

Love the Functionality, Hate the Speed, Now What?



WordCamp Phoenix 2017



JOSHUA EICHORN

CTO, PAGELY



My WordPress Experience

- **Built Pagely's Hosting Stack**
- **Worked on our devops team**
- **Solved many performance problems**

Just another slow wordpress
site

RECENT POSTS

Hello Wordcamp

RECENT COMMENTS

ARCHIVES

October 2017

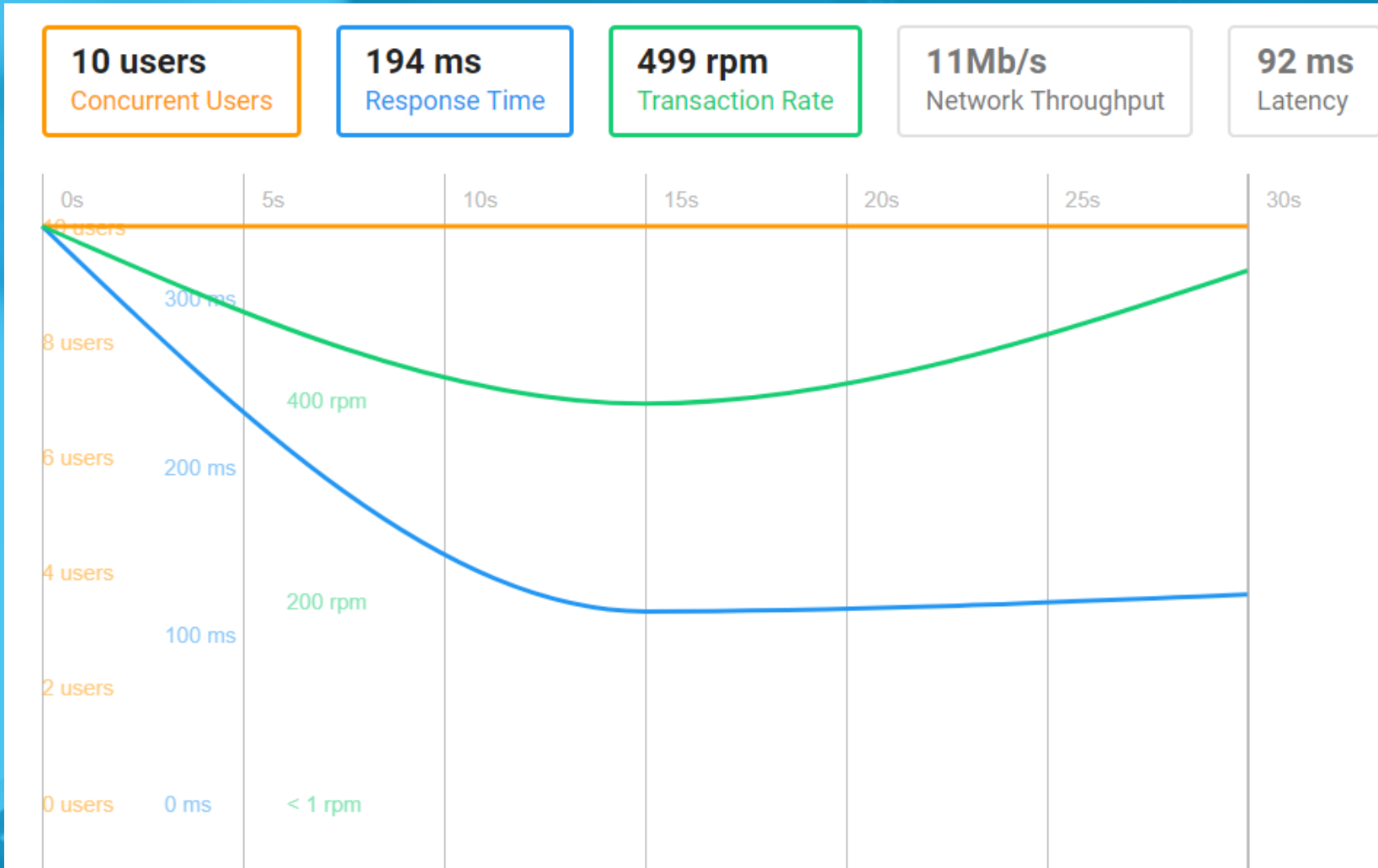
I Built a wordpress site

Hello Wordcamp

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Its performance didn't suck



Then I made the site a bit cooler

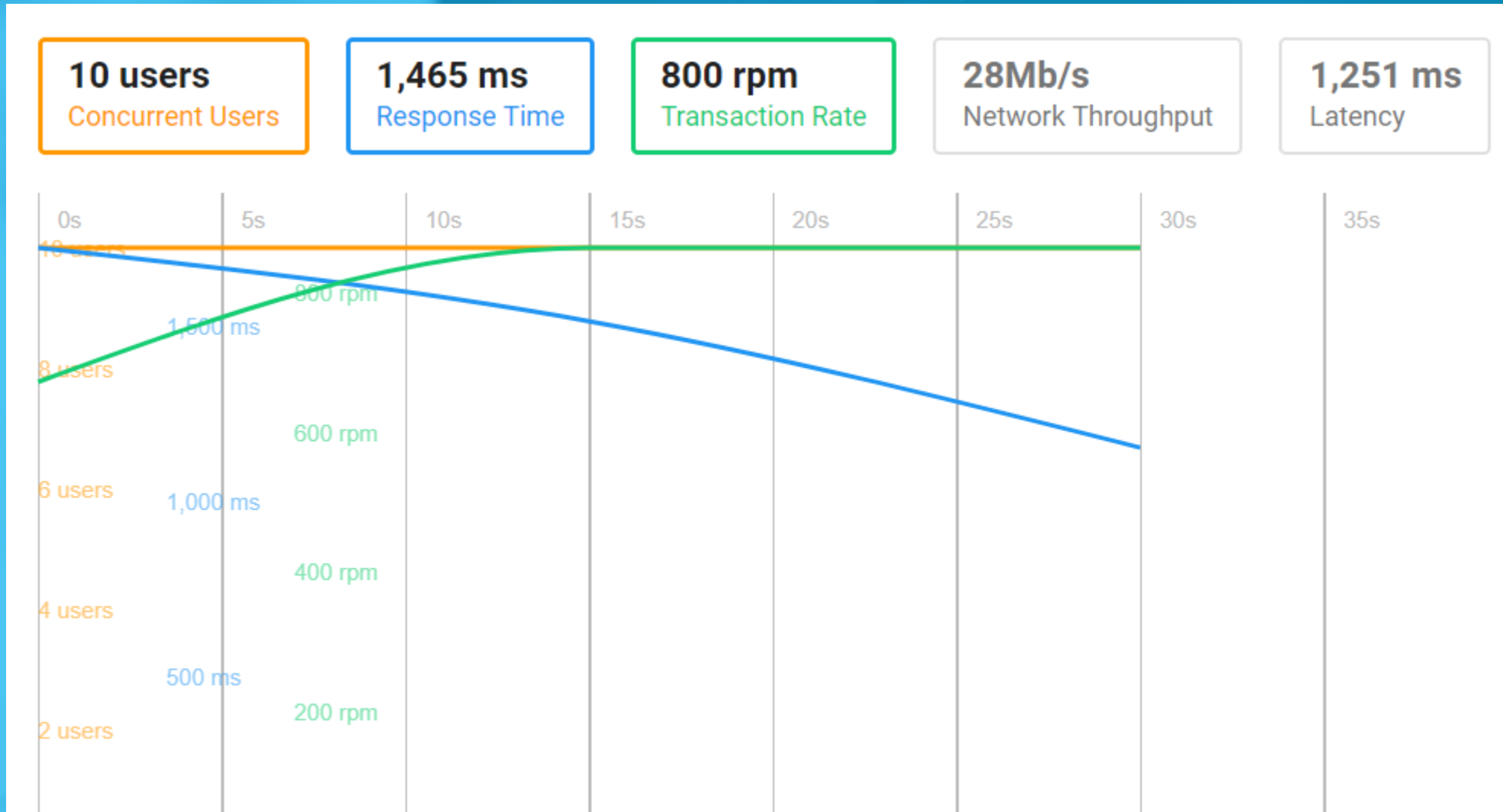
All (8) | Active (8) | Must-Use (1)

Bulk Actions ▼

<input type="checkbox"/> Plugin	Description
<input type="checkbox"/> Akismet Anti-Spam Settings Deactivate	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam . Your site is fully configured and being protected, even while you sleep. Version 4.0 By Automatic View details
<input type="checkbox"/> All In One SEO Pack Upgrade to Pro Documentation Support Forum SEO Settings Deactivate	Out-of-the-box SEO for your WordPress blog. Features like XML Sitemaps, SEO for custom post types, SEO for blogs or business sites, SEO for ecommerce sites, and much more. More than million downloads since 2007. Version 2.4.2 By Michael Torbert View details Feature Request/Bug Report
<input type="checkbox"/> Content Desirability Deactivate	Adds a score to your posts showing how desirable they are Version 0.1.0 By Joshua Eichorn
<input type="checkbox"/> Easy FancyBox Settings Deactivate	Easily enable the FancyBox jQuery extension on all image, SWF, PDF, YouTube, Dailymotion and Vimeo links. Also supports iFrame and inline content. Version 1.6 By RavanH View details
<input type="checkbox"/> Gutenberg Deactivate	Printing since 1440. This is the development plugin for the new block editor in core. Meant for development, do not run on real sites. Version 1.5.1 By Gutenberg Team View details
<input type="checkbox"/> Simple Share Buttons Adder Deactivate Settings	A simple plugin that enables you to add share buttons to all of your posts and/or pages. Version 7.3.7 By Simple Share Buttons View details
<input type="checkbox"/> WordPress Importer Deactivate	Import posts, pages, comments, custom fields, categories, tags and more from a WordPress export file. Version 0.6.3 By wordpressdotorg View details
<input type="checkbox"/> Yet Another Related Posts Plugin Deactivate Settings	Adds related posts to your site and in RSS feeds, based on a powerful, customizable algorithm. Version 4.4 By Acknowledge View details

Its performance isn't so great now

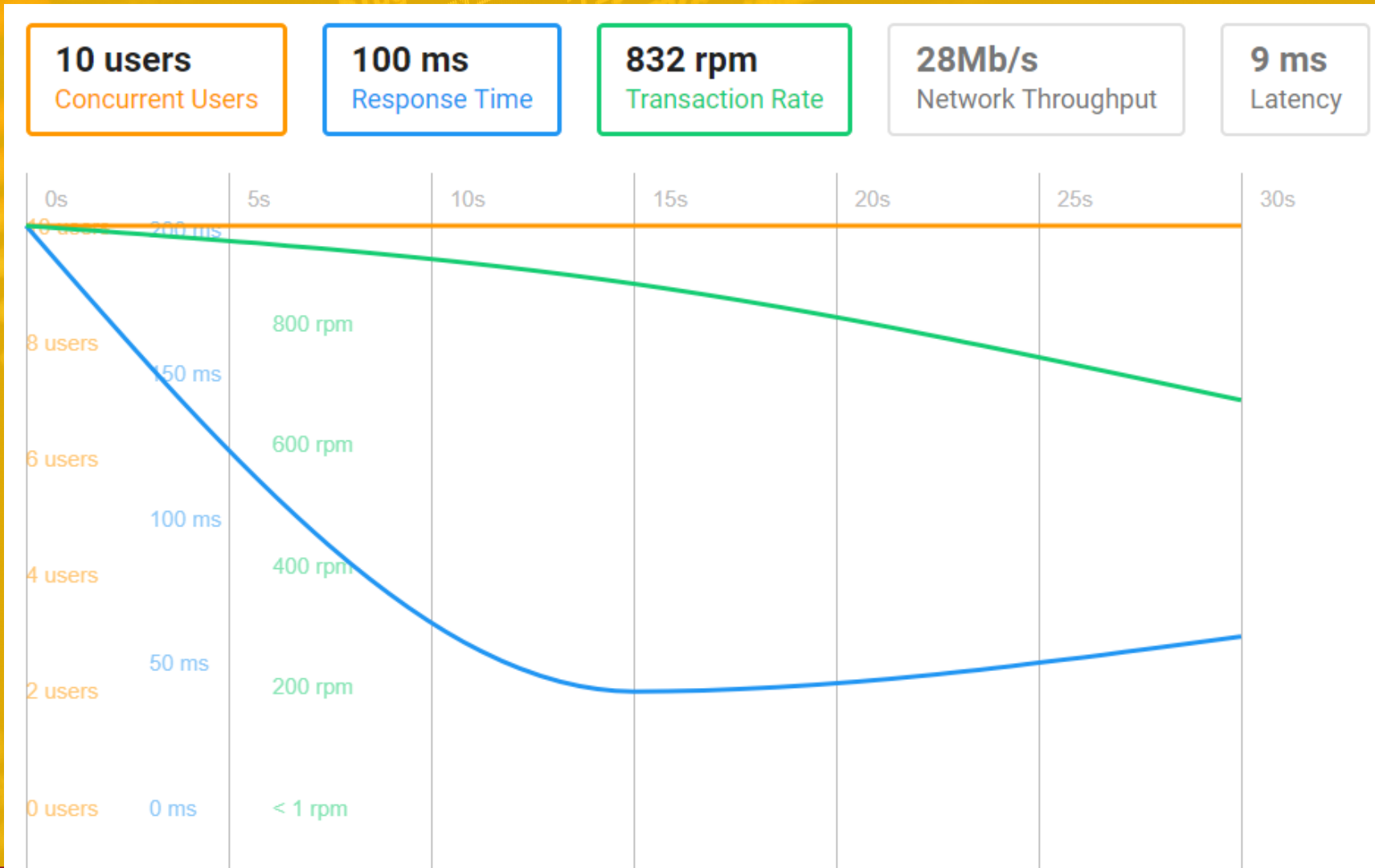
```
350  
351  
352 /* =Menu  
353 -----  
354  
355  
356 display: inline-block  
357 height: 60px  
358 float: right;  
359 margin: 11px 28px 0px 0px;  
360 max-width: 800px;  
361 )
```



A photograph of a modern glass skyscraper at night. The building's facade is composed of a grid of dark window frames. Through the windows, the interior of the building is visible, showing multiple floors with office spaces. The interior lights are a mix of warm white, pink, and blue, creating a vibrant, multi-colored glow. The building is set against a dark night sky. A large, semi-transparent blue banner is overlaid across the middle of the image, containing the text "Now What?".

Now What?

Full Page Caching to the Rescue



Maybe Not



- Low cache hit rate
- Traffic from bots
- Logged in requests
- Full cache purge

Red Flag



Page rendering time
over 2 seconds

Lets do some math

- Server has 2 cores
- Each core can render 4 WordPress pages at once
- Each page takes 2 seconds to complete

$$\begin{aligned} \text{RPS Max} &= (2 * 4) / 2 \\ \text{RPS} &= 4 \end{aligned}$$

A photograph of a modern glass skyscraper at night, with many windows illuminated from within. The image is overlaid with a semi-transparent blue gradient. In the center, the text "Now What?" is written in a large, white, sans-serif font.

Now What?



Lets find whats slow

Slow is normally IO

Slow Database query
Slow http request

Its sometimes doing tons of fast things
Its less commonly doing something slow in
PHP land

**Revert changes
one at a time
to pinpoint a slow plugin**

- 1. APM like New Relic**
- 2. Query Monitor, etc**
- 3. P3 profiler**
- 4. PHP profilers, Xdebug**

**Tools help pinpoint
a slow line of code**

Query Monitor

→ ↻ ⓘ phx-wordcamp-2017.exampledomain.website/hello-wordcamp/

🌐 Phx Wordcamp 2017 Test Site ✍️ Customize 💬 0 ➕ New ✎ Edit Post 1.69s 6,094KB 0.0466s 99Q SEO

Phx Wordcamp 2017 Test Site

Just another slow wordpress site

Search ...

Hello W

Lorem ipsum del

Everything you can see

Phx Wordcamp 2017 Test Site

Just another slow wordpress site

RECENT POSTS

- Queries
- Queries by Caller
- Queries by Component
- Request
- Rewrite Rules
- Template: single.php
- Scripts
- Styles
- Hooks & Actions
- Languages
- HTTP API Calls (1)
- Transient Updates
- Environment
- `is_single()`
- `is_singular()`

Wordcamp

37

met, consectetur adipiscing elit, sed do
eiu...unt ut labore et dolore magna aliqua. Ut
eni...quis nostrud exercitation ullamco laboris
nis...modo consequat. Duis aute irure dolor in
renrehenderit in voluptate velit esse cillum dolore eu fugiat nulla

Digging in, found the slow

	HTTP Request	Status	Call Stack	Component	Timeout	Time
		All ▾		All ▾		
1	GET http://cd.pagely.com ?url=http%3A%2F%2Fphx-wordcamp-2017.e xampldomain.website%2Fhello-wordcam p%2F	200 OK	<pre>1. get_template_part('content','') /wordpress-4.8.2/wp-includes/general-template.php:167 2. locate_template() /wordpress-4.8.2/wp-includes/template.php:647 3. load_template('~wp-content/themes/twentyfifteen/content.php') /wordpress-4.8.2/wp-includes/template.php:690 4. the_content() /wordpress-4.8.2/wp-includes/post-template.php:240 5. apply_filters('the_content') /wordpress-4.8.2/wp-includes/plugin.php:203 6. Content_Desirability->the_content() wp-content/plugins/content-desirability/class-content-desirability.php:38 7. Content_Desirability->render() wp-content/plugins/content-desirability/class-content-desirability.php:53 8. Content_Desirability->callApi() wp-content/plugins/content-desirability/class-content-desirability.php:43 9. wp_remote_get() /wordpress-4.8.2/wp-includes/http.php:170 10. WP_Http->get() /wordpress-4.8.2/wp-includes/class-http.php:612</pre>	Plugin: content-desirability	5	1.5074
						1.5074

A photograph of a modern glass skyscraper at night, with interior lights glowing through the windows. The image is overlaid with a semi-transparent blue filter. In the center, the text "Now What?" is written in a large, white, sans-serif font.

Now What?

Lets fix it.

If its not your code

FORK IT.

Plugins hosted in github makes it simple

**At a minimum rename the folder,
update comments**

Goal is to get your fixes upstream

CAUTION



Running patched plugins without renaming is a good way to have auto updates ruin your day

Fixing a slow API call

You can't make the API faster, so limit its effects

Options:

- Caching around the API call
- Use a work queue

Caching using transients

Transients are the generic WordPress caching API

https://codex.wordpress.org/Transients_API

Stores data in object cache if available or wp_options

Example Before

```
public function render()
{
    if (is_singular('post'))
    {
        return "<div class='ContentDesirability'>".$this->callApi(get_permalink())."</div>";
    }
    return "";
}

public function callApi($permalink)
{
    $score = wp_remote_get('http://cd.pagely.com?url='.urlencode($permalink));
    return $score['body'];
}
```


Example After

```
public function render()
{
    if (is_singular('post'))
    {
        return "<div class='ContentDesirability'>".$this->callApiCached(get_permalink())."</div>";
    }
    return "";
}
```

```
public function callApiCached($permalink)
{
    // transient keys have a max length of 172 chars, so md5 to get a constant length
    $key = 'Content_Desirability-'.md5($permalink);
    $content = get_transient($key);
    if ($content === false)
    {
        $content = $this->callApi($permalink);
        set_transient($key, $content);
    }

    return $content;
}
```

Things to think about

Default to set_transient is no expiration

Often we are getting timely data

Set expirations as long as possible

Dynamic expirations are often the solution

Setting an Expiration

```
public function callApiCached($permalink)
{
    // transient keys have a max length of 172 chars, so md5 to get a constant length
    $key = 'Content_Desirability-'.md5($permalink);
    $content = get_transient($key);
    if ($content === false)
    {
        $content = $this->callApi($permalink);
        $expiration_seconds = 4 * 60 * 60;
        // after a week set expiration to 24 hours
        if (get_the_date('U') < (time()-86400*7))
            $expiration_seconds = 24 * 60 * 60;
        set_transient($key, $content, $expiration_seconds);
    }

    return $content;
}
```

Transient Caching Results

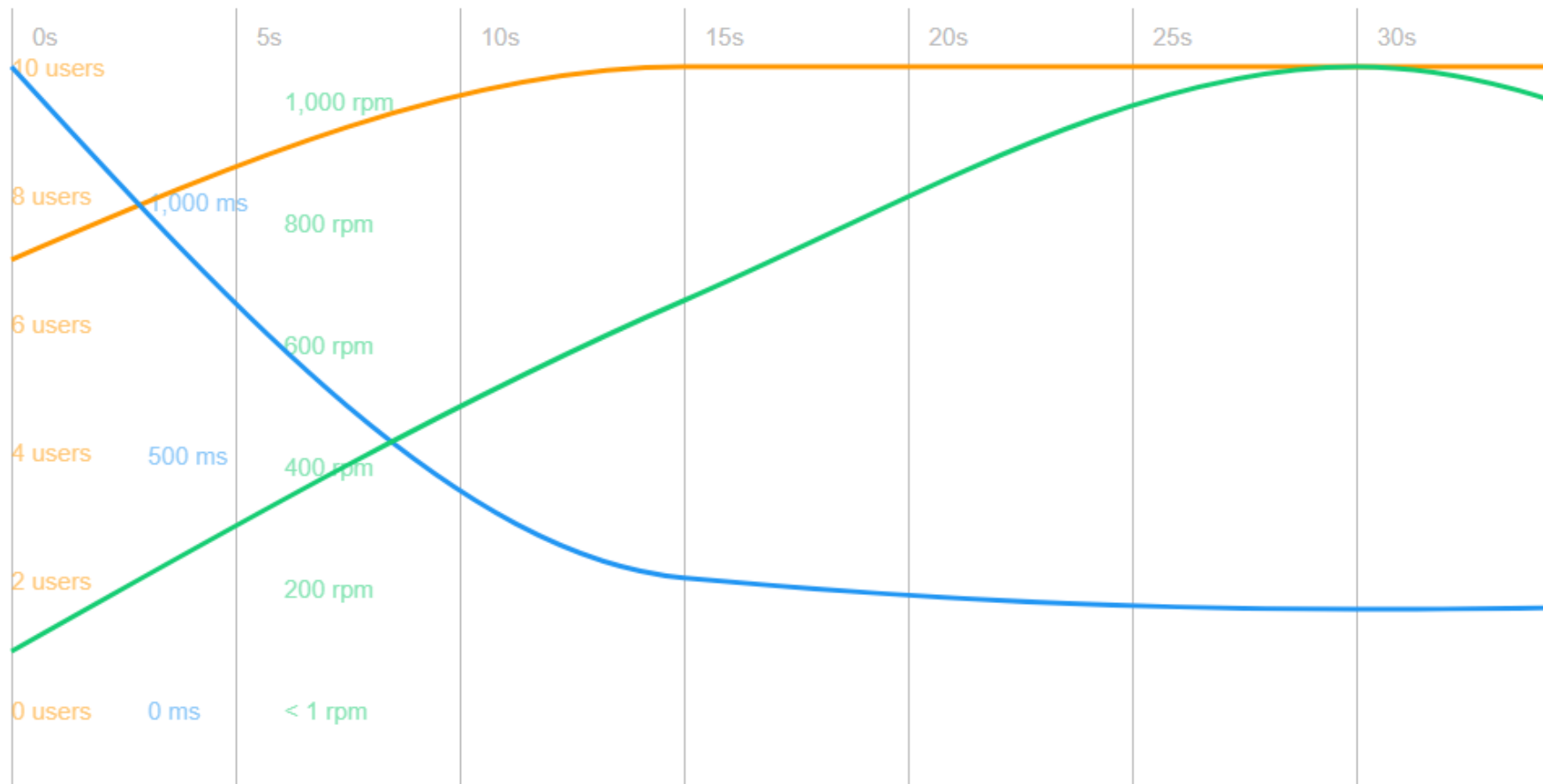
9 users
Concurrent Users

486 ms
Response Time

600 rpm
Transaction Rate

22Mb/s
Network Throughput

205 ms
Latency



Problem Solved?

Only if we can get a high enough cache hit rate

Long tail content

Bots crawling the site

What if the api gets slower

Moving the API call out

1. If you have the data, display it
2. **If you don't queue a request to load it,**
Make API request outside of the normal flow
3. Update the data in transients

Some projects to look at

WP Minions: <https://github.com/10up/WP-Minions>

WP Background Processing:

<https://github.com/A5hleyRich/wp-background-processing>

Lets build a basic queue

Store jobs in an option

Process them using wp-cron

Limitations

Keep the max queue size relatively small

Will have a low max processing rate

Will be prone to race conditions

Setup the cron

```
add_filter('cron_schedules', array($this, 'cron_schedules'));
if (!wp_next_scheduled('content_desirability_cron'))
    wp_schedule_event( time(), 'five_minutes', 'content_desirability_cron' );
}
```

```
public function cron_schedules($schedules)
{
    $schedules['five_minutes'] = [
        'interval' => 300,
        'display' => esc_html__( 'Every Five Minutes' ),
    ];
    return $schedules;
}
```

```
function content_desirability_cron() {
    $plugin = new Content_Desirability();
    $plugin->apiCron();
}
```


Update render to handle no value

```
public function render()
{
    if (is_singular('post'))
    {
        $score = $this->callApiCachedQueued(get_permalink());
        if ($score)
            return "<div class='ContentDesirability'>$score</div>";
    }
    return "";
}
```

Return a cache value or queue

```
public function callApiCachedQueued($permalink)
{
    $key = 'Content_Desirability-'.md5($permalink);
    $content = get_transient($key);
    if ($content === false)
    {
        $queue = get_option('cd_queue');
        if (!$queue)
            $queue = [];
        $queue[$key] = $permalink;

        if (count($queue) > 100)
            $queue = array_slice($queue, 0, 100);
        update_option('cd_queue', $queue, false);
    }
    return $content;
}
```


Make API calls in the cron

```
public function apiCron()
{
    $start = microtime(true);
    $queue = get_transient('cd_queue');
    if (!$queue)
        return;

    $expiration_seconds = 24 * 60 * 60;
    foreach($queue as $key => $permalink)
    {
        $content = $this->callApi($permalink);
        set_transient($key, $content, $expiration_seconds);
        $this->removeFromQueue($key);

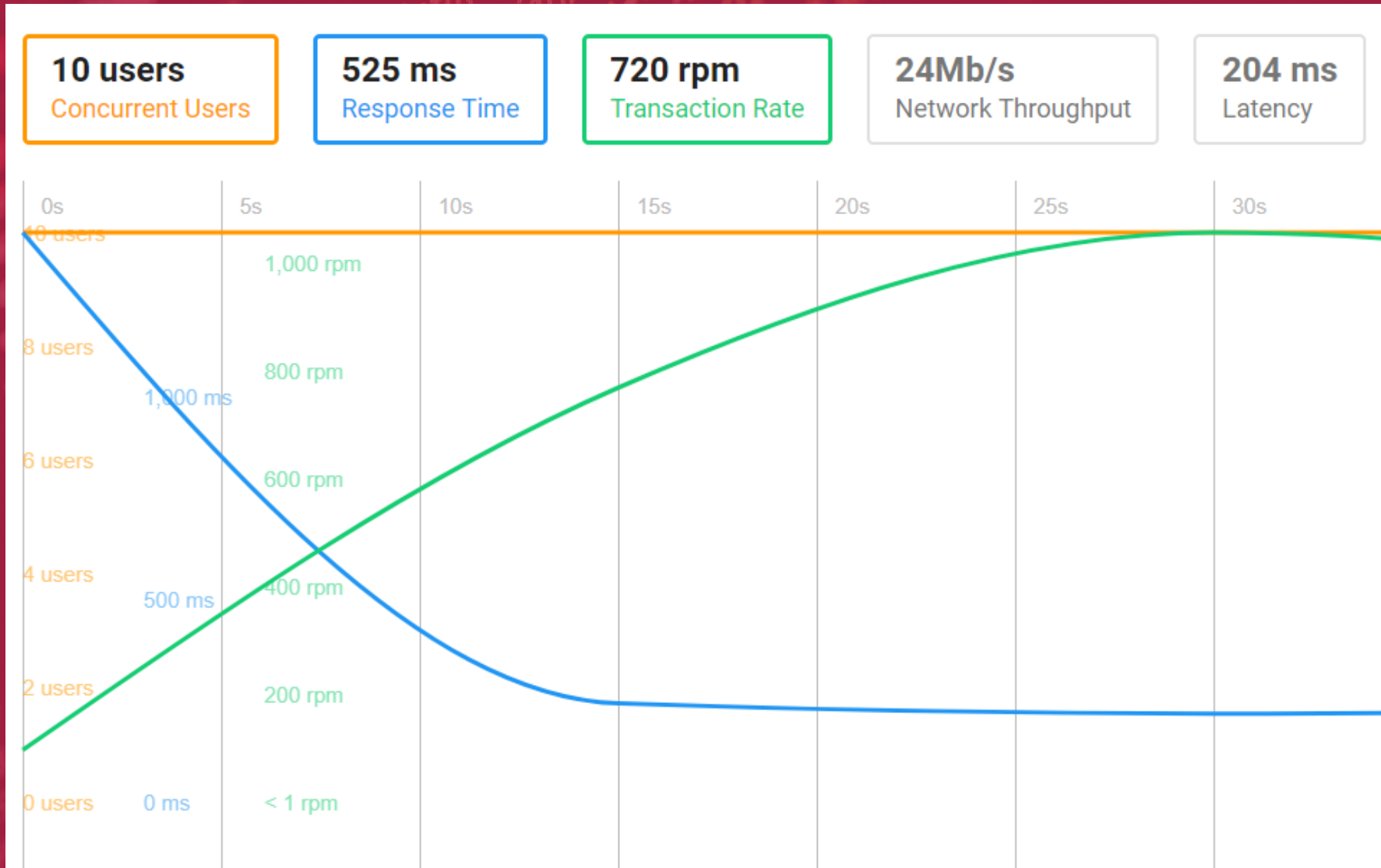
        // after 10 seconds stop
        if (microtime(true)-$start > 10)
            break;
    }
}
```

Remove tasks from the queue

```
public function removeFromQueue($key)
{
    // reduce races, by dropping the in-memory object cache
    // most likely won't be compatible with drop-in object-cache.php (memcache, redis etc)
    global $wp_object_cache;
    $wp_object_cache->cache = [];

    $queue = get_option('cd_queue');
    unset($queue[$key]);
    update_option('cd_queue', $queue, false);
}
```


Queue Results



Problem Solved?

As long as our queue can keep up

High rate sites will need to use something like WP Minions

The fixes we tried

Added a full page cache

Patched the plugin

Caching around API calls

Moving API calls to a work queue

What I normally do

Use full page caching and Transient Caching

For really high traffic sites use WP Minions

Thanks 😊

