

МІНІСТЕРСТВО ОСВІТУ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ

МАТЕРІАЛИ ДЕВ'ЯТОЇ
МІЖНАРОДНОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ ТА МОЛОДИХ ВЧЕНІХ



ПРИСВЯЧЕНА 55-РІЧЧЮ
ІНСТИТУТУ КОМП'ЮТЕРНИХ СИСТЕМ

“Сучасні інформаційні технології 2019”

“Modern Information Technology 2019”



NetCracker®



23-24 травня

Одеса
«Екологія»
2019

УДК 004.05

МЕТОД ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ WEB-САЙТА

Лозинский В. О., Тищенко А. И.

ст. преп. каф. СПО Городничая Е. А., ст. преп. каф. СПО Онищенко Т.В.
Одесский Национальный Политехнический Университет, УКРАИНА

АННОТАЦИЯ. В представленной работе описан метод повышения производительности web-сайта. Данный метод позволяет, в рамках заданной последовательности шагов при загрузке web-страницы, уменьшить время выполнения определенных шагов путем изменения принципов функционирования web-сайта.

Введение. Исследования компании KissMetrics демонстрируют следующую статистику: 47% респондентов отметили, что время загрузки web-страницы не должно превышать двух секунд, 40% респондентов сообщили, что не будут посещать web-сайт, для которого время загрузки web-страницы превышает показатель в три секунды [0]. На основании полученной информации можно сформулировать следующий вывод: увеличение времени загрузки web-страницы приводит к уменьшению численности аудитории web-сайта.

Цель работы. Цель данной работы состоит в разработке метода повышения производительности web-сайта. Для определения эффективности применения разработанного метода, необходимо выполнить эксперимент, в ходе которого определить количественный показатель снижения времени загрузки web-сайта.

Основная часть работы.

Для определения метода повышения производительности web-сайта, следует изучить последовательность шагов при загрузке web-страницы:

14. Выполнение браузером DNS-запроса по указанному доменному имени.

15. В случае, если web-сайт найден, инициируется TCP-подключение к серверу, на котором данный web-сайт расположен.

16. На основании установленного TCP-соединения формируется HTTP-соединение.

17. Браузер выполняет HTTP-запрос кода web-страницы, выполняется загрузка web-страницы на основании полученного от сервера HTTP-ответа.

18. Браузер выполняет парсинг HTML-кода. Парсинг – это процесс выполнения синтаксического анализа кода [0].

19. Динамически загружаются файлы внешних ресурсов.

20. Встроенный в браузер интерпретатор исполняет код, написанный на языке программирования JavaScript.

На основании анализа последовательности шагов при загрузке web-страницы был разработан метод повышения производительности web-сайта:

1. Сокращение количества HTTP-запросов

Для загрузки таких компонент web-сайта, как медиа файлы, таблицы стилей, javascript-скрипты, формируется и выполняется отдельный HTTP-запрос. Как следствие, имеем ситуацию, при которой увеличение количества компонент web-страницы приводит к увеличению количества времени, необходимого для отображения страницы. Сокращение количества HTTP-запросов позволяет решить данную проблему. Способ решения - фильтрация файлов для отбора критически важных.

2. Сжатие и объединение файлов

Сжатие файлов включает в себя удаление форматирования, пробелов и повторяющихся участков кода. Наличие повторяющихся фрагментов кода приводит к увеличению итогового размера файла. Файлы, которые соответствуют таблицам стилей нуждаются в предварительном сжатии и последующей конкатенации в один файл, который будет подключаться тегом «style» в html коде. Полученные в результате выполнения данного этапа файлы будут иметь меньший размер, что приведет к уменьшению времени их загрузки.

3. Использование асинхронной загрузки

Файлы с расширениями .css и .js, могут быть загружены двумя различными способами: синхронно или асинхронно. Синхронная загрузка выполняется последовательно, что может привести к следующей проблеме: файл большого объема, который находится в начале стека, не позволит загружаться файлам, которые расположены следующими в стеке, пока не загрузится в полном объеме. Избежать данной ситуации можно с помощью асинхронной загрузки, которая позволяет загружать файлы одновременно.

4. Минимизация времени загрузки первого байта

Время до первого байта, или TTFB, - это время, которое затрачивается браузером на ожидание получения первого байта информации со стороны сервера. Компания Google рекомендует ограничивать TTFB значением 200 мс [0]. Если TTFB web-сайта превышает 200 мс, то для уменьшения этого значения следует устранить следующие причины: динамическое генерирование web-страниц, настройки сервера, предусматривающие временные резервы.

5. Реализация кеширования

Кэш - промежуточный буфер информации, доступ к которому осуществляется быстрее, чем к серверной стороне. В кэш сохраняется информация, которая запрашивалась пользователем ранее с наибольшей частотой. Предполагается, что именно эта информация будет запрошена пользователем в последующем [0]. В том случае, если пользователь посещает web-сайт при включенном режиме кеширования, элементы отображаемой web-страницы сохраняются в кэше или же во временном хранилище. Это означает, что при повторном запросе web-страницы, браузер, при проверке кэша, обнаружит наличие web-страницы и не будет обращаться на серверную сторону путем HTTP-запросов для получения информации, тем самым уменьшая время загрузки.

6. Выделение в процессе загрузки web-сайта дочерних процессов.

Web-страница состоит из 3 основных частей: верхней части, основной части и нижней части. Каждая из этих частей может загружаться независимо от двух остальных. Как следствие, можно выполнять загрузку основной и нижней частей после полной загрузки верхней части. Это позволит уменьшить минимально допустимое время для взаимодействия пользователя с web-интерфейсом. Полученный временной буфер будет использован для загрузки основной и нижней частей web-страницы.

Для определения эффективности представленного метода было проведено следующий эксперимент: анализировалось время загрузки web-страницы до реализации разработанного метода и после выполнения реализации. В результате выполнения эксперимента было получено следующие данные: первоначальное время загрузки web-страницы составляло 13,8 секунды, а после реализации метода повышения производительности было получено результат 3,5 секунды.

Выводы. В данной работе был разработан метод повышения производительности web-сайта. Метод состоит из 6 этапов, сформулированных на основании проведенного анализа последовательности шагов при загрузке web-страницы. Для оценки эффективности использования представленного метода был проведен эксперимент, в результате выполнения которого были получены следующие результаты: время загрузки web-сайта удалось уменьшить на 75% (в 4 раза).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. How Loading Time Affects Your Bottom Line [Электронный ресурс]. – Режим доступа: URL: <https://neilpatel.com/blog/loading-time/>. – Назва з екрана.
2. Парсинг [Электронный ресурс]. – Режим доступа: URL: <https://www.ipipe.ru/info/parsing.html>. – Назва з екрана.
3. Improve Server Response Time [Электронный ресурс]. – Режим доступа: URL: <https://developers.google.com/speed/docs/insights/Server>. – Назва з екрана.
4. Что такое кэш [Электронный ресурс]. – Режим доступа: URL: <https://ktonanovenkogo.ru/voprosy-i-otvety/chto-takoe-kesh-kak-ochistit-v-opere-mozile-xrome-brauzerax.html>. – Назва з екрана.