(Or Any CMS)

WordPress Optimization For Shared Hosting Environments

Paul O'Flaherty (@PaulOFlaherty)

Scrw Media

WHAT & WHO CAN BENEFIT

- What is our Goal?
  - To make WordPress run as fast as possible on a Shared Host.
  - To do it without spending money – only free / open source solutions.
- Who Is This Presentation For?
  - Everyone (Beginners/ Pros/ Users of Other CMS's)
  - Solutions work on shared and dedicated environments.

Causes Of A Slow Site / Server Load

- High traffic volume (a wonderful problem)
- Large page size
- Complicated pages
- Database calls (of the unnecessary variety)
- Plugins and 3rd party widgets
- No caching

Page Size And Complexity What Can We Do?

- Reduce Complexity – Minimize HTTP Requests
  - Think about what features you really need
  - Less widgets / less plugins
  - Less iframes
  - Get rid of 3rd party widgets and tracking services
  - Style sheets at the top
  - Scripts at the bottom
  - Less images

CSS Sprites

- Use CSS sprites to reduce the number of images on your site without sacrificing variety.
- Reduces the number of HTTP requests to pull the images.

BEFORE

Number of HTTP requests:
10

Total size of the images:
20.5 KB

EXAMPLE

AFTER

Number of HTTP requests:
1

Total size of the images:
13 KB

EXAMPLE

Compress Your Images

- Compress your images
- Compress your images
- Compress your images

- Don't use larger images than needed.
  - If your blog has a content area 600 pixels wide why are you uploading an image 2400 pixels wide?
  - Does it need to be a JPEG, PNG, or GIF???

Size Matters

600x 401 Pixels – 520 KB PNG

It's Not The Size, It's What You…

600x 401 Pixels – 36.1 KB JPG

Experiment With Image Types

- Experiment with formats and compression
- Choose the right image type for the job
- Does my picture of Dexter & Masuka really need to be 2MB?

Culling The Database Calls

- Plugins – If you don't need them get rid of them
  - Do you really need to show off how many visitors are on your page?
  - Do you need 20 related posts in your feed or at the end of your post?
- Widgets – Do you really need it?
  - Can the output be cached?
- Optimize Code
  - Reduce the number of recurring DB calls by coding them out
- Cache
  - Cache your pages for better performance
  - Use an object cache

CODE OUT THE CALLS

- WordPress makes a lot of unnecessary DB calls.
  - Template path
  - Favicon
  - Graphics
  - Links
- These calls are made to facilitate designers and developers..

CODE OUT THE CALLS

Code from Header.php in the WordPress Default Theme - Twenty Eleven

```
69
70  <body <?php body_class(); ?>>
71  <div id="page" class="hfeed">
72      <header id="branding" role="banner">
73              <hgroup>
74                  <h1 id="site-title"><span><a href="<?php echo esc_url( home_url( '/' ) ); ?>" title="<?php echo esc_attr( get_bloginfo(
    'name', 'display' ) ); ?>" rel="home"><?php bloginfo( 'name' ); ?></a></span></h1>
75                  <h2 id="site-description"><?php bloginfo( 'description' ); ?></h2>
76              </hgroup>
77
```

Same section of code but with unnecessary calls coded out.

```
69
70  <body <?php body_class(); ?>>
71  <div id="page" class="hfeed">
72      <header id="branding" role="banner">
73              <hgroup>
74                  <h1 id="site-title"><span><a href="http://mysite.com" title="My Site" rel="home">MY SITE</a></span></h1>
75                  <h2 id="site-description">Site Description</h2>
76              </hgroup>
77
78              <?php
```

Offload the work

Offload parts of your site to 3rd party services –

- Comments – Disqus, Facebook etc…
- Random / featured posts
- Hot posts
- Feedburner
- Imgur / Flickr

- Ajax Libraries

Cons of offloading –

- Reliant on a third party service
- If they're slow, so are you.. If go down, so may your site

Offload the work

A number of the javascript libraries distributed with WordPress are also hosted on Google's **AJAX Libraries CDN**.

This provides numerous potential performance benefits:

- increases the chance that a user already has these files cached
- takes load off your server
- uses compressed versions of the libraries (where available)
- Google's servers are set up to negotiate HTTP compression with the requesting browser
- Decreased latency

Caching

- Browser Caching
  - Essentially helps a server reduce the number of requests that each visitors browser make to your site by caching the files on the users computer so they don't download them again.
- Server Caching
  - Server caching caches the generated pages on the server so they don't have to be reassembled when requested by a new user.

Browser Caching - Expires or Cache-Control Header

- Browser caching works best with repeat visitors.
- We're telling the users browser to hang on to files such as images for set period of time instead of downloading them again.
- Browser caching is done in your .htaccess file
- Expires can also be set in the header

Browser Caching

Inside the .htaccess file

```
<IfModule mod_expires.c>
    ExpiresActive On
    ExpiresByType text/html M3600
    ExpiresByType text/css M3600
    ExpiresByType application/x-javascript M3600
    ExpiresByType image/bmp M3600
    ExpiresByType image/gif M3600
    ExpiresByType image/x-icon M3600
    ExpiresByType image/jpeg M3600
</IfModule>
```

Server Caching

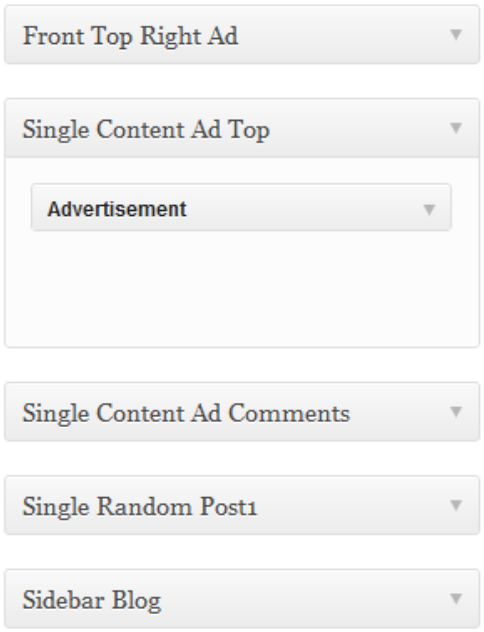
- Caching saves pages that have been accessed so they can be served to another visitor without rendering them again
- There are many caching plugins out there, but for typical shared host environments we recommend and use WP-Super Cache
- When properly set up the performance is increase is phenomenal

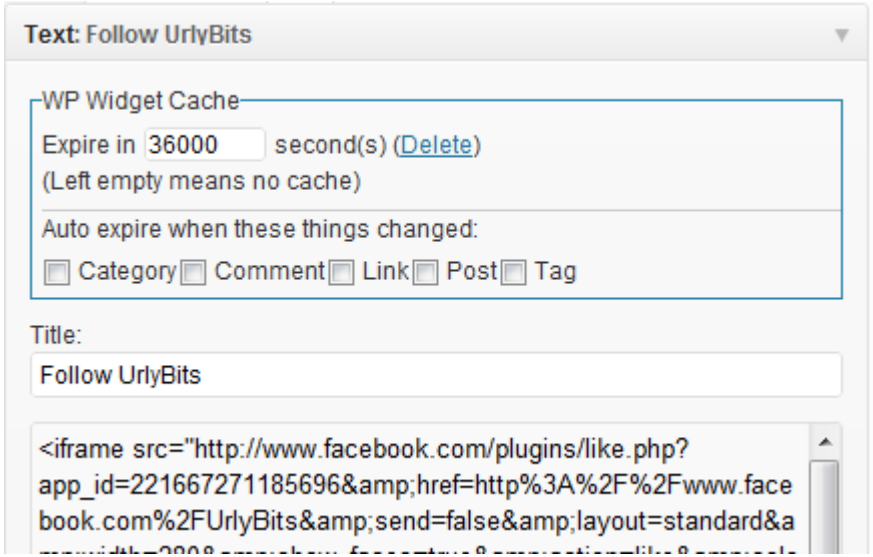
Setting Up WP Super Cache

- Use mod_rewrite to serve cache files.
- Compression allows you to send a zipped file to the users browser rather than individual page parts
- Preload allows you to cache your pages in advance of anybody visiting them
  - 2 ways to do this – Do it in bulk or just set to preload mode – we just set to preload (less server load)
  - Remove bots from the set of excluded user agents
  - Set Your Expiration – Default is 3600 – we use 172800 (2 days)

Why An Object Cache?

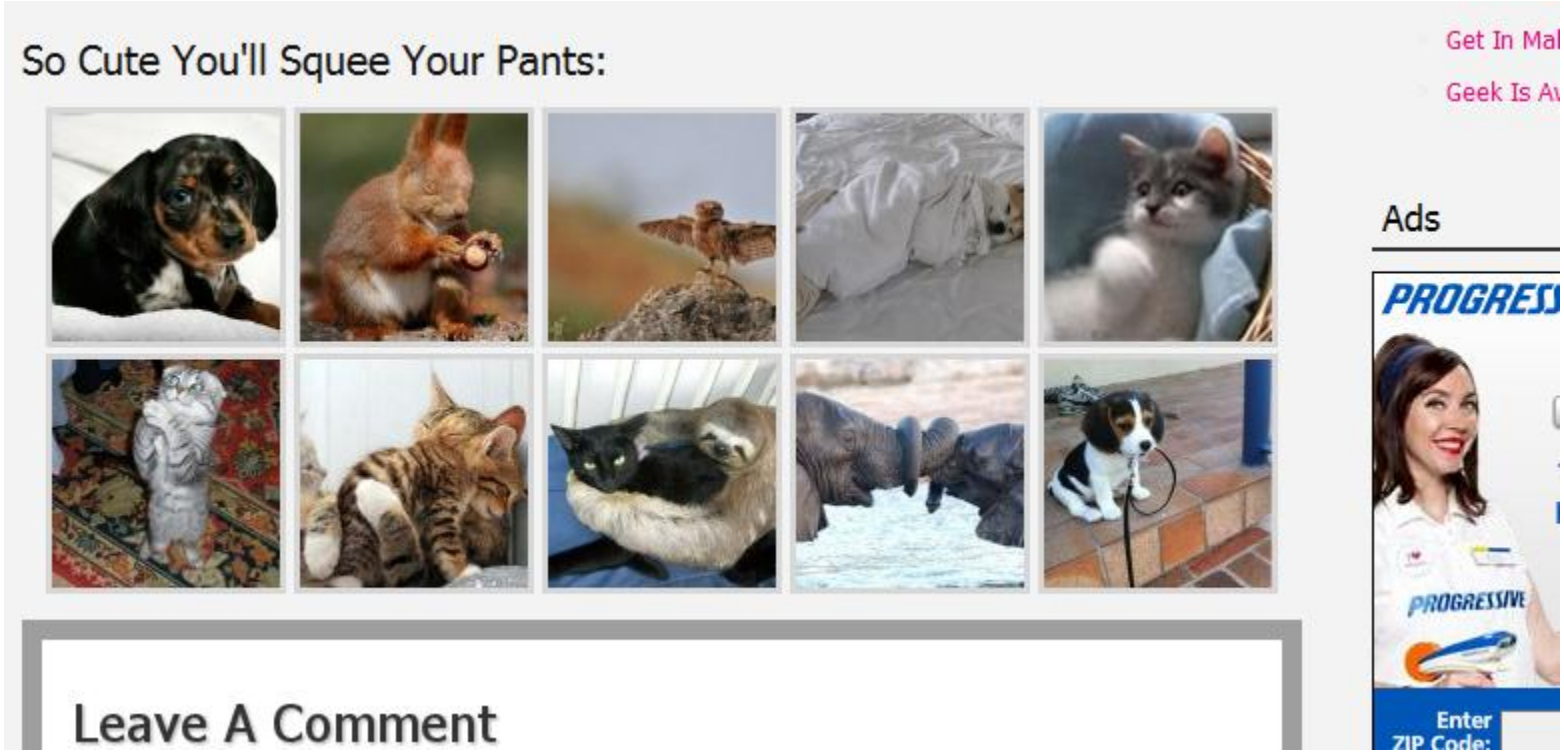
- Cache Widgets independent of the main cache
- Allows for more dynamic content while reducing server load





OBJECT CACHE TO THE RESCUE

- Plugins and widget output typically ISN'T CACHED
- They run EVERY TIME a page is loaded



NOW THAT YOU HAVE A CACHE

- Minify
  - WP Minify grabs JS/CSS files in your generated WordPress page and passes that list to the Minify engine. The Minify engine then returns a consolidated, minified, and compressed script or style for WP Minify to reference in the WordPress header.
  - In plain English – strips out all the white space, comments and bloat from your CSS and JS files allowing them to be sent to the user quicker

PIPELINING WITH MULTIPLE HOSTNAMES

- The Poor Mans CDN
- Split Components Across Domains to reduce DNS lookups

www.yoursite.com/images/a.png --->
www.yoursite.com/images/b.png --->

www.yoursite.com
assets1.example.com/a.png
assets2.example.com/b.png

- Beware the DNS lookup penalty
- Use a maximum of 4 domains / subdomains (2 recommend)

Final Tips

- Remove revisions
- Optimize database
- Permalink structure
  - No → http://yoursite.com/%postname%/
  - Yes → http://yoursite.com/%post_id%
- Backup everything

- Pay For A CDN

- WordPress Codex is your friend.

Thank You For Your Time

- My name is Paul O'Flaherty, thank you for listening and feel free to get in touch if you have any questions.

- Blog – http://pauloflaherty.com
- Twitter – @pauloflaherty
- Facebook – PaulOFlahertyOnline
- Google+ http://ongpl.us/pof

- Download the PDF http://pauloflaherty.com/wptmu

Resources

- CSS Sprite Tutorial
  - Building Faster Websites With CSS Sprites
  - CSS Sprites: What They Are, Why They're Cool, and How To Use Them
  - Radical Image Optimization Tool
- Caching
  - WP Super Cache
  - Widget Cache
- Minify – WP Minify
- 3 Reasons You Should Let Google Host jQuery For You – Dave Ward
- Use Google Libraries
- WP Parallel Loading System
- Yahoo Best Practices For Speeding Up Your Website
- WordPress Optimization – WordPress Codex
- Mark Jaquith – Theme and Plugin Security.

Attribution

- “The Flash” by Robert Ball
- CSS Sprite images – CSS Tricks.com