



# WordPress® on cPanel

Performance Optimization Guide



## Overview

The speed and performance of your website is just as important as the content, product, and/or service you are selling. While WP Toolkit is a powerful tool when it comes to managing WordPress installations, there is a lot to be said about actually fine-tuning the server in order to get the best possible performance from the hosted websites.

With this in mind, we set out to establish a baseline for WordPress on a typical cPanel server using an out-of-the-box configuration, and then quantify the impact of various changes and improvements that could be made to the server.

## The Test Process & Environment

All tests were conducted on a VPS provided by Digital Ocean, with the following specifications:

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>■ CentOS 7</li> <li>■ Two CPU cores</li> <li>■ 4GB RAM</li> <li>■ 80GB Disk</li> </ul> | <ul style="list-style-type: none"> <li>■ cPanel V98.0.12</li> <li>■ PHP 7.4.25</li> <li>■ PHP-FPM enabled</li> </ul> |
|---|--|

All tests were performed using WordPress version 5.8.1 installed via WP Toolkit. The site was configured with WooCommerce and populated with mock data to emulate a typical eCommerce website.

One of the best ways to speed up any WordPress website is to use some sort of caching. Caching stores your WordPress posts and pages as static files and serves those files to users (rather than rendering those pages for each request), saving a considerable amount of processing power while speeding up the end user experience. Our tests focused on three different caching options:

- OPcache compiles human-readable PHP code to opcode that is understood by the server when any PHP file loads for the first time on a web page. The opcode is saved to the server's memory for faster loading at each subsequent visit. OPcache is installed by default on any server running PHP version 5.5 and above.
- cPanel NGINX® sits between a client and the server, saving copies of all the content it sees. If a client requests content the cache has previously stored, the cache returns the content directly without contacting the server. NGINX can be installed and managed via the NGINX Manager within WHM®.
- LiteSpeed with LSCache is a WordPress plugin built as an all-in-one site acceleration tool with a server level cache. To use LiteSpeed with LSCache, simply install the plugin via WP Toolkit in the same manner as any other plugin.

## The Results:

	OPcache	cPanel NGINX	LiteSpeed with LSCache
Low Traffic Throughput	137% increase	300% increase	300% increase
High Traffic Throughput	53% increase	Nearly 1000% increase	Nearly 1000% increase
Average Response Time	60% reduction	90% reduction	90% reduction
Time-to-First-Byte	50% reduction	90% reduction	90% reduction
Fully Loaded Time	50% reduction	50% reduction	60% reduction
Content Load Time	40% reduction	50% reduction	60% reduction

Some basic configuration changes also resulted in significantly improved performance:

- Apache Worker MPM is a Multi-Processing Module that implements a hybrid multi-process, multi-threaded server that results in a larger number of server requests to be handled with fewer system resources. The Worker MPM is enabled by appropriately editing the Apache configuration and restarting the httpd service.
- `memory_limit` is a value within the `php.ini` file that controls the maximum amount of memory, in bytes, that a script is allowed to allocate. You can increase this limit simply by editing your server's `php.ini` file or using MultiPHP INI Editor in WHM.

The Results:

Apache Worker MPM	PHP Memory increase – 256MB <code>memory_limit</code>
<ul style="list-style-type: none"> <li>■ 15% lower response time.</li> <li>■ 30% improvement in TTFB.</li> <li>■ 20% improvement in FLT.</li> </ul>	<ul style="list-style-type: none"> <li>■ 45% reduction in failure rate during high traffic testing.</li> <li>■ 10% increase in high traffic throughput.</li> <li>■ 25% lower maximum response time.</li> <li>■ 30% improvement in TTFB.</li> </ul>

While all of the above are positive results on their own, the benefits do compound as well. For example, by implementing NGINX with worker processes matching the number of CPU cores, utilizing OPcache, increasing the PHP memory limit, and switching to the Worker MPM for Apache, the performance improvements are notable:

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>■ 260% increase in low traffic throughput.</li> <li>■ 1200% increase in high traffic throughput.</li> <li>■ 75% reduction in TTFB.</li> <li>■ 55% reduction in FLT.</li> </ul> | <ul style="list-style-type: none"> <li>■ 80% reduction in Backend load time.</li> <li>■ 60% reduction in ContentLoad time.</li> <li>■ 50% reduction in response time deviation and maximum response time.</li> </ul> |
|---|--|

Although there is a nearly infinite number of potential modifications that could be made to increase the performance of a server, all of the tools and features used in the above example do come included with cPanel & WHM® at no additional cost.

When you combine the above information along with WP Toolkit, you'll optimize both the performance and management of your WordPress websites. Enable WP Toolkit on your cPanel server today.