

ИССЛЕДОВАНИЕ МЕТОДОВ ОПТИМИЗАЦИИ WEB-САЙТОВ НА БАЗЕ CMS DRUPAL

Аннотация. Предложены методы ускорения работы сайта на базе CMS Drupal. Используются различные уровни кэширования, настройка специализированных модулей, оптимизация баз данных и программного кода.

Ключевые слова: сайт, база данных, оптимизация, кэширование, модуль, производительность, ядро, CMS, Drupal, PHP, MySQL.

Введение. При современном развитии интернета вопрос ускорения работы сайтов является очень актуальным. Большое количество сайтов создается с помощью систем управления содержимым (CMS), которые позволяют уменьшить время разработки сайта и упрощают его сопровождение. Однако недостатком таких сайтов является невысокое их быстродействие, так как система накапливает много ресурсов и кода, что плохо влияет на скорость доступа к сайту.

Поэтому, стоит задача исследовать средства и методы повышения быстродействия работы сайтов, как общие, так и специфические для CMS Drupal – одной из самых распространенных систем управления содержимым.

Постановка задачи. Необходимо исследовать методы ускорения работы сайта на базе CMS Drupal, определить преимущества и недостатки каждого метода и комбинаций методов.

Основная часть. Drupal представляет собой многомодульную платформу с открытым кодом, предназначенную для совместного управления содержимым сайта многими пользователями. Интегрированная среда и шаблонная система тем оформления полностью написаны на PHP.

Важное средство оптимизации сайта – это оптимизация кода PHP. Ее можно разделить на несколько методов.

1. Применение кэширования. В целом ряде случаев кэширования просто незаменимо. Суть кэширования сводится к созданию кэша

из наиболее часто используемых и наиболее редко изменяемых данных.

2. Оптимальное применение баз данных (БД). Следует помнить, что каждое подключение к БД, каждый запрос, каждая лишняя таблица в запросе и т.д. – все это создает ощутимую нагрузку.

3. Буферизация, сжатие. В процессе выполнения все данные, которые выводятся пользователю, тут же ему и отправляются. Выполнения кода приостанавливается, пока данные не будут забраны. Если в процессе работы скрипт отправляет готовый HTML-код через функцию echo, например, или любым другим методом, то скрипт передает данные по частям, каждый раз приостанавливая выполнение. Лучшим методом является вывод данных в буфер с дальнейшей их передачей в сжатом виде.

4. Использование оптимизирующих модулей. Модули ZendOptimizer и eAccelerator главным образом занимаются оптимизацией и динамичным кэшированием откомпилированного PHP кода. Часто позволяют поднять производительность почти в десятки раз. Их применение имеет смысл в первую очередь на достаточно больших проектах, хотя можно оптимизировать и любые другие сайты на PHP.

5. Ограничение использования функций готовых библиотек. Функции всегда требуют некоторых дополнительных ресурсов. Подключение готовых библиотек «съедает» львиную долю производительности.

Однако для сайтов, разработанных на базе системы управления содержимым Drupal, первым шагом является оптимизация сайта с помощью внутренних возможностей системы.

Для этого необходимо соблюдать основные шаги улучшения скорости загрузки сайта для пользователей:

- 1) отключение всех неиспользуемых модулей;
- 2) использование встроенного кэширования Drupal;
- 3) настройка компрессии страниц.

При использовании компрессии, страницы сайта передаются в сжатом виде, что увеличивает нагрузку на сервер, но снижает время загрузки страниц посетителями сайта.

Второй шаг – оптимизация Drupal с помощью модулей.

Модуль `AuthenticatedUserPageCaching` (`Authcache`) позволяет кэшировать страницы, как для анонимных пользователей, так и для авторизованных пользователей более качественно, чем встроенное кэширование `Drupal`.

Модуль `Boost` позволяет кэшировать статические страницы, добавляя значительную производительность и масштабируемость для сайтов, которые получают главным образом анонимный трафик.

Модуль `Speedy` предназначен для ускорения производительности интерфейса на сайте. Он обеспечивает уменьшенную версию основных файлов `JavaScript`, которые еще не минимизированы.

Третий шаг – оптимизация базы данных.

При частом обновлении `CMSDrupal` содержимого таблиц `БД MySQL`, особенно служебных таблиц, последние подвергаются фрагментации, что снижает быстродействие сервера `БД` и скорость выполнения запросов на чтение. Для решения проблемы необходимо периодически дефрагментировать таблицы `MySQL` с помощью команды `OPTIMIZE TABLE`. Для оптимизации таблиц `БД` используется специализированный модуль `Drupaldb_maintenance`.

Четвертый шаг – специфические настройки модулей.

Значительно ускорить работу `Drupal`-сайта при высокой посещаемости, а также уменьшить дисковое пространство можно, уменьшив размер ведения логов и статистики.

Пример реализации. Рассмотренные методы ускорения работы сайта были протестированы на примере интернет-магазина, написанного на базе `CMS Drupal`. Чтобы понять, как сайт будет выдерживать нагрузки и запросы от многих пользователей, воспользовались сервисом тестирования сайтов <https://loadimpact.com/>. Этот сервис имитирует запросы к `URL` большим количеством пользователей одновременно анализирует трафик, время отклика сайта и время загрузки страницы отдельных составных частей трафика, поступающего с сайта.

Перед оптимизацией сайта, было проведено тестирование для определения быстродействия страниц. Получили среднее время загрузки страницы сайта, при постепенном увеличении пользователей онлайн, достигает от 15 до 25 секунд. Настройки тестов во всех случаях тестирования приведены в табл.1.

На первом шаге ускорения работы сайта подключили и настроили оптимизирующие компоненты `CMS Drupal`: встроенные воз-

возможности ядра и модули оптимизации и кэширования. Провели анализ быстродействия страницы сайта после этих действий (табл.2).

Таблица 1

Конфигурация сервиса тестирования

URL	http://ant-shop.com.ua/
Пользователи	SimulatedBrowserUsers
Количество пользователей	50
Время выполнения теста	5 минут
Место, с которого приходят запросы пользователей	Эшбурн, США

Таблица 2

Конфигурация тестирования быстродействия. Шаг 1

Параметр конфигурации	Статус	Комментарии
Пользователь	<i>anonymous</i>	Кэш работает
Встроенный кэш ядра Drupal	<u><i>включен</i></u>	Кэширование блоков и страниц для анонимных пользователей в базе данных
Модуль Boost	<i>выключен</i>	Создание статических страниц для анонимных пользователей
Модуль AuthCache	<u><i>включен</i></u>	Кэширование страниц для зарегистрированных и анонимных пользователей
Модуль FileCache	<u><i>включен</i></u>	Размещение кэшированных страниц в файлах, а не в базе данных
Модуль Speedy	<u><i>включен</i></u>	Уменьшенные и оптимизированные файлы JavaScript
Оптимизация таблиц базы данных MySQL	<i>Не проведена</i>	
Оптимизация CSS и JavaScript ядром Drupal	<u><i>включен</i></u>	Предоставляет файлы стилей и скриптов одним файлом вместо нескольких
Оптимизация кода PHP	<i>Не проведена</i>	

Включение кэширования, как встроенного в ядре системы, так и в специализированных модулях, и оптимизация файлов CSS

иJavaScripty скорили работу сайта в 2-3 раза от начальных значений. Теперь загрузка страницы составила примерно 5 секунд.

На следующем шаге были включены дополнительные модули оптимизации кэширования, оптимизированы таблицы в базе данных и коды PHP нескольких модулей. Настройка и конфигурирование модулей для тестов приведена в табл.3.

Таблица 3

Конфигурация тестирования быстродействия. Шаг 2

Параметр конфигурации	Статус	Комментарии
Пользователь	<i>admin</i>	Кэш работает.
Встроенный кэш ядра Drupal	<u><i>включен</i></u>	Кэширование блоков и страниц для анонимных пользователей в базе данных
Модуль Boost	<u><i>включен</i></u>	Создание статических страниц для анонимных пользователей
Модуль AuthCache	<u><i>включен</i></u>	Кэширование страниц для зарегистрированных и анонимных пользователей
Модуль FileCache	<u><i>включен</i></u>	Размещение кэшированных страниц в файлах, а не в базе данных
Модуль Speedy	<u><i>включен</i></u>	Уменьшенные и оптимизированные файлы Javascript
Оптимизация таблиц базы данных MySQL	<u><i>Проведена</i></u>	
Оптимизация CSS и JavaScript ядром Drupal	<u><i>включен</i></u>	Предоставляет файлы стилей и скриптов одним файлом вместо нескольких
Оптимизация кода PHP	<u><i>Проведена</i></u>	

Результаты более чем удовлетворительные, среднее время загрузки страницы сократилось до 3 секунд (рис. 1).

Оптимизация базы данных, кода PHP, включение статического кэширования файлов с помощью специализированного модуля Boost и других модулей помогли дополнительно оптимизировать сайт.

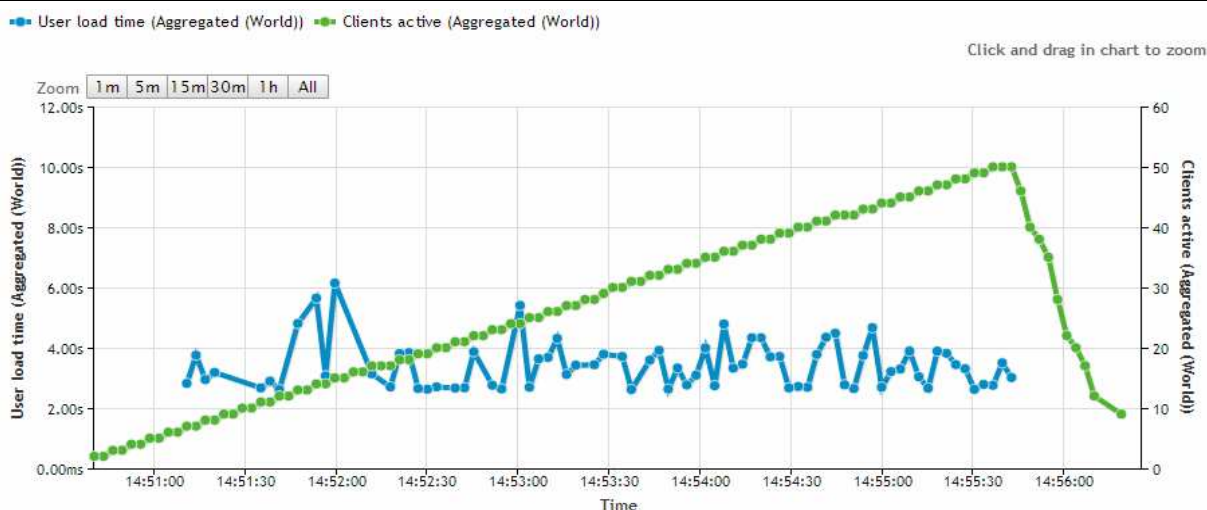


Рисунок 1 – Время загрузки страницы в зависимости от количества посетителей

Выводы. После выполнения оптимизации и внедрения методов повышения быстродействия, установки дополнительных модулей, оптимизации базы данных и PHP кода, получены такие результаты:

- время загрузки страницы в целом уменьшалось с 15-25 секунд до 3 секунд;
- количество запросов к серверу уменьшилось вместе с кэшированием страницы.

В результате проведенного исследования сделаны следующие выводы о методах оптимизации сайтов на базе CMS Drupal и порядке их применения.

В первую очередь должны быть использованы методы внутренних возможностей системы – настройка встроенного кэша и оптимизация ядра. Затем подключаются и настраиваются специализированные модули, проводится оптимизация баз данных. И только потом, в случае недостаточных результатов, - оптимизация с помощью модификации существующего программного кода.

ЛИТЕРАТУРА

1. Тодд Томлинсон. CMS Drupal 7. Руководство по разработке системы управления веб-сайтом. Пер. с англ. – М.: Издательский дом «Вильямс». – 2011. – 560 с.
2. Victor Kane. Leveraging Drupal: getting your site done right. – Wrox, 2009. – 480p.