

Speeding up WordPress from the backend

Backend (PHP, MySQL, nginx) tricks & tips to speed up a WordPress setup.

This is not a step-by-step tutorial and I don't have the promise that these will always work. They worked for me. Follow the links and the "more" links for detailed setup and step-by-step tutorials.

Cache in WordPress itself

Enable WordPress built-in object caching^[^1] in `wp-config.php`:

```
/* Cache */  
define ( 'WP_CACHE', true );
```

php

This, on it's own is not a really big win, but depending on the theme & the plugins is can be a good addition. If you want a really powerful addition, install an Object Cache module along with a user object cache PHP module (see below), for example APC Object Cache Backend^[^2] or APCu Object Cache Backend^[^3], if you have APC or APCu^[^4] installed. (The second on is the user cache only version of APC intended to be used with the new OpCache^[^5].)

PHP

opcode cache

Install an opcode cache plugin for PHP unless you're using PHP 5.5 which has Opcache built-in. For PHP 5.4, I'd recommend the opcache module as well, since APC is considered to be unstable for 5.4. Another choice could be Xcache[⁶].

More:

- <http://php.net/manual/en/book.opcache.php>

Store PHP sessions in memcached

By default PHP is using the disk; putting session data into memcached can speed up PHP significantly[⁷] and also solves loadbalancing issues. You'll need to install the memcache PHP extension and replace the original lines with these in php.ini:

```
session.save_handler = memcache
session.save_path = "tcp://127.0.0.1:11211"
```

ini

More information:

- <http://php.net/manual/en/memcached.sessions.php>

Put PHP-FPM tmp directory to memory

Linux systems have tmpfs[⁸] which is basically virtual filesystem in the memory - obviously faster than disk by magnitudes, but limited by the RAM.

If you have enough RAM, put the PHP-FPM temp directory into it (add this to the pool configuration file, so this is per pool):

```
php_admin_value[open_basedir] = /var/www/:/dev/shm/
php_admin_value[upload_tmp_dir] = /dev/shm/
```

ini

More on SHM:

- <http://www.cyberciti.biz/tips/what-is-devshm-and-its-practical-usage.html>

nginx

Fire up file cache

nginx has built in open file cache (caching descriptors, file headers, etc, but not the files themselves, that's either the job of the filesystem or Varnish^[^9])

Add this to `nginx.conf`:

```
##  
# file cache  
##  
open_file_cache max=2048 inactive=20s;  
open_file_cache_valid 30s;  
open_file_cache_min_uses 2;  
open_file_cache_errors on;
```

nginx:

More information:

- http://wiki.nginx.org/HttpCoreModule#open_file_cache
- <http://www.nginxtips.com/nginx-open-file-cache/>

MySQL

###Add (or add more) query cache Enable query cache in `my.cnf` if you haven't done that already. While this is not always as useful as it seems, for WordPress, it can make a significant difference.

```
## QUERY CACHE ##  
query_cache_type = 1  
query_cache_limit = 1M  
query_cache_size = 256M
```

nginx

More:

- <http://www.cyberciti.biz/tips/enable-the-query-cache-in-mysql-to-improve-performance.html>
- https://dev.mysql.com/doc/refman/5.6/en/server-system-variables.html#sysvar_query_cache_type

Use tmpfs as MySQL temporary folder (warning, this can be dangerous!)

In case you're extremely sure you have enough memory and the SHM segment is large enough (see "Put PHP-FPM tmp directory to memory" above) to handle the largest table's sorting, force MySQL to use the memory as temporary folder:

Change `tmpdir` in `my.cnf` to:

```
tmpdir = /dev/shm
```

apache

More information & discussion:

- <http://openquery.com.au/blog/experiment-mysql-tmpdir-on-tmpfs>

These tips usually speed up WordPress even if you're not using a cache plugin - this is highly recommended though, even if all these tricks are applied.** **

And remember: always test your setup after, to see if everything is working as expected!

Links

1. http://codex.wordpress.org/Class_Reference/WP_Object_Cache
2. <http://wordpress.org/plugins/apc/>
3. <http://wordpress.org/plugins/apcu/>
4. <http://pecl.php.net/package/APCu>
5. <http://php.net/manual/en/book.opcache.php>
6. <http://xcache.lighttpd.net/>
7. <http://www.dotdeb.org/2008/08/25/storing-your-php-sessions-using-memcached/>
8. <https://www.kernel.org/doc/Documentation/filesystems/tmpfs.txt>
9. <https://www.varnish-cache.org/>

Created by Peter Molnar <mail@petermolnar.net>, published at 2014-05-30 09:02 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/speeding-wordpress-backend/> , licensed under CC-BY-4.0 .