



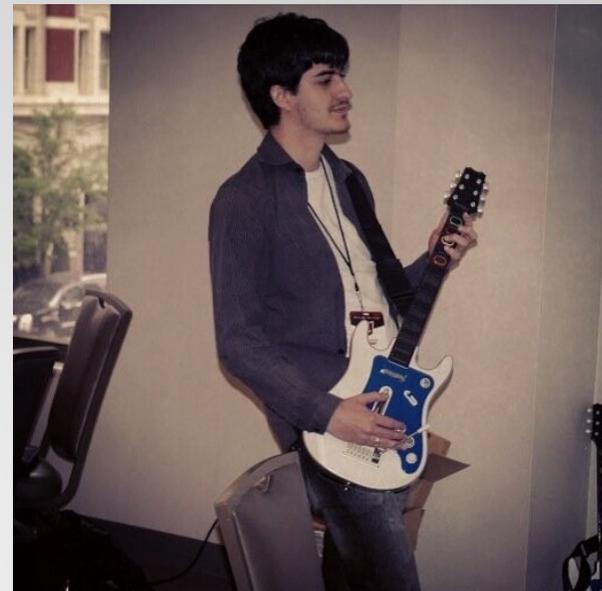
WORDCAMP
SÃO PAULO
2014

MELHORANDO O DESEMPENHO DO SEU WORDPRESS

JULIAN FERNANDES • TIAGO HILLEBRANDT

Julian Fernandes

- Technical Support Team Lead na Copyblogger Media, LLC
- Trabalha diariamente com uma estrutura com mais de 10.000 sites **WordPress**
- Blogueiro viciado em performance e benchmarks



 [julianfernandes](#)

 [julianfern](#)

 julian@copyblogger.com

 [julian_fern](#)

 [julianfernandes.com](#)



WORDCAMP
SÃO PAULO
2014



Tiago Hillebrandt

- **DevOps Engineer** na **Copyblogger Media, LLC**
- Trabalha diariamente com uma estrutura contendo mais de 10.000 sites **WordPress**
- **Colaborador** de diversas comunidades de **software livre**, como Ubuntu, WordPress, Bash, dentre outras
- Doteiro e violonista amador nas horas vagas



tiagohillebrandt



tiagoscd



tiagohillebrandt.dev



tiagohillebrandt.eti.br



tiago@tiagohillebrandt.com



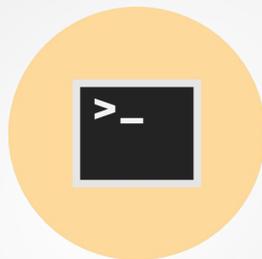
WORDCAMP
SÃO PAULO
2014

Introdução

Nesta palestra vamos apresentar algumas técnicas para melhorar o desempenho do seu site WordPress em três lados:



DATABASE-SIDE



LINUX-SIDE



PHP-SIDE



SPDY: HTTPS eficiente

- Um protocolo de rede criado pelo Google com o objetivo de deixar a web mais rápida
- Limpa e compacta os cabeçalhos de requisição e resposta, o que diminui o uso de banda e o tempo de carregamento das páginas
- Possibilita o download de múltiplos elementos com uma única requisição
- Com a utilização do chamado *server push* é possível enviar um arquivo mesmo antes dele ser requisitado, como o style.css do seu tema



SPDY: HTTPS eficiente

```
julianfernandes — julianfernandes.com + (-) - VIM — vim — bash
1 server {
2     listen          443 ssl spdy;
3     ssl              on;
4     ssl_certificate  /etc/nginx/ssl/julianfernandes.com/julianfernandes.com.crt-combined;
5     ssl_certificate_key /etc/nginx/ssl/julianfernandes.com/julianfernandes.com.key;
6     keepalive_timeout 70;
7
8     spdy_headers_comp 9;
9
10    ssl_session_cache shared:SSL:20m;
11    ssl_session_timeout 10m;
12
13    ssl_prefer_server_ciphers on;
14    ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
15    ssl_ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AES
16    GCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS;
17
18    ssl_stapling on;
19    ssl_stapling_verify on;
20    ssl_stapling_resolver http://rapidssl-ocsp.geotrust.com;
21    resolver 8.8.8.8 8.8.4.4 valid=300s;
22    resolver_timeout 10s;
23
24    add_header Strict-Transport-Security "max-age=31536000";
25
26    access_log /var/log/nginx/julianfernandes.com.access.log vhost;
```

16,1 Top



W3 Total Cache

- Excelente plugin de cache para **WordPress**
- Traz diversas funções de cache, CDN e minify, focadas em melhorar o desempenho do seu site
- Mais de 3,8 milhões de *downloads* via WordPress.org
- Desenvolvido pela **W3 EDGE**



W3 Total Cache

Por que devemos usá-lo?

- **Reduz o uso de recursos** do servidor
- Se bem configurado, seu site poderá **carregar até 10x mais rápido**
- **SEO**: a velocidade de carregamento do site influencia no posicionamento do site no Google e outros buscadores
- Pode **reduzir o uso de largura de banda** devido a compressão de conteúdo

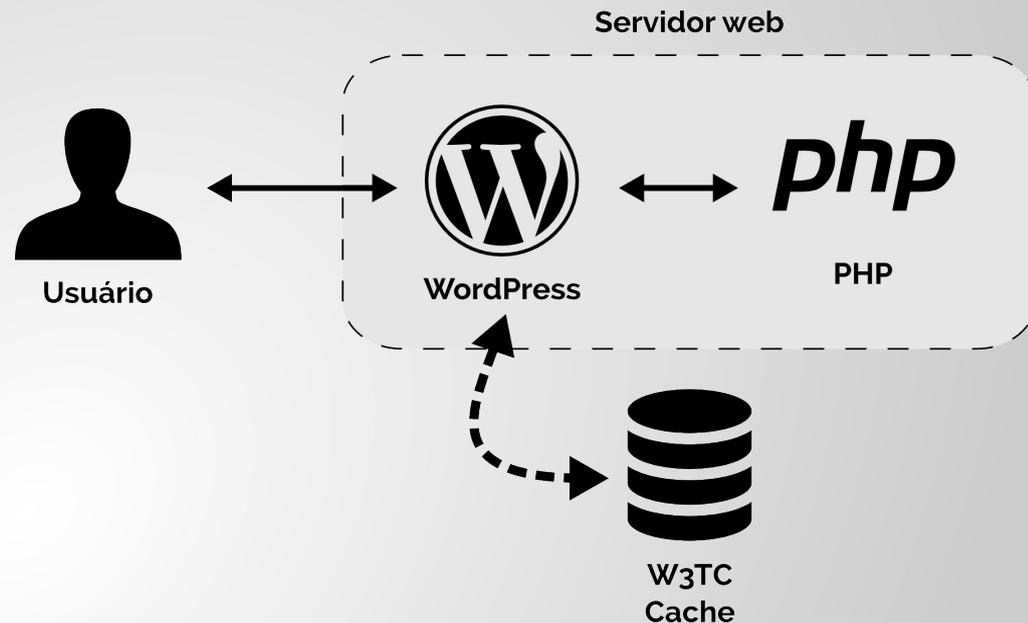


WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Page Cache: o que é?

- Cache baseado em URLs
- No primeiro acesso a determinada URL, um arquivo estático com o conteúdo é criado dentro do cache
- Os próximos acessos para mesma URL são carregados do cache, reduzindo o uso de recursos do PHP e MySQL



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Page Cache: Disk

- **Disk Basic:** carrega os arquivos do *cache* por meio de scripts PHP
- **Disk Enhanced:** carrega os arquivos do cache sem quaisquer scripts PHP, tende a ser mais efetivo que o *Basic*



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Page Cache: OpCode

- **Alternative PHP Cache (APC):** excelente ferramenta de OpCode. No entanto foi descontinuada desde o PHP 5.5, em virtude do Zend Optimizer
- **eAccelerator:** o desenvolvimento do eAccelerator está parado há mais de 2 anos, portanto não utilize esta opção
- **XCache:** excelente ferramenta de OpCode. O projeto existe faz mais de 8 anos, no entanto é mantido somente por uma pessoa
- **WinCache:** opção disponível para servidores IIS rodando PHP. Para habilitar este método, é necessário instalar a extensão *WinCache* no servidor



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Page Cache: Memcached

- Armazena os dados na memória, mantendo o nome, valor, tempo de expiração, tamanho e algumas *flags* específicas
- Eficiente método para cache de objetos
- Ao utilizar esta opção, provavelmente você aumentará o uso da memória do seu servidor



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Page Cache: qual método devo escolher?

- Os métodos **OpCode** e **Memcached** tendem a ser mais rápidos, pois o *cache* carregará diretamente da memória. Portanto, são uma boa opção se você dispor de bastante memória para dedicar à eles
- Já os métodos **Disk**, apesar de serem um pouco mais lentos, podem se tornar mais viáveis em servidores compartilhados ou VPSs com pouca memória disponível, ou ainda caso queira poupar memória para processar outras informações
- Se ainda assim está na dúvida de qual escolher, você pode testar a performance do seu site método-por-método usando as dicas que daremos em breve



W3 Total Cache

Browser Cache

- É o cache presente nos navegadores web
- No W3TC é possível determinar, por exemplo, quanto tempo as imagens, arquivos JS, CSS, HTML, dentre outros, ficarão cacheados no navegador do usuário
- Permite a compressão dos cabeçalhos HTTP via gzip



W3 Total Cache

Object Cache

- O WordPress possui uma classe de *object cache*, no entanto ela não é persistente
- Em outras palavras, a informação carregada é mantida na memória somente até o fim da requisição atual



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Object Cache

- No W3TC, o *object cache* armazena uma cópia do objeto, seus atributos e valores em formato JSON
- Além disso, a informação passa a ser mantida de forma permanente, pois não é perdida ao término da requisição
- É possível determinar um intervalo para limpeza do cache, bem como uma lista de objetos que não devem ser cacheados



W3 Total Cache

Database Cache

- Cache de *queries* específicas do MySQL
- Permite determinar o tempo de *caching*
- Também é possível determinar termos específicos que, se presentes no SQL, não farão o cache da *query*
- Armazena a tabela, campos e valores em formato JSON



W3 Total Cache

Fragment Cache

- Essa tecnologia permite fazer *caching* de partes do site por meio de *hooks* específicos do tema
- Atualmente somente os temas usando **Genesis Framework** são compatíveis com essa tecnologia no W3TC



WORDCAMP
SÃO PAULO
2014

W3 Total Cache

Minify

- Remove caracteres “desnecessários”, acarretando na redução do tamanho do arquivo
- Deve ser utilizado com cautela
- Permite a seleção dos arquivos que devem minificados e combinados



W3 Total Cache

CDN

- Indicado para quem recebe tráfego de diversas localizações (geograficamente distribuído)
- Reduz o uso de largura de banda do servidor onde seu site está hospedado
- Serviços de CDN recomendados:
 - Amazon CloudFront
2 datacenters no Brasil, custo variável por mês (on-demand)
 - CloudFlare
1 datacenter no Brasil, custo fixo por mês



WORDCAMP
SÃO PAULO
2014

StudioPress Accelerator

- Solução de page caching, resultando em mais velocidade para seu site e menos carga no seu servidor
- Também indicado para quem deseja reduzir o uso de largura de banda do servidor onde o site está hospedado
- É possível determinar o continente onde o serviço estará localizado
- Proteção contra ataques de força bruta
- Baixo custo mensal para manter o serviço

Autoloaded Options

O que é a Options API?

- Atualmente a **Options API** possibilita o armazenamento de dados dentro da tabela **wp_options** do WordPress

wp_options

- option_id
- option_name
- option_value
- autoload

- Todas as opções com *autoload="yes"* são carregadas automaticamente para dentro do *object cache*, por meio da função `wp_load_alloptions()`, a cada *page load*



WORDCAMP
SÃO PAULO
2014

Autoloaded Options

Como as *options* podem ser um problema?

- Quando o **número de autoloaded options é alto**, isso influencia diretamente na performance do site, tendo em vista que elas são carregadas automaticamente a cada requisição
- Quando o **tamanho das autoloaded options é grande**, isso implica no desempenho, pois muitos dados terão de ser carregados durante a consulta ao MySQL



WORDCAMP
SÃO PAULO
2014

Autoloaded Options

Como detectar se elas são um problema?

- Execute a seguinte query em seu banco de dados:

```
mysql> SELECT COUNT(*) AS 'Total', ROUND(SUM(LENGTH(option_value)) / POWER(1024, 2), 2) AS 'Tamanho (Mb)' FROM wp_options WHERE autoload='yes';
+-----+-----+
| Total | Tamanho (Mb) |
+-----+-----+
| 234 | 0.27 |
+-----+-----+
1 row in set (0.00 sec)

mysql> █
```

- E então use esta tabela como referência:

Total < 1000 && Tamanho < 0,80Mb => OK

Total < 1000 && Tamanho >= 0,80Mb => Executar *query 1*

Total > 1000 && Tamanho < 1,00Mb => Executar *query 2*

Total > 1000 && Tamanho >= 1,00Mb => Executar as duas



WORDCAMP
SÃO PAULO
2014

Autoloaded Options

Query 1

```
mysql> SELECT option_name, ROUND(LENGTH(option_value) / POWER(1024,2), 2) AS tamanho FROM wp_options WHERE autoload = 'yes' ORDER BY tamanho DESC LIMIT 0,15;
```

option_name	tamanho
wpseo_titles	0.97
rewrite_rules	0.01
siteurl	0.00
home	0.00
blogname	0.00
blogdescription	0.00
users_can_register	0.00
admin_email	0.00
start_of_week	0.00
use_balanceTags	0.00
use_smilies	0.00
require_name_email	0.00
comments_notify	0.00
posts_per_rss	0.00
rss_use_excerpt	0.00

15 rows in set (0.00 sec)



WORDCAMP
SÃO PAULO
2014

Autoloaded Options

Query 2

```
mysql>
mysql> SELECT SUBSTRING(option_name, 1, 8) AS prefixo, COUNT(*) AS total FROM wp_options GROUP BY SUBSTRING(option_name, 1, 6) ORDER BY total DESC LIMIT 0,15;
```

prefixo	total
_transie	25
default_	9
_site_tr	9
widget_a	8
comments	7
wpseo_in	6
mailserv	4
theme_mo	3
image_de	3
uploads_	3
thumbnai	3
posts_pe	2
recently	2
thread_c	2
close_co	2

```
15 rows in set (0.00 sec)
```



WORDCAMP
SÃO PAULO
2014

Transients API

O que é?

- Possibilita armazenar dados em cache na base de dados
- Muito similar a **Options API**, usando inclusive a própria **wp_options** para armazenar os dados
- Possui nome personalizado, tempo de expiração e é carregado automaticamente
- Se bem usados, podem acarretar em grandes melhorias de performance



WORDCAMP
SÃO PAULO
2014

Transients API

Funções da API

- Funções disponíveis no WordPress:
 - `set_transient()`
 - `get_transient()`
 - `delete_transient()`
- Funções para multisites:
 - `set_site_transient()`
 - `get_site_transient()`
 - `delete_site_transient()`



Transients API

Quando os transients passam a ser um problema?

- Quando o **número de transients é muito alto**, a performance do site tende a ser afetada devido ao grande número de requisições ao banco para carregá-los
- Quando o **tamanho dos dados armazenados nos transients** é muito grande, as requisições ao banco tendem a levar certo tempo para processar, prejudicando a performance do site



WORDCAMP
SÃO PAULO
2014

Transients API

Como determinar se estão ajudando ou prejudicando meu site?

- Você pode verificar isso com esta *query*:

```
mysql> SELECT COUNT(*) AS total FROM wp_options WHERE option_name LIKE '%_transient%';
+-----+
| total |
+-----+
|    34 |
+-----+
1 row in set (0.00 sec)
```



WORDCAMP
SÃO PAULO
2014

Transients API

Limpendo os transients

- É possível limpar todos os transients do banco de dados rodando a seguinte *query*:

```
mysql>  
mysql> DELETE FROM wp_options WHERE option_name LIKE '%_transient%';  
Query OK, 34 rows affected (0.01 sec)
```



WORDCAMP
SÃO PAULO
2014

WordPress Scheduled Events

O que é?

São ações programadas para serem executadas automaticamente pelo WordPress.

Exemplos:

- Posts agendados
- Exclusão de *transients* expirados da `wp_options`
- Envio de emails de renovação de assinaturas
- Criação de backups do site
- Verificação de *updates* para temas e plugins



WORDCAMP
SÃO PAULO
2014

WordPress Scheduled Events

Como funciona?

- Por padrão ele rodará apenas se sua página tiver acessos, um método não muito confiável
- Carrega em todos os acessos, e caso um evento agendado trave por algum motivo, o acesso do seu visitante pode ficar consideravelmente lento
- Pode ser implementado no crontab do seu servidor, um método muito confiável



WORDCAMP
SÃO PAULO
2014

WordPress Scheduled Events

Como implementar a melhor solução?

Para ativar o *crontab* precisaremos desativar o cron padrão do WordPress. Isso pode ser feito adicionando a seguinte linha ao *wp-config.php*:

- `define('DISABLE_WP_CRON', 'true');`



WORDCAMP
SÃO PAULO
2014

WordPress Scheduled Events

Como implementar a melhor solução?

```

juliafernandes - wp-config.php = (/var/www/juliafernandes.com) - VIM - ssh - bash
<?php

/** Random modifications to WordPress */
define('WP_CACHE', true);
define('W3TC_PRO', true);
define('FORCE_SSL_LOGIN', true);
define('FORCE_SSL_ADMIN', true);
define('WP_SITEURL', 'https://www.juliafernandes.com');
define('WP_HOME', 'https://www.juliafernandes.com');
define('COOKIE_DOMAIN', 'www.juliafernandes.com');
define('WP_POST_REVISIONS', false);
define('MEDIA_TRASH', false);

define('DISABLE_WP_CRON', 'true');

/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, WordPress Language, and ABSPATH. You can find more information
 * by visiting {@link http://codex.wordpress.org/Editing_wp-config.php Editing
 * wp-config.php} Codex page. You can get the MySQL settings from your web host.
 */
"wp-config.php" [readOnly] 104L, 3807C                               14,1                               Top

```



WORDCAMP
SÃO PAULO
2014

WordPress Scheduled Events

Como implementar a melhor solução?

Execute o seguinte comando no seu servidor, substituindo `dominio.com` pelo domínio do seu site:

- ```
echo “*/10 * * * * www-data wget -o /dev/null -O /dev/null
http://dominio.com/wp-cron.php” > /etc/cron.d/wordpress
```

O comando acima irá executar os eventos agendados do Wordpress a cada dez minutos via *crontab*



WORDCAMP  
SÃO PAULO  
2014

# FV Gravatar Cache

- Apesar de ser um ótimo serviço, o Gravatar acaba tendo sérios problemas de performance por vezes
- Gravatar é salvo e hospedado no seu servidor
- Permite total controle do cache de navegador, permitindo a integração com o W3 Total Cache
- Cache criado de duas formas, no envio do comentário e via cron job



# Efficient Related Posts

- Posts relacionados configurados no dashboard do WordPress
- Lista de itens ao invés de miniaturas
- WordPress Related Posts by Zemanta é uma alternativa com miniaturas

---

## Related

Bookmarked: This Weeks Highlights  
of the Web  
In "Useful Links"

Das passende Related Posts Word-  
Press-Plugin finden  
In "Plugins"

Genericons: A free GPL-licensed  
Icon Font For WordPress Projects  
In "WordPress"



WORDCAMP  
SÃO PAULO  
2014

# Genesis Simple Share

- Elimina a dependência dos lentos servidores do Facebook, Twitter e Google
- Hospeda tudo no seu servidor e permite total controle do cache de navegador

## Why Copyblogger Is Killing Its Facebook Page

by ERIKA NAPOLETANO



WORDCAMP  
SÃO PAULO  
2014

# Pingdom para otimização front-end

- Ótima ferramenta para descobrir problemas de performance front-end e corrigi-los
- Mantem histórico dos testes para análise de melhorias



The screenshot shows a Pingdom test report for the website <http://websynthesis.com>. The test was conducted on October 17 at 22:16:30. The report includes a thumbnail of the website, a performance score of 89/100, 61 requests, a load time of 767ms, and a page size of 1.0MB. A comparison statement indicates the website is faster than 93% of all tested websites. Social sharing buttons for Twitter, Facebook, and Email are present, along with a 'DOWNLOAD HAR' button.

| Perf. grade | Requests | Load time | Page size |
|-------------|----------|-----------|-----------|
| 89/100      | 61       | 767 ms    | 1.0MB     |

Your website is faster than 93% of all tested websites

DOWNLOAD HAR | Tweet | Post to Timeline | Email



WORDCAMP  
SÃO PAULO  
2014

# Pingdom para otimização front-end

|     |                                                                                                      |          |  |
|-----|------------------------------------------------------------------------------------------------------|----------|--|
| JS  | <a href="#">synthesis-scripts.js?ver=3.9.1</a><br>websynthesis.com/wsa/wp-content/theme...           | 927 B    |  |
| JS  | <a href="#">backstretch.js?ver=3.9.1</a><br>websynthesis.com/wsa/wp-content/theme...                 | 2.0 kB   |  |
| JS  | <a href="#">synthesis-home.js?ver=3.9.1</a><br>websynthesis.com/wsa/wp-content/theme...              | 823 B    |  |
| JS  | <a href="#">vil7fkv.js</a><br>use.typekit.net/                                                       | 9.1 kB   |  |
| JS  | <a href="#">conversion.js</a><br>www.googleadservices.com/pagead/                                    | 4.1 kB   |  |
| CSS | <a href="#">d?3bb2a6e53c9684ffdc9a9ef61a5b2a62506...</a><br>use.typekit.net/c/0e969c/1w;museo-san... | 202.2 kB |  |
| JS  | <a href="#">analytics.js</a><br>www.google-analytics.com/                                            | 11.3 kB  |  |
| →   | <a href="http://googleads.g.doubleclick.net/pa...">http://googleads.g.doubleclick.net/pa...</a>      | 0 B      |  |



WORDCAMP  
SÃO PAULO  
2014

# Load testing com Blitz.io

- Ferramenta paga para load testing de websites, servidores e aplicações web
- Perfeita para descobrir gargalos e otimizar servidores web, além do planejamento de capacidade



WORDCAMP  
SÃO PAULO  
2014

# Load testing com Blitz.io

## ANALYSIS

This rush generated **56,866** successful hits in **60 seconds** and we transferred **284.45 MB** of data in and out of your app. The average hit rate of **947.77/second** translates to about **81,887,040** hits/day.

The average response time was **20 ms**.

You've got bigger problems, though: **1.94%** of the users during this **rush** experienced timeouts or errors!

■ **HITS 98.06%** (56866)  
■ **ERRORS 1.94%** (1127)  
■ **TIMEOUTS 0.00%** (0)

### RESPONSE TIMES

FASTEST: **16** MS

SLOWEST: **51** MS

AVERAGE: **20** MS

### TEST CONFIGURATION

REGION: **VIRGINIA**

DURATION: **60** SECONDS

LOAD: **999-1000** USERS

### OTHER STATS

AVG. HITS: **948** /SEC

DATA TRANSFERRED:

**284.45** MB



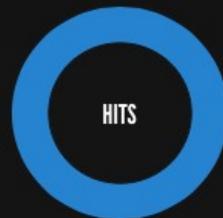
WORDCAMP  
SÃO PAULO  
2014

# Load testing com Blitz.io

## HITS

This rush generated **56,866** successful hits. The number of hits includes all the responses listed below. For example, if you only want **HTTP 200 OK** responses to count as Hits, then you can specify `--status 200` in your rush.

| CODE | TYPE | DESCRIPTION | AMOUNT |
|------|------|-------------|--------|
| 200  | HTTP | OK          | 56866  |



■ HTTP 200 OK **100%** (56866)

## ERRORS

The first error happened at **2.5** seconds into the test when the number of concurrent users was at **999**. Errors are usually caused by resource exhaustion issues, like running out of file descriptors or the connection pool size being too small (for SQL databases).

| CODE | TYPE | DESCRIPTION        | AMOUNT |
|------|------|--------------------|--------|
| 23   | TCP  | Connection timeout | 1127   |



■ CONNECTION TIMEOUT **100%** (1127)



WORDCAMP  
SÃO PAULO  
2014

# Outras estratégias de performance

- Lazy loading
- Otimização de imagens
- Genesis Framework ou otimização de temas
- Desenvolvimento de temas com otimização front-end em mente



WORDCAMP  
SÃO PAULO  
2014



WORDCAMP  
SÃO PAULO  
2014

Dúvidas?