут быть: React.js, Vue.js, Angular.js, Bases.js и другие. Все эти библиотеки и фреймворки используют разный подход в разработке, но во многом они схожи, если говорить о модульном и компонентном подходе [7], который включает в себя разработку страниц приложения как некий набор самостоятельных компонентов, которые можно использовать неоднократно в рамках

2 <https://interactive-plus.ru>

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

проекта, или по-другому – пере используемые компоненты и объединять в модули.

Навигация внутри приложения происходит с помощью других библиотек [5], например: React-Router, VueRouter и других. Они позволяют сопоставлять компонентам пути и указывать, где именно стоит отобразить нужный компонент.

Управление данными, происходит с помощью состояний. Начальное состояние задается при вызове нужного компонента (при навигации). Оно может быть в виде статических данных, а может наполняться данными, которые получаем с сервера. При таком подходе, все операции над данными не требуют дополнительного обращения к серверу, так как все операции выполняются на стороне клиента, а на сервер передаем уже обработанные данные. Состояния хранятся в так называемом «хранилище данных (store)» [8], которое не привязано к какому-то конкретно компоненту, а может вызываться из любой точки приложения.

3. Сравнение скорости загрузки

Сравнение скорости загрузки производится на проекте «Вулкан-М» в двух его реализациях: серверное приложение и одностраничное веб-приложение. В качестве примера для сравнения скорости, были выбраны следующие критерии:

1. Первоначальная загрузка сайта.
2. Переход между страницами.

Первоначальная скорость загрузки серверного приложения составила 7.12 сек и представлена на рисунке 1.

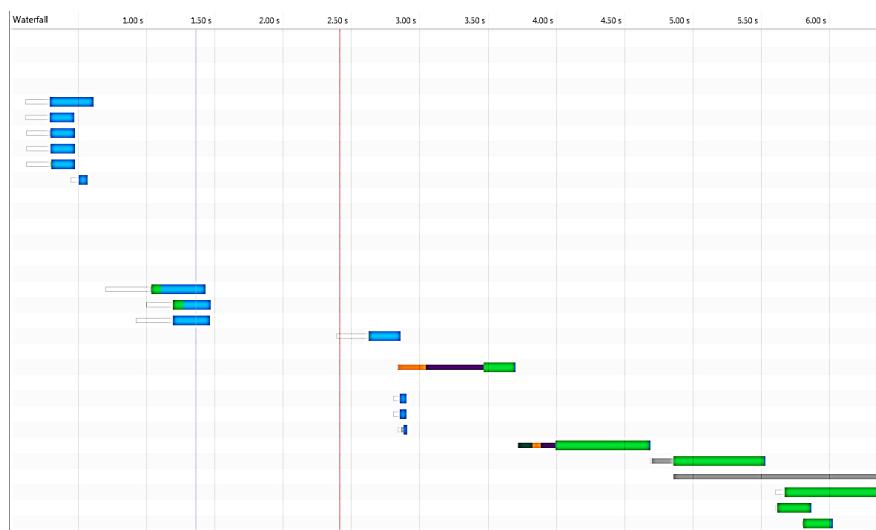


Рис. 1. Скорость загрузки серверного приложения

Первоначальная скорость загрузки одностраничного веб-приложения состояла 4.15сек и представлена на рисунке 2.

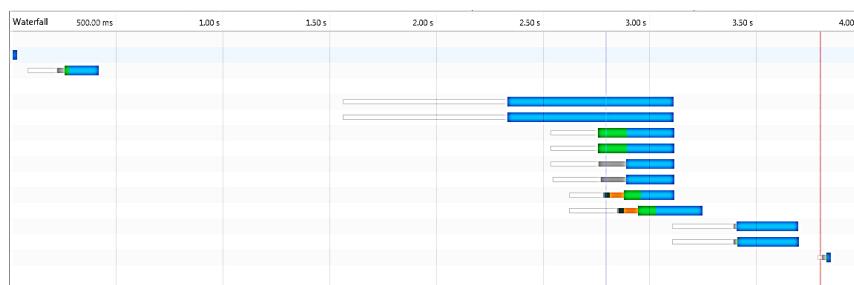


Рис. 2. Скорость загрузки одностраничного веб-приложения

Переход по страницам в серверном приложении похож на рисунок 1. Это связано с тем, что при переходы по страницам, каждый раз приходят данные с сервера (стилевые, скриптовые и сама HTML-страница), данный график представлен на рисунке 3.

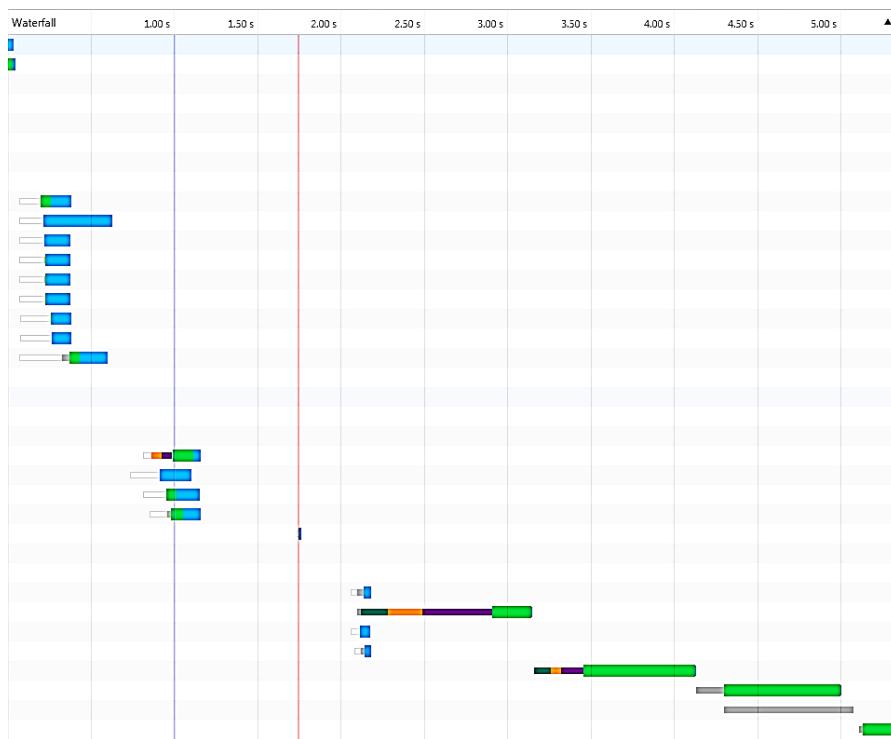


Рис. 3. Переход по страницам серверного приложения

В случае со SPA [1], виден значительный прирост в скорости загрузки. На рисунке 4 видно, что с сервера получаются только необходимые данные из базы, а стилевые, скриптовые и сама HTML-страница больше не приходят.

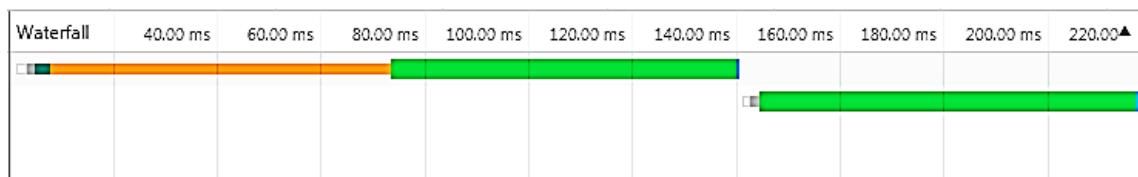


Рис. 4. Переход по страницам одностраничного веб-приложения

Заключение

Применение SPA-архитектуры в проекте «Вулкан-М», значительно снизило скорость загрузки страниц, тем самым повысило скорость и эффективность работы пользователю. Так же применение такой архитектуры упростило задачу по поддержке и обновлению проекта.

Список литературы

1. ХабраХабр, крупнейшие блог-посты в сфере ИТ в рунете // HabraHabr. – 2014 [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru>

2. Calabonga, блог по программированию и технологиям // Calabonga. – 2017 – [Электронный ресурс]. – Режим доступа: <https://www.calabonga.net>
3. Tproger, одно из крупнейших объединений программистов в рунете // Tproger. – 2017 [Электронный ресурс]. – Режим доступа: <https://tproger.ru>