

WPengine®

15 WAYS TO OPTIMIZE WORDPRESS FROM THE INSIDE OUT

WHITE PAPER

By Janna Pyles

TABLE OF CONTENTS

- Introduction..... 3**
- 7 Key Ways to Improve Load Time on Your Site’s Server..... 3**
 - Leverage Page Cache.....3
 - Use Object Cache4
 - Reduce Database Bloat.....4
 - Keep WordPress Clean4
 - Use Relevant Plugins.....4
 - Use a CDN.....5
 - Aim for Scalability.....5
- 8 Ways to Improve Your Page Render Time..... 5**
 - Leverage Browser Caching.....5
 - Enable GZIP Compression.....5
 - Combine and Minify Static Resources6
 - Use a CDN.....6
 - Compress Images.....6
 - Defer Javascript Parsing.....6
 - Lazy-Load Long Pages6
 - Reduce External Calls.....7
- Author Bio 8**
- About WP Engine 9**



Introduction

Speed is an increasingly important demand for today's websites. [Studies have shown](#) that today, 25 percent of users will only wait two to four seconds for a page to load before leaving. At WP Engine, we're in the business of making your site blazing fast so your content can reach your audience quickly and effectively.

Therefore, we've put together this list of ways you can improve the time it takes for your page to load.

To start, site performance can be broken down into two main segments:

- The amount of time it takes the server to generate the page
- The amount of time it takes the browser to download the page

In this white paper, we'll be focusing on the things you can do to improve how quickly your site's server is able to generate a page to send to a web browser.

7 Key Ways to Improve Load Time on Your Site's Server



Here are some ways to make WordPress lightweight, speedy, and scalable.

In this section we'll be focusing on the things you can do to improve how quickly your server is able to generate a page to send to its web browser.

1. Leverage Page Cache

One of the main differences between HTML and PHP sites is that with HTML sites, the web browser simply has to request a static file to serve the page. With PHP (which WordPress uses), the page is dynamically built when it is requested by executing PHP code and database queries.

While this means PHP-based sites are computationally more expensive on a server, it also means the type of content your site can serve is more dynamic and personalized to your users. This

is why many choose to use WordPress as a content management system (CMS).

Since it's more taxing on your server to build a dynamic web page for each user that comes to your site, we highly recommend using a **page cache system**. This means each page that's requested is stored as a static file in a caching layer on your site's server, to serve to visitors for a specific period of time. This offloads much of the work the server has to do to generate your site's pages when they are visited.

At WP Engine, we use page cache as part of our [Evercache](#) system, with a cache expiration time of ten minutes. This means that when a page is requested for the first time, it spends time generating the page with PHP and database queries and then stores a copy of that page in cache. For the next ten minutes, when that page is requested, it will be served from the cache layer instead of regenerating it again. After ten minutes, the cache is purged and the process starts over.

2. Use Object Cache

Aside from generating the page with PHP, your WordPress site also depends on a database to create pages and store information. When information is requested from the database, running those database queries can also take more time and power on the server's end.

When you use a layer of the **object cache**, it allows you to store WordPress database query results in the server's memory. So instead of running a new request for the same data to be processed by the database each time, WordPress is able to get this information from **server memory**.

WP Engine uses an object cache layer to store repeated query results. All sites will have the option to enable this caching layer from their User Portal under *Installs » Utilities*.

3. Reduce Database Bloat



Keeping your database in tip-top shape should also be a priority for your site. The less database bloat you have, the faster your site can perform.

This means cutting down on things like post revisions, orphaned post/postmeta and comment/commentmeta rows, as well as reducing autoloaded rows in `wp_options`.

In WordPress, the `wp_options` database table has a lot of your site's configuration settings like: `siturl`, active themes, active plugins, cron jobs, and more. These are important settings which probably should be set to automatically load on every page.

By default, most options stored in `wp_options` are set in the "autoload" column to "yes," which means they have to load on every single requested page. However, plugins or themes may be adding rows and transients to your `wp_options` table, which may not be information that needs to load on every single page.

It's good practice to periodically check through your `wp_options` table to see if there are any rows added by themes or plugins which might be very long, and may be causing a delay in site load time.

4. Keep WordPress Clean



WordPress is a very versatile CMS, but sometimes customization can be very costly to the amount of time it takes to load your site. It's best to keep configuration files, like the `.htaccess` file and `wp-config.php` files, close to their default state. These files have to be loaded on every page load too, so reducing things like rewrites and server directives in the `.htaccess` file, in particular, can be helpful.

WP Engine offers the option to set redirects and rewrites in our User Portal. Redirects set here will be executed before the request is sent to be processed by PHP. This translates to less heavy lifting for the server to redirect the request.

5. Use Relevant Plugins

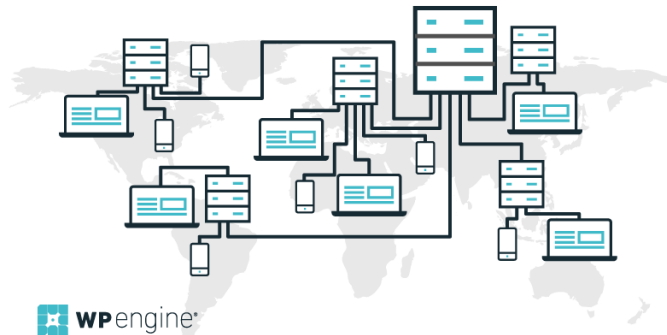
Since WordPress is open source, there are thousands of plugin and theme options to choose from when it comes to your site.

When you're deciding on which plugin to use for a site function, check out the plugin's page and see how often it's updated, and how often support threads are resolved.

If updates are fairly frequent and the authors still seem engaged in support threads, this is a good indication that this plugin is being actively maintained. This is good because more updates often mean more optimizations to the way the plugin functions. It also means the authors are more likely to help patch any security vulnerabilities the plugin may encounter.

Using relevant plugins also means to cut down on the plugins your site doesn't need or use. If a plugin is inactive and you don't know when you would use it again, it's best to delete it entirely.

6. Use a CDN



Using a CDN refers to serving your site's static content like images, CSS, and JavaScript from a network of servers spread across the world. This enables your site to deliver these files faster to everyone, no matter where your visitors are geographically located.

CDN technology is offered at WP Engine, which can be enabled right through your User Portal without using a plugin to rewrite your site's source code. But other plugins can also integrate with CDN for minification or caching purposes.

There are also CDN options which involve routing your site through a reverse proxy and leveraging caching at this level. Offloading content through this type of CDN can be valuable for sites with extremely high levels of traffic. They allow for custom configuration of caching for pages and files as well if you want a more granular approach to caching. By using this kind of CDN, it helps offload a good portion of the server processing to cache it instead.

7. Aim for Scalability

With WordPress, it's important to weight content that's dynamic against content that's quick and lightweight for your server. Offloading dynamic actions that are unique to visitors to be loaded by JavaScript or other browser-level interactions is one way to help ensure your quest for dynamic content isn't over-burdening your site's server.

It's also important to keep your site as cacheable as possible. Limit the amount of PHP actions and queries being called on your site. This means, reducing actions that have to be repeated on every page load, like a page-view counter or a share counter. Instead, try using caching settings for these features to only refresh them every few minutes, or every hour.

If your blog has many older archives of articles, this could also mean something as simple as being crawled by Google could add up to

server load time, because of the number of uncached pages being requested. You can try increasing your cache-expiration settings in your site's .htaccess file, to help ensure that fewer old pages are going to the server uncached.

8 Ways to Improve Your Page Render Time



In this section we'll be focusing on the things you can do to improve the render time of your page in the visitor's browser. Specifically, the elements which might be causing your users to see a blank white screen before your site starts to show content (or "render").

1. Leverage Browser Caching

When your web server has HTTP headers set up to specify cache expiration time, it also includes browser directives on how long the webpage should be cached in your visitor's browser. This tells your visitor's browser to download the elements of your website (like CSS, JavaScript, and images) from their machine's local disk rather than from the network. Since this means the browser has fewer network connections to make, this will help ensure your webpage loads faster for them.

It's also important to ensure your HTTP headers include an expiration date so the browser knows when to get the resources from the network as if they were new, rather than getting them from their local machine.

At WP Engine, we take care of this piece for you since our servers are already optimized to serve WordPress websites. By default, pages are set to expire every ten minutes, and static resources like images, CSS and JavaScript are set to expire every 30 days.

2. Enable GZIP Compression

GZIP compression, also known as "browser compression" is a way of compressing text-based content like your site's code, into a more lightweight and easily-understandable set of characters for your browser to read.

If you host your site at WP Engine, our server configuration already handles GZIP compression for your website. However, if you host

elsewhere you may want to use a plugin that helps optimize this for you, like [Check and Enable GZIP Compression](#). This can also be [set in your site's .htaccess file](#) manually if you prefer.

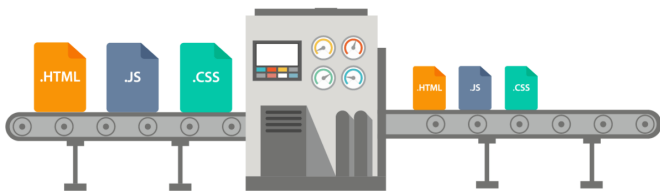
Remember, GZIP Compression can only work on pages and files served from your site's server. You cannot control whether external resources like ads, external web fonts, or external images are using GZIP.

3. Combine and Minify Static Resources

Minification and combining multiple static resources into one helps for a few reasons: the browser has fewer items to download, and the items that it does download are smaller. This equates to a faster page render time since the browser can download the page content faster.

There are several plugins available to help with optimizing these resources, but the ones we've seen work most successfully have been [Better WordPress Minify](#) and [Autoptimize](#). Both have the option to be used with a CDN URL as well so these resources can be loaded from a server that's geographically close to the end user.

4. Use a CDN



As previously discussed, a CDN can help by loading your site's static resources like images, CSS, and JavaScript from a network of servers located all around the world. This means when a user in Germany requests an image, it loads the static files from a CDN server closest to that user. CDN is especially important for sites with a worldwide audience.

At WP Engine, CDN is available for an extra cost on Personal-level plans and included on Professional and Business-level plans.

5. Compress Images

Unoptimized images can easily be the largest source of bloat in your site's load time. You can use a page test like WP Engine's Page Performance tool, or [webpagetest.org](#) to see a waterfall view of your site's load time. If the longest-loading elements are images, or the tools recommend optimizing images, it may be time to look at an image compression plugin.

Two of the most common ones are [WP Smush](#), and [EWWW Image Optimizer Cloud](#). These plugins both offer options for lossy compression, which allows you to compress the image size, while

still retaining the same image quality. Not only does this help your page load faster, it also reduces bandwidth usage and helps increase functions like copying your site, or creating backups.

6. Defer Javascript Parsing



If you've been using page test tools to test your website's speed, you may have run across this recommendation which can be difficult to understand. If you look at your page's waterfall view using a tool like [webpagetest.org](#) or [Pingdom](#), you'll likely see that there's a number of JavaScript files (.js files) loading before your "start render" line. This is known as "render-blocking JavaScript".

The core function of JavaScript is to perform an action on a webpage, like a popup or rotating images in your slider. In reality, these actions don't need to be loaded until your site fully loads the content and styles. So by "Defer JavaScript Parsing," these tools are really saying, "load this stuff later on in your page instead of at the top."

There are a few plugins out there which can help you in deferring this JavaScript, but one we've seen work well is [Above the Fold](#). This plugin also plays nice with Autoptimize and Better WordPress Minify!

7. Lazy-Load Long Pages

For one-page sites and sites which have a long home page, Lazy Loading can be a real time saver. Lazy Loading essentially prevents the elements lower down on your page from being loaded until the visitor scrolls down to see them. By not loading all of the content of your long page at once, this allows your site to begin rendering faster. A common plugin used for this would be [BJ Lazy Load](#).

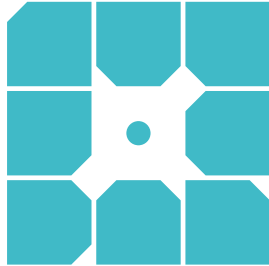
8. Reduce External Calls

When a webpage is loaded, every element on that page has to be requested by your visitor's web browser. Some sites may have a large number of ads loading, external stylesheets and fonts, or embedded/iframes of elements from other sites. With every external request, a DNS lookup has to be made by the browser and it must download those elements from those external sites. The DNS lookup time can eventually add up to a lot of browser load time. And, if those sites are taking a long time to respond it could affect your own page time as well. Reducing the number of these calls can improve your site load time.



ABOUT THE AUTHOR: **JANNA PYLES**

Janna Pyles is WP Engine's Site Performance Subject Matter Expert. She has lived in Austin since 2014 and enjoys hiking with her dog, painting, and both technical and free form writing.



About WP Engine

WP Engine powers amazing digital experiences for websites and applications built on WordPress. The company's premium managed hosting platform provides the performance, reliability and security required by the biggest brands in the world, while remaining affordable and intuitive enough for smaller businesses and individuals. Companies of all sizes rely on WP Engine's award-winning customer service team to quickly solve technical problems and create a world-class customer experience. Founded in 2010, WP Engine is headquartered in Austin, Texas and has offices in Limerick, Ireland, San Francisco, California, San Antonio, Texas, and London, England.

www.wpengine.com

