

## ПОВЫШЕНИЕ СКОРОСТИ ЗАГРУЗКИ ВЕБСАЙТА

Мойсенович А. В.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Дубинко Н. А. – к.п.н., доцент

Скорость загрузки страниц сайта является важным показателем влияющий на удобство использования сайта пользователями [1]. В последнее время происходит стремительный рост мобильного интернета и всё больше и больше людей посещают сайты с мобильных устройств скорость соединения с интернетом которых может не превышать 54 Кбит/с [2]. Большинство пользователей покинут вебсайт, если его время загрузки превысит 2 - 3 сек, а идеальными будут показатели равные времени реакции пользователя - примерно 0.5 сек [2].

Целью данной работы является разбор и анализ основных путей повышения скорости загрузки страниц интернет вебсайта.

К основным путям по повышению скорости загрузки сайта относятся:

1. сокращение времени ответа сервера;
2. применение кеша браузера;
3. оптимизация изображений;
4. сокращение CSS, JavaScript, HTML;
5. использование статического и динамического сжатия для статических ресурсов.

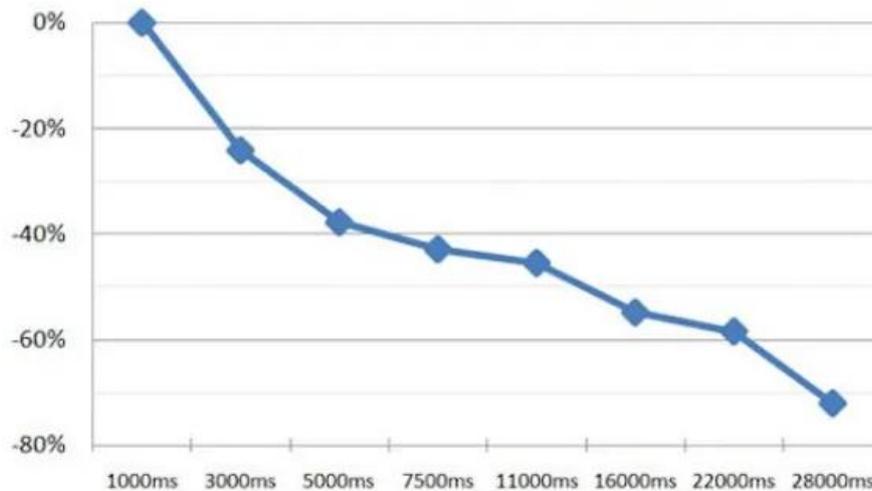


Рисунок 1 – Снижение конверсии в зависимости от времени загрузки сайта

Сокращение время ответа от сервера представляет собой оптимизацию показателя TTFB (Time to first byte) время, до приёма первого байта информации от сервера [2]. Проверить это значение можно при помощи стандартных средств разработчика браузера. Идеальными значениями для данного показателя являются <50 мс. и должны быть незначительно выше времени передачи пакетов от клиента до сервера [2]. Негативное воздействие на время ответа сервера оказывают:

1. медленная работа с базой данных;
2. нехватка ресурсов (слабый процессор или недостаточный объем оперативной памяти);
3. неправильная конфигурация сервера;
4. отсутствие кэширования.

Применение кеша браузера для меморизации статических ресурсов является распространенным способом оптимизации производительности современных веб приложений [2]. Основными способами кэширования являются применения отправляемых сервером браузеру специальных заголовков:

1. cache-control – представляет собой директиву для механизма кэширования браузера состоящую из 4 сегментов: возможность кэширования ресурса, время кэширования, ревалидация и перезагрузки, дополнительных параметров;
2. expires – описывает дату и время по истечению которых кеш считается устаревшим;
3. etag – содержит уникальный идентификатор ресурса, который позволяет проверить соответствие версии. Например, можно использовать хеширование, для получения идентификатора файла;
4. vary – позволяет указать, какие заголовки нужно использовать для определения возможности использования закэшированного ресурса.

Отдельно стоит упомянуть про современный способ кеширования ресурсов при недоступности веб сервера или сетевого подключения – ServiceWorker. Технология пришла на замену manifest файлам и построена на базе WebWorkers [2]. Идея заключается в создании отдельного воркера с возможностью подписи на запросы получения ресурсов с целью их перехвата, анализа, и возвращением закешированного результата до того, как запрос уйдёт от клиента.

Оптимизация изображений и видео файлов заключается в их предварительной обработке с целью повышения сжатия специальными средствами, однако это только часть решения [2]. HTML5 стандарт привнёс много новшеств и возможностей по созданию так называемых responsive (отзывчивых) изображений при помощи атрибутов srcset и sizes для тегов img:

```

```

Рисунок 2 – пример img тега

Атрибут srcset позволяет указать название изображения, а также его ширину в пикселях. Sizes в свою очередь указывает соотношение между размером дисплея и размером области, которое занимает изображение. Имея все эти данные, браузер может выбрать, какое изображение загружать для определённого размера экрана до его загрузки, что позволяет использовать разные картинки для больших дисплеев и мобильных устройств.

Использование статического или динамического сжатия представляет собой архивирование файлов в результате сборки проекта или непосредственно сервером по мере возникновения необходимости в файле. Если браузер, вместе с запросом на файл отправит заголовок Accept-Encoding содержащий значение gzip, сервер попытается вернуть сжатую версию файла. Динамическое сжатие выполняется самим веб сервером и не требует дополнительных действий со стороны программиста, однако добавляет нагрузку на сам сервер, который теперь будет выполнять архивирование на лету. Статическое сжатие подразумевает наличие подготовленных, архивированных файлов в корне вебсайта. Архивирование представляет собой мощный инструмент, который, в некоторых случаях, позволяет уменьшить размер статических файлов сайта более чем на 60%. Также, хотелось бы заметить, что во время статической компрессии, возможно использование архиватора с более высокой степенью сжатия, что усилит достигнутый эффект. Низкая скорость мешает пользователю быстро получить доступ к необходимой информации, что обычно приводит к росту показателей отказа, а современные поисковые системы такие как Google, Yandex, Bing учитывают показатель скорости загрузки при ранжировании сайтов в поисковой выдаче [3].

Таким образом, для улучшения скорости загрузки вебсайта необходимо:

1. использовать современные HTML5 теги и атрибуты для изображений и видео;
2. увеличить время жизни кеша и добавить процесс ревалидации;
3. применить статическое сжатие.

**Список использованных источников:**

1. Вайнштейн, Л. А. Эргономика: учебное пособие / Л. А. Вайнштейн. Минск: ГИУСТ БГУ, 2010. – 399 с.
2. Методика повышения скорости загрузки сайтов, сделанных на системах управления содержимым. // Системы обработки информации. Выпуск 4 (150). Харьков 2017. Харьковский университет воздушных сил имени Ивана Кожедуба. с. 92-96.
3. Эргономика информационных систем: пособие / Т. В. Калилец, В. С. Осипович, И. Ф. Киринович, В. В. Савченко, К. Д. Яшин.– Минск, БГУИР, 2017. – 73 с.