

Page Load Time Worksheet

Source: Drupal Performance Agency

Pages: 6

Last Modified: 10:50 PM Jan 12, 2008

License: Creative Commons Attribution-ShareAlike2.0

Description: *There are a number of ways to improve the amount of time it takes visitors to load your web page.*

This worksheet was designed by the Drupal Performance Agency. For more information, email perf@tag1consulting.com.

Measurements

There are many tools that can be used to measure page load performance on your website. Here are some that we recommend.

I. YSlow FireFox add-on

The YSlow FireFox add-on is maintained by Yahoo. It analyses your web page and offers a simple to understand report explaining why your website is slow to load. The tool requires that the Firebug FireFox add-on also be installed. Visit <http://developer.yahoo.com/yslow/> for complete details on using the FireFox add-on.

II. Load Time Analyzer FireFox add-on

The Load Time Analyzer Firefox add-on is attributed to Google. Through a relatively spartan user interface, it records how many objects FireFox has loaded, and how long they take, displaying this data in a simple graph. This is a useful way to determine which elements on your web page are consistently taking the longest time to load. Visit <https://addons.mozilla.org/en-US/firefox/addon/3371> to download the add-on.

III. Third-party monitoring solutions

There are a number of third-party monitoring solutions available. They offer the advantage of retaining historical data that can be used to graph how any changes make have affected site performance. They can also offer a view of your site's performance from any networks throughout the world.

General Improvements

I. Reduce the number of HTTP requests on each page

Each object on your web page requires a new connection from a visiting web browser to your web server. By reducing the number of objects on your web page, you can drastically improve the page load times for your website.

1. Aggregate and compress your CSS files

Drupal 5 includes core support for CSS aggregation and compression.

2. Aggregate and compress your JavaScript files

Drupal 6 gains core support for JavaScript aggregation and compression. Back ported patches for Drupal 5 exist.

II. Set far-future expire headers on CSS, JavaScript, images, etc

Set a far future Expires header on all objects on your web pages. This will tell visiting web browsers that they don't need to validate their cached copies with each page load, with the potential to greatly improve page load times for repeat visitors. The header can expire in a few days, or even months or years in the future. Should you need to change this cached object before the cache expires, you will need to change the name of the object or the path to the object, causing web browsers to treat it as a new object.

III. Serve objects from multiple domains

One pages on which there will always be a lot of objects, you can sometimes see a performance gain by serving the content from multiple domains. (For example, server1.sample.com, server2.sample.com, server3.sample.com, even if all are served from the same physical server with the same IP address). Web browsers limit the number of active connections allowed from a single domain, thus by serving content from multiple domains you can cause web browsers to download more objects on a given page at the same time. Note that on the first visit to your page, the client web browser has to do a DNS lookup for each domain that you use, so serving objects from too many domains can actually cause a slowdown. It is generally recommended to use anywhere from two to four domains, depending on how many objects you are serving per page.

IV. Decrease the size of files on your web page

A significant percentage of Internet browse the Web using slow dial up connections. Large objects on your web pages will cause your pages to load slowly for these visitors. One way to compress content served from your web site is through the use of

mod_deflate, discussed below in the Apache section.

V. Reduce the number of cookies used by your website

Cookies and all their contents are transmitted along with every HTTP request in a single domain, including requests for CSS, JavaScripts, and images. Minimizing the number and size of cookies is therefore a good idea. You can also serve CSS, JavaScript and Images from another domain to prevent the cookies and their content from being included in these HTTP requests.

VI. Utilize a Content Delivery Network

If your website reaches a global audience, you may benefit from using a CDN, or Content Delivery Network, allowing your visitors to download content from a network located physically closer to them.

Drupal

Drupal is a modular content management system written in PHP. Drupal was originally released as an open source project in 2001. Visit <http://drupal.org/> for more information on Drupal.

I. Enable only required modules

There are a wealth of contributed modules freely available from <http://drupal.org/> for extending Drupal's core functionality. Each module you install is more code that has to be parsed by PHP. Also, many contributed modules have not been carefully audited for performance or security concerns.

II. Enable the page cache

Drupal core provides simple page caching functionality for anonymous visitors. Essentially, once Drupal creates a complete page for an anonymous visitor, it caches this page in the database. Future anonymous visitors to the same page view the cached version, saving Drupal from having to recreate the page each time. When site content is updated, Drupal flushes the page cache to prevent serving stale data.

1. Visit "Administer >> Site Configuration >> Performance" to enable the page cache.
2. If your modules do not use the Drupal `_init()` or `_exit()` hooks, you may be able to enable "Aggressive" caching for even better performance.
3. If content on your website is constantly changing, you may need to enforce a

minimum cache lifetime to see any benefit from the page cache.

III. Enable CSS aggregation and compression

Drupal 5 core provides an option for aggregating and compressing CSS files, at the same location for enabling the page cache.

IV. Enable a file cache module

1. Boost module (<http://drupal.org/project/boost>)

The boost module can greatly improve Drupal performance by writing a static version of your Drupal generated pages, allowing the Apache web browser to serve them without any PHP overhead.

2. Fastpath_fscache (http://drupal.org/project/fastpath_fscache)

The fastpath_fscache module moves Drupal's cache from the database to the file system, making it possible to server pages without accessing the database. However, there has not yet been an official stable release of the module.

Apache

The Apache HTTP Server boasts being the most popular web server on the Internet since April of 1996. The recommended current stable branch is the 2.2.x branch. Visit <http://httpd.apache.org/> for more information on Apache.

I. Enable HTTP keepalives

Configuration option: KeepAlive

This Apache configuration options is enabled by default in the latest versions of Apache. It allows web browsers to make multiple requests through a single connection to the web server, rather than having to make a new connection to the server for each request.

II. Enable gzip compression

There are several Apache modules the provide content compression.

1. mod_deflate (included with Apache 2.x)
2. mod_gzip

III. Increase the number of standby processes

Configuration options: MinSpareServers, MaxSpareServers

As you are using PHP, you are most likely using the Apache pre-fork MPM. This is a

non-threaded implementation of Apache that pre-forks child processes to handle incoming connection. Each active connection to your web server is handled by a separate child process. If you have a very active web server handling large numbers of simultaneous connections, you should increase the number of spare servers that are waiting to handle new connections. By default, the Apache pre-fork MPM keeps a minimum of 5 spare servers waiting for new connections, and a maximum of 10 spare servers. On a very active web server, you may want to increase the minimum spare servers as high as 32, and the maximum spare servers as high as 64.

IV.Reduce Apache's memory footprint

You can see how much memory each Apache process is using by running “ps auxww”, or by running “top”. There is a performance hit for allocating memory, so you want to make your Apache process as lean as it can be.

1. Compile Apache statically
2. Only enable modules that you actually need. (Use “*apache -M*” and “*apache -l*” to list all dynamic and static modules currently loaded by Apache. Disable any that you do not actually need.)

PHP

PHP is the scripting language utilized by the Drupal CMS. Visit <http://www.php.net/> for more information on PHP.

I. Enable an opcode cache

PHP is a scripting language. Each time your web page is visited, the Zend Engine compiles your PHP script into PHP opcode (“operation codes”. The compiled binary opcode is what is actually executed, then it is flushed from memory to be compiled the next time your web page is visited. Opcode caches retain the compiled opcodes, reusing them on each page visit and greatly reducing the overhead of PHP. Unfortunately, no existing opcode cache is completely bug free, and they will occasionally cause Apache to segfault resulting in blank white pages being served. When this happens, you need to gracefully restart Apache to restart the failed child process. Scripts exist to automate this graceful restart process.

1. APC, the “Alternative PHP Cache”, is often a favorite because it is developed and maintained by the authors of PHP. (<http://us2.php.net/apc>)
2. eAccelerator is another favorite, as it often out performs the competition.

(<http://eaccelerator.net/>)

3. Xcache is a newer solution that has a reputation for being both fast and very stable, tending to cause less Apache segfaults than other opcode caches. It also offers a nice administrative view of cache usage, helpful in tuning it for optimal performance.

(<http://xcache.lighttpd.net/>)

4. Zend Platform is a commercial offering (formerly known as Zend Cache and Zend Accelerator) claiming advanced features for developers, targeting Enterprise Scalability. (<http://www.zend.com/en/products/platform/>)

II. Reduce PHP's memory footprint

You can view all features currently compiled into your PHP installation by running `phpinfo()`. You can reduce PHP's memory footprint by recompiling PHP, disabling any features that you are not using.