inpsyde.

How to Scale WordPress

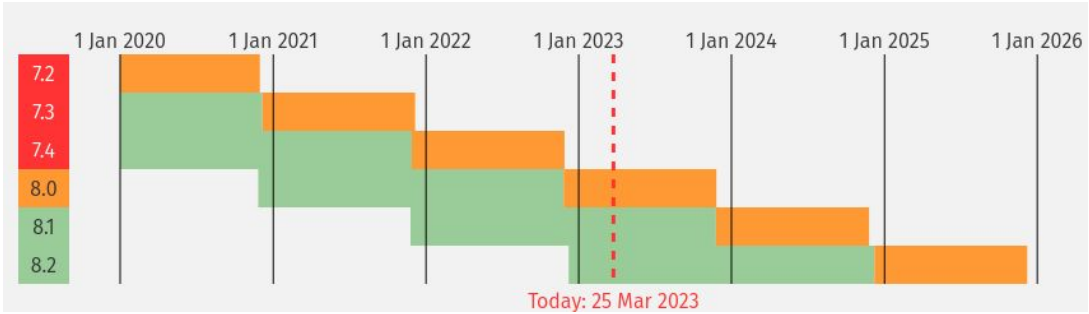
Robert Windisch - CIO - Inpsyde

@nullbytes

Scaling WordPress

PHP8

- *It's supported! PHP 5/7 isn't.*
- *Performance!*
- *Enables better Code*
- *Update only on testing environment!*
- *You want to move to PHP 8.x asap*
- *Plugins/Themes that breaks can be indicator of performance hogs*



https://www.php.net/supported-versions.php

Update WordPress to save the planet

WordPress 6.2 brings some major performance wins to WordPress core, visible in the benchmarks for both Web Vitals and Server Timing metrics. Performance is further improved for block themes performance with ~20% faster TTFB and ~14% faster LCP. On pages with hero images, the LCP improvements are even greater at ~19%.

- The new filter `pre_wp_load_alloptions` allows short-circuiting the loading of WordPress's autoloaded options with custom logic. View ticket #56045.
- The results of the `get_adjacent_post()` function are now being cached. View ticket #41131.
- Cache keys for `WP_Term_Query` are now based on SQL without placeholders so that they can actually result in cache hits. View ticket #57298.
- `WP_Query` is now no longer priming post caches twice. View ticket #57373.
- Lazy-loading term metadata from the cache is now faster due to using `wp_cache_get_multiple()`. View ticket #57150.
- The results of `wp_get_global_settings()` are now cached within a single request, resulting in a faster response time of ~8% for WordPress core. View ticket #57502.

https://make.wordpress.org/core/2023/03/09/wordpress-6-2-field-guide/

Update WordPress to save the planet

Priorities for 2023

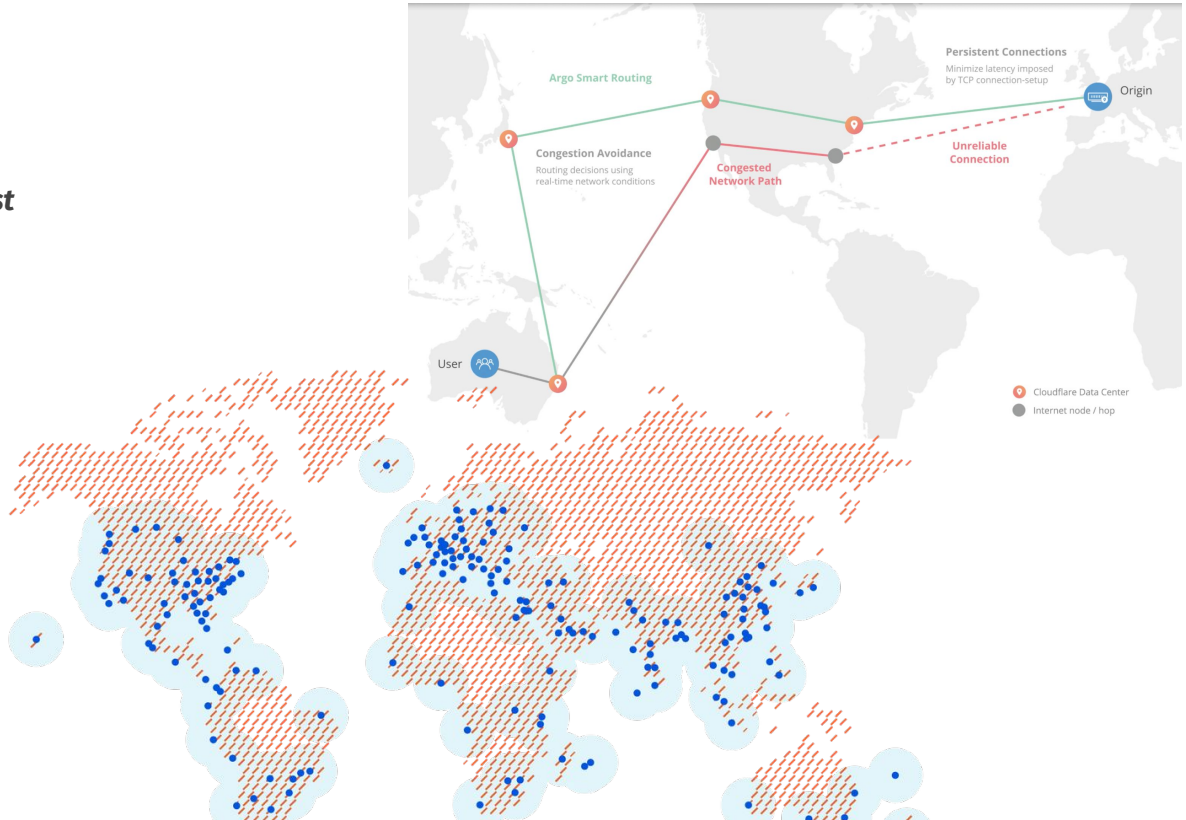
The priorities on this roadmap are grouped in a few broader categories, each dedicated to optimizing a certain aspect of WordPress performance.

- Server response time
- Database optimization
- JavaScript & CSS
- Images
- Measurement
- Ecosystem tools

https://make.wordpress.org/performance/roadmap-2023/

Content Delivery Network

- *Reverse Proxy*
- *Not on the same server*
- *Closer to the visitor -> faster request*
- *Can also mitigate Attacks*
- *Your server can be offline*
- *Only use your server run PHP*



Core Web Vitals

- *Largest Contentful Paint (LCP)*
- *First Input Delay (FID)*
- *Cumulative Layout Shift (CLS)*
- *Everything influences that:*
    a. *How much is going on the server*
    b. *How fast is PHP*
    c. *Object caching*
    d. *Cold Caches*
    e. *MySQL*

https://web.dev/vitals/

Object Cache

- *Server that serves data from Memory*
- *Memory is faster than disk*
- *Memcached or Redis Plugins for WordPress*
- *Perfect base to scale to multiple servers*
- *WordPress will speed up right away*
- *Look out for the right Invalidation*
- *Object caching in code*
- *Be careful with E-Commerce Pages*

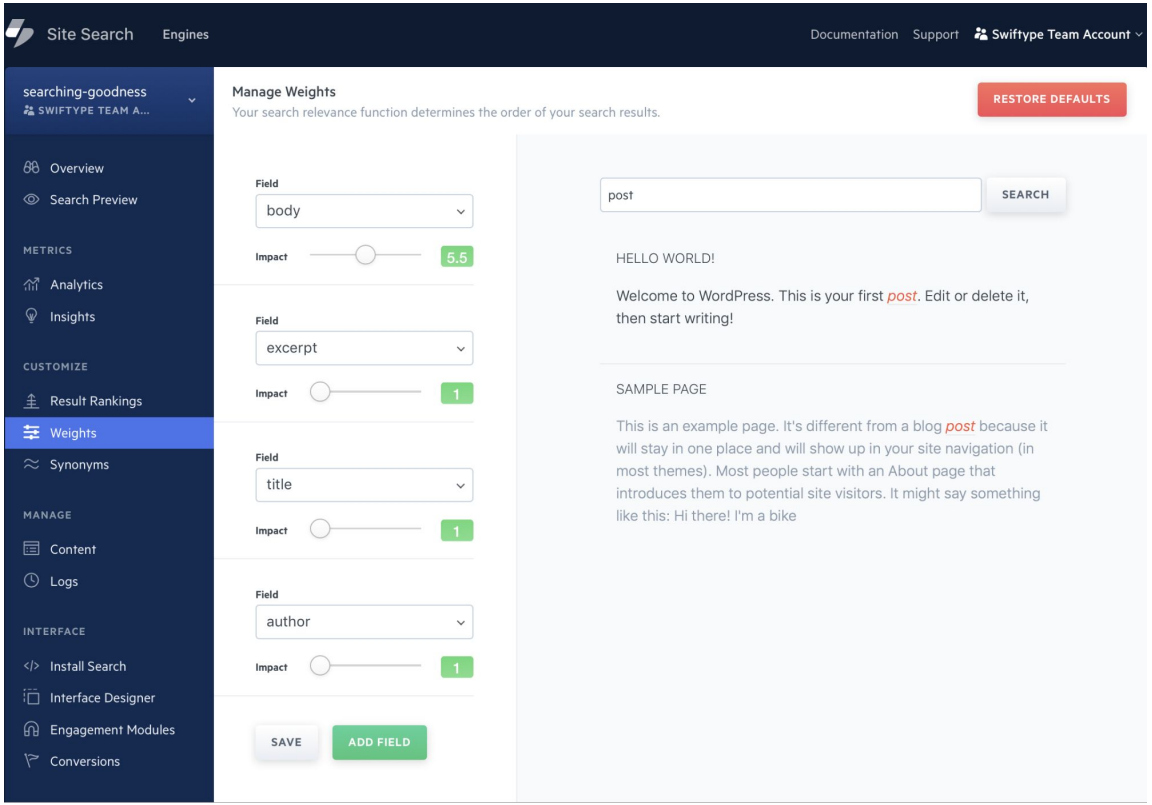
Advanced Optimization

Hard Steps for WordPress Scaling

- *ElasticSearch and Custom Development*
- *Write MySQL queries on a different server*
- *Use a Load Balancing*
- *Debug your Performance with tools like New Relic or Blackfire.io*
- *Review every line of plugins and themes*
- *Exit early and other coding patterns*

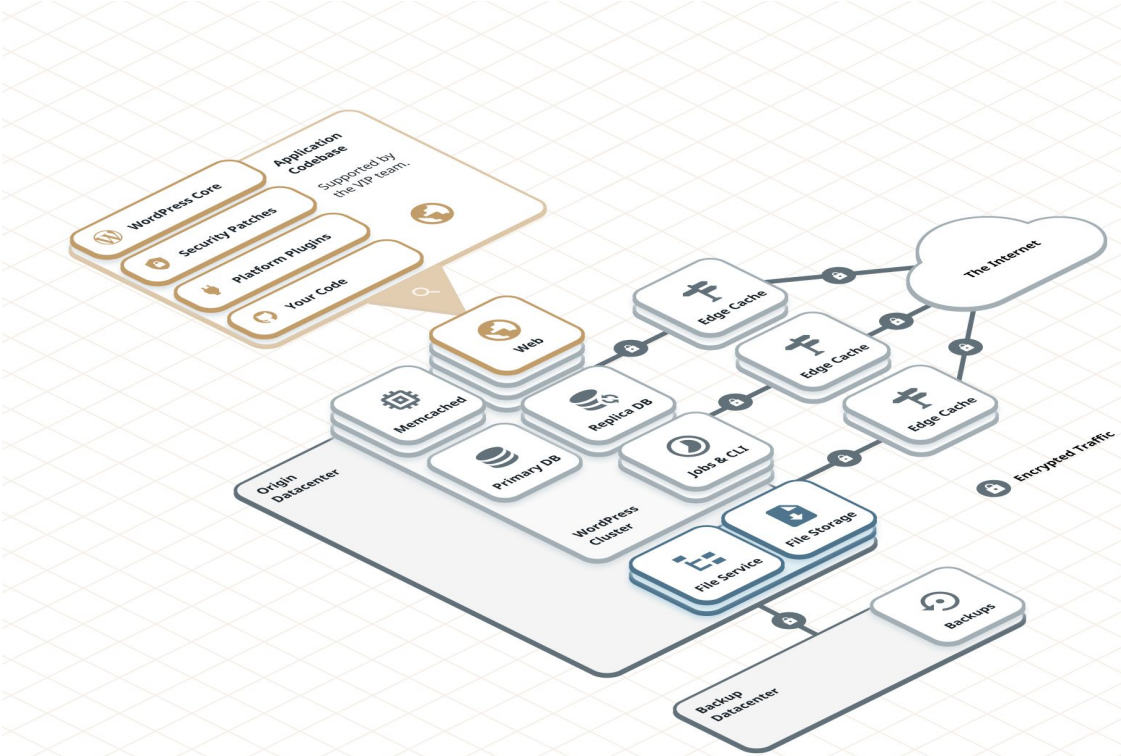
Elastic

- ***State of the Art Search***
- ***WordPress is not for searching***
- ***External Server***
- ***Facet Search***
- ***Weight***
- ***Synonyms***



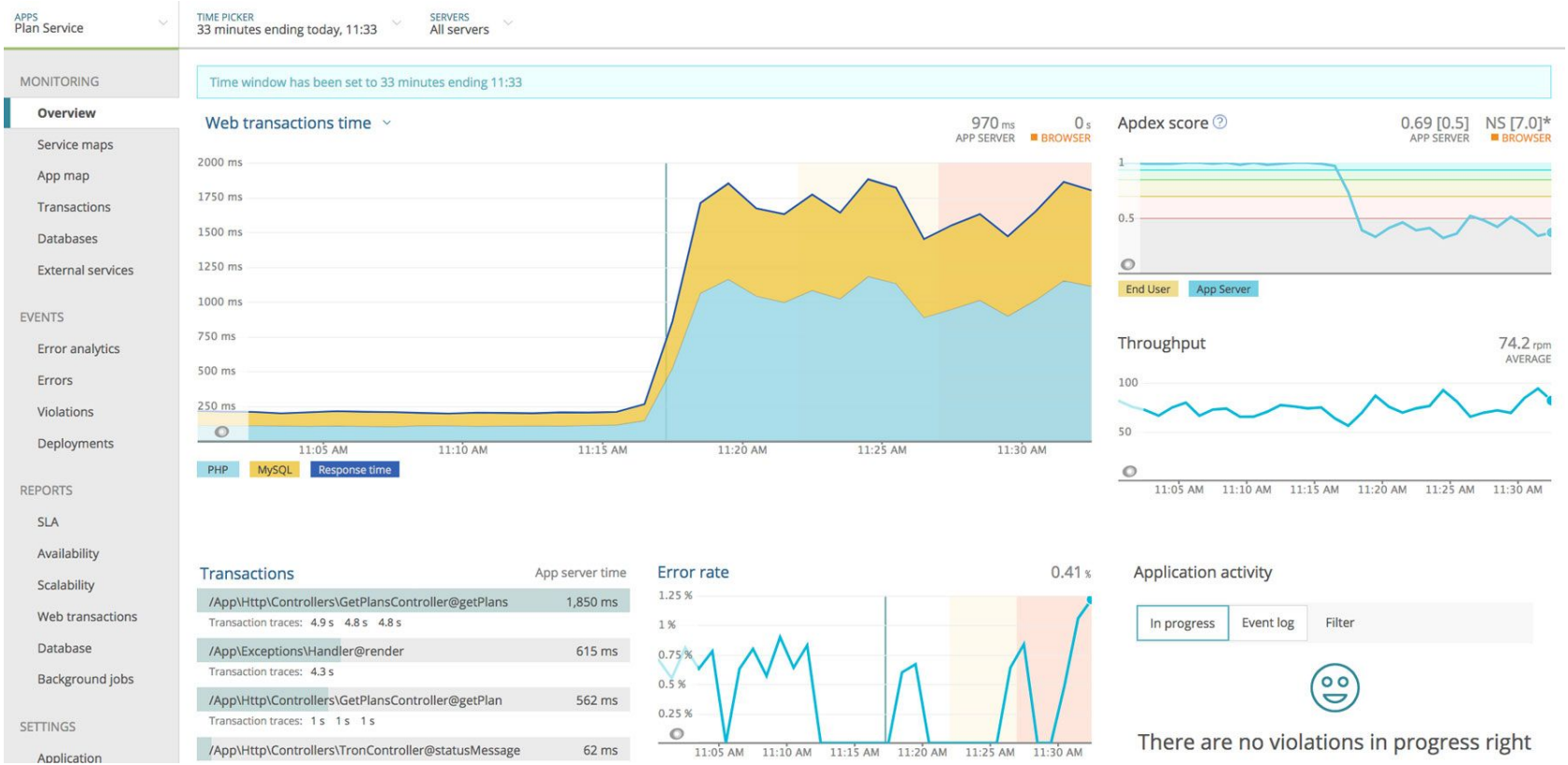
MySQL / Loadbalancing

- ***MySQL Read only Server***
- ***That's how the big hosting does it***



https://wpvip.com/documentation/developing-with-vip/the-vip-platform/

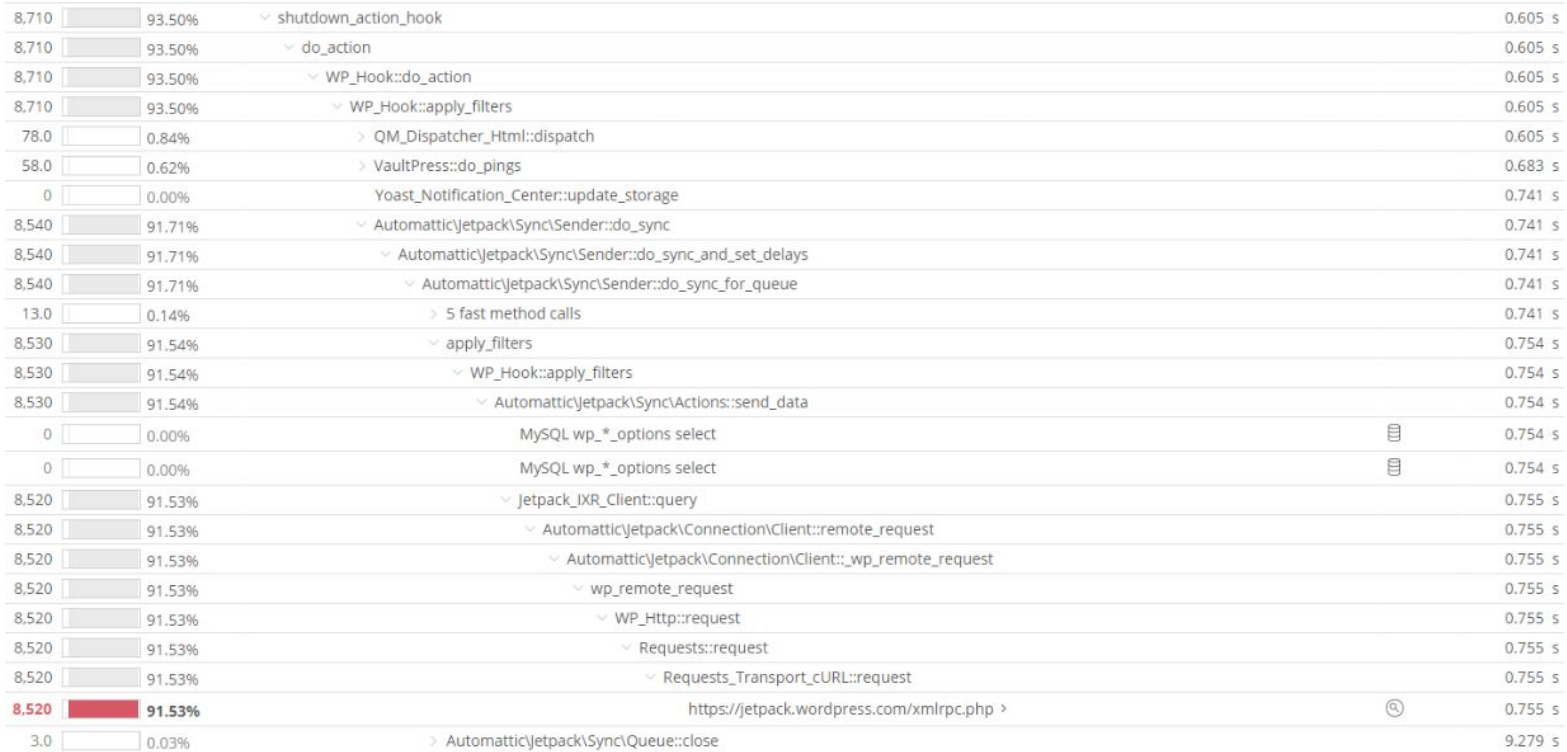
New Relic


APPS
Plan Service
TIME PICKER
33 minutes ending today, 11:33
SERVERS
All servers
MONITORING
Overview
Service maps
App map
Transactions
Databases
External services
EVENTS
Error analytics
Errors
Violations
Deployments
REPORTS
SLA
Availability
Scalability
Web transactions
Database
Background jobs
SETTINGS
Application
Time window has been set to 33 minutes ending 11:33
Web transactions time
970 ms
APP SERVER
0 s
BROWSER
2000 ms
1750 ms
1500 ms
1250 ms
1000 ms
750 ms
500 ms
250 ms
11:05 AM
11:10 AM
11:15 AM
11:20 AM
11:25 AM
11:30 AM
PHP
MySQL
Response time
Apdex score
0.69 [0.5]
APP SERVER
NS [7.0]*
BROWSER
1
0.5
End User
App Server
Throughput
74.2 rpm
AVERAGE
100
50
11:05 AM
11:10 AM
11:15 AM
11:20 AM
11:25 AM
11:30 AM
Transactions
App server time
/App\Http\Controllers\GetPlansController@getPlans
1,850 ms
Transaction traces: 4.9 s 4.8 s 4.8 s
/App\Exceptions\Handler@render
615 ms
Transaction traces: 4.3 s
/App\Http\Controllers\GetPlansController@getPlan
562 ms
Transaction traces: 1 s 1 s 1 s
/App\Http\Controllers\TronController@statusMessage
62 ms
Error rate
0.41 %
1.25 %
1 %
0.75 %
0.5 %
0.25 %
11:05 AM
11:10 AM
11:15 AM
11:20 AM
11:25 AM
11:30 AM
Application activity
In progress
Event log
Filter
There are no violations in progress right

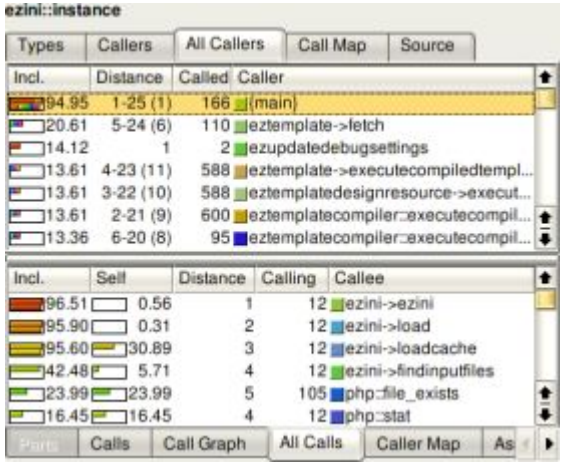
New Relic

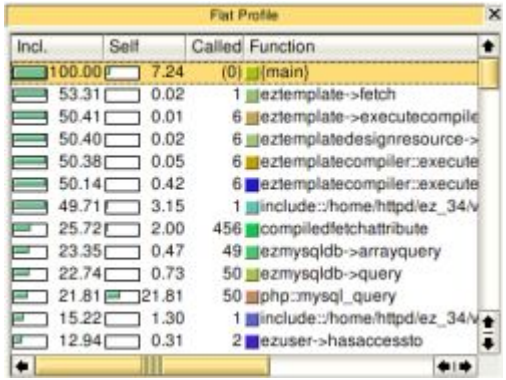
Duration (ms)	Duration (%)	Segment	Drilldown	Timestamp
9,310	100.00%	**/wp-admin/nav-menus.php**		0.000 s
22.0	0.24%	15 fast method calls		0.001 s
18.0	0.19%	do_action		0.024 s
1.0	0.01%	SuperSecrectCompanyNamespace :D\Auth\App\CoreServiceProvider::SuperSecrectCompanyNamespace :D\Auth\App\{closure}		0.043 s
0	0.00%	WPSEO_Sitemaps::init_sitemaps_providers		0.044 s
1.0	0.01%	Automattic\WP\Cron_Control\Events_Store::get_option		0.044 s
81.0	0.87%	do_action		0.045 s
12.0	0.13%	12 fast method calls		0.126 s
89.0	0.96%	do_action		0.139 s
27.0	0.29%	19 fast method calls		0.228 s
206	2.21%	do_action		0.255 s
2.0	0.02%	do_action		0.461 s
0	0.00%	Jetpack::admin_body_class		0.463 s
6.0	0.06%	_wp_menu_output		0.463 s
9.0	0.10%	do_action		0.471 s
1.0	0.01%	7 fast method calls		0.480 s
122	1.31%	do_accordion_sections		0.481 s
2.0	0.02%	9 fast method calls		0.603 s

New Relic


8,710 93.50% shutdown_action_hook 0.605 s
8,710 93.50% do_action 0.605 s
8,710 93.50% WP_Hook::do_action 0.605 s
8,710 93.50% WP_Hook::apply_filters 0.605 s
78.0 0.84% QM_Dispatcher_Html::dispatch 0.605 s
58.0 0.62% VaultPress::do_pings 0.683 s
0 0.00% Yoast_Notification_Center::update_storage 0.741 s
8,540 91.71% Automattic\Jetpack\Sync\Sender::do_sync 0.741 s
8,540 91.71% Automattic\Jetpack\Sync\Sender::do_sync_and_set_delays 0.741 s
8,540 91.71% Automattic\Jetpack\Sync\Sender::do_sync_for_queue 0.741 s
13.0 0.14% 5 fast method calls 0.741 s
8,530 91.54% apply_filters 0.754 s
8,530 91.54% WP_Hook::apply_filters 0.754 s
8,530 91.54% Automattic\Jetpack\Sync\Actions::send_data 0.754 s
0 0.00% MySQL wp_*_options select 0.754 s
0 0.00% MySQL wp_*_options select 0.754 s
8,520 91.53% Jetpack_IXR_Client::query 0.755 s
8,520 91.53% Automattic\Jetpack\Connection\Client::remote_request 0.755 s
8,520 91.53% Automattic\Jetpack\Connection\Client::_wp_remote_request 0.755 s
8,520 91.53% wp_remote_request 0.755 s
8,520 91.53% WP_Http::request 0.755 s
8,520 91.53% Requests::request 0.755 s
8,520 91.53% Requests_Transport_cURL::request 0.755 s
8,520 91.53% https://jetpack.wordpress.com/xmlrpc.php 0.755 s
3.0 0.03% Automattic\Jetpack\Sync\Queue::close 9.279 s

Xdebug





https://xdebug.org/docs/profiler

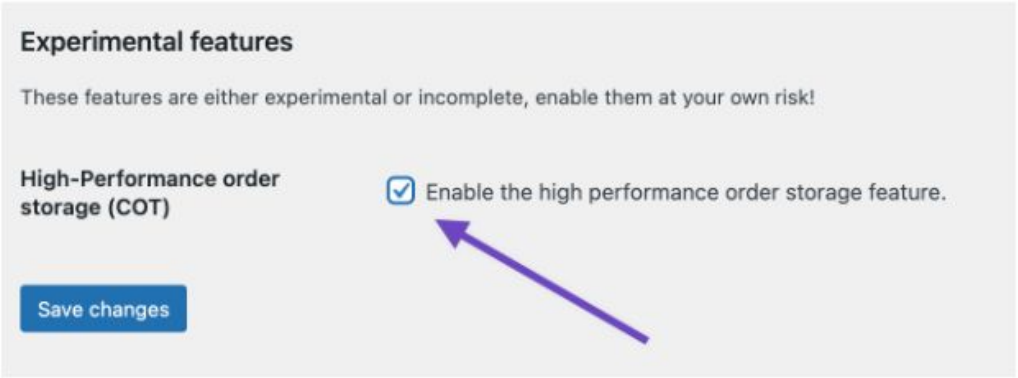
Review the Code

- ***Not everything should be loaded with the init hook***
- ***Read every line of code!***

Scaling WooCommerce

High-Performance Order Storage

- *Will be active with WooCommerce 8*
- *Custom tables*
- *Test your plugins*



https://woocommerce.com/document/high-performance-order-storage/

inpsyde.

Thank you

Robert Windisch - CIO - Inpsyde

@nullbytes