

# Концептуальная модель загрузки веб-страницы

С. В. Жуков, О. А. Ковалева, С. В. Ковалев

Тамбовский государственный университет им. Г. Р. Державина, Тамбов, Россия

**Аннотация.** В статье рассматривается разработанная авторами концептуальная модель загрузки веб-страницы. Модель состоит из трех сущностей: Backend (характеризующаяся метриками: Page creation time, Number of SQL queries, SQL query execution time, Cache size, Page size, Number of errors), environment (характеризующаяся метриками: DNS, TLS, Connection, Load speed), frontend (характеризующаяся метриками: FCP, LCP, TBT, CLS, SI). Представленная концептуальная модель предлагает рассматривать процесс загрузки страницы, включающий все этапы формирования от запроса страницы пользователем до отрисовки полученного контента. Ее применение в процессе оптимизации скорости загрузки веб-страницы дает новый взгляд на возможные проблемы веб-проекта. В работе представлен пример применения данной модели для решения актуальной задачи по оптимизации скорости загрузки страницы.

**Ключевые слова:** концептуальная модель, backend, environment, frontend, веб-страница.

DOI 10.14357/20718632240203

EDN LWCJNC

## Введение

В современном обществе веб-пространство является важной площадкой для размещения информации, обмена мнениями, оказания и заказа услуг. Интернет-ресурсы стремятся предоставить разнообразный, качественный контент и при этом обеспечить максимально комфортный пользовательский опыт. В данном контексте, одним из определяющих факторов успешного веб-сайта становится скорость загрузки его страниц. В статье [1] ее отмечают в качестве одного из критериев, определяющих эффективность SEO. В работе [2] автор отмечает, что оптимизация скорости загрузки страницы является одной из важнейших работ по внутренней оптимизации сайта, так как позволяет сайту занимать более высокие строчки в выдаче поисковых систем. В статье [3] авторы также указывают на важную роль скорости загрузки страницы для продвижения мобильных версий веб-сайтов. В работе [4]

по результатам интернет-опроса 1319 пользователей скорость загрузки страниц была признана важным элементом веб-сайта, первым среди его функциональных элементов. Эти исследования подтверждают важность влияния скорости загрузки страницы на пользовательский опыт взаимодействия с веб-сайтом.

Сервисы аналитики для веб-сайтов, например, Яндекс Метрики или Google Analytics, позволяют веб-мастерам увидеть, что на сайте есть проблема с долгой загрузкой страниц. Для того чтобы детально рассмотреть причину этой проблемы часто применяют специальные веб-сервисы вроде PageSpeed Insights, WebPageTest или GTmetrix. Они дают оценку скорости загрузки страницы и рекомендации для ее улучшения. Однако они не могут проанализировать весь процесс загрузки страницы.

Браузер инициирует процесс загрузки веб-страницы, формируя запрос к серверу. Этот запрос содержит информацию о необходимом

ресурсе, например, URL страницы, заголовки с дополнительными параметрами, такими как язык, предпочтения сжатия и другие. При помощи DNS серверов браузер определяет целевой сервер, и устанавливает с ним соединение. Веб-сервер получает HTTP-запрос и начинает обработку. Этот этап включает в себя поиск запрошенного ресурса, его сбор и формирование ответа. В случае динамических веб-страниц, сервер может взаимодействовать с базой данных или выполнить другие вычислительные задачи. После обработки запроса сервер формирует HTTP-ответ, который включает в себя заголовки с метаданными и тело ответа с самим контентом. Этот ответ может содержать HTML-код, стили, скрипты, изображения и другие ресурсы, необходимые для полного отображения веб-страницы. Ответ от сервера передается по сети к браузеру пользователя. Время передачи данных зависит от различных факторов, таких как пропускная способность сети, удаленность сервера, наличие сжатия данных и другие параметры. После получения данных браузер начинает процесс отображения веб-страницы. Он разбирает HTML-код, загружает стили и скрипты, обрабатывает изображения.

В современных работах на тему увеличения скорости загрузки страницы авторы рассматривают в основном вопросы, связанные с оптимизацией процессов отображения страницы в браузере пользователя. Эти вопросы поднимались как в работе [5], где были предложены 4 подхода в оптимизации загрузки HTML страниц, так и в работе [6], где были предложены такие методы уменьшения времени загрузки веб-страницы как корректировка структурного HTML файла, объединение файлов стилей, использование механизмов сжатия контента (GZIP сжатие веб-сервером). В работе [7] выделены 11 этапов повышения скорости загрузки сайта, из которых только два относились к процессам формирования страницы на стороне сервера. В статьях [8–10] авторы рассматривали возможность повышения скорости отображения страниц путем применения методов Text-First и Layout-First, ускоряющих первую «отрисовку» контента на экране, сокращения размеров критического CSS кода за счет использования серверного визуализатора Essential, устранения

блокировки рендеринга JavaScript и CSS файлами. В работе [11] был представлен алгоритм позволяющий определить, какой код JavaScript можно безопасно отложить для повышения скорости отображения страницы.

Альтернативный подход к оптимизации скорости загрузки страницы был рассмотрен в работе [12]. Авторы реализовали свой веб-ресурс по архитектуре SPA (Single Page Applications) с использованием современных компонентов и инструментов, что позволило значительно улучшить показатели скорости загрузки ресурса по сравнению со старой версией сайта. Однако такая переработка для классического MPA (Multi Page Application) ресурса сопряжена с большими трудозатратами и проблемами в SEO продвижении, поэтому широко не применяется. В статье [13] авторы показали как применение HTTP/2 может улучшить производительность веб-страницы, и как это улучшение связано с ее структурой. В работе [14] рассматривалась оптимизация производительности веб-приложения с использованием метода быстрой загрузки базы данных «ключ-значение».

В ряде работ [15–18] для оценки скорости загрузки страницы часто применяется один инструмент, фиксирующий в основном метрики «отрисовки» страницы в браузере, который подробно описан в работе [19]. Однако он не может показать проблемы при формировании страницы на стороне сервера. Существуют и другие инструменты, дающие оценку сайту, в том числе по скорости загрузке страниц. Подробнее они рассматриваются в работе [20].

Веб-сервисы не могут проанализировать этапы формирования страницы на сервере, а серверные скрипты аналитики не дают полных отчетов о поведении страницы в браузере, что не позволяет использовать один инструмент для всестороннего изучения проблем, связанных со скоростью загрузки страницы. Ускорение загрузки веб-страницы требует не только оптимизации клиентской стороны веб-сайта, но также оптимизации процессов на сервере, а также контроля оптимальной доставки контента до конечного пользователя. Для того чтобы не упустить важных факторов, влияющих на скорость загрузки страницы, необходима модель, способная учитывать все этапы формирования, транспортировки и обработки данных.

Поэтому целью данной работы является разработка концептуальной модели, которая включает в себя анализ времени, затраченного на генерацию контента на сервере, передачу данных по сети и обработку на стороне клиента.

Разработка данной модели является актуальной задачей, т.к. позволяет систематизировано изучать различные показатели скорости загрузки страницы, проверяя каждую метрику в модели. Поддержание высокой скорости загрузки страниц - важная задача т.к. от нее зависит ранжирование сайта в поисковой выдаче и пользовательский опыт посетителей сайта. Влияние скорости загрузки страницы на эти факторы описано в статье [21].

Для достижения поставленной цели необходимо рассмотреть этапы формирования страницы, и на их основании выделить ряд метрик, характеризующих процесс в целом. На основании этих данных можно сформулировать концептуальную модель загрузки страницы. Для демонстрации принципа работы с ней нужно привести примеры поиска проблем в работе веб-сайта.

## 1. Принцип построения концептуальной модели

Современный процесс загрузки веб-страниц представляет собой сложный механизм, включающий несколько ключевых этапов. Эти этапы включают формирование запроса, обработку на стороне сервера, передачу данных и отображение контента на стороне клиента. В данном разделе рассмотрим каждый этап подробно, чтобы понять какие показатели при загрузке страницы мы можем измерить и на какие из них можно влиять.

Загрузка веб-страницы начинается с того, что пользователь запрашивает страницу через ввод URL в адресную строку браузера или переходом по гиперссылке. Так начинается первый этап получения страницы – отправка запроса на сервер. В идеале он не должен занимать много времени, но может столкнуться с задержками из-за сетевых факторов.

Первый шаг при отправке запроса — поиск адресата запроса. Например, при посещении <https://some-site.ru>, HTML-страница находится на сервере с IP адресом 94.172.199.42. Для человека сайты идентифицируются доменными име-

нами, в то время как для компьютеров они идентифицируются IP-адресами. Поэтому при первом посещении сайта происходит поиск IP адреса в системе доменных имен (DNS). DNS-серверы, подобно телефонной книге, содержат базу данных публичных IP-адресов и соответствующих им имен хостов. Они служат для конвертации доменных имен в IP-адреса по запросу.

В процессе поиска DNS-браузер взаимодействует с сервером, запрашивая соответствие IP-адреса для <https://some-site.ru>. Если найдено соответствие, оно предоставляется, в противном случае, появляется сообщение об ошибке. Полученный IP-адрес может быть сохранен в кэше, ускоряя последующие посещения сайта, так как не требуется нового поиска DNS. Запрос DNS осуществляется только при первом посещении. Затем следует установка соединения через TCP [22].

Когда соединение TCP установлено, можно начинать обмен данными по TLS. Этот дополнительный этап нужен для безопасных соединений. Он потребует дополнительных 5 шагов, начиная от выбора шифра до установки безопасного соединения. Важно отметить, что, несмотря на замедление процесса отправки запроса, стоит подождать, пока будет обеспечена безопасность соединения при помощи шифрования, так как оно гарантирует надежность соединения и защиту данных. Кроме того, некоторые поисковые системы (например Google) учитывают поддержку https соединения при ранжировании сайта в поисковой выдаче [23].

Когда сервер получает от браузера запрос, содержащий детальное описание того, что требуется, начинается процесс обработки, который в значительной степени зависит от используемого серверного программного обеспечения, такого как Nginx или Apache. Эти программы, играют ключевую роль в обработке запросов и взаимодействии с программами-обработчиками, такими как PHP, Ruby и т.д.

Веб-сайты чаще всего реализуются с использованием PHP, одного из наиболее распространенных языков программирования для веб-разработки. Для хранения данных веб-ресурса часто прибегают к использованию баз данных, таких, как, например, MySQL, PostgreSQL и MongoDB. Сложные вычисления, результаты работы внешних сервисов, высоко нагруженные

SQL запросы – все это факторы, увеличивающие время формирования страницы. Для устранения их влияния применяют технологию кэширования. Кэш — это временное хранилище данных, позволяющее ускорить доступ к ресурсам. Например, веб-сервер может временно сохранять результаты выполнения запросов или часто используемые данные, что существенно снижает время ответа на повторные запросы. В случае возникновения ошибок в процессе формирования страницы веб-сервер или программа-обработчик в большинстве случаев создают специальные файлы, называемые логами, с описанием произошедших сбоев. Результат работы веб-сервера отправляется пользователю в ответ на его запрос.

Браузер получает ответ от сервера спустя некоторое время после его отправки. Время, затрачиваемое на транспортировку ответа, может варьироваться в зависимости от состояния сети и удаленности сервера. Чтобы браузер смог отобразить html-страницу, он должен выполнить ряд вычислений.

Прежде всего, нужно построить дерево DOM (Document Object Model). Оно создается из HTML-разметки и представляет собой структуру и содержание страницы. Параллельно с этим процессом работает сканер предзагрузки, позволяющий начать загружать важные ресурсы страницы (вроде CSS и JavaScript) до того момента, как их найдет основной поток HTML парсера.

Следующий шаг в отображении страницы — это построение дерева CSSOM. Браузер преобразует CSS-файлы в CSSOM для их понимания и обработки. Каждый набор правил в CSS считывается браузером, и на основе селекторов создается дерево узлов. Этот процесс аналогичен формированию DOM в HTML, но применительно к CSS. Процесс построения происходит быстро [24].

После него начинается работа с JavaScript. По окончании загрузки JavaScript код должен быть интерпретирован, скомпилирован, обработан и исполнен. Скрипты преобразовываются в абстрактное синтаксическое дерево (AST). Некоторые браузеры берут Abstract Syntax Tree и передают его в интерпретатор, который преобразует дерево в байт-код. Байт-код исполняется в основном потоке. Весь этот процесс называется компиляцией.

Финальные этапы рендеринга включают стилизацию, компоновку (layout), «отрисовку» (paint) и, в некоторых случаях, композицию (composition). На предыдущем этапе CSSOM- и DOM-деревья объединяются в дерево рендера, которое затем используется для определения расположения каждого видимого элемента. После этого элементы отображаются на экране. В определенных сценариях контент может быть вынесен на отдельные слои и объединен (composition), что повышает производительность, позволяя «отрисовывать» содержимое экрана на графическом процессоре вместо центрального процессора, освобождая основной поток. В случае если какие-то части JavaScript-кода были отложены до завершения загрузки страницы, то они начнут исполняться после данного этапа [25].

Говоря о скорости загрузки страницы нужно отметить, что с точки зрения пользователя время, затрачиваемое браузером до полной «отрисовки» страницы, и время, воспринимаемое пользователем как окончание загрузки страницы, не равны. Поэтому для оценки того, насколько быстро загружается страница с точки зрения пользователя, существует ряд метрик, оценивающих не фактическое время завершения загрузки страницы, а то, как пользователь воспринимает процесс ее загрузки. Так как эти метрики включают в себя и общую оценку скорости загрузки страницы, в модели целесообразнее использовать их.

## 2. Концептуальная модель загрузки страницы

Рассмотрев ключевые этапы формирования страницы можно перейти к описанию концептуальной модели, представленной на Рис. 1.

Концептуальная модель загрузки страницы — это абстрактное представление о том, как происходит процесс загрузки страницы и предназначена для более глубокого понимания данного процесса, помогает его анализировать и работать над его улучшением. Концептуальная модель визуализирует основные сущности, участвующие в отображении страницы, и описывает их метрики. На Рис. 1 она представлена визуально в виде ER-диаграммы.

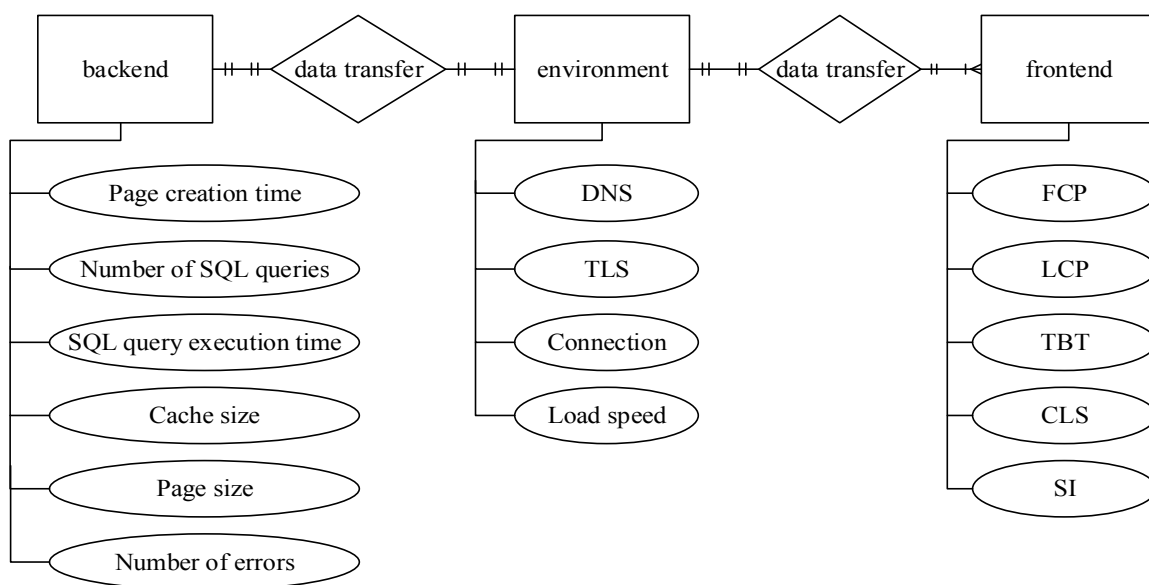


Рис. 1. ER-диаграмма концептуальной модели загрузки веб-страниц

Модель состоит из трех сущностей: frontend объединяет все метрики, связанные с обработкой страницы в браузере, environment – все метрики, связанные с передачей запроса от клиента к серверу и обратно, backend – все метрики, связанные с формированием страницы на сервере.

Для использования модели, следует проследить цепочку формирования страницы по ER-диаграмме. Сначала один из множества пользователей посылает запрос одному серверу (множество сущностей frontend связаны с конкретным environment). В рамках передачи запроса на сервер мы можем отследить DNS – время, затраченное на поиск IP адреса сервера, TLS – время, затраченное на установку защищенного подключения, Connection – время, затраченное на отправку запроса к IP сайта, Load speed – скорость передачи данных. После передачи данных через environment запрос приходит на backend.

Обработка запроса на сервере (в сущности backend) характеризуется следующими параметрами: Page creation time – общим временем формирования страницы; Number of SQL queries – количеством SQL запросов, выполненных для формирования страницы; SQL query execution time – общим временем, потраченным на выполнение SQL запросов; Cache size – размером кэша, используемого для формирования страницы; Page size – конечным размером страницы

после сжатия; Number of errors – количеством ошибок, полученных при формировании страницы. После формирования ответа запрос возвращается к пользователю через сущность environment.

Пользователь получает данные страницы в ответ на свой запрос. Эти данные обрабатываются в сущности frontend. При обработке формируются новые запросы к серверу, проходящие по вышеописанным шагам. Первый запрос страницы характеризуется метриками из Google Lighthouse v10: FCP – First Contentful Paint; LCP – Largest Contentful Paint; TBT – Total Blocking Time; CLS – Cumulative Layout Shift; SI – Speed Index.

Проверив каждую из указанных метрик, можно найти аномалии: слишком низкие или слишком высокие показатели. Совокупность нескольких аномалий может свидетельствовать о проблемах при загрузке страницы в области той сущности, в которой выявлена аномалия. Задача веб-разработчика – проверить причины аномалий и устранить их при необходимости.

Суммарное время ожидания ответа от сервера (общее время формирования страницы плюс время на отправку и получения запроса) должно быть около 240мс. Приемлемые значения для метрик сущности frontend зависят от текущей актуальной Lighthouse и приведены в документации данного инструмента [26].

3. Применение разработанной модели

Для демонстрации применения концептуальной модели загрузки страницы рассмотрим процедуру проверки скорости загрузки веб-ресурса по шагам на примере веб-сайта интернет-магазина. Сайт работает на базе CMS «1С-Битрикс: Управление сайтом» версии 23.500.0, запущенном в «1С-Битрикс: Веб-окружение» версии 7.5.2 с php 8.1.22, MySQL 5.7.40-43 на VPS сервере с операционной системой CentOS 7, 8 Гб оперативной памяти, 4 ядра 3.6 ГГц.

Сначала проверим показатели сущности environment. Для этого можно использовать панель разработчика браузера или один из сервисов проверки веб-страниц, отображающий подробности запросов, посылаемых браузером, например, GTmetrix или WebPageTest. Рассмотрим такую детализацию на примере последнего.

На Рис. 2, видно, что при загрузке веб-сайта происходит дополнительный запрос к серверам центра сертификации, а также тратится время на установку TLS соединения - крайне высокое, около секунды. Такое поведение запроса сигнализирует о том, что в настройке сертификата сайта допущена ошибка. Проверим это с помощью сервиса LeaderSSL. На Рис. 3 представлен результат проверки сайта.

Подтвердилась ошибка настройки SSL сертификата для установки защищенного соединения. В данном случае не была задана полная цепочка сертификатов.

После проверки ресурсов одной страницы можно воспользоваться сервисом для проверки доступности сайта из разных регионов и разных стран. Например, можно использовать сервис Ping-Admin. Если других аномалий не обнаружено, переходим к проверке сущности backend.

Для проверки большинства метрик стоит поискать инструменты, разработанные под тот программный продукт, на базе которого работает веб-сайт. Если таких инструментов не предусмотрено, их придется разработать. Для примера рассмотрим такой инструмент отладки, предоставляемый Битриксом. В панели управления можно активировать режим отладки, после чего система покажет детальную статистику по использованным ресурсам для создания каждого компонента страницы. Кроме того, можно посмотреть общее время создания страницы, количество и продолжительность SQL запросов, размер кэша. По каждому пункту можно получить детализацию. На рассматриваемом сайте была обнаружена аномалия – долгое время создания страниц Page creation time - 726 мс (Рис. 4). При этом SQL запросы и шаблон сайта

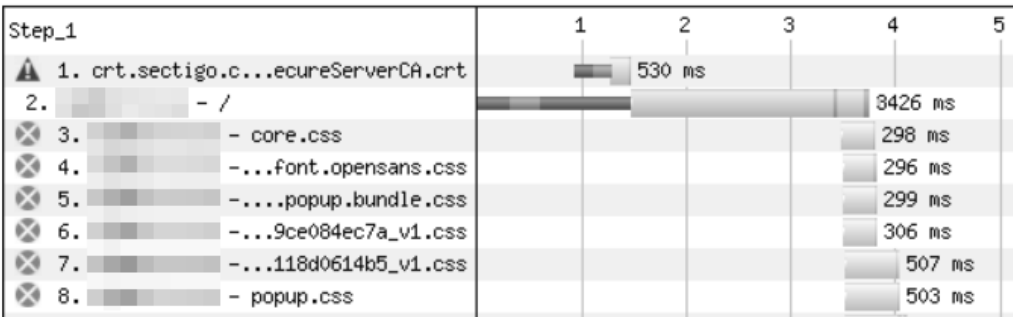


Рис. 2. Детализация запросов, отправляемых браузером для загрузки веб-страницы

✖ Невозможно получить сертификат локального эмитента. Возможно, сертификат сайта или промежуточные сертификаты установлены некорректно.

🔒 Сервер поддерживает устаревший (менее безопасный) протокол SSL v2. Настоятельно рекомендуем отключить его

Информация о сервере

IP адрес(а)

-

Домен		
Сертификат	Альтернативные имена (SANs)	
	Тип сертификата	Подтверждение организации (OV)
Подлинность	Действителен с	✓ 12 октября 2021

Рис. 3. Проверка SSL-сертификата через сервис LeaderSSL

Страница: <span style="background-color: #cccccc; border: 1px solid #000; padding: 2px;"> </span> Автокеширование компонентов: включено.						
	Исполнение	сек.	Компоненты	сек.	Запросы	сек.
<b>Страница</b>	<b>100.00%</b>	<b>0.7268</b>	<b>40</b>	<b>0.1510</b>	<b>283</b>	<b>0.0841</b>
<b>Пролог</b>	<b>83.74%</b>	<b>0.6086</b>	<b>30</b>	<b>0.1376</b>	<b>227</b>	<b>0.0671</b>
<b>Ядро</b>	<b>63.96%</b>	<b>0.4649</b>	<b>0</b>	<b>0.0000</b>	<b>153</b>	<b>0.0472</b>
<b>Агенты</b>	<b>0.00%</b>	<b>0.0000</b>	<b>0</b>	<b>0.0000</b>	<b>0</b>	<b>0.0000</b>
<b>Шаблон</b>	<b>19.78%</b>	<b>0.1437</b>	<b>30</b>	<b>0.1376</b>	<b>74</b>	<b>0.0198</b>

Рис. 4. Режим отладки в CMS Битрикс, часть окна детализации времени создания страницы (на примере главной страницы сайта)

```

2023-12-19 16:33:25 Host: [redacted] UNKNOWN [ErrorException] E_WARNING Memcached::pconnect(): Can't connect to 127.0.0.1:11216, Connection refused (111) (0)
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/configurator/memcachedconnectionconfigurator.php:73
#0: Bitrix\Main\Data\Configurator\MemcachedConnectionConfigurator->createConnection()
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/memcachedconnection.php:31
#1: Bitrix\Main\Data\MemcachedConnection->connectInternal()
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/connection.php:53
#2: Bitrix\Main\Data\Connection->getResource()
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/cacheengine/memcached.php:61
#3: Bitrix\Main\Data\CacheEngine\Memcached->__construct(array)
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/cache.php:99
#4: Bitrix\Main\Data\Cache::createCacheEngine(array)
/var/www/www-root/data/www/[redacted]/bitrix/modules/main/lib/data/cache.php:156

```

Рис. 5. Часть лога CMS Битрикс

не имели существенного влияния на скорость формирования страницы.

По детализации видно, что больше всего времени заняла обработка файлов ядра CMS, что нетипично для данной системы. Для уточнения проблемы проверим логи системы. Часть логов приведена на Рис. 5. В логах видим ошибку работы системы: не получается подключить Memcached.

После того как проверка сущности backend завершена и все возможные аномалии устранены, переходим к проверке сущности frontend. Для нее лучше всего использовать сервис PageSpeed Insights. Он работает на базе Google Lighthouse и позволяет не только измерить нужные метрики, но и получить рекомендации по устранению аномалий.

Сервис показывает множество проблем и возможных путей их устранения. Результаты теста могут отличаться от страницы к странице из-за разного наполнения контентом. Значения метрик в серии тестов одной страницы могут незначительно отличаться из-за непостоянства скорости загрузки ресурсов и порядка их обработки.

Продemonстрируем замеры, сделанные только для главной страницы сайта. Она состоит из 16 блоков (каждый блок формируется своим

компонентом), не включая «шапку» и «подвал». От начала до конца страницы расположены: слайдер с баннерами, ключевые преимущества магазина в виде векторных иконок и подписями, три блока, презентующих основные разделы каталога товаров (по типу товара, по производителю, по применению) в виде иллюстраций с текстом, три слайдера с товарами (акции, популярное, новинки), текстовый блок, иллюстрированный процесс производства товара, форма обратной связи, текстовые блоки с малыми изображениями, блок ссылок на социальные сети, динамические карты с отметками расположения физических магазинов в разных городах, облако тегов из разделов сайта. К странице подключены системы веб-аналитики, онлайн чат. Всего для загрузки страницы требуется загрузка 226 ресурсов из которых 76.4% (по объему) приходится на ресурсы сайта.

Для примера рассмотрим исправление одного показателя из метрик – CLS. Он увеличивается, если во время загрузки страницы макет смещается. В данном случае проблема была в том, что первый слайдер с баннерами на первом экране не отображался до тех пор, пока не будут загружены его скрипты. После появления слайдер занимал значительную часть экрана,

Табл. 1. Отчет PageSpeed Insights для мобильного и десктопного устройства о скорости загрузки страницы до и после исправления верстки слайдера

Тест	Платформа	FPC	SI	TBT	LCP	CLS	Итоговый балл
До исправления верстки слайдера	Mobile	3,0	11,3	2240	18,5	1.485	7
	PC	1,0	2,9	100	2,0	0.778	58
После исправления верстки слайдера	Mobile	3,6	10,2	1 490	8,2	0.057	33
	PC	1,0	3,1	180	2,0	0.003	77

смещая все нижестоящие элементы страницы вниз. После исправления верстки слайдера показатель CLS заметно улучшился, что продемонстрировано в Табл. 1. Выполняя рекомендации сервиса, следует привести все метрики сущности настолько близко к оптимальным показателям, насколько это возможно.

## Заключение

Роль пользовательского опыта в контексте продвижения веб-сайтов и общей эффективности взаимодействия с ними высока. Одним из существенных факторов, влияющих на пользовательский опыт, является скорость загрузки веб-страницы. Всесторонняя оценка процесса формирования веб-страницы важна для улучшения данного показателя.

В работе были рассмотрены этапы формирования веб-страницы, на основании которых были выделены метрики для оценки процесса загрузки страницы, а также приведены метрики, предлагаемые сервисом Google Lighthouse для анализа скорости загрузки веб-страницы с точки зрения пользователя. Используя эти данные авторами статьи была разработана концептуальная модель загрузки страницы. Визуально модель отображена в виде ER-диаграммы. Поиск проблем в скорости загрузки сайта с помощью данной модели позволяет не пропускать важные аспекты формирования страницы, что приводит к увеличению эффективности поиска. В качестве примера применения модели был продемонстрирован процесс поиска факторов, влияющих на замедление скорости загрузки веб-страницы.

В дальнейшем данная модель может использоваться как основа для формирования

чек-листа проверки скорости загрузки веб-ресурсов, написанных на разных CMS системах, фреймворках, языках программирования.

## Литература

- Özkan, B. Evaluating the websites of academic departments through SEO criteria: a hesitant fuzzy linguistic MCDM approach / B. Özkan, E. Özceylan, M. Kabak, Dağdeviren M. // *Artificial Intelligence Review*. – 2020. – Т. 53. – С. 875-905. – DOI 10.1007/s10462-019-09681-z.
- Жеребцова, К. О. SEO-продвижение сайта: специфика, механика, особенности / К. О. Жеребцова // *Экономика в теории и на практике: актуальные вопросы и современные аспекты: сборник статей IX Международной научно-практической конференции*, Пенза, 25 июня 2021 года. – Пенза: Наука и Просвещение, 2021. – С. 33-38.
- Элиханов, В. Г. Мобильное позиционирование сайта / В. Г. Элиханов, Э. Р. Гузиева, З. М. Мушурова // *Экономика: вчера, сегодня, завтра*. – 2020. – Т. 10, № 9-1. – С. 567-572. – DOI 10.34670/AR.2020.79.17.066.
- Полухин, Н. В. Информационные предпочтения пациентов в контексте коммуникации на веб-сайтах медицинских организаций / Н. В. Полухин, Н. В. Эккерт, В. В. Козлов // *Siberian Journal of Life Sciences and Agriculture*. – 2021. – Т. 13, № 5. – С. 226-246. – DOI 10.12731/2658-6649-2021-13-5-226-246.
- Бирюков, М. А. Обзор методов увеличения скорости загрузки веб-страниц / М. А. Бирюков, О. А. Якубова // *Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022): XI Международная научно-техническая и научно-методическая конференция*, Санкт-Петербург, 15–16 февраля 2022 года. Том 1. – Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2022. – С. 160-165.
- Горячкин, Б. С. Повышение эффективности работы с веб-ресурсом за счет инструментария системного программиста / Б. С. Горячкин, Т. И. Ханмурзин // *Динамика сложных систем - XXI век*. – 2022. – Т. 16, № 3. – С. 26-39. – DOI 10.18127/j19997493-202203-03.
- Воронова, А. М. Технология повышения скорости загрузки сайта / А. М. Воронова // *Наука XXI века: Вызовы, становление, развитие: Сборник статей VIII Меж-*



- дународной научно-практической конференции, Петрозаводск, 10 января 2023 года. – г. Петрозаводск: Международный центр научного партнерства «Новая Наука» (ИП Ивановская И.И.), 2023. – С. 267-272.
8. Vogel, L. User Acceptance of Modified Web Page Loading Based on Progressive Streaming / L. Vogel, T. Springer // Web Engineering. – 2022. – Т. 13362. – С. 391-405. – DOI 10.1007/978-3-031-09917-5\_27.
  9. Vogel, L. Speed Up the Web with Universal CSS Rendering / L. Vogel, T. Springer // Web Engineering. – 2023. – Т. 13893. – С. 191-205. – DOI 10.1007/978-3-031-34444-2\_14.
  10. Vogel, L. An In-Depth Analysis of Web Page Structure and Efficiency with Focus on Optimization Potential for Initial Page Load / L. Vogel, T. Springer // Web Engineering. – 2022. – Т. 13362. – С. 101-116. – DOI 10.1007/978-3-031-09917-5\_7.
  11. Kloos, J. Deferrability Analysis for JavaScript / J. Kloos, R. Majumdar, F. McCabe // Hardware and Software: Verification and Testing. – 2017. – Т. 10629. – С. 35-50. – DOI 10.1007/978-3-319-70389-3\_3.
  12. Технологии оптимизации работы сайта на примере аналитической системы публикационной активности пензенского государственного университета / Р. А. Торопкин, Я. В. Зиновьев, Н. С. Рассказов, М. А. Митрохин // Модели, системы, сети в экономике, технике, природе и обществе. – 2020. – № 4(36). – С. 71-78. – DOI 10.21685/2227-8486-2020-4-7.
  13. Zarifis, K. Modeling HTTP/2 Speed from HTTP/1 Traces / K. Zarifis, M. Holland, M. Jain, E. Katz-Bassett, R. Govindan // Passive and Active Measurement. – 2016. – Т. 9631. – С. 233-247. – DOI 10.1007/978-3-319-30505-9\_18.
  14. Priya, M.D. Optimization of Web Applications Using Eager Loading / M.D. Priya, B. Sabarinathan, M. Surya // Proceedings of 6th International Conference on Recent Trends in Computing. – 2021. – Т. 177. – С. 711-722. – DOI 10.1007/978-981-33-4501-0\_66.
  15. Шконда, Д. Н. Тестирование производительности интернет-страниц / Д. Н. Шконда, М. Е. Щелкунова // Наука, инновации и технологии: от идей к внедрению: Материалы Международной научно-практической конференции, Комсомольск-на-Амуре, 07–11 февраля 2022 года. – Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2022. – С. 75-77.
  16. Слободянюк, А. С. Проведение нагрузочного тестирования сайта ВКонтакте / А. С. Слободянюк, М. Е. Щелкунова // Вестник научного общества студентов, аспирантов и молодых ученых. – 2023. – № 1. – С. 93-97.
  17. Буйневич, М. В. Анализ результатов аудита сетевых информационных ресурсов МЧС России / М. В. Буйневич, А. В. Максимов, А. В. Вострых // Научно-аналитический журнал "Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России". – 2020. – № 1. – С. 101-110.
  18. Солозобов, О. А. Состояние и проблемы технической оптимизации сайтов российских букмекерских контор для мобильного поиска / О. А. Солозобов // Вестник науки и образования. – 2019. – № 4-2(58). – С. 37-42.
  19. Толстых, М. А. Применение инструмента Lighthouse для анализа качественного состояния web - приложения / М. А. Толстых // Концепции, теория и методика фундаментальных и прикладных научных исследований: сборник статей Международной научно-практической конференции, Екатеринбург, 05 февраля 2021 года. – Уфа: Общество с ограниченной ответственностью "Аэтерна", 2021. – С. 67-69.
  20. Namoun, A. A Review of Automated Website Usability Evaluation Tools: Research Issues and Challenges / A. Namoun, A. Alrehaili, A. Tufail // Design, User Experience, and Usability: UX Research and Design. – 2021. – Т. 12779. – С. 292-311. – DOI 10.1007/978-3-030-78221-4\_20.
  21. Хайрутдинов, К. М. Анализ влияния оптимизации загрузки веб-страниц на пользовательский опыт / К. М. Хайрутдинов // Проблемы и перспективы развития АПК региона: Материалы Межвузовской студенческой научно-практической конференции, Пермь, 06 декабря 2022 года – 06 2023 года. – Пермь: ИПЦ «Прокрость», 2023. – С. 427-430.
  22. Arika O. How web browsers work - navigation (part 1, with illustrations). URL: <https://dev.to/arikaturika/how-web-browsers-work-part-1-with-illustrations-1nid> (дата обращения: 04.02.2024).
  23. Павел Н. Влияние HTTPS на ранжирование в поисковых системах. URL: <https://www.demis.ru/articles/vliyanie-https-na-ranzhirovanie-saita/> (дата обращения: 04.02.2024).
  24. Koray. T. How does a browser create a web page? URL: <https://www.oncrawl.com/technical-seo/how-does-a-browser-create-a-web-page/> (дата обращения: 04.02.2024).
  25. Populating the page: how browsers work. URL: [https://developer.mozilla.org/en-US/docs/Web/Performance/How\\_browsers\\_work](https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work) (дата обращения: 04.02.2024).
  26. Overview. URL: <https://developer.chrome.com/docs/lighthouse/overview> (дата обращения: 04.02.2024).

**Жуков Станислав Владимирович.** Тамбовский государственный университет им. Г.Р. Державина, Тамбов, Россия. Аспирант. Область научных интересов: информационные технологии, математическое и компьютерное моделирование динамических систем. E-mail: [i@coder-stas.ru](mailto:i@coder-stas.ru)

**Ковалева Ольга Александровна.** Тамбовский государственный университет им. Г.Р. Державина, Тамбов, Россия. Профессор кафедры математического моделирования и информационных технологий, доктор технических наук, доцент. Область научных интересов: информационные технологии, математическое и компьютерное моделирование динамических систем. E-mail: [solomina-oa@yandex.ru](mailto:solomina-oa@yandex.ru)

**Ковалев Сергей Владимирович.** Тамбовский государственный университет им. Г.Р. Державина, Тамбов, Россия. Профессор кафедры математического моделирования и информационных технологий, доктор технических наук, доцент. Область научных интересов: информационные технологии, математическое и компьютерное моделирование динамических систем. E-mail: sseedd@mail.ru

## Conceptual Model of Web Page Loading

S. V. Zhukov, O. A. Kovaleva, S. V. Kovalev

Derzhavin Tambov State University, Tambov, Russia

**Abstract.** The article discusses the conceptual model of web page loading developed by the authors. The model consists of three entities: Backend (characterized by the metrics: Page creation time, Number of SQL queries, SQL query execution time, Cache size, Page size, Number of errors) environment (characterized by the metrics: DNS, TLS, Connection, Load speed), frontend (characterized by metrics: FCP, LCP, TBT, CLS, SI). The presented conceptual model proposes to consider the process of loading a page, including all stages of its formation from the user requesting a page to rendering the received content. Its use in the process of optimizing the loading speed of a web page gives a new look at possible problems of a web project. The paper presents an example of using this model to solve the current problem of optimizing page loading speed.

**Keywords:** conceptual model, backend, environment, frontend, web page.

DOI 10.14357/20718632240203

EDN LWCJNC

## References

1. Özkan B., Özceylan E., Kabak M., Dağdeviren M. Evaluating the websites of academic departments through SEO criteria: a hesitant fuzzy linguistic MCDM approach. *Artificial Intelligence Review*. 2020; 53:875-905. doi:10.1007/s10462-019-09681-z
2. Zherebtsova K.O. SEO website promotion: specifics, mechanics, features. *EKONOMIKA v TEORII i na PRAKTIKE: AKTUALNIE VOPROSI i sovremennie ASPEKTI*. 2021; 9:33-38. (In Russ).
3. Elikhanov V.G., Guzueva E.R., Mutsurova Z.M. Mobile site positioning. *Ekonomika: vchera, segodnya, zavtra*. 2020; 10(9-1):567-572. (In Russ). doi:10.34670/AR.2020.79.17.066
4. Polukhin N.V., Ekkert N.V., Kozlov V.V. Information preferences of patients in the context of communication on the websites of medical organizations. *Siberian Journal of Life Sciences and Agriculture*. 2021; 13(5):226-246. (In Russ). doi:10.12731/2658-6649-2021-13-5-226-246
5. Biryukov M., Yakubova O. Review of methods to increase web page loading speed. *Aktualnie problemi infotelekomunikatsii v nauke i obrazovanii (APINO 2022)*. 2022; 1:160-165. (In Russ).
6. Goryachkin B.S., Khanmurzin T.I. Increasing the efficiency of working with a web resource using system programmer tools. *Dinamika slozhnykh sistem - XXI vek*. 2022; 16(3):26-39. (In Russ). doi:10.18127/j19997493-202203-03
7. Voronova A.M. Technology to increase website loading speed. *Nauka XXI veka: Vizovi, stanovlenie, razvitie*. 2023; 8:267-272. (In Russ).
8. Vogel L., Springer T. User Acceptance of Modified Web Page Loading Based on Progressive Streaming. *Web Engineering*. 2022; 13362:391-405. doi:10.1007/978-3-031-09917-5\_27
9. Vogel L., Springer T. Speed Up the Web with Universal CSS Rendering. *Web Engineering*. 2023; 13893:191-205. doi:10.1007/978-3-031-34444-2\_14
10. Vogel L., Springer T. An In-Depth Analysis of Web Page Structure and Efficiency with Focus on Optimization Potential for Initial Page Load. *Web Engineering*. 2022; 13362:101-116. doi:10.1007/978-3-031-09917-5\_7
11. Kloos J., Majumdar R., McCabe F. Deferrability Analysis for JavaScript. *Hardware and Software: Verification and Testing*. 2017; 10629:(35-50). doi:10.1007/978-3-319-70389-3\_3
12. Toropkin R.A., Zinovev YA.V., Rasskazov N.S., Mitrokhin M.A. Technologies for optimizing website performance using the example of the analytical system of publication activity of Penza State University. *Modeli, sistemi, seti v ekonomike, tekhnike, prirode i obshchestve*. 2020; 4(36): 71-78. (In Russ). doi:10.21685/2227-8486-2020-4-7
13. Zarifis K., Holland M., Jain M., Katz-Bassett E., Govindan R. Modeling HTTP/2 Speed from HTTP/1 Traces. *Passive and Active Measurement*. 2016; 9631:233-247. doi:10.1007/978-3-319-30505-9\_18

14. Priya M.D., Sabarinathan B., Surya M. Optimization of Web Applications Using Eager Loading. *Proceedings of 6th International Conference on Recent Trends in Computing*. 2021; 177:711-722. doi:10.1007/978-981-33-4501-0\_66
15. Shkonda D.N., Shchelkunova M.E. Testing the performance of Internet pages. *Nauka, innovatsii i tekhnologii: ot idei k vnedreniyu*. 2022; 75-77 (In Russ).
16. Slobodyanyuk A.S., Shchelkunova M.E. Carrying out load testing of the VKontakte website. *Vestnik nauchnogo obshchestva studentov, aspirantov i molodikh uchenikh*. 2023; 1:93-97. (In Russ).
17. Buinevich M.V., Maksimov A.V., Vostrykh A.V. Analysis of the results of an audit of network information resources of the Russian Ministry of Emergency Situations. 2020; 1:101-110. (In Russ).
18. Solozobov, O.A. The state and problems of technical optimization of Russian bookmaker websites for mobile search. *Vestnik nauki i obrazovaniya*. 2019; 4-2(58):37-42. (In Russ).
19. Tolstikh, M.A. Using the Lighthouse tool to analyze the quality of a web application. *KONTSEPTyII, TEORIYa i METODIKA FUNDAMENTALNIKh i PRIKLADNIKh nauchnikh issledovaniy*. 2021; 67-69. (In Russ).
20. Namoun A., Alrehaili A., Tufail A. A Review of Automated Website Usability Evaluation Tools: Research Issues and Challenges. *Design, User Experience, and Usability: UX Research and Design*. 2021; 12779:292-311. doi:10.1007/978-3-030-78221-4\_20
21. Khairutdinov, K.M. Analysis of the impact of web page loading optimization on user experience. *Problemi i perspektivi razvitiya APK regiona*. 2023; 427-430. (In Russ).
22. Arika O. How web browsers work - navigation (part 1, with illustrations). Available from: <https://dev.to/arikaturika/how-web-browsers-work-part-1-with-illustrations-1nid> [Accessed February 04, 2024].
23. Pavel N. The impact of HTTPS on search engine rankings. Available from: <https://www.demis.ru/articles/vliyanie-https-na-ranzhirovanie-saita/> [Accessed February 04, 2024].
24. Koray. T. How does a browser create a web page? Available from: <https://www.oncrawl.com/technical-seo/how-does-a-browser-create-a-web-page/> [Accessed February 04, 2024].
25. Populating the page: how browsers work. Available from: [https://developer.mozilla.org/en-US/docs/Web/Performance/How\\_browsers\\_work](https://developer.mozilla.org/en-US/docs/Web/Performance/How_browsers_work) [Accessed February 04, 2024].
26. Overview. Available from: <https://developer.chrome.com/docs/lighthouse/overview> [Accessed February 04, 2024].

**Zhukov Stanislav V.** Postgraduate student, Derzhavin Tambov State University, 33, Internatsionalnaya St., 392000, Tambov, Russia. E-mail: [i@coder-stas.ru](mailto:i@coder-stas.ru)

**Kovaleva Olga A.** Dr. Sc., Tech., Professor, Derzhavin Tambov State University, 33, Internatsionalnaya St., 392000, Tambov, Russia. E-mail: [solomina-oa@yandex.ru](mailto:solomina-oa@yandex.ru)

**Kovalev Sergei V.** Dr. Sc., Tech., Professor, Derzhavin Tambov State University, 33, Internatsionalnaya St., 392000, Tambov, Russia. E-mail: [sseedd@mail.ru](mailto:sseedd@mail.ru)