UDK 004.42:004.453
DOI 10.7251/ZRSNG1708059D
COBISS.RS-ID 7298072

Unapređenje performansi eCommerce sistema zasnovanog na "Oxid eSales" frejmvorku

Performance improvement of eCommerce System based on "Oxid eSales" Framework

Miloš Dobrojević, www.magma.rs

Sažetak **- Oxid eSales je eCommerce frejmvork otvorenog koda izrađen u PHP programskom jeziku i koji koristi MySQL/MariaDB sistem za upravljanje bazom podataka. Platforma je popularna na nemačkom govornom području i okuplja veliki broj kompanija i nezavisnih programera koji nude sopstvene module i šablone (templejte) kojima se menjaju ili dopunjuju kako osnovni set funkcionalnosti, tako i dizajn odnosno korisnički interfejs platforme. Iako je ovakav biznis model opšteprihvaćen u IT industriji, često se dešava da ponuđena rešenja ne ispunjavaju očekivane tehničke standarde, što se između ostalog ispoljava i kroz umanjene radne performanse onlajn prodavnice. Uzrok tome mogu biti neoptimizovani kod i/ili SQL upiti, što iziskuje povećanje procesorskog vremena i utroška radne memorije, a pokušaji naknadne optimizacije su otežani glomaznim sistemom fajlova.**

Ključne riječi **– Onlajn prodavnice; Oxid eSales; Optimizacija softvera; Optimizacija baze podataka; PHP framework**

Abstract – Oxid eSales is an open source eCommerce framework built with PHP programming language and uses MySQL/MariaDB system for database management. This platform is very popular in areas with german speaking population, gathering around significant number of companies, coders and designers who offer their own products such as modules and themes, aimed to improve framework's basic set of functionalities, user interface or general appearance. Although this is a common business model in IT industry, solutions found on the market place may not comply with expected technical standards, resulting in reduced performance of online shop. This may be caused by unoptimized code and/or SQL queries, which in turn invoke increased processor time and memory footprint required for application to work. Subsequent attempts for optimization often prove to be difficult due to massive file system.

Keywords **– eCommerce solutions; Oxid eSales; Software optimization; Database optimization; PHP framework**

I. OXID ESALES PHP FRAMEWORK

Online shopping or eCommerce allows consumers to buy goods or services directly from a seller over the Internet. Consumers browse one or more websites, or alternatively use specialized search engines in search for a desired product. Nowadays, consumers can shop online using a range of different electronic equipment, such as desktop computers, laptops, tablet devices, smartphones and smartTVs.

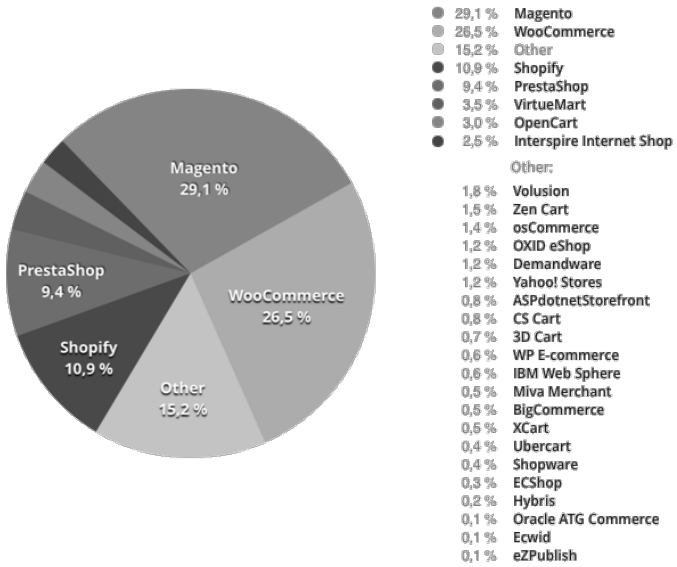

Figure 1 - Shopping Cart Software market share

Although global market is oversaturated with numerous specialized software solutions, a.k.a. shopping cart software, Magento, WooCommerce, Shopify and PrestaShop (the big four) are dominant taking almost 3/4 of global market, Figure 1. [1, 2]

However, there are other eShop software solutions, not as popular on global scale as the big four, but may be dominant in geographical territories, languages or niches.

One of the popular eCommerce solutions within german language speaking population in Europe (Germany, Austria, Switzerland and northern Italy) is Oxid eShop, available on the market as open-source software [8] in Community, Professional and Enterprise editions. [3] It is built in PHP

mdobrojevic@sinergija.edu.ba

programming language, uses MySQL or MariaDB database server and requires Apache web server [4] in order to run.

II. OXID ARCHITECTURE

A. MVC and modular structure

Rather than using an established third-party framework (e.g., Zend Framework), Oxid eSales utilizes its own MVC compliant framework based on modular architecture and basic principles of OOP (*Object Oriented Programming*). It heavily uses getter and setter methods, meaning that object attributes are not directly accessible and must be addressed via get() or set() methods. [5]

View classes are located in the /oxid/views directory and corresponding model classes in the /oxid/core directory. Both class groups can be overloaded as modules, which provides great possibility for code modification.

MVC (*model-view-controller*) is software architectural pattern which divides a software application into three interconnected parts. With Oxid eSales, MVC is based on the following structure: [5]

- Model → /core
- View → /out
- Controller → /views

B. File system

Although elaborate and rather massive, consisting of cca. 470 directories and subdirectories, and some 2,600 files in the basic installation (Community Edition v4.9.x), Oxid file system is quite simple and easy for understanding. There are only ten base directories in the root folder:

- admin
- application
- bin
- core
- documentation
- export
- log
- modules
- out
- tmp

If developed "by the book", meaning to avoid to modify the core code and properly override built-in classes, the custom code goes into the following subdirectories:

- ./application/views
- ./modules
- ./out

Similarily to Magento, custom code (PHP) is grouped into modules, and custom design (images, CSS and JavaScript) is grouped into theme directories.

C. Module Interface

In order to override a view or core class, a new class must be defined as extension of an existing class. New class must be saved within the /oxid/modules directory. Of course, new moudule has to be activated from the administration section.

D. Database structure

Oxid database in this particular project consists of 143 tables. Out of them, 68 tables belong to standard Oxid CE installation, 8 tables belong to custom modules developed by a third party, and the rest are DB table views.

Majority of tables (56) use the MyISAM storage engine. Tables intended for products, orders, logs, SEO and shopping cart content are set to InnoDB storage engine. One table dedicated for captha data is stored in the memory.

E. Templates

The OXID eShop uses the Smarty template engine. [6, 7] Each template associated with corresponding view is located in the following directories:

- */oxid/out/basic/tpl* (front-end) and
- */oxid/out/admin/tpl* (back-end).

Template files consists of HTML code and embedded Smarty tags, thus enabling almost unlimited possibilities for visual customization.

III. OXID EXCHANGE MARKETPLACE

Oxid eSales platform extensions are available on the Oxid eXchange marketplace (*http://exchange.oxid-esales.com*) as free or propriaty solutions. The end users benefit from the implementation of flexible, fully customisable solutions from the know-how of over 100 certified OXID solution partners and from ongoing direct contact with the company's professional support and development teams. [8] Being an open source software, users benefit from fast, innovative turnaround, high quality development and dependable, long-term investment security.

Available extensions are grouped into different categories and subcategories according to their purpose. Basic categories are:

- Shop Management
- System Integration
- Order and Delivery
- Frontend and User Experience
- Market Places
- Marketing and Campaigns
- International Trade
- Legal and Security
- OXID Products

As stated before, Oxid eSales is mostly marketed on territories with german language speaking population, thus the documentation and supporting forums are generally in german language too. With relatively scarce documentation in english language, this in turn provides to be rather challenging for non german speaking developers.

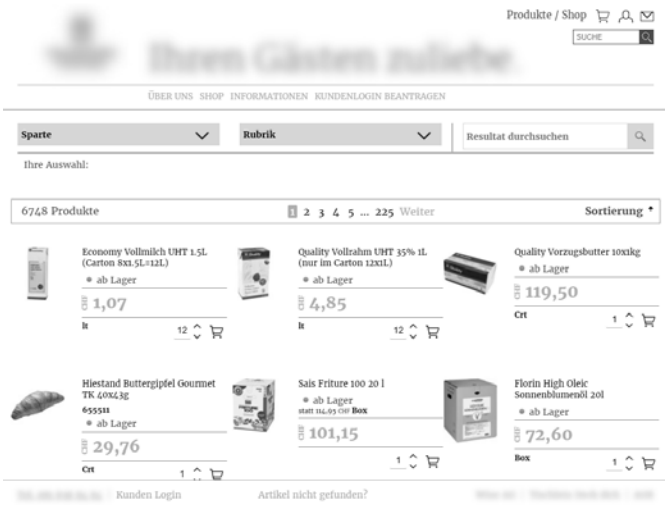

Image 1 - Webshop index page

IV. CASE STUDY: FOOD AND BEVERAGE ECOMMERCE SOLUTION

Initial eCommerce solution analysed in this paper was custom developed, by a third party, for a wholesale company in the food and beverages industry, based in Switzerland.

With cca. 6,500 articles on the stock, the company is specialized in supplying hotels and restaurants with various foods and drinks.

A. *Initial configuration*

Solution was based on the Oxid eShop v4.9.6 community edition, with the custom theme and several custom modules aquired from an independent software company:

- Administrator login
- Custom prices
- Custom quantities
- Custom stock management
- Ajax content filtering
- Customer orders listing

B. *Documentation*

Although the custom source code itself was neat and easy to read, it was not properly commented. In overall, project was poorly documented, without possibility to communicate with previous developer(s).

C. *Usage scenario*

This eCommerce solution has a specific usage scenario, considering that goods are ordered by chefs, using tablet devices or smartphones while on the workplace in the restaurant or hotel kitchen.

Due to work conditions, including food manipulation and processing, increased humidity and temperature, handling the tablet device in order to purchase products may prove to be difficult:

- Web shop browsing
- Products search

Image 2 - Webshop content filtration by section, category or a kewywords

- Sorting and filtering search results
- Payments

D. *User interface optimization*

Given conditions demand user interface (UI) which is easy to comprehend and to work with. Ajax controlled products search and filtering is a straightforward improvement of the webshop usability, Image 2.

Instead of optimizing webshop for a good search engine ranking (*SEO, Search Engine Optimization*) or overstuffing product pages with excessive details, content was kept clean and optimized for returning customers who are familiar with the webshop offering and who have exactl purchase lists.

Regular customers have a range of products they purchase on a regular basis. Thus a custom module "Customer orders listing" provides on a single web page a listing of all products that have been previously purchased by a given user account.

E. *Performance issues*

For a longer period of time, company who owns and operates this webshop was getting frequent complaints from the customers on webshop performance, especially on the "Customer orders listing" module which took exceptionally long times to generate web page, or simply just returns a blank screen.

Given information was scarce, reffering to page generation times between one and three minutes.

V. PERFORMANCE IMPROVEMENT

A. *Performance testing*

cl=oxwcategorytree(show/hide)
Memory usage: 7.832 MB (peak: 7.845 MB)
System memory usage: 8 MB (peak: 8 MB)
Executed: 2017-10-21 23:00:21
Timestamp: 1508619621.7578
Execution time:0.3012

Profile oxNew:	3.87706s	1287.06%	67	*	0.05787s
Profile buildTree:	0.57239s	190.01%	1	*	0.57239s
Profile loadinglists:	0.39777s	132.05%	2	*	0.19889s
Profile executeMaintenanceTasks:	0.34457s	114.39%	1	*	0.34457s
Profile process:	0.2553s	84.75%	2	*	0.12765s
Profile _getLangTranslationArray:	0.0717s	23.8%	8	*	0.00896s
Profile oxAutoload:	0.07122s	23.64%	66	*	0.00108s
Profile getDynamicUrl:	0.05828s	19.35%	2	*	0.02914s
Profile seoencoder_loadFromCache:	0.02111s	7.01%	3	*	0.00704s
Profile fround:	0.00036s	0.12%	16	*	2.0E-5s
Profile isSearchEngine:	9.0E-5s	0.03%	1	*	9.0E-5s
Profile oxviewconfig::setViewConfigParam:	5.0E-5s	0.02%	4	*	1.0E-5s
Profile oxviewconfig::getViewConfigParam:	3.0E-5s	0.01%	2	*	1.0E-5s

Image 3 - Segment in Oxid debugger info

Webshop owner did not provided neither a list of customers nor their complaints. Thus for initial performance testing, builtin Oxid debugger was used, Image 3.

Testing was performed on a developer machine with the following configuration:

- 4 cores Intel® Celeron® processor, 1.8GHz/4GB RAM
- Windows 10 operating system
- XAMPP platform
  - Apache 2.3 web server
  - PHP 5.6
  - MariaDB 10.1.21 SQL server

Table 1 - Page generation and fetch times in seconds

	Page generation	Cache fetch
Index page	10.17	0.43
Category page	4.74	4.21
Product page	4.03	3.21

Table 2 - Page generation times for Customer orders listing

	Page generation
Empty list	0.72
10 products	0.695
20 products	3.25

Table 3 - Table structure, module Customer orders listing

Field	Type	Null	Def	Extra
ID	int(11)	No		auto_increment
Adress_ID	int(11)	No	0	
Artikel_ID	int(11)	No		
Barcode	varchar(16)	Yes		
Bezeichnung	varchar(50)	Yes		
Menge	float	Yes	1	
Sorter	smallint(6)	Yes	0	
ausWeb	smallint(6)	Yes	0	
CANDELETE	smallint(6)	Yes	0	

B. Performance analysis

Preliminary results acquired with Oxid debugger proved that average page generation time varies for different webshop sections, in two distinct cases, Table 1:

- Dynamic page generation (first run, page was not cached yet)
- Page refresh (every other pass, with page already cached)
- For module Customer orders listing, demo user account was created with 20 products, mimicking previous orders.

C. Finding the critical usergroup

Upon several runs, test results for moderate products lists up to 20 products (Table 2) showed nothing unusual - page generation times were in the same range as other sections of the webshop (Table 1).

In the next step, it was necessary to discover the critical group of users with large sets of previously ordered products in order to further analyze page generation times, Listing 1.

```
SELECT
        id,
        count( id ) AS counter,
        Adress_ID,
        Artikel_ID
FROM `anwordersatz`
GROUP BY Adress_ID
ORDER BY counter DESC
```

Listing 1 - Who are the customers with large number of previously ordered products

id	counter	Adress_ID	Artikel_ID
1593	966	10361	1083
7182	897	10036	4308
22023	577	10646	1460
2305	571	10643	85
15239	557	10340	4308
52303	545	11550	12029
12027	543	10085	1083
15765	529	10008	1083

Listing 2 - SQL query result (partial) - Customers ordered by the number of previously ordered products

SQL Query (Listing 1) provided result with several hundreds of customers, the biggest customer having 966 different products in its previous orders.

D. Code performance analysis

```
project_root
├- ...
├- modules
|   ├- ...
|   ├- 3rd_party_company_name
|   |   ├- ...
|   |   └- orders_listing_module
|   |       ├- controllers
|   |       ├- core
|   |       ├- models
|   |       ├- out
|   |       ├- translations
|   |       └- views
...
```

Image 4 - Oxid eSales - Custom module directory structure

As shown on Image 4, 3rd party modules are placed into modules subdirectory. Target module consists of 27 files and 56 directories.

Source code mapping was done using a tool initially developed for Magma CMS™ content management system which provides mapping of variables, constants and functions throughout the filesystem.

In order to pinpoint slow SQL queries or places in the code which are causing slow execution time, a simple technique of injecting time markers was used, Listing 3 and Listing 4.

```php
<?php

class tInfo
{
        public static $microtime;
}
```

Listing 3 - PHP class, tInfo

```
tInfo::$microtime['render_3'] = microtime(1);
```

Listing 4 - Time marker placement, example

```
Array
(
    [marker_17] => 1509142291.6914
    [marker_18] => 1509142611.7658
    [marker_19] => 1509142612.6476
)

Array
(
    [marker_17] => 1509141466.7476
    [marker_18] => 1509141470.4244
    [marker_19] => 1509141471.1637
)
```

Image 5 - Time marker readings

E. Results

Several iterations were required in order to pinpoint a segment of code which was causing slow page generation (320 seconds, for list of 647 products), placed between time markers 17 and 18, Image 5.

Upon further investigation using similar technique, it was trivial to locate poorly optimized queries and functions located in another custom module, made by the same 3rd party company.

After the problematic piece of code was rewritten and optimized, execution time of the same section was reduced under 4 seconds (80 times faster than orginal code), Image 5.

VI. CONCLUSION

Although a simple tecnique and tools were used in this research, they provided clean and unambiguous solution resulting in acceptable page generation time, and even more importan, in overall customer satisfaction.

The problem was caused by module(s) developed by a 3rd party company. The original source code was correct, but unoptimised for clients with large lists of previously ordered products.

Properly set use-case scenario is a paramount in software development. Although correctly set for a specific customers in a very specific working environment, an oversight in code development process caused customer's dissatisfaction, and furthermore, reduction in company's turnover.

LITERATURE

[1] N. Henderson, "Magento, WooCommerce Lead Ecommerce Platform Market Share: Report", theWhir, 2016. Access: Oct. 4 2017. Available on http://www.thewhir.com/web-hosting-news/magento-woocommerce-lead-ecommerce-platform-market-share-report

[2] "Magento 2 Contributes to the Global Ecommerce Platforms Market", aheadWorks, 2016. Access: Oct. 4 2017, available on https://blog.aheadworks.com/magento-2-contributes-to-the-global-ecommerce-platforms-market/

[3] “Server and system requirements”, Oxid eSales, 2017. Available on https://www.oxid-esales.com/en/support-services/documentation-and-help/oxid-eshop/installation/oxid-eshop-new-installation/server-and-system-requirements.html

[4] "System requirements OXID eShop Community Edition", Oxid eSales, 2017. Access on Oct. 5, 2017. Available on https://www.oxid-esales.com/en/support-services/documentation-and-help/oxid-eshop/installation/oxid-eshop-new-installation/server-and-system-requirements/system-requirements-ce.html

[5] "MVC, where is the Model?", Oxid eSales Forum, 2010. Access: Oct. 15, 2017. Available on: http://forum.oxid-esales.com/showthread.php?t=4205

[6] A. Ziethen, "PHP Module Programming with OXID eShop CE", Developer.com, 2010. Available on https://www.developer.com/lang/php/article.php/3857246/PHP-Module-Programming-with-OXID-eShop-CE.htm

[7] "Smarty Grundlagen", Dokumentation und Hilfe, Oxid eSales. Available on http://www.oxid-esales.com/de/support-services/dokumentation-und-hilfe/archiv-oxid-eshop/design-anpassen/templates/smarty-grundlagen.html

[8] Oxid sSales AG, LinkedIn, visited on Oct. 15 2017. Available on https://www.linkedin.com/company/oxid-esales-ag