WordPress*formance:* *cacheando, que es gerundio*

Optimiza *WordPress para mejorar su* ***Rendimiento***

@JavierCasares @WPHostingTeam

¡Hola!

Soy **Javier Casares**

JavierCasares.com
WPSysAdmin.com
@JavierCasares

Representante global
WordPress Hosting Team
make.WP.org/hosting

WordPress Hosting team

make.wordpress.org/hosting

We work to improve WordPress' end-user experience across hosting environments through industry collaboration and user education. Come join us!

The team meets in the #hosting-community Slack channel each week.

IMPORTANTE

Cada infraestructura y tecnología es un mundo.

Cada servicio, *demonio* o servidor es un mundo.

Lo que diga no podrá ser utilizado en mi contra.

Que nadie se moleste si hay crítica y autocrítica al sector.

Agenda

-Qué es WPO
-Herramientas para medir
-Cómo medir
-Partes de WordPress
-Servidores y Servicios
-Caché
-Datos

Qué es WPO

Web Performance Optimization

Web Performance Optimizacion

- Web:

Hace referencia a un sitio de Internet.

- Performance:

Rendimiento y velocidad de un sitio web.

- Optimization:

Mejorar lo que hay, al máximo posible.

El guepardo (cheetah) es el animal que representa el *Web Performance Optimization*.

Es el animal terrestre más veloz, ya que alcanza una velocidad punta de 115 km/h en carreras de hasta cuatrocientos o quinientos metros.

WPO vs. SEO

El SEO, aunque es una optimización, se usa muchas veces para engañar a los motores de búsqueda.

El WPO no permite engaños. Son 100% datos, es objetivo. Es transparente, abierto, público y ético. Reduce el coste energético de Internet.

SEO onPage + WPO = éxito.

Herramientas para medir

Herramientas

Sólo hacen falta 3 herramientas:

- Tu navegador y sus herramientas para Webmasters (normalmente, en el F12).
- WebPageTest (webpagetest.org).
- PageSpeed Insights (pagespeed.web.dev)

Cómo medir

Las herramientas hay que interpretarlas

Las “cosas”

- Time to First Byte (TTFB):
El tiempo que se tarda desde que se envía la solicitud hasta que se reciben los primeros datos desde el servidor.

- DOMContentLoaded:
Como momento en el que se carga el HTML y permite trabajar con ello, sin necesidad de esperar a los CSS o scripts.

Las “cosas”

- First Contentful Paint (FCP):
Momento en el que se cargan os primeros elementos del DOM (texto, imágenes o cualquier contenido).

- Time to Interactive (TTI):
Momento en el que el sitio es completamente interactivo y usable por el usuario.

¿Hay más?

Sí, pero sin obsesionarse.

Que una herramienta te diga que algo está mal, no significa que esté mal.

Las herramientas analizan “por página” y no “por sitio”, y eso es un error.

Partes de WordPress

El ZIP

WordPress es un fichero ZIP descargable.

Incluye ficheros PHP, además de HTML, CSS, JavaScript, imágenes…

Almacena los datos en una base de datos.

La base de datos

Para funcionar, WordPress necesita una base de datos, compatible MySQL.

Este es un primer punto a optimizar.

El PHP

El lenguaje en el que está programado es PHP, que es el que realiza las operaciones más complejas y cálculos.

Este es otro punto a optimizar.

Los ficheros “estáticos”

Existen ficheros que en general no varían, como pueden ser estilos, scripts o los media que subimos nosotros mismos (imágenes, documentos…)

Estos ficheros también se pueden optimizar.

Servidores y Servicios

Qué hace falta para que WordPress funcione

@JavierCasares @WPHostingTeam

Servidor Web

Es la capa que hay entre el usuario y la web.

- Apache
- nginx
- LiteSpeed

PHP

Es lo que ejecuta y procesa los cálculos.

- PHP <= 7.3 (5.6.20 a 7.3.x) 
- **PHP = 7.4 (7.4.x)** 
- PHP >= 8.0 (8.0.x a 8.1.x) 

+info: Comparando WordPress de PHP 5.6 a PHP 8.1

SQL

Es donde se almacena la información.

-MySQL 8
-MariaDB 10.2-10.5
-MariaDB 10.6-10.7 😨 *(utf8mb3)*

Caché

El secreto de WordPress

@JavierCasares @WPHostingTeam

Caché de PHP

Caché de PHP

El código PHP de WordPress se queda precalculado.

Se suele usar PHP OPcaché.

Plugin para gestionarlo:

- WP OPcache

Caché de PHP

Para funcionar es necesario tener las extensiones de PHP-OPcache.

apt-get –y install php-opcache

Se activa en el php.ini.

opcache.enable=1

Caché de PHP

```
[opcache]
opcache.enable=1
opcache.memory_consumption=128
opcache.interned_strings_buffer=16
opcache.max_accelerated_files=10240
opcache.max_wasted_percentage=15
opcache.revalidate_freq=60
opcache.fast_shutdown=0
opcache.enable_cli=1
opcache.validate_timestamps=0
```

Caché de PHP

En Plesk, cPanel y otros viene instalado, pero no siempre activado por defecto.

Hay que activarlo por sitio.

Si lo activas, deberías tener el plugin activado y no actualizar por Git, FTP o cualquier sistema que no sea el propio WordPress.

Caché de Base de Datos

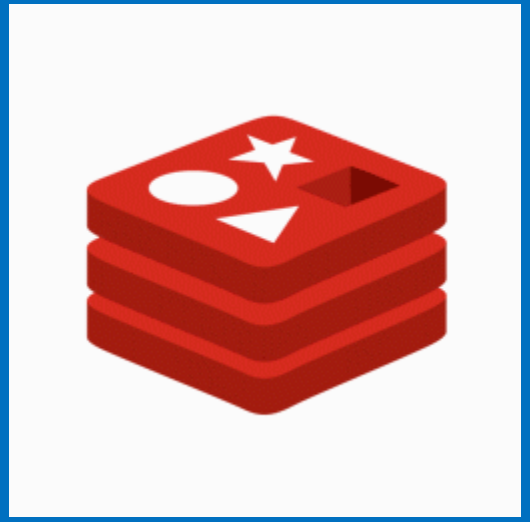
Caché de Objetos

Las consultas a la base de datos se quedan pre calculadas.

Se puede usar memcached o Redis (preferible).

Plugin para gestionarlo:

- Redis Object Cache



Caché de Objetos

Nuestro servidor ha de tener Redis activo, o la posibilidad de conectar a un Redis externo.

apt-get –y install redis-server php-redis

Para que funcione en WordPress hay que activar el ‘object-cache.php’. Lo más fácil, activar el plugin de Redis.

Caché de Objetos

Configuración del redis.conf.

maxmemory ***512m***
maxmemory-policy ***allkeys-lfu***
maxclients ***4096***

Caché de Objetos

En paneles tipo Plesk o cPanel no viene ni se suele permitir instalar de forma sencilla.

Estos paneles suelen ser para “hosting compartido”.

Generaría “problemas de seguridad” (entre comillas) si el que gestiona los sitios es el usuario final.

Caché de Web / Página

Caché de Página

El código HTML a visitar se guarda en el disco.

Se almacena, habitualmente, en el disco duro.

Plugin para gestionarlo:
- WP Super Cache

Caché de Página

Para que funcione sólo hay que activar un plugin de caché de página.

En Apache HTTPD o LiteSpeed no es necesaria configuración extra.

En nginx sí.

Caché de Página

```
set $cache_uri $request_uri;
if ($request_method = POST) { set $cache_uri 'null-cache’; }
if ($query_string != "") { set $cache_uri 'null-cache'; }
if ($request_uri ~* "(/wp-admin/|/xmlrpc\.php|/wp-(.+)\.php|/feed/|index\.php|sitemap(_index)?\.xml|[a-z0-9_-]+-sitemap([0-9]+)?\.xml)") { set $cache_uri 'null-cache’; }
if ($http_cookie ~* "comment_author|wordpress_[a-f0-9]+|wp-postpass|wordpress_logged_in") { set $cache_uri 'null-cache’; }
set $cachefile "/wp-content/cache/supercache/$http_host/$cache_uri/index.html";
if ($https ~* "on") { set $cachefile "/wp-content/cache/supercache/$http_host/$cache_uri/index-https.html"; }
```

Caché de Página

El objetivo de este plugin es:

1. El usuario visita la página. Si existe en el disco se sirve, si no, se genera y almacena.
2. Se establece el tiempo máximo de vida de la página (cuanto más, mejor). 86400s
3. Si la página cambia, WordPress (y el plugin) eliminarán de la caché la página.

Caché de Usuario / Navegador

Caché de Navegador

Le decimos al navegador que reaproveche las cosas ya visitadas.

Si te descargas una imagen, y en la siguiente página se usa esa misma imagen ¿hemos de volverla a descargar?

Se mandan cabeceras de Apache / nginx con las reglas de uso.

Caché de Navegador

No viene por defecto configurado, es un trabajo manual.

¿Qué es lo que vamos a cachear y que es bastante inmutable? Imágenes, CSS y JavaScript (tiene truco).

Es importante usar un plugin de “optimización”.

Caché de Navegador

1. Los CSS y JavaScript con parámetros, no se cachean:
   <link href='/wp-includes/css/dashicons.min.css**?ver=5.9.1**'>
2. Hemos de convertirlo a algo “único”.
   <link href="/wp-content/cache/css/css_166867d0f9ac1fb5a.css">

Caché de Navegador

Ejemplo de caché en el navegador (nginx):

```
location ~ \.(?:gif|ico|jpeg|jpg|png|svg|webp)$" {
  etag on;
  expires max;
  add_header Cache-Control "public, no-transform, inmutable";
}
```

Caché de Navegador

```
-/favicon\.(ico|png)
-.(?:bmp|gif|ico|jpeg|jpg|png|svg|svgz|webp)$
-.(?:mid|midi|mp3|ogg|wav)$
-.(?:mp4|ogv|webm)$
-.(?:doc|docx|ppt|pptx|rtf|xls|xlsx)$
-.(?:bz2|gz|rar|tar|tgz|zip)$
-.(?:css|js)$
-.(?:atom|rss)$
-.(?:json|xml)$
-.(?:eot|otf|ttf|woff|woff2)$
```

Otras cachés

CDN

Es una caché externa a WordPress.

Sólo deben cachearse los estáticos, por ejemplo, creando un *estaticos.example.com*.

WordPress lo permite hacer de forma nativa.

https://www.wpsysadmin.com/blog/2019/08/25/como-configurar-bien-una-cdn-en-wordpress/

CDN

¿Qué mirar? “options” + cambios

upload_url_path -> https://estaticos.example.com

```
wp search-replace 'https://www.example.com/wp-
content/uploads/'
'https://estaticos.example.com/'
```

Caché de objetos, en disco

Podemos usar el plugin Docket Cache.

```
cd wp-content/
mount -t tmpfs -o size=512m tmpfs ./cache/docket-cache
```

https://www.wpsysadmin.com/rendimiento/#cache-de-objetos-en-disco

Caché de traducciones

Podemos usar el mu-plugin WordPress Translation Cache.

A partir de ese momento, los ficheros .PO/.MO de las traducciones tanto de WordPress como de themes y plugins comenzarán a ser cacheados.

NOTA: Se cachea en sistemas de OPcache.

https://www.wpsysadmin.com/blog/2021/05/19/cache-traducciones/

Datos

Probemos herramientas

clouding.io VPS: 1 CPU, 2 GB RAM y 10 GB NMVe

Base

-Servidor recién instalado
-WordPress recién instalado
-Creación de 2.500 entradas con FakerPress
-Muestra 1.000 entradas en la página principal
-Contenidos completos, incluidas imágenes y otros elementos.

Datos (primera visita)

	First Byte	TBT	Time	Requests	Total KBytes	URL
Sin Caché	1,787	7,065	12,831	607	73,006	WPT
Navegador	1,668	5,717	12,379	606	73,002	WPT
PHP	**0,757**	5,746	11,652	606	73,002	WPT
Objetos	0,784	6,930	12,037	606	73,002	WPT
Página	**0,107**	4,651	11,794	606	73,002	WPT

Datos (segunda visita)

	First Byte	TBT	Time	Requests	Total KBytes	URL
Sin Caché	0,602	5,173	8,215	78	11,415	WPT
Navegador	0,609	5,041	8,402	**2**	**1,134**	WPT
PHP	0,590	5,100	8,731	2	1,134	WPT
Objetos	0,786	4,799	8,879	2	1,134	WPT
Página	**0,079**	5,267	8,505	2	1,134	WPT

¡Gracias!

ahora, consultorio gratis

Encuéntrame en: JavierCasares.com

Más información en: WPSysAdmin.com

WordPress Hosting: make.wp.org/hosting

@JavierCasares @WPHostingTeam