Optimizing
Magento Performance

Optimizing Magento Performance

Author: Michael Zhalevich, aheadWorks.com

According to the recent research, 64% of dissatisfied online shoppers said they were less likely to visit a slow retailer again, 62% were less likely to purchase from the site again and 48% would purchase from a competitor.

Such statistics always makes developers puzzle – from one hand, the website owner wants the application to be multifunctional and universal, from the other – it shouldn't be slow. And the problem is always to make a solid extension work as fast as possible.

Performance optimization is a good way out in such situation. Optimizing a website, you should consider providing two factors:

1. acceptable pages loading speed;
2. acceptable pages loading speed when N people use the same application simultaneously.

The basic advice regarding response times that has been about the same for forty years:
0.1 second is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.

1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.

10 seconds is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Considering all this information, let's try to provide the pages loading speed within 0.1 – 1 second with the load of 20 competitive requests by optimizing Magento performance.

What we have

Server:
· CPU: Intel® Xeon® CPU X3320 @ 2.50GHz
· Memory: 4GB
· HDD: 1HDD, non-RAID
Magento CE 1.7.0.0

What we should do

1. Select a tool to measure Magento website performance

http_load
http_load runs multiple http fetches in parallel, to test the throughput of a web server. It runs in a single process, so it doesn't bog down the machine. It can be configured to do https fetches as well.

Apache HTTP server benchmarking tool
It is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving.

Siege
Siege is an http load testing and benchmarking utility. It was designed to let web developers measure their code under duress, to see how it will stand up to load on the internet. Siege supports basic authentication, cookies, HTTP and HTTPS protocols. It lets its user hit a web server with a configurable number of simulated web browsers.

Apache JMeter
Apache JMeter is open source software, a 100% pure Java desktop application designed to load test functional behavior and measure performance. It may be used to test performance both on static and dynamic resources. Apache JMeter can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.

2. Select OS for server

Here you can select whatever you like or what your hosting provider offers:

· BSD: FreeBSD, OpenSD, etc.
· Debian: Debian, Ubuntu
· Gentoo, Slackware
· RHEL

3. Select webserver with PHP

Apache + mod_php
It is the most popular application environment

Nginx + Apache + mod_php
Nginx as a proxy before Apache doesn't influence the pages generation speed but gives the increase of loading speed and economy of system resources

Nginx + php-fpm
This variant is less popular as none of the web hosting control panels like cPanel work flawless with Nginx. But our tests showed that exactly this environment was 30% faster than Apache

aheadWorks
www.aheadworks.com
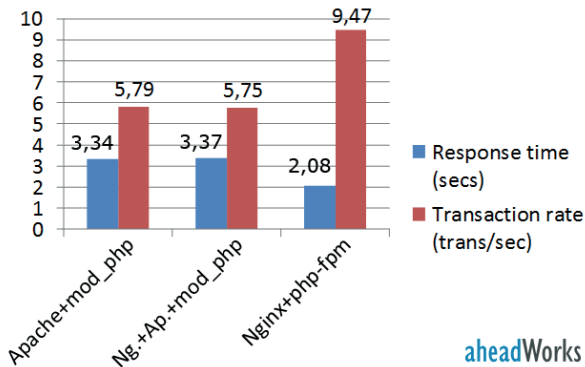
To avoid many serious mistakes in correct webserver configuration, be careful with Nginx settings and don't forget to:

Forbid access to Magento catalogs

```
location/app/{deny all;}
location ~ ^.ht {deny all;}
location /.svn/ {deny all;}
```

Disable logging where it is not necessary

```
location/media/catalog/{
access_log off;
}
location /js/ {
access_log off;
}
```

Turn on compression for js and css

```
location/media/js/{
gzip on;
gzip_min_length 1000;
gzip_disable "MSIE[1-6]\.";
expires 5d;
access_log off;
}
```

Enlarge the FastCGI response time

```
fastcgi_read_timeout
```
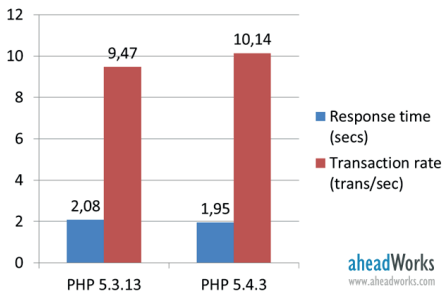
4. Select PHP 5.3.* or PHP 5.4.*

There are 2 obvious advantages of moving to PHP 5.4.*:

· it is faster;
· later you can use new language constructions.

But don't forget about the disadvantage you can face:

· problems when migrating to new version

The tests showed the following results:
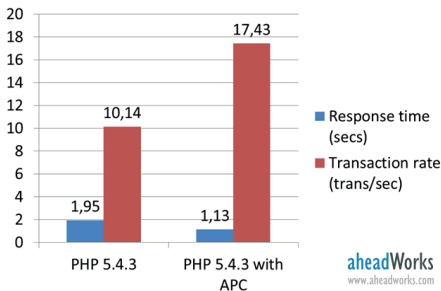

5. Select opcode cacher

Open Source
- · APC
- · Xcache
- · eAccelerator

Proprietary
- · IonCube
- · Zend Server

As you can see below, the opcode cacher usage gives the speed increase of about 70%:

6. Change the number of child-processes in PHP-FPM depending on your hardware

7. Move users sessions to memcached

8. Move cache to APC or memcached

9. Set the higher values in the settings of PHP – realpath_cache_size and realpath_cache_ttl

Let's sum up what we've got after all these steps on optimization:
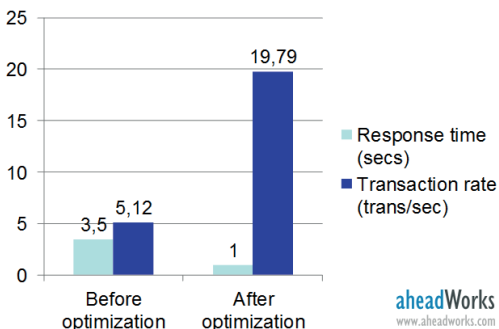
As you can see, server environment optimization resulted in reducing rendering speed in 3.5 times as well as decreasing the price of the required hardware.

aheadWorks

Magento Extensions & Development

ecommerce.aheadWorks.com

20000+
CUSTOMERS WORLDWIDE

60+
MAGENTO EXTENSIONS

100+
CUSTOM PROJECTS

40+
MAGENTO THEMES

INDUSTRY PARTNER

SOLUTION PARTNER