

Архитектор информационных систем

Николай Мацевский
Евгений Степанищев
Глеб Кондратенко

Реактивные веб-сайты

Клиентская оптимизация в алгоритмах и примерах

Учебное пособие



Интернет-Университет
Информационных Технологий
www.intuit.ru



БИНОМ.
Лаборатория знаний
www.lbz.ru

Москва, 2010

ОГЛАВЛЕНИЕ

5	ВВЕДЕНИЕ
7	Об авторах
7	Благодарности
6	Как ускорить работу сайта для посетителя
11	Глава 1. ОБЗОР МЕТОДОВ КЛИЕНТСКОЙ ОПТИМИЗАЦИИ
11	1.1. Клиентская оптимизация
12	1.2. Анализ веб-страниц
20	1.3. Уменьшение размера загружаемых объектов
28	1.4. Уменьшение количества запросов
40	1.5. Увеличение скорости отображения веб-страниц
43	1.6. Оптимизация структуры веб-страниц
49	Глава 2. АЛГОРИТМИЗАЦИЯ СЖАТИЯ ТЕКСТОВЫХ ФАЙЛОВ
49	2.1. Методы сжатия, поддерживаемые браузерами
52	2.2. Проблемы в браузерах, прокси-серверах и firewall
55	2.3. Настройка веб-серверов Apache, nginx и lighttpd
65	2.4. Собственная реализация сжатия со стороны сервера
71	2.5. Альтернативные методы сжатия
75	Глава 3. АЛГОРИТМЫ УМЕНЬШЕНИЯ ИЗОБРАЖЕНИЙ
76	3.1. Уменьшаем GIF (Graphics Interchange Format)
82	3.2. Оптимизируем JPEG (Joint Photographic Experts Group)
92	3.3. Оптимизируем PNG (Portable Network Graphics)
107	3.4. Оптимизируем SVG (Scalable Vector Graphics)
114	3.5. Средства онлайн-оптимизации
119	Глава 4. УМЕНЬШЕНИЕ КОЛИЧЕСТВА ЗАПРОСОВ
119	4.1. Автоматическое объединение текстовых файлов
127	4.2. Алгоритм разбора и сбора CSS Sprites
135	4.3. CSS Sprites и data:URI, или Microsoft и весь остальной мир
138	4.4. Автоматизация кроссбраузерного решения для data:URI
142	4.5. Автоматизация кэширования

156	Глава 5. ОПТИМИЗАЦИЯ СТРУКТУРЫ ВЕБ-СТРАНИЦ
156	5.1. Динамические стили: быстро и просто
160	5.2. Оптимизация CSS-структуры
167	5.3. Пишем эффективный CSS
171	5.4. Стыкуем асинхронные скрипты
177	5.5. Стыкуем компоненты в JavaScript
181	5.6. Что такое CDN и с чем его едят
185	5.7. Практическое использование CDN на примере Google Apps
192	Глава 6. ТЕХНОЛОГИИ БУДУЩЕГО
193	6.1. Профилируем JavaScript
197	6.2. Проблемы при оценке производительности браузеров
208	6.3. Перспективы «быстрого» JavaScript
214	6.4. Реализация логики CSS3-селекторов
220	6.5. API для CSS-селекторов в браузерах
224	6.6. Canvas: один шаг назад, два шага вперед
234	6.7. Вычисляем при помощи Web Workers
239	6.8. Клиентские хранилища
251	Глава 7. АВТОМАТИЗАЦИЯ КЛИЕНТСКОЙ ОПТИМИЗАЦИИ
251	7.1. Обзор технологий
256	7.2. Установка Web Optimizer
263	7.3. Настройка Web Optimizer
269	7.4. Примеры использования Web Optimizer
272	7.5. Решаем проблемы с установкой Web Optimizer
279	Глава 8. ПРАКТИЧЕСКОЕ ПРИЛОЖЕНИЕ
279	8.1. Разгоняем ASP .NET: 100 баллов и оценка «А» в YSlow
285	8.2. Разгоняем Drupal
300	8.3. Разгоняем Wordpress
304	8.4. Разгоняем Joomla! 1.5
308	8.5. Разгоняем Joostina
315	8.6. Пара советов для Ruby on Rails
318	8.7. Разгоняем jQuery
328	8.8. Клиентская оптимизация для произвольного сайта
335	ЗАКЛЮЧЕНИЕ
335	В качестве послесловия

Введение

Дорогой читатель, спасибо тебе за неустанный интерес к клиентской оптимизации, имеющей такое большое значение при оценке качества веб-проекта, для продвижения сайта и увеличения доходности интернет-подразделения любой компании.

Первая книга о клиентской оптимизации, «Разгони свой сайт» (<http://speedupyourwebsite.ru/>), вызвала многочисленные (все положительные) отклики, и это вдохновило нас на написание продолжения. К слову, мы рекомендуем прочесть первую книгу тем, кто еще этого не сделал. Это поможет глубже понять материал, изложенный в данной книге, легче вникнуть в суть излагаемых здесь концепций.

Книга, которую ты сейчас держишь в руках, призвана пролить свет на те стороны клиентской оптимизации, которые остались в тени после выхода первой книги. Мы надеемся, что она будет интересна как клиентским оптимизаторам со стажем, так и начинающим специалистам, желающим расширить свой кругозор.

Книга «Реактивные веб-сайты» содержит большое количество теоретического материала о клиентской оптимизации, но акцент в значительной мере сделан на описании прикладных методов оптимизации, а также на их квинтэссенции — автоматизации. Большая часть материала этой книги посвящена именно внедрению клиентской (и частично серверной) оптимизации при разработке веб-сайтов, а почти вся седьмая глава — приложению для автоматического ускорения сайтов, Web Optimizer (<http://www.web-optimizer.ru/>).

Кроме того, в этой книге мы немного заглянули в будущее и постарались описать те аспекты производительности, которые будут актуальны буквально через год-другой, когда большинство браузеров станут настолько быстрыми, что обычные методы оптимизации потеряют свою первоначальную эффективность. О перспективах оптимизации рассказывается в шестой главе.

Наконец, практическое приложение (оно приведено в восьмой главе) получилось достаточно объемным и охватывает множество текущих систем, применяемых для разработки сайтов малой и средней сложности. Это позволит использовать данное издание в полной мере как справочник с пошаговым руководством к действию: что и как нужно сделать, чтобы сайт работал «быстрее молнии».

На этих страницах опубликован первоклассный материал от профессиональных специалистов по ускорению сайтов и прикладным техникам оптимизации. Это знаменательно, что данное издание наконец-то увидело свет и ты держишь его в своих руках, читатель!

Об авторах

Данная книга не смогла бы охватить всех заявленных тем, если бы не авторский коллектив, вложивший все лучшее в это издание. Каждый из авторов внес существенный вклад в создание финальной версии этой книги.

Николай Мацевский, основатель и генеральный директор инновационной компании «ВЕБО», главной целью которой является разработка решений для оптимизации производительности веб-сайтов. Николай является одним из лучших специалистов в области клиентской и серверной производительности. Именно благодаря ему увидели свет сотни ценных русскоязычных статей, посвященных производительности веб-сайтов, первая книга по клиентской оптимизации «Разгони свой сайт», а также приложение для автоматического ускорения сайтов Web Optimizer.

Евгений Степанищев, сотрудник компании Яндекс подготовил две потрясающие главы на тему прикладного сжатия текстовой информации и оптимизации изображений, а также раздел шестой главы, посвященный клиентским хранилищам. Его познания в области графических спецификаций оказались настолько велики, что позволили предсказать несколько оптимизационных техник задолго до их появления.

Глеб Кондратенко, сотрудник компании Acronis, подготовил главу с обзором методов клиентской оптимизации, — прекрасное подспорье для тех, кто еще не знаком с основами данного технологического направления. Помимо этого Глеб свел воедино структуру книги, определил ее формирование. Этот поистине титанический труд позволил выпустить весь нижеизложенный материал в слаженной концепции.

Благодарности

В первую очередь авторы выражают свою благодарность Сергею Чикуненку за консультирование и часть материалов по оптимизации изображений (особенно PNG), Игорю Сысоеву за консультирование по pngix и его материалы по проблемам поддержки сжатого контента в браузерах и прокси-серверах, Виталию Харисову за неоценимую помощь в прояснении вопросов быстродействия отрисовки страниц в различных браузерах и применения CSS-логики.

Также, благодаря тесному сотрудничеству с Русланом Синицким (aka sirus, <http://fullajax.ru/#:developers>) удалось создать уникальный инструмент для автоматического создания кроссбраузерных изображений

в формате `data:URI` (подробнее об этом рассказывается в четвертой главе). Ольга Абанова (<http://www.getincss.ru/>) поделилась ценным материалом о конкурентной загрузке стилей, а Денис Абушаев (<http://mearion.blogspot.com/>) — своими советами касательно ускорения загрузки приложений на Ruby on Rails.

Отдельно хочется поблагодарить следующих специалистов за консультирование и предоставление материалов для разделов книги: «Разгоняем Drupal» — Елену Цапину, «Разгоняем Joomla!» — Николая Кирша, и «Разгоняем jQuery» — Олега Смирнова. Они самостоятельно собрали весь материал и позволили читателям получить прикладные советы по оптимизации данных систем и библиотек «из первых рук».

Это просто замечательно, что коллектив высокопрофессиональных авторов смог предоставить действительно интересный материал, который будет полезен широкому кругу читателей, — и донести его в максимально доступном формате.

Как ускорить работу сайта для посетителя

С каждым годом Интернет растет вширь и вглубь. Увеличивается пропускная способность каналов, пользователи переходят с коммутируемого доступа на безлимитный. Сайты становятся больше по размеру, больше по наполнению и сложнее во взаимодействии. Размеры загружаемых файлов при этом увеличиваются многократно, а время ожидания пользователей не уменьшается.



За последние 5 лет средний размер веб-страниц вырос втрое (по данным исследования Akamai), а за последний год — в полтора раза (по данным [webo.in](http://www.webo.in)). При этом каждая страница использует в среднем по 60 объектов, что крайне негативно сказывается на общем времени загрузки. Только порядка 5–10% от общего времени загрузки приходится на серверную часть. Все остальное составляет именно клиентская архитектура.

Что обычно видит пользователь, заходя на ваш сайт? И как долго он это видит? 75% посетителей уйдут после 10 секунд. При этом наиболее характерным временем ожидания будет 4 секунды: если за это время сайт загружается у 90% пользователей, то вы счастливый владелец быстрого интернет-ресурса.

Однако и здесь дорога каждая миллисекунда. Недаром высоконагруженные проекты типа Google, Amazon, Flickr, Netflix, Яндекс, ВКонтакте и Одноклассники так серьезно подходят к вопросу скорости загрузки сайтов. За каждым потерянным моментом времени кроется определенная сумма денег. Это именно то место, где время тождественно деньгам.

В чем проблема?

Основное время при загрузке страницы уходит именно на клиентскую часть. Серверные затраты обычно крайне малы и составляют от 50 до 500 мс. Среднему пользователю на самом деле абсолютно все равно, сколько страница будет создаваться на сервере, если он увидит ее через полсекунды. В этом случае фокус смещается именно на клиентскую, а не серверную оптимизацию.

Характер проблем варьируется от сайта к сайту. Иногда он заключается в особенности интернет-подключения основной массы пользователей ресурса (например, если широко используются модемы или GPRS). Иногда — в сложности самого сайта и неоправданном использовании ресурсов сети. Иногда — в неграмотном использовании клиентских технологий и большого количества разнородных решений. Но все эти проблемы можно решить.

Ключевые моменты оценки качества веб-проектов

Говоря о скорости загрузки, нельзя не обозначить ее роль в оценке технологического качества любого Интернет-проекта. При этом стоит обратить внимание и на следующие моменты (которые можно достаточно быстро проверить с помощью бесплатных инструментов).

- Совместимость сайта с браузерами.
- Соответствие стандартам.
- Семантика HTML-кода.
- Доступность сайта для пользователей.
- Скорость работы на стороне сервера.
- Скорость работы на стороне браузера.

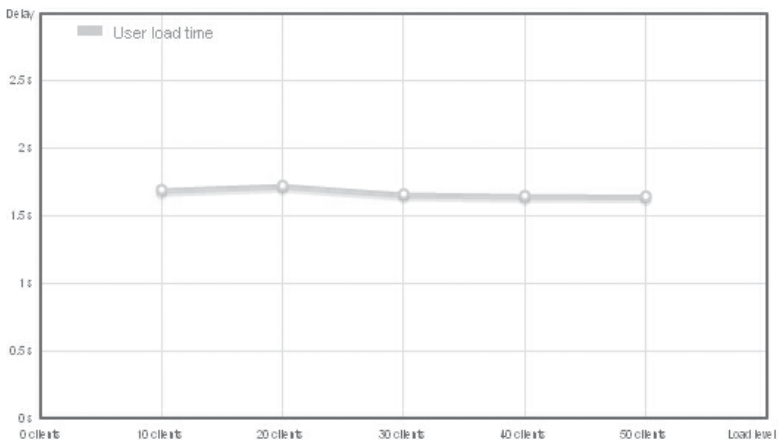
Скорость работы сайта

Скорость работы оказывает ключевое влияние на конвертацию посетителей в доход сайта. Особенно это характерно для интернет-магазинов. Как мы можем проверить эту скорость? Для серверной стороны существует инструмент host-tracker.com, с помощью которого можно

установить характерное время доступа к сайту из различных точек земного шара. Также он позволяет оценить характерное время создания страницы на сервере (если минимальное время доступа к сайту больше 1с, то уже стоит бить тревогу). Время создания страницы на сервере можно оценить и с помощью webo.in/my/action/timings/ (необходима регистрация).

Для проверки отказоустойчивости сайта стоит воспользоваться сервисом Load Impact (<http://www.loadimpact.com/>). Бесплатный анализ позволяет проверить до 50 одновременных посещений (каждое из которых может открывать несколько десятков страниц на сайте). Основным показателем устойчивости сайта к нагрузкам будет вид кривой, которая получится после проведения теста. Если график ровный или время открытия страницы несколько падает при росте посещаемости — значит, с сайтом все нормально. Если нет, то нужно принимать решение об оптимальном потоке посетителей, которых сайт сможет обслужить.

Оценка отказоустойчивости сайта. Источник: www.loadimpact.com



Для проверки качества скорости загрузки клиентской составляющей можно воспользоваться либо YSlow под Firebug для Firefox (оценка должна быть не менее 80, лучше всего не менее 90), либо бесплатным инструментом webo.in (простая оценка не ниже 70, лучше не ниже 80). Подробнее об инструментах для анализа клиентской производительности веб-сайтов рассказано в первой главе.

На что влияет качество сайта

В первую очередь, на стоимость его поддержки и развития. Сайт, сделанный грамотно и по всем стандартам, гораздо проще развивать, поскольку времени на написание дополнительного кода при наличии хорошей, стандартной базы уходит значительно меньше. Это и масштабируемость проекта, когда разработчики предусматривают большой «запас прочности» для ежедневных посещений, а в пиковые часы сайт работает в штатном режиме, не требуя дополнительного масштабирования.

Скорость создания HTML-страниц и общее время загрузки очень сильно влияет на доступность сайта. Ключевым параметром здесь является «загрузка за 4 секунды» и более быстрая загрузка, чем у основных конкурентов. В отсутствие основных преимуществ на рынках с высокой конкуренцией скорость работы сайта может оказаться решающим фактором при принятии решения со стороны конечного пользователя.

Для решения проблем, связанных с медленной скоростью загрузки сайта, прежде всего нужно начинать с профессионального подхода при разработке какого-либо веб-приложения. Он обязательно должен опираться на использование веб-стандартов, в том числе и в области скорости загрузки и взаимодействия с пользователем.

Для полноценной оптимизации крупные компании разрабатывают специализированные решения. Однако даже в этом случае они опираются на многочисленный свод правил, описывающих наиболее оптимальное взаимодействие браузера с пользователем. Для среднего размера сайтов, использующих какое-либо стандартное решение для обслуживания контента, стоит обратить свое внимание на решения по автоматизации клиентской оптимизации. Подробнее они описаны в седьмой главе.

Наконец, при разработке сайтов стоит руководствоваться принципом «мягкой деградации», согласно которому нужно наращивать функциональность в зависимости от способности браузера, а не наоборот. Например, можно использовать самые передовые технологии при верстке, которые поддерживаются не очень большим числом браузеров, но должны будут поддерживаться значительной их долей, скажем, через год. Тогда определенные решения будут выглядеть безупречно у небольшого числа пользователей, у остальных — хорошо или приемлемо. Но с прогрессом Интернета число последних будет стремительно уменьшаться.

Также стоит упомянуть про возможность использования распределенной сети серверов для сведения времени ответа к минимуму. В этом могут помочь существующие системы распределенных вычислений и обработки запросов, такие как Amazon S3, Google Apps, Microsoft Azure. Также аналогичная система есть и на российских просторах. Это первая в России CDN — NGENIX (<http://ngenix.net/>). Подробнее о CDN рассказывается в пятой главе.

Глава 1. Обзор методов клиентской оптимизации

1.1. Клиентская оптимизация

Клиентская оптимизация — это оптимизация процесса загрузки клиентским приложением содержимого веб-страниц. Основная цель такой оптимизации — достижение максимальной скорости загрузки страниц сайта браузером клиента, ведь даже незначительные изменения времени загрузки могут иметь серьезные последствия для задачи, возложенной на сайт.

При построении высокопроизводительных сайтов должен присутствовать и клиентский, и серверный подход, они во многом дополняют друг друга. Главное отличие клиентского подхода состоит в том, что в качестве объекта оптимизации рассматриваются страницы сайта, получаемые браузером клиента, состоящие из HTML-документа, содержащего вызовы внешних объектов, а также сами внешние объекты (чаще всего это файлы CSS, файлы JavaScript и изображения).

Может показаться, что клиентская оптимизация является лишь составляющей частью серверной оптимизации, однако это не так. Различные технологические решения клиентской области сайта при одинаковой нагрузке на сервер могут обеспечивать совершенно разные характеристики клиентского быстродействия.

При исключении из рассмотрения всех факторов, относящихся к серверному программному обеспечению и каналу передачи данных, можно заключить, что увеличение скорости загрузки страницы на различных стадиях загрузки принципиально возможно за счет ограниченного количества методов. Об этих методах и пойдет речь далее.

1.2. Анализ веб-страниц

Большинство приведенных в книге методов оптимизации являются универсальными и могут быть применены практически в любом случае, на любом сайте. Но только выбор наиболее подходящего плана оптимизации может привести к наилучшему результату при решении каждой конкретной задачи.

Перед оптимизацией сайта необходим тщательный анализ его клиентской производительности, а также четко сформулированная цель оптимизации, ведь в подобном усовершенствовании важен только результат, а не процесс.

Процедуру анализа веб-сайта можно разделить на несколько основных стадий: анализ веб-страниц и их компонентов, анализ стадий загрузки веб-страниц и анализ характеристик браузеров, при помощи которых веб-страницы обычно загружаются.

1.2.1. Определение цели оптимизации

Целью клиентской оптимизации может быть решение подобных задач:

- достижение минимально возможного времени загрузки какой-либо конкретной страницы;
- достижение минимально возможного времени загрузки группы страниц, просматриваемых в произвольном порядке;
- обеспечение минимально возможного времени с момента запроса страницы до момента появления у пользователя возможности просматривать страницу и взаимодействовать с ней.

Это далеко не полный перечень возможных целей. Иногда и вовсе требуется достигать компромисса и выбирать между несколькими взаимоисключающими вариантами оптимизации. В таких ситуациях лучше иметь максимум возможной информации о ваших веб-сайтах и их посетителях.

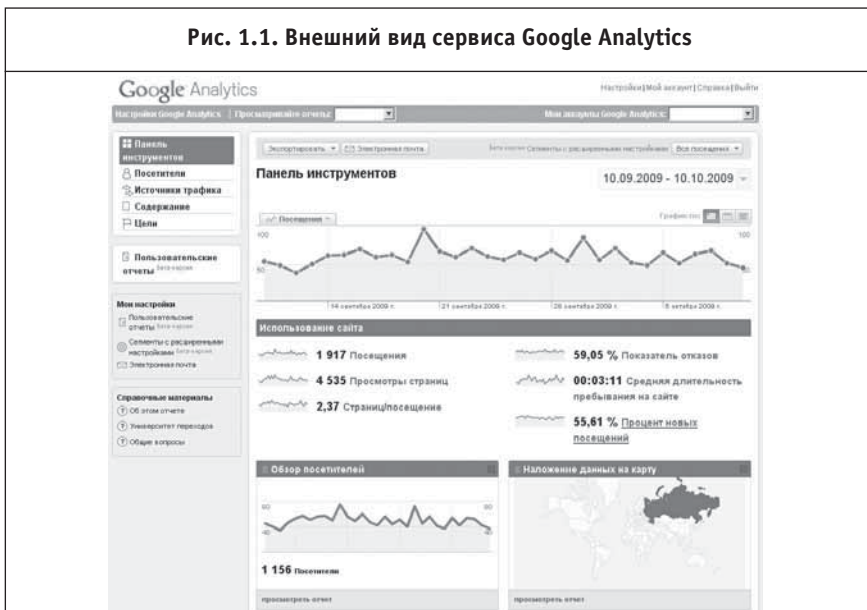


Определить список «критических» страниц, на которых необходим максимальный эффект оптимизации, можно при помощи систем сбора и анализа статистики. Необходимо также учитывать назначение и специфику оптимизируемого сайта или сервиса.

Как правило, оптимизация требуется на главной странице сайта и других страницах с высокой посещаемостью, но это не всегда так. В качестве примера можно привести страницы оформления заказа на коммерческом сайте. На них может приходиться лишь 5% от общего числа посетителей сайта, однако если они будут загружаться слишком медленно, посетители могут так и не стать клиентами.

Google Analytics (<http://www.google.com/analytics/>)

Рис. 1.1. Внешний вид сервиса Google Analytics



Google Analytics — один из лучших среди бесплатных сервисов для сбора и анализа статистики. С его помощью можно узнать о посетителях сайта почти все: страницы, с которых они переходили на сайт, время и длительность посещений, наиболее посещаемые страницы и последовательности посещений, параметры программного и аппаратного обеспечения и т. п.

Сервис работает по тому же принципу, что и большинство Интернет-счетчиков: специальный код устанавливается на всех страницах сайта и регистрирует каждое посещение, собирая все данные о нем.

Яндекс.Метрика (<http://metrika.yandex.ru/>)

Относительно молодой, но активно развивающийся русскоязычный сервис для оценки посещаемости сайтов и анализа поведения пользователей на нем. Позволяет получить детальную информацию об источниках перехода на сайт, числе возвратов, просматриваемом содержимом, географии и демографии посещений, программных и аппаратных характеристиках компьютеров пользователей.

Для сбора всей упомянутой выше информации достаточно лишь установить определенный код на всех страницах анализируемого сайта.

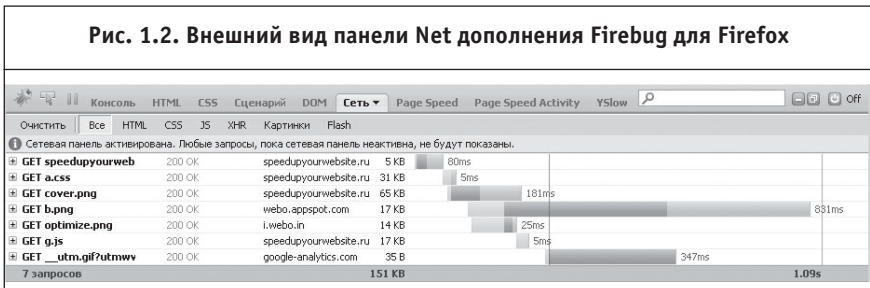
1.2.2. Анализ заголовков, компонентов и стадий загрузки страницы

Firebug (<http://getfirebug.com/>)

Одним из наиболее популярных среди веб-разработчиков средств для анализа и разработки веб-страниц является дополнение Firebug для браузера Firefox.

Панель Net в дополнении Firebug позволяет получить весьма детальную диаграмму загрузки страницы. По диаграмме можно определить стадии загрузки страницы, понять порядок загрузки объектов и выяснить, какие объекты блокируют, замедляют загрузку страницы. Кроме того, на этой панели можно получить детальную информацию о размере и заголовках самого документа, а также всех загруженных внешних объектов.

Рис. 1.2. Внешний вид панели Net дополнения Firebug для Firefox



Firebug также позволяет изменять DOM-дерево и CSS-свойства страницы без ее перезагрузки, сразу отражая результат изменений на странице, а также предоставляет обширные возможности для отладки и профилирования кода JavaScript. Все эти возможности являются прекрасным подспорьем во время работ по оптимизации сайта.

Стоит заметить, что аналогичные Firebug инструменты существуют во всех широко распространенных браузерах. В браузере Safari схожей

[. . .]