

TESTING GUIDELINE MAGENTO SPEED

1. Introduction

In this document the guidelines for testing the speed optimization of a Magento webshop are described. These can be used for approving the project as well as not interpreting a problem on the local side as the server being slow.

2. Requirements

Broadband internet connection (at least 1mbps)
There should be no other downloads on the connection

3. Testing goals

The load time should be < 2 seconds for all pages on the Magento webshop. This doesn't include the page parse time.

4. Testing Tool

The actual load time can be measured in Firefox using an addon <https://addons.mozilla.org/en-US/firefox/addon/lori-life-of-request-info/>

This addon will show you 4 different numbers in the bottom status bar once installed.

The first is the response start time. Ideally this will be between 0.2-1.5 seconds

The second is the response end time. This should be no more than 0.5-1 second if you are using a connection of the required speed. It has nothing to do with the server when it comes to Magento because Magento starts its response output only after all of the code has been parsed. The servers are on 100mbit connection, so server speed also doesn't affect this. The only thing that affects this is the users connection and the browser parse time. The browser parse time shall be between 1-2 seconds unless tested on a very old computer which means the response end time should be no more than 2+1=3 seconds on a good connection. This is not the same as checking Wordpress or Intershop. In Wordpress or Intershop the response end time can be affected by the php page parsing time because response starts as soon as the php processing starts unlike Magento where response output starts only after entire php code is run (that is how MNC and Zend framework work).

5. Testing Steps

1. All browser cache and data should be cleared for optimum testing (pages might load faster without this giving wrong impression of actual speed).
2. Load any 2 pages (this is the first load and will not count towards the <2 seconds load time)
3. All pages after step 2 should have a load time < 2 seconds
4. If response start time > 1.5 seconds, page failed test (currently in the unoptimized version, no page satisfies this on moderate browser load)
5. If response end time > 3 seconds please check your connection for overload in downloads and go back to step 4 after fixing this problem.
6. Ideally the third and last number (x KB, y req) should be as small as possible, however is limited to amount of content (images etc) on a page
7. If the pages satisfy 4, you will want to run load tests if possible. I will be running load tests myself, but if possible you can have 25-50 or more people browsing the site at the same time. This shouldn't affect the above numbers.

Over 50 simultaneous users the numbers will increase gradually, and if you plot the delay over the number of simultaneous users, it should grow exponentially in an ideal system. It does so up to 50 users too, but the effects will be too small to notice and the slope of such a graph will be very small.

A straight line graph with a very small slope means that you have not tested with enough simultaneous users, which usually means the site can handle a load more than expected. A straight-line graph with a high slope means that the testing has failed and needs more optimization on the server end.

Ideally, the graph should be exponential up to 400-1000 simultaneous requests after which it will start to become a straight-line graph. This is absolutely normal because one server can handle only so many loads. A rough estimate of this gives 400 simultaneous users browsing the site for 1 hour x a good load for 10 hours a day x 30 days = 120,000 users per month.

The graph should not be of a negative slope or of a low positive slope. It just means you aren't putting enough load on the server to actually run an effective test. This could also be due to the fact that the connection that you are testing on isn't strong enough to handle the downloads (if you are running all the tests on a single connection).

If you are running multiple tests on a single connection simultaneously, your connection should be able to handle the load. This means that you need 1mbps per test that you are running as per above requirements. For 50 simultaneous tests on a single connection you will need a 50 mbps connection for accurate testing, or else you are generating false results by overloading your connection and not the server.