Advanced performance optimizations

By Peter Elmered

@pelmered

Optimal Norge AS
About me

Peter Elmered

E-commerce / Web Developer
at Optimal Norge
(Mostly Magento, Wordpress & WooCommerce)
optimalnorge.no

Plugin developer

Big Wordpress and open source fan

Twitter & GitHub: pelmered

BTW: Oslo is a great place to live and work with web development
Contact me if you are interested. I'll help you get settled in.
Why should I care about performance?
Why should I care about performance?

- User experience
- Conversion
- SEO
- Etc....
What makes a website REALLY fast?
http://homeautomateit.com
Tested from on March 5 at 10:14:36

Perf. grade 100/100 Requests 15 Load time 365ms Page size 173.5kB

Your website is faster than 98% of all tested websites

DOWNLOAD HAR Tweet Post to Timeline Email
Yes... it's CACHING!

WordCamp
Norrköping 2014
25-27 april
Höökparken, Östergötland
What is cache?

To store frequently accessed data in a faster storage medium for faster access

Or to temporarily store data in a precomputed format that is faster to read or use
The important thing is:

It speeds things up.
Some principles of caching on the web

- Cache everything that is possible to cache (should probably be everything in one way or another)
- Cache where it's fast for the user to access
  - Close to user (low latency)
  - On fast storage mediums
    - RAM is fast, HDD... not so much.
  - Store the data that requires as little processing as possible
    - Faster and saves server resources
Typical types of cache for a WP site

- Browser cache
- Opcode cache
- Object cache
- Database cache
- Database engine cache (query cache, buffer pool etc.)
- Full page cache (Generated HTML)
Full page cache

- Saves the generated pages (HTML)
- Serves subsequent request to the same page from cache
- Expires after a set time (TTL) or when you update a post.
Normal request
0.5 – 5+ s

Full page cache
1 ms

Request
Web server
PHP
MySQL
Disk

Request
Page cache
That was easy!

Just use page cache for everything

Right...?
Well... Almost

This is where it starts to get a bit more complicated 😊
The problem with e-commerce sites and other sites with user specific content

- You can't serve the exact same HTML to every customer
- Carts
- Login / user details
- Personalized content
The problem with e-commerce sites
User logged in

User logged out
WooCommerce fragments to the rescue!

1. Normal request served by page cache
2. AJAX request served dynamically by WP. Updates HTML / DOM
3. Store user data in the browser's session storage
WooCommerce fragments

- AJAX request first page load of the session
- Loads user content straight from session storage in the browser for any subsequent page loads
- This is done in milliseconds after document.load() event and will not be noticable in most cases
WooCommerce fragments

- Included in WooCommerce by default
- Easy to use

```php
<?php
add_filter('add_to_cart.fragments', 'my_cart_fragment');

function my_cart_fragment( $fragments ) {
    global $woocommerce;

    ob_start();
    ?>
    <!-- Dynamic Cart HTML -->
    <?php
    $fragments['#header-cart'] = ob_get_clean();
    return $fragments;
}
```
Demo

WordCamp
Norrköping 2014
25-27 april
Högskolan i Östergötland
Session storage

- Browser support. All major browsers including IE8+.
- Covers more than 90% of the internet users.
- For unsupported browsers there will be an AJAX-request for every page load.
# Typical load times

<table>
<thead>
<tr>
<th>WC Fragment cache. Supported browser</th>
<th>First page load</th>
<th>Page ready</th>
<th>Personalized content ready</th>
<th>Subsequent page loads</th>
<th>Personalized content ready</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5s</td>
<td></td>
<td>1.8s</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WC Fragment cache. Unsupported browser</th>
<th>First page load</th>
<th>Page ready</th>
<th>Personalized content ready</th>
<th>Subsequent page loads</th>
<th>Personalized content ready</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5s</td>
<td></td>
<td>1.8s</td>
<td>0.5</td>
<td>1.8s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No page cache</th>
<th>First page load</th>
<th>Page ready</th>
<th>Personalized content ready</th>
<th>Subsequent page loads</th>
<th>Personalized content ready</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.5s</td>
<td></td>
<td>1.5s</td>
<td>1.5s</td>
<td>1.5s</td>
</tr>
</tbody>
</table>
Server performance and scalability

- Reduces server load greatly.
  - Instead of generating the full page for every request, the server only need to generate selected parts once per session or on change.

- Easier and cheaper to scale.
For purging the page cache when you save or publish a post you need two things:

- the Nginx helper plugin
  https://wordpress.org/plugins/nginx-helper/
- This Nginx module
  https://github.com/FRiCKLE/ngx_cache_purge
Cache is only fast when it's warm...
Cache is only fast when its warm

- Use a script to crawl your site to keep important pages in cache

- I use a modified version of Ramon Fincken's (MijnPress.nl) plugin “Warm Cache”
  - I will see if I can submit a pull request to this plugin. Otherwise I will just publish my code on GitHub.

http://wordpress.org/plugins/warm-cache/
Don't forget to prevent this request from being cached

Nginx

```perl
if ($arg_warm_cache != "") {
    set $skip_cache 1;
}
...

location ~ .php$ {
    ...

    fastcgi_cache_bypass $skip_cache;
    fastcgi_no_cache $skip_cache;
}
```
Some additional performance tips
Change DB Engine to InnoDB

- Much better performance, especially if you have a lot of writes

- Increase InnoDB buffer pool
  Should be able to fit most of the database in the pool (Buy more RAM!)

- The only major drawback compared to MyISAM is the lack of full text index (effects search performance)
  Can be solved by adding software like Solr or ElasticSearch or 3rd-party services like loop54.com
Do not use bad plugins

- Find bad or misbehaving plugins
  - Divide and conquer
  - P3 - Plugin Performance Profiler
  - Profiling tools like Xdebug or New Relic
- Remove plugins that isn't used
- Many plugins can be replaced by small code snippets or by external tools
Handling images

- Lossless compression of images on upload
  - EWWW Image Optimizer
    https://wordpress.org/plugins/ewww-image-optimizer/

- Are your clients uploading images straight from the camera?
  - Resize them to a sane size automatically with Imsanity
    http://wordpress.org/plugins/imsanity/
Questions?

This presentation will be available later.
I will also publish my Nginx config files.

@pelmered / Optimal Norge AS