

ИССЛЕДОВАНИЕ МЕТОДОВ КЭШИРОВАНИЯ CMS MAGENTO

Пантыкин Д.С.¹, Амирагян Л.М.² E mail: Pantykin627@scientifictext.ru

¹Пантыкин Денис Сергеевич – студент,
кафедра математического обеспечения и применения электронно-вычислительных машин;

²Амирагян Лусине Меружаевна – студент,
кафедра инженерной экономики,

Институт управления в экономических, экологических и социальных системах
Южный федеральный университет, г. Таганрог

Аннотация: в статье рассмотрены методы кэширования одной популярной системы управления интернет-магазинами Magento Community Edition доступные для разработчика, влияние их на скорость работы сайта и подробная информация по каждому инструменту. Приведены способы уменьшения времени генерации страниц для конечного пользователя, такие как: объединение JS и CSS файлов, использование плоских таблиц ключевых сущностей в базе данных, объединение исходных файлов системы. Рассмотрены доступные виды кэширования: XML-файлов, переводов, коллекций и настроек интерфейса конечного приложения.

Ключевые слова: web, cms, magento, cache, скорость работы интернет-магазина, интернет-магазин.

RESEARCH METHODS FOR CACHE CMS MAGENTO

Pantykin D.S.¹, Amiraghyan L.M.²

¹Pantykin Denis Sergeevich – student,
DEPARTMENT OF SOFTWARE ENGINEERING;

²Amiraghyan Lusine Meruzhaevna – student,

DEPARTMENT OF ENGINEERING ECONOMICS, INSTITUTE OF MANAGEMENT IN ECONOMIC, ENVIRONMENTAL
AND SOCIAL SYSTEMS OF THE SOUTHERN FEDERAL UNIVERSITY, TAGANROG

Abstract: the article deals with methods of caching a popular online store management system Magento Community Edition available for the developer, their influence on the speed of the site and detailed information on each tool. There are ways of reducing the page generation time for the end user, such as the union JS and CSS files, the use of flat tables of key entities in the database, the union of the original system files. We consider the available types of caching: XML-files, translation, collection and final application interface settings.

Keywords: web, cms, magento, cache, the speed of online shopping, online stores.

УДК 004.031.42

Все большую роль на рынке занимает электронная коммерция. С каждым годом количество людей, которые используют Интернет-ресурсы для покупки различных товаров растет. Но при посещении сайта мы не задумываемся, как все устроено. Для покупателей важны следующие факторы:

1. Наличие и разнообразие товаров необходимой категории;
2. Цена;
3. Акции и скидки;
4. Интуитивно-понятный интерфейс;
5. Скорость работы сайта.

Если первые три фактора – это проблемы конкретной организации-владельца Интернет-магазина, их политики ведения бизнеса и экономической составляющей, то для создания интерфейса и поддержания приемлемой скорости работы сайта можно составить список рекомендации для разработчиков.

При разработке нового сайта продаж, поднимается другой вопрос – ускорение процессов разработки и уменьшение вероятности возникновения ошибок. Для этого на рынке появились различные системы для создания Интернет-магазинов:

1. Системы управления содержимым (CMS), которые поставляются в виде дистрибутива.
2. SaaS-решения, предоставляющие услуги по созданию Интернет-магазинов.

Выбор CMS Magento[1] обусловлен богатым набором базовых функций: многоязыковая система, многоуровневое ценообразование, возможность использования нескольких валют, большое сообщество и, как следствие, разнообразие доступных модулей, расширяющих возможности системы, большое внимание к безопасности системы, множество тематических форумов и готовых решений на возникающие проблемы.

Как известно скорость работы сайта влияет не только на конверсию, но и на ранжирование результатов в выдаче различных поисковых систем[2]. Поэтому разработчики CMS Magento включили в базовую версию системы инструменты, позволяющие уменьшить скорость отклика сайта:

1. Cache;

2. Объединение JS/CSS файлов;
3. Flat catalog;
4. Компиляция.

Cache в CMS Magento подразделяется на несколько видов:

1. Configuration cache;
2. Layouts cache;
3. Block HTML output;
4. Translations;
5. Collections data;
6. EAV types and attributes;
7. Web services cache;

Configuration Cache и Layouts cache.

Все настройки модулей и тем интерфейса содержатся в XML-файлах, редактируя их, мы можем подключать различные файлы для отображения, подключать каскадные таблицы стилей, устанавливать настройки для генерации страниц в панели администратора и т.д. Минимальное количество XML-файлов для каждого модуля – 3. При выключенной опции, все файлы собираются в единый при каждом запросе, что создает большую нагрузку на сервер. Разумеется, на этапе разработки, это оправдано. Но на итоговом проекте мы включаем данный вид кеширования. Кроме того в любой момент мы можем перезагрузить его, например, после добавления нового модуля или темы.

Block HTML output. Кеширует шаблоны, которые используются в теме. По факту, кеширует только стандартный шаблон меню.

Translations. Множественная поддержка языков осуществляется через CSV-файлы, которые содержат строки на исходном языке и перевод на нужный язык. Стоит использовать даже на одноязычном сайте.

Collections data. Позволяет кешировать коллекции, которые используют метод `initCache()`. Это стоит учитывать при разработке модулей для данной системы.

EAV types and attributes. Не используется.

Web services cache. CMS Magento предоставляет возможности API сервиса. Для его настройки необходимо создавать XML-файлы в создаваемом модуле. Множество модулей порождает множество файлов с конфигурациями определения и переопределения API-функций для REST и SOAP подходами, который необходимо объединять в один при каждом запросе.

Объединение JS/CSS файлов. Так как все JS и CSS файлы подключаются в XML-файлах, которые потом анализируются ядром CMS Magento. У системы есть возможность автоматически объединять эти файлы. Кроме этого, хорошей практикой является удаление лишних символов из JS и CSS файлов после этапа разработки или после внесения правок.

Flat Catalog. По умолчанию CMS Magento использует EAV(Entity-Attribute-Value) модель базы данных для некоторых сущностей. Эта такой подход к организации хранения данных, когда сущность к которой относится атрибут, параметры атрибута и его значение разнесены в разные таблицы. В данной системе по такому принципу выстроены следующие сущности: продукты, категории, пользователи и адреса пользователей.

Такое хранение данных является очень гибким – добавление 1 атрибута к сущности это всего лишь новая строка в таблице значений атрибутов. Но такой подход увеличивает количество запросов к базе данных и, как следствие, увеличивает нагрузку на сервер и время генерации страницы. Использование опции “Flat Catalog” позволяет хранить информацию о сущностях в одной таблице, что существенно уменьшит нагрузку на базу данных и уменьшит время отклика сайта.

Компиляция.

Объединяет некоторые файлы ядра в один и помещает их и все остальные исполняемые файлы в одну директорию `includes/src`. Таким образом, существенно уменьшается количество обращений интерпретатора PHP к файловой системе при работе системы.

На диаграмме 1 представлены итоги тестов, выполненных с различными конфигурациями. Все тесты проводились в следующем окружении:

- macOS Sierra, 10.12.1
- 24gb 1600 MHz, Core i5 (I5-4670), 3.4 GHz
- Standard Hard Drive: 1 TB (7200 RPM)
- PHP Version 5.6.25
- Mysql version: 5.7.15
- Apache/2.4.23 (Unix)
- Magento ver. 1.9.2.2
- Кол-во продуктов: 1000
- Кол-во категорий: 50

Тест выполнялся на домашней странице с 30 товарами. Для каждой конфигурации замер выполнялся 5 раз и вычислялось среднее значение. Список тестируемых конфигураций:

1. Без кэша;
2. Cache;
3. Cache + объединение JS/CSS файлов;
4. Cache + объединение JS/CSS файлов + Flat Catalog;
5. Cache + объединение JS/CSS файлов + Flat Catalog + Компиляция;

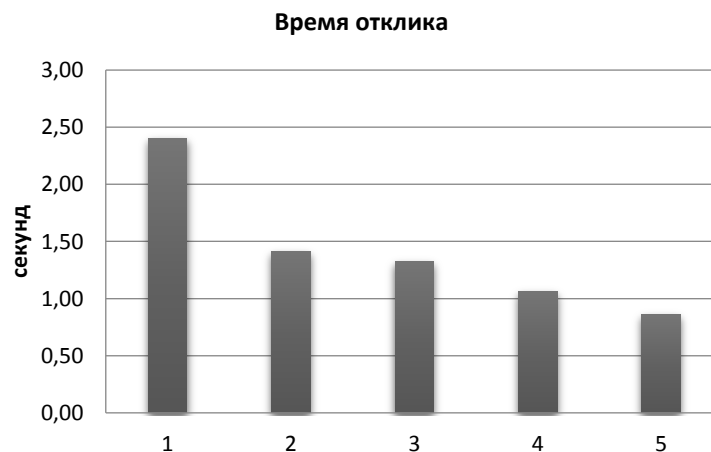


Рис. 1. Результаты теста

Список литературы / References

1. [Электронный ресурс]: eCommerce Software & eCommerce Platform Solutions // CMS Magento. Режим доступа: Magento.com/ (дата обращения: 05.03.2017).
2. Маркетинг → 30+ кейсов: как скорость сайта влияет на конверсию. [Электронный ресурс]: Николай Мациевский, Технический директор. Айри.рф. Режим доступа: <https://habrahabr.ru/post/300210/> (дата обращения: 04.03.2017).
3. Разработка → Увеличение производительности Magento. [Электронный ресурс]: Валентин Климов, Lead Technical Architect. Режим доступа: <https://habrahabr.ru/post/282025> (дата обращения: 05.03.2017).