

IMPROVING WORDPRESS PERFORMANCE

Xdebug and PHP profiling

WordCamp Athens 2017

Otto Kekäläinen

Seravo.com

@ottokekalainen

seravo



Otto Kekäläinen

- Linux and open source advocate
- Contributed to WordPress Core, translations, Linux, Docker, Nginx, Redis, MariaDB...
- CEO, sysadmin and developer at [Seravo.com](https://seravo.com) – WordPress hosting and upkeep

Enterprise grade hosting and upkeep for WordPress

seravo

Let's
Encrypt

 MariaDB

NGINX

php7

 redis

 WP
CLI

WORDPRESS SUCCESS FACTORS

1. Easy to use
2. Easy to extend



COMMON CHALLENGES

1. Security
2. Speed



A WEB FULL OF WRONG ADVICE

Most of the guides and tutorials on security and speed lack *evidence*.



I've done mythbusting at many WordCamps with

WordPress Security 101

seravo.com/wordpress-security-101/

A long-exposure photograph of a highway at night. The image shows multiple lanes of traffic with light trails from cars, creating a sense of motion. Streetlights are visible along the road, and the overall scene is dark with bright highlights from the lights.

Now it is time to make sense out of

WordPress Speed

Performance

A background image of a desk with a laptop, a potted plant, and a ukulele. The text is overlaid on the image in four white boxes.

STEP 1: Measure

STEP 2: Optimize

STEP 3: Validate

STEP 4: Rinse and repeat



STEP 1: Measure

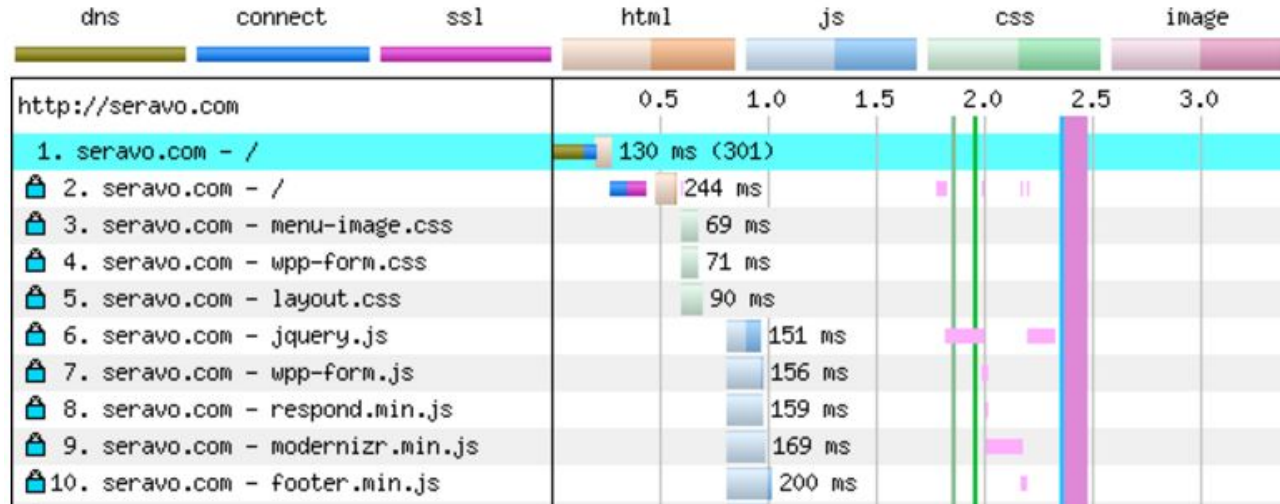
Find out your **baseline** to make sure your optimizations later at least do not worsen the performance!

FULL PAGE LOAD

- Online tools
 - WebPageTest.org
 - GTMetrix.com
 - Pingdom Tools
 - Yellow Lab Tools
 - Pagelocity
 - KeyCDN Performance Test
 - ...
- Visualize:
 - HTML load time
 - CSS rendering
 - JavaScript loading
 - image files download
 - ...

HOW FAST IS WORDPRESS?

= How fast PHP code generates the HTML



= HTTP request time

CURL

```
ssh example.com
```

```
curl -s -o /dev/null \  
      -w "%{time_total}\n" https://example.com/
```

0,235

<https://curl.haxx.se/docs/manpage.html#-w>

CURL WITH NO CACHE

```
curl -s -o /dev/null \  
-w "%{time_total}\n" \  
-H "Pragma: no-cache" https://example.com/
```

<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>

CURL LOOP TO DETECT VARIATION

```
export LC_NUMERIC=C
for i in {1..20}
do
    curl -so /dev/null -w "%{time_total}\n" http://localhost/
done | awk '{ sum += $1; n++; print $1 } END { if (n > 0) print
"AVG: " sum / n; }'
```

0.209

0.107

0.152

AVG: 0.1378

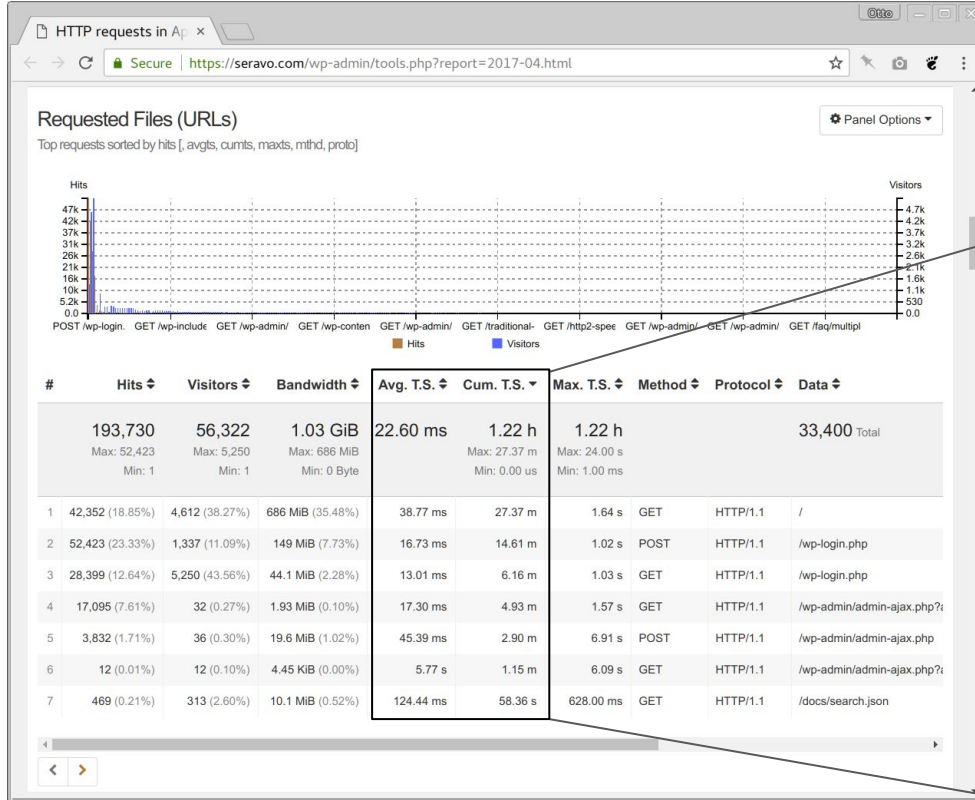
LOG HTTP REQUEST TIME

```
[29/May/2017:10:02:45 +0300] "POST /wp-admin/admin-ajax.php
HTTP/1.1" 200 58 "Mozilla/5.0 (KHTML, like Gecko)
Chrome/58.0.3029.110 Safari/537.36" - - 0.028
```

nginx.conf


```
log_format extensive '$host '
    '$remote_addr - $remote_user [$time_local] '
    '"$request" $status $body_bytes_sent '
    '"$http_referer" "$http_user_agent" '
    '$upstream_cache_status - '
    '$request_time';
```


ANALYZE WITH GOACCESS



- Average load time
- Cumulative load time = sum of all loads

| Avg. T.S. ↕ | Cum. T.S. ▼ |
|--------------|-------------|
| 22.60 ms | 1.22 h |
| Max: 27.37 m | |
| Min: 0.00 us | |
| 38.77 ms | 27.37 m |
| 16.73 ms | 14.61 m |
| 13.01 ms | 6.16 m |

A photograph of two dogs playing on a sandy beach. The dog on the left is a brindle-colored dog, possibly a Weimaraner, with its mouth open and looking towards the right. The dog on the right is a dark-colored dog, possibly a Weimaraner, with its mouth open and looking towards the left. The background shows a blurred beach and ocean under a blue sky.

STEP 2: Optimize

Find and solve the bottleneck

QUICK AND DIRTY: WP-CLI LOOP

```
for p in $(wp plugin list --fields=name --status=active)
do
    echo $p
    wp plugin deactivate $p
    for i in {1..5}
    do
        curl -so /dev/null -w "%{time_total}\n" \
            -H "Pragma: no-cache" http://localhost/
    done
    wp plugin activate $p
done
```

QUICK AND DIRTY: WP-CLI LOOP

- Baseline ~550 ms
- Deactivating wp-to-twitter or polylang does not have an effect
- When deactivating advanced-custom-fields-pro load times drop to ~65 ms



```
Plugin 'wp-to-twitter' deactivated.  
Success: Deactivated 1 of 1 plugins.  
0.551  
0.549  
0.508  
0.527  
0.529  
Plugin 'wp-to-twitter' activated.  
Success: Activated 1 of 1 plugins.  
advanced-custom-fields-pro  
Plugin 'advanced-custom-fields-pro' deactivated.  
Success: Deactivated 1 of 1 plugins.  
0.066  
0.063  
0.063  
0.064  
0.063  
Plugin 'advanced-custom-fields-pro' activated.  
Success: Activated 1 of 1 plugins.  
polylang  
Plugin 'polylang' deactivated.  
Success: Deactivated 1 of 1 plugins.  
0.505  
0.499  
0.509
```


A LITTLE MORE DEPTH: DEBUG BAR

The screenshot shows the WordPress Debug Bar interface. At the top, it displays site information: 'wceu2017-demo', PHP version '7.0.16-4+deb.sury.org~trusty+1', MySQL version '5.5.5', and Memory Usage '9 044 984 bytes'. A 'Debug' button is visible in the top right corner.

Below the site information, a summary of actions is shown:

- UNIQUE ACTIONS: 515
- TOTAL ACTIONS: 4551
- ACTION: 16
- SLOWEST ACTION: 543.22ms

A dropdown menu is open, listing various categories: Queries, WP Query, Request, Object Cache, Cron, Transients, and Slow Actions (highlighted in yellow).

The 'Slow Actions' section is expanded, showing a table of actions:

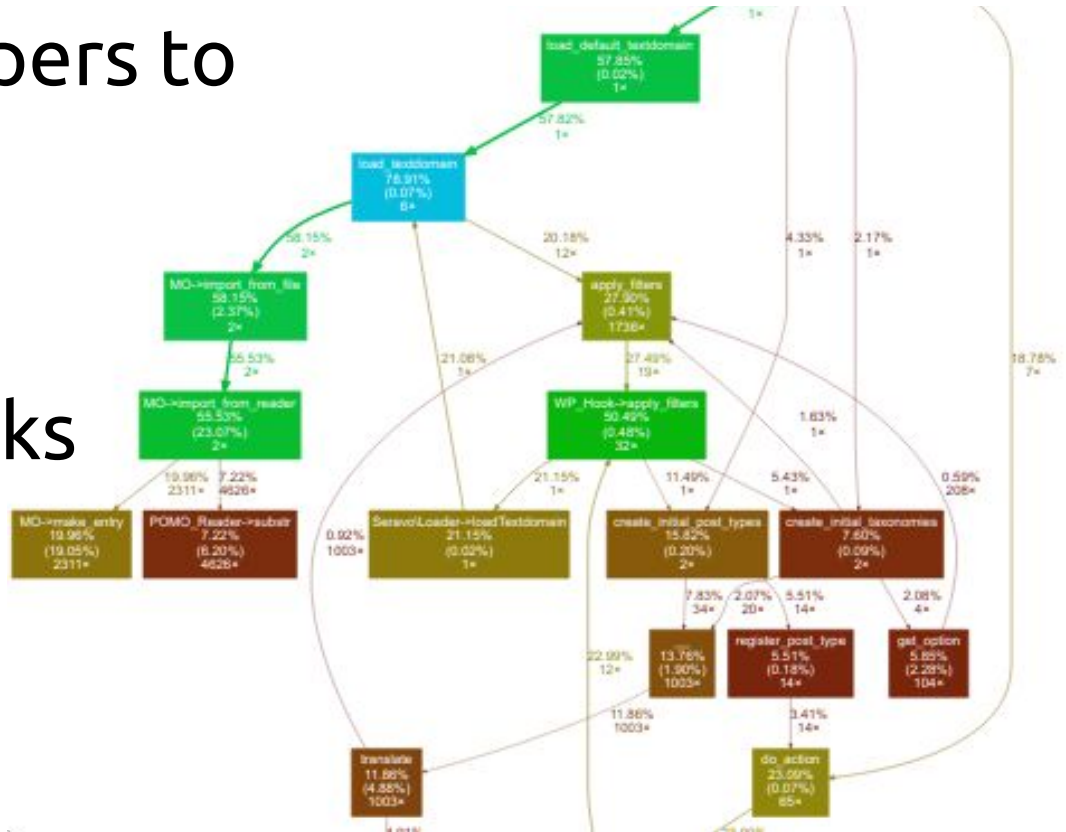
| Action or Filter | Callbacks | Calls | Per Call | Total |
|------------------|-----------|-------|----------|----------|
| wp_footer | 6 | 1 | 543.22ms | 543.22ms |
| admin_bar_menu | 17 | 1 | 159.98ms | 159.98ms |
| init | 19 | 1 | 144.05ms | 144.05ms |
| wp_head | 33 | 1 | 102.62ms | 102.62ms |



THE PROFESSIONAL WAY: XDEBUG

A tool for developers to

- analyze PHP execution
- find bottle necks
- xdebug.org



XDEBUG INSTALLATION

```
$ sudo apt-get install php-xdebug
```

```
$ nano /etc/php/fpm/conf.d/20-xdebug.ini
```

```
; Enable Xdebug
```

```
zend_extension=xdebug.so
```

```
; Enable php profiling with get param XDEBUG_PROFILE=1
```

```
xdebug.profiler_output_dir=/tmp
```

```
xdebug.profiler_output_name=cachegrind.out.%t.%p
```

```
xdebug.profiler_enable_trigger=1
```

```
$ sudo service restart php-fpm
```

PROFILING RUN OF WORDPRESS FRONT PAGE

```
/tmp $ curl -I http://localhost/?XDEBUG_PROFILE=1 \  
      -w "%{time_total}\n"
```

```
0.611
```

```
/tmp $ ll -h
```

```
11M  cachegrind.out.1455198789.5601
```

```
/tmp $ head cachegrind.out.1455198789.5601
```

```
version: 1
```

```
creator: xdebug 2.2.3
```

```
cmd: /data/wordpress/htdocs/index.php
```

```
part: 1
```

```
positions: line
```

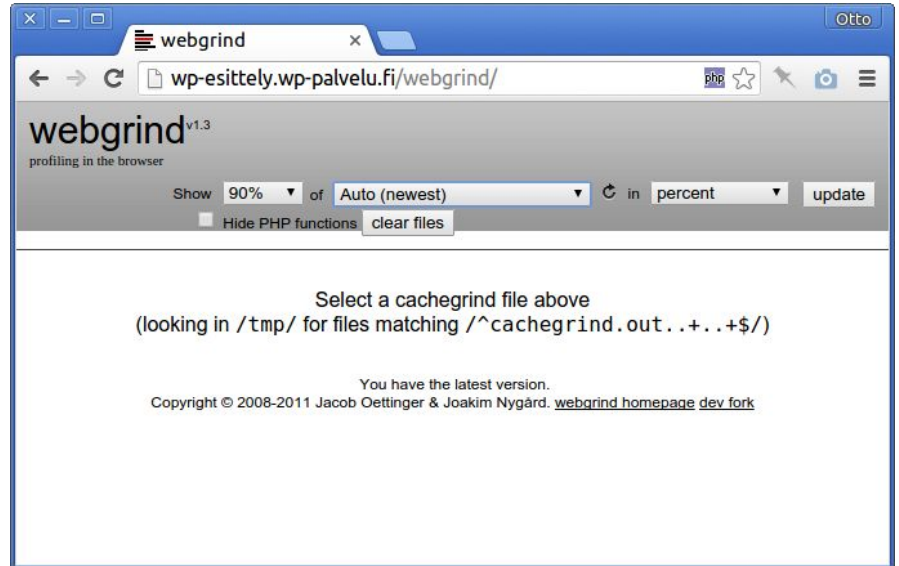
```
...
```


WEBGRIND INSTALLATION

```
$ cd /data/wordpress/htdocs
```

```
$ git clone https://github.com/jokkedk/webgrind
```

```
$ sudo apt-get install graphviz
```



PREINSTALLED IN VVV AND SERAVO VAGRANT

```
laptop$ vagrant ssh
```

```
vagrant$ xdebug_on      # Varying Vagrant Vagrants
```

```
vagrant$ wp-xdebug-on # Seravo Vagrant
```

Go profiling!



WEBGRIND UI EXPLAINED

webgrind^{v1.3}
profiling in the browser

Show **98%** of **index.php (cachegrind.out.otto)** in **milliseconds**

Hide PHP functions

/data/wordpress/htdocs/index.php
cachegrind.out.otto @ 2016-01-25 14:15:56



3063 different functions called in **6718** milliseconds (1 runs, **649** shown)

Filter: (regex too) Invert filter

| Function | Invocation Count | Total Self Cost | Total Inclusive Cost |
|---------------------------------|------------------|-----------------|----------------------|
| MO->import_from_reader | 6 | 527 | 1173 |
| Calls | | | |
| MO->make_entry @ 250 | 6992 | 438 | |
| Translation_Entry->key @ 251 | 6992 | 76 | |
| POMO_Reader->substr @ 244 | 6998 | 60 | |
| POMO_Reader->substr @ 245 | 6998 | 55 | |
| POMO_FileReader->read_all @ 232 | 6 | 5 | |
| php::unpack @ 236 | 6998 | 4 | |
| Translations->set_headers @ 248 | 6 | 3 | |
| php::unpack @ 237 | 6998 | 2 | |

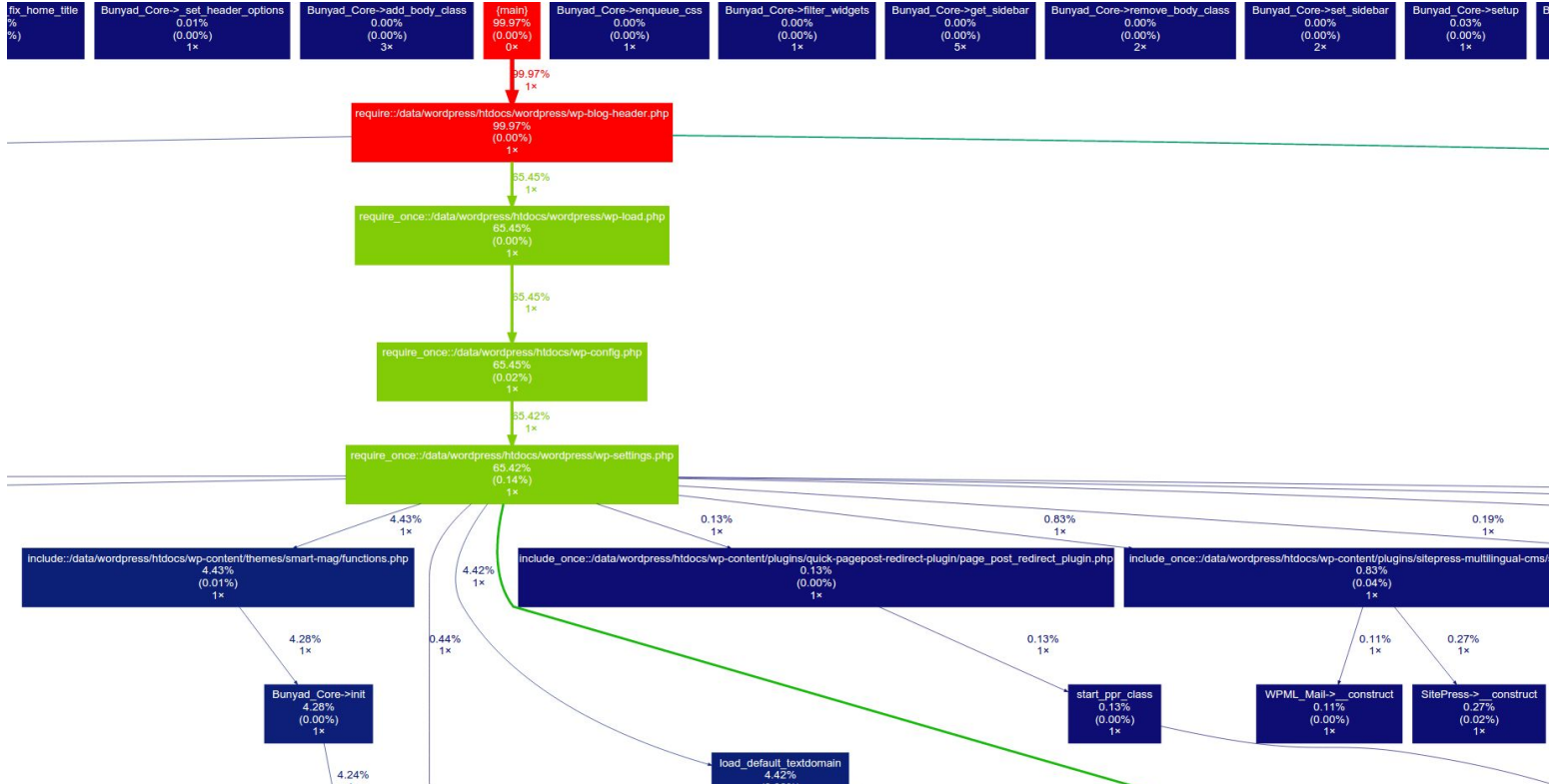
FILTER FOR USUAL SUSPECTS

- load
- open
- curl
- query

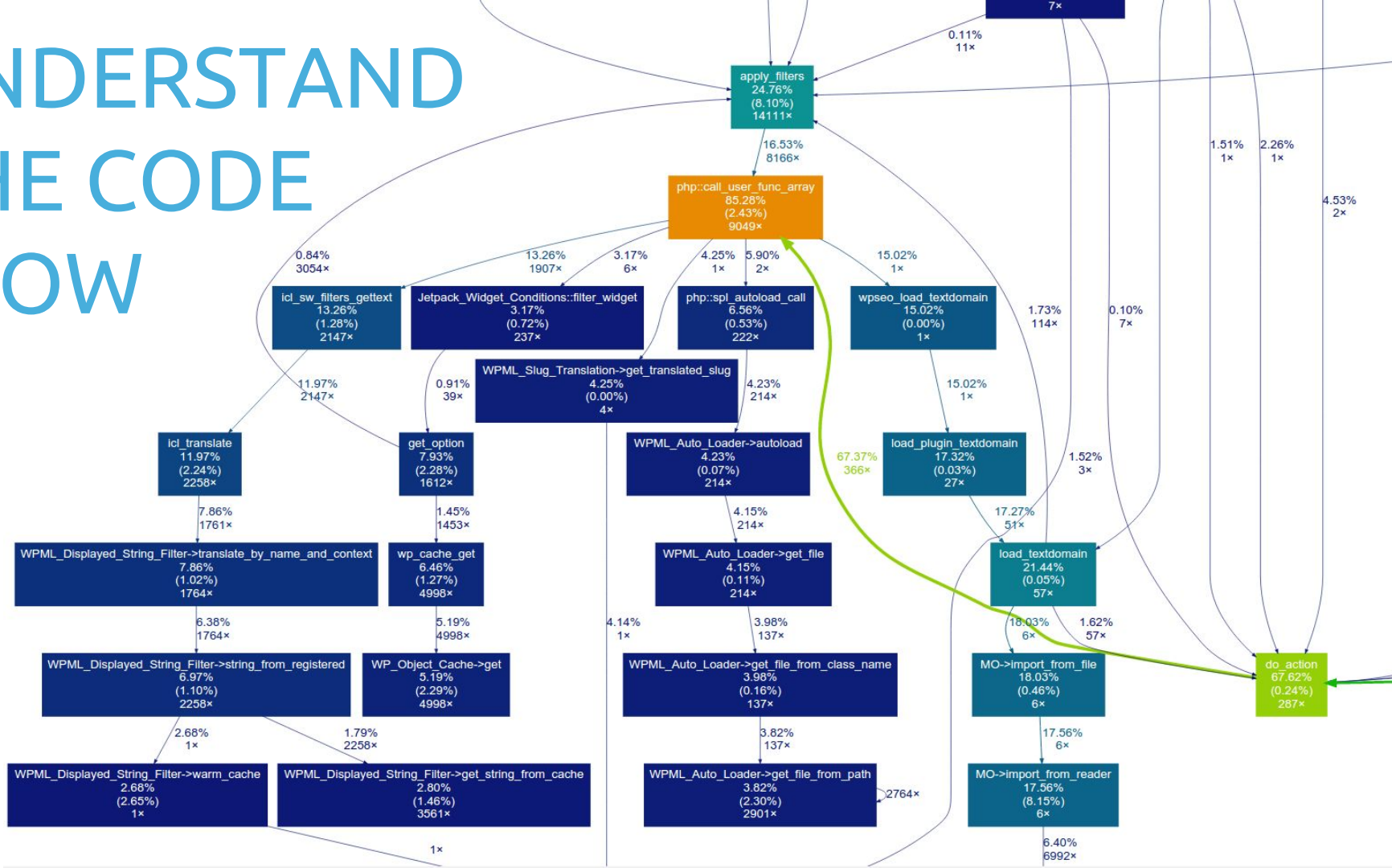
Filter: (regex too) Invert filter

| Function | Invocation Count | Total Self Cost |
|--|------------------|------------------------|
| ▼ WPML_Auto_Loader->get_file_from_path | 2901 | 1 |
| Calls | Count | Total Call Cost |
| ▶ WPML_Auto_Loader->get_file_from_path @ 157 | 2764 | 135 |
| php::glob @ 154 | 3180 | 73 |
| php::is_file @ 151 | 3233 | 25 |
| WPML_Auto_Loader->get_base_dirs @ 140 | 137 | 0 |
| ▶ WPML_Auto_Loader->build_dir @ 146 | 469 | 0 |
| Called From | Count | Total Call Cost |
| ▶ WPML_Auto_Loader->get_file_from_class_name @ 114 | 137 | 252 |
| ▶ WPML_Auto_Loader->get_file_from_path @ 157 | 2764 | 135 |
| ▶ wp_load_alloptions | 1162 | |
| ▶ Composer\Autoload\ClassLoader->findFile | 948 | |
| ▶ Composer\Autoload\ClassLoader->findFileWithExtension | 530 | |
| ▶ Composer\Autoload\ClassLoader->loadClass | 948 | |
| ▶ WC_Autoloader->autoload | 145 | |
| ▶ WPML_Auto_Loader->get_file_from_class_name | 137 | |

SHOW CALL GRAPH



UNDERSTAND THE CODE FLOW





Typical issues and solutions

FREQUENT OR LONG DB QUERIES?

| Function | Invocation Count | Total Self Cost | Total Inclusive Cost |
|----------------------------|------------------|-----------------|----------------------|
| ▶ php::mysqli_query | 331 | 52.09 | 52.09 |
| ▶ wpdb->query | 328 | 3.00 | 55.58 |
| ▶ get_option | 2109 | 2.25 | 6.45 |
| ▶ translate | 5858 | 2.07 | 3.59 |
| ▶ wfConfig::loadAllOptions | 100 | 2.06 | 8.88 |
| ▶ php::inet_pton | 2157 | 1.99 | 2.21 |
| ▼ wpdb->get_results | 175 | 1.36 | 47.39 |

| Calls | Count | Total Call Cost |
|-------------------------------------|-------|-----------------|
| ▶ wpdb->query @ 2381 | 175 | 45.87 |
| ▶ wpdb->check_safe_collation @ 2376 | 175 | 0.16 |
| ▶ php::get_object_vars @ 2409 | 5120 | 0.00 |

| Called From | Count | Total Call Cost |
|--|-------|-----------------|
| ▶ Avada_Upgrade->clear_twitter_widget_transients @ 465 | 1 | 40.40 |
| ▶ wfConfig::loadAllOptions @ 160 | 100 | 3.85 |
| ▶ update_meta_cache @ 826 | 66 | 2.98 |

Fix Avada theme upgrade check

SWITCH TO POLYLANG FOR FASTER PAGE LOADS

| ▶ Translation_Entry->key | | | | 8727 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------|-----------------|--|------|-------------|-------|-----------------|--|----|-----|---|----|----|--|------|----|---|---|---|--|---|---|------------------------------|---|---|--------------------------------------|---|---|---|---|---|--|---|---|---------------------|---|---|--------------------|---|---|--|---|---|--|---|---|
| ▶ WPML_Displayed_String_Filter->truncate_name_and_context | | | | 3665 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WP_Object_Cache->build_key | | | | 4624 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Displayed_String_Filter->translate_by_name_and_context | | | | 2309 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ icl_sw_filters_gettext | | | | 2198 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ _wp_filter_build_unique_id | | | | 2947 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ wp_cache_get | | | | 4615 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Displayed_String_Filter->string_from_registered | | | | 2309 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ php::glob | | | | 3094 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▼ wpdb->get_results | | | | 70 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table><thead><tr><th>Calls</th><th>Count</th><th>Total Call Cost</th></tr></thead><tbody><tr><td>▶ wpdb->query @ 2322</td><td>70</td><td>246</td></tr><tr><td>▶ wpdb->check_safe_collation @ 2317</td><td>69</td><td>4</td></tr><tr><td>▶ php::get_object_vars @ 2350</td><td>6416</td><td>1</td></tr></tbody></table> | | | | | Calls | Count | Total Call Cost | ▶ wpdb->query @ 2322 | 70 | 246 | ▶ wpdb->check_safe_collation @ 2317 | 69 | 4 | ▶ php::get_object_vars @ 2350 | 6416 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Calls | Count | Total Call Cost | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ wpdb->query @ 2322 | 70 | 246 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ wpdb->check_safe_collation @ 2317 | 69 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ php::get_object_vars @ 2350 | 6416 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table><thead><tr><th>Called From</th><th>Count</th><th>Total Call Cost</th></tr></thead><tbody><tr><td>▶ WPML_Displayed_String_Filter->warm_cache @ 140</td><td>1</td><td>182</td></tr><tr><td>▶ WPML_Admin_String_Filter->get_registered_string @ 144</td><td>16</td><td>77</td></tr><tr><td>▶ WPML_Element_Translation->maybe_populate_cache @ 196</td><td>37</td><td>31</td></tr><tr><td>▶ WPML_Slug_Translation->get_translated_slug @ 98</td><td>1</td><td>8</td></tr><tr><td>▶ WPML_Displayed_String_Filter->get_original_value @ 215</td><td>1</td><td>5</td></tr><tr><td>▶ WP_Query->get_posts @ 3623</td><td>3</td><td>4</td></tr><tr><td>▶ SitePress->get_language_tag @ 3839</td><td>2</td><td>3</td></tr><tr><td>▶ quick_page_post_recs->get_main_array @ 1466</td><td>1</td><td>2</td></tr><tr><td>▶ WPML_Element_Translation->prefetch_ids @ 161</td><td>4</td><td>2</td></tr><tr><td>▶ icl_st_init @ 153</td><td>1</td><td>1</td></tr><tr><td>▶ get_pages @ 4545</td><td>1</td><td>1</td></tr><tr><td>▶ WPML_Term_Translation->maybe_warm_term_id_cache @ 95</td><td>1</td><td>1</td></tr><tr><td>▶ WC_Comments::wp_count_comments @ 254</td><td>1</td><td>1</td></tr></tbody></table> | | | | | Called From | Count | Total Call Cost | ▶ WPML_Displayed_String_Filter->warm_cache @ 140 | 1 | 182 | ▶ WPML_Admin_String_Filter->get_registered_string @ 144 | 16 | 77 | ▶ WPML_Element_Translation->maybe_populate_cache @ 196 | 37 | 31 | ▶ WPML_Slug_Translation->get_translated_slug @ 98 | 1 | 8 | ▶ WPML_Displayed_String_Filter->get_original_value @ 215 | 1 | 5 | ▶ WP_Query->get_posts @ 3623 | 3 | 4 | ▶ SitePress->get_language_tag @ 3839 | 2 | 3 | ▶ quick_page_post_recs->get_main_array @ 1466 | 1 | 2 | ▶ WPML_Element_Translation->prefetch_ids @ 161 | 4 | 2 | ▶ icl_st_init @ 153 | 1 | 1 | ▶ get_pages @ 4545 | 1 | 1 | ▶ WPML_Term_Translation->maybe_warm_term_id_cache @ 95 | 1 | 1 | ▶ WC_Comments::wp_count_comments @ 254 | 1 | 1 |
| Called From | Count | Total Call Cost | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Displayed_String_Filter->warm_cache @ 140 | 1 | 182 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Admin_String_Filter->get_registered_string @ 144 | 16 | 77 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Element_Translation->maybe_populate_cache @ 196 | 37 | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Slug_Translation->get_translated_slug @ 98 | 1 | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Displayed_String_Filter->get_original_value @ 215 | 1 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WP_Query->get_posts @ 3623 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ SitePress->get_language_tag @ 3839 | 2 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ quick_page_post_recs->get_main_array @ 1466 | 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Element_Translation->prefetch_ids @ 161 | 4 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ icl_st_init @ 153 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ get_pages @ 4545 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_Term_Translation->maybe_warm_term_id_cache @ 95 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WC_Comments::wp_count_comments @ 254 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ WPML_String_Translation->get_current_string_language | | | | 2490 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ Translations->translate | | | | 1733 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▶ add_filter | | | | 2838 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

EXTERNAL HTTP REQUESTS ON EVERY PAGE LOAD?

| Function | Invocation Count | Total Self Cost | Total Inclusive Cost |
|---|------------------|-----------------|----------------------|
| php::curl_exec | 1 | 963 | 963 |
| Called From | | | |
| Count | | Total Call Cost | |
| WP_Http_Curl->request @ 224 | 1 | 963 | |
| MO->import_from_reader | 7 | 394 | 856 |
| MO->make_entry | 7013 | 301 | 306 |
| apply_filters | 7224 | 179 | 4525 |
| Calls | | | |
| Count | | Total Call Cost | |
| wp_cron @ 525 | 1 | 975 | |
| icl_sw_filters_gettext @ 235 | 1481 | 568 | |
| WooCommerce->init @ 525 | 1 | 565 | |
| wpseo_init @ 525 | 1 | 393 | |
| WPML_Admin_Texts->icl_st_translate_admin_string @ 235 | 14 | 347 | |

DEVELOPERS: PLEASE LEARN TO USE THE WP TRANSIENT API!

Most DB queries and external PHP::curl request can be cached easily:

```
@@ -12,7 +12,12 @@ class LeadsiusAPIGateway {  
  
    public function getAllForms()  
    {  
-        $this->allForms = $this->makeAPICall('webforms.json');  
+        if ( false === ( $this->allForms = get_transient( 'leadsius-apina' ) ) ) {  
+            // It wasn't there, so regenerate the data and save the transient  
+            $this->allForms = $this->makeAPICall('webforms.json');  
+            set_transient( 'leadsius-apina', $this->allForms, 1 * HOUR_IN_SECONDS );  
+        }  
+  
        if($this->allForms->total > $this->allForms->page_size){
```



Hunt down that
rare beast


```
$ for i in {1..99};  
  do curl -IL -H "Pragma: no-cache"  
  -w "%{time_total}\n" -o /dev/null  
  -s "http://localhost/?XDEBUG_PROFILE=1";  
done
```

```
$ ll -Sh /tmp
```

```
-rw-r--r--  111M cachegrind.out.1455200976.5601  
-rw-r--r--   91M cachegrind.out.1455200984.5601  
-rw-r--r--   89M cachegrind.out.1455200972.5604  
-rw-r--r--   89M cachegrind.out.1455200964.5604  
-rw-r--r--   88M cachegrind.out.1455200973.5604  
-rw-r--r--   87M cachegrind.out.1455200963.5601  
-rw-r--r--   87M cachegrind.out.1455200967.5601
```



STEP 3: Validate



STEP 4: Rinse & repeat



DEMO: How fast can we make
Twentyseventeen?

STEP 1: MEASURE

| ◆ | Function | | Invocation Count | ◆ | Total Self Cost | ▼ |
|---|---------------------------------|--|------------------|---|-----------------|---|
| ● | ▶ MO->import_from_reader | | 2 | | 21.61 | |
| ● | ▶ MO->make_entry | | 2311 | | 17.85 | |
| ● | ▶ POMO_Reader->substr | | 4626 | | 5.81 | |
| ● | ▶ translate | | 1003 | | 4.57 | |
| ● | ▶ Translation_Entry->key | | 2972 | | 4.13 | |
| ● | ▶ Translations->translate | | 661 | | 3.28 | |
| ● | ▶ MO->import_from_file | | 2 | | 2.22 | |
| ● | ▶ get_option | | 104 | | 2.13 | |
| ● | ▶ WP_Hook->add_filter | | 507 | | 1.83 | |
| ● | ▶ __ | | 1003 | | 1.78 | |
| ● | ▶ add_filter | | 507 | | 1.21 | |
| ● | ▶ Translations->translate_entry | | 661 | | 1.02 | |

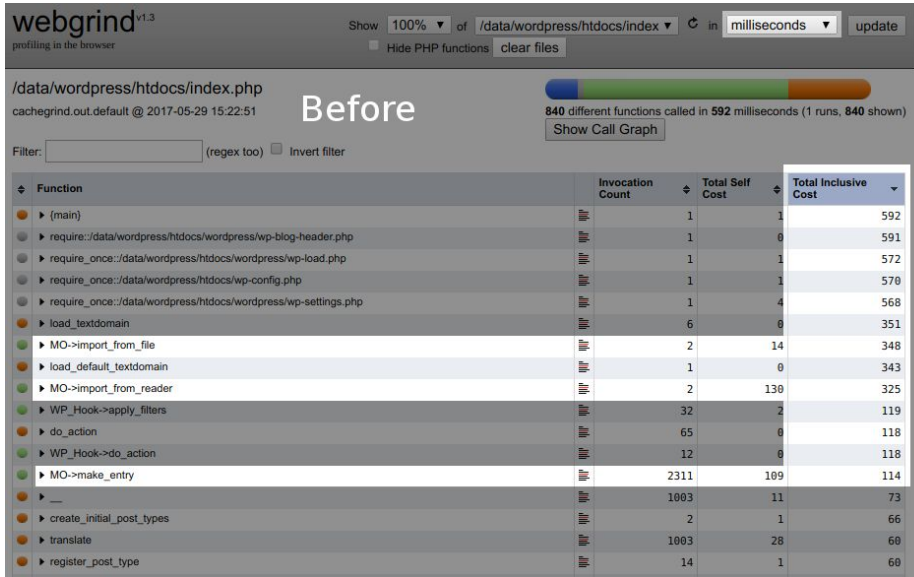
STEP 2: OPTIMIZE

Translation functions slowest. WP does not use native gettext (<https://core.trac.wordpress.org/ticket/17268>).

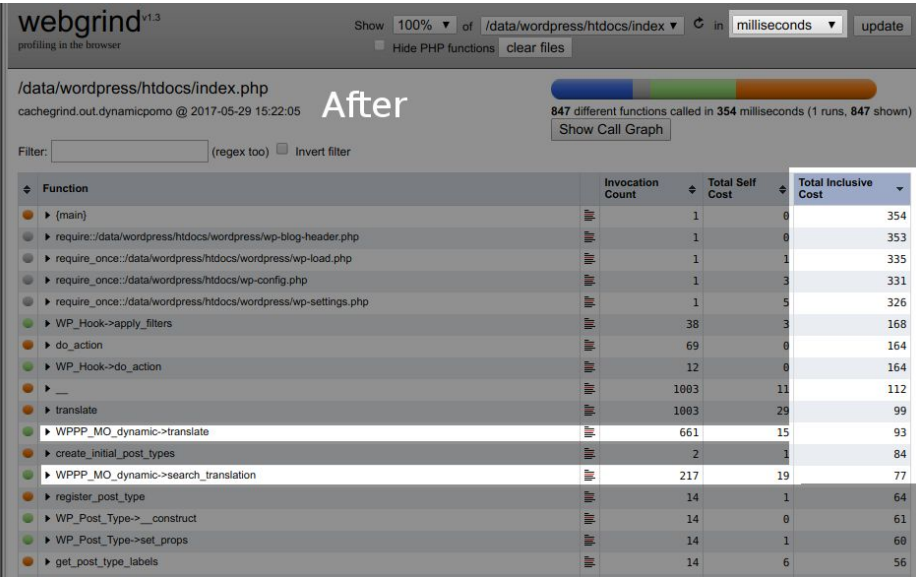
Solution:

```
composer require aucor/dynamic-mo-loader
```

STEP 3: VALIDATE



Before: 500-600 ms



After: 300-400 ms

SEE IT YOURSELF!

github.com/ottok/wceu2017-demo



REMEMBER

- Nginx access logs easy
- Xdebug **never** in production
- xhprof/upprofiler can be production
- PCEL APD (2004)
- `memory_get_usage()`, `microtime()`



THANK YOU!



seravo

SERAVO.COM

facebook.com/Seravocom

Twitter: [@Seravo](https://twitter.com/Seravo) [@ottokekalainen](https://twitter.com/ottokekalainen)