Caching, Optimization, Scaling Big

BY DARKO GJORGJIJOSKI

WORDCAMP SKOPJE, OCTOBER, 2018

About me

Darko Gjorgjijoski

Full stack web developer & devops

https://darkog.com

WordCamp Skopje

What we will talk about?

1. Caching
   a.) Within the code
      - Non-Persistent cache
      - Persistent cache
   b.) Outside the code
      - Caching plugins
         - W3 Total Cache

What we will talk about?

2. Optimization / PageSpeed
   - WordPress plugins

3. Scaling
   - Tips
   - Possible server setup for scaling

WordCamp Skopje

Caching

a. **What is caching and how it works?**

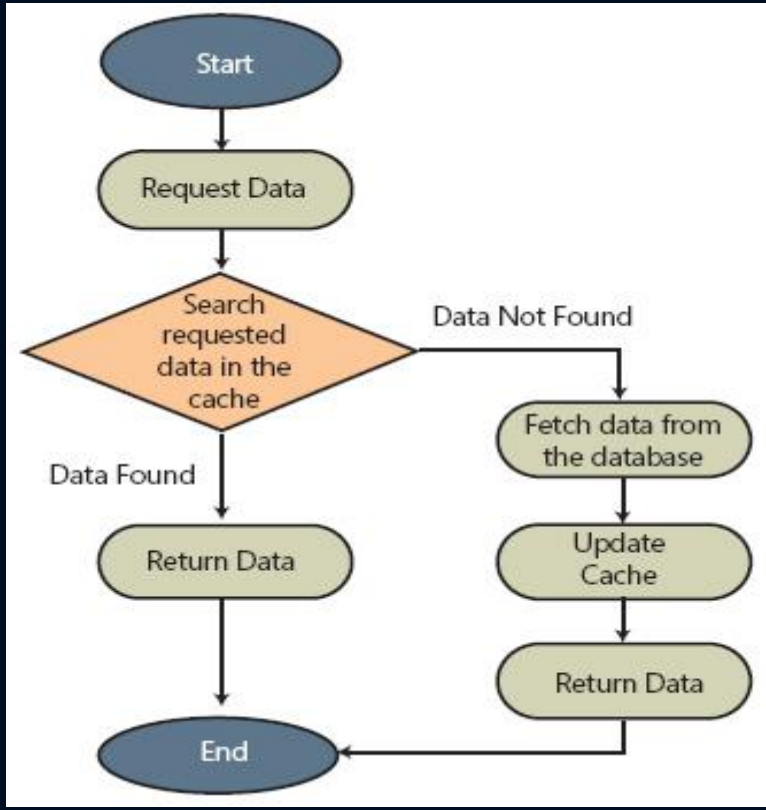
b. **Advantages of caching**

- Avoid complex/slow database SQL queries
- Less external api calls and more bandwidth saved
- Saves server resources
- Faster and better experience for the users

c. **Disadvantages**

- Outdated data sometimes
- Can be headache for developers

d. **Caching in WordPress**



WordCamp Skopje

Caching :: Within the code

Non-persistent cache

By default **WP Object Cache** class/functions is not persistent.

WP Object Cache documentation / functions

https://codex.wordpress.org/Class_Reference/WP_Object_Cache

Found in wp-includes/cache.php

```
1.  wp_cache_add($key, $value, $group = ‘’, $expire = 0)
2.  wp_cache_set($key, $data, $group = ‘’, $expire = 0)
3.  wp_cache_get($key, $group = ’’, $force = false, &$found = null)
4.  wp_cache_delete($key, $group = ’’)
5.  wp_cache_flush()
```

WordCamp Skopje

Caching :: Within the code

Persistent Cache

1. Make WP Object Cache implmenetation persistent.

   We can override the wp_cache_* functions by placing **object-cache.php** dropin in wp-content directory and provide own implementation that will be persistent and store/read the cached data on disk, redis, memached and other persistent storage.

   Some plugins already overriding the wp_cache_* functions by providing own implementation of the wp_cache_* functions through object-cache.php dropin. Such examples are W3 Total Cache, WP_SuperCache, Memcached Object Cache…

Caching :: Within the code

Persistent Cache

2. Transients API

Simple and standardized way of storing cached data in the database temporarily by giving it a custom name and a timeframe after which it will expire and be deleted.

Few notes...

- Transients are a type of cache, not data storage
- Transients are powered by the same get_option()/update_option() backend for storing permanent values and are stored in wp_options table
- Transients will expire! The expired transients will be autodeleted.
- Transients can disappear at any time, and you cannot predict when this will occur

Name	Type
option_id	bigint(20)
option_name	varchar(191)
option_value	longtext
autoload	varchar(20)

WordCamp Skopje

Caching :: Within the code

Persistent Cache

2. Transients API

Transient API documentation / functions

https://codex.wordpress.org/Transients_API

Found in wp-includes/option.php

```
1.  get_transient( $transient )
2.  set_transient( $transient, $value, $expiration )
3.  delete_transient( $transient  )
```

WordCamp Skopje

Caching :: Within the code :: Examples

1. WP Object Cache


```php
function get_movies_reviews( $genre, $year ) {
    $cache_key = 'movie_review_'. $genre . '_' . $year;
    $results   = wp_cache_get( $cache_key );
    if ( false === $results ) {
        $results = get_posts( array(
                'post_type'      => 'movie_review',
                'tax_query'      => array(
                    'relation' => 'AND',
                    array(
                        'taxonomy' => 'genre',
                        'field'    => 'slug',
                        'terms'    => $genre,
                    ),
                    array(
                        'taxonomy' => 'year',
                        'field'    => 'slug',
                        'terms'    => $year,
                    ),
                )
            )
        );
        wp_cache_set( $cache_key, $results, 'my_plugin', 60 * 30 );
    }
    return $results;
}
```

Caching :: Within the code :: Examples

2. Transients API


```
function get_movies_reviews( $genre, $year ) {
    $cache_key = 'movie_review_'. $genre . '_' . $year;
    $results   = get_transient( $cache_key );
    if ( false === $results ) {
        $results = get_posts( array(
                'post_type'      => 'movie_review',
                'tax_query'      => array(
                    'relation' => 'AND',
                    array(
                        'taxonomy' => 'genre',
                        'field'    => 'slug',
                        'terms'    => $genre,
                    ),
                    array(
                        'taxonomy' => 'year',
                        'field'    => 'slug',
                        'terms'    => $year,
                    ),
                )
            )
        );
        set_transient( $cache_key, $results, 60 * 30 );
    }
    return $results;
}
```

Diff

Diff

WordCamp Skopje

Caching :: Outside the code

1. Using plugins

- W3 Total Cache
- WP Super Cache
- WP Rocket
- Comet Cache
- WP Fastest Cache
- Redis Object Cache

2. WebServer Cache

Some web hosting companies provide their own caching systems out of the box.

3. Browser Cache

WordCamp Skopje

Optimization

1. Google PageSpeed Insights

What is covered?

- Minify CSS/JS files
- Minify HTML
- Eliminate Render-blocking JavaScript and CSS
- Optimize Images
- Enable Compression
- Leverage browser caching
- Reduce server response time

Check your site!

https://developers.google.com/speed/pagespeed/insights/

WordCamp Skopje

Optimization

PageSpeed Insights

https://wpdev2.ink/ ANALYZE

Mobile Desktop

88 / 100 Suggestions Summary

Consider Fixing:

Eliminate render-blocking JavaScript and CSS in above-the-fold content
Show how to fix

Reduce server response time
Show how to fix

Leverage browser caching
Show how to fix

Minify CSS
Show how to fix

Minify JavaScript
Show how to fix

5 Passed Rules

Show details

Download optimized image, JavaScript, and CSS resources for this page.

WordCamp Skopje

Optimization

2. WordPress Plugins for optimization

- Better WordPress Minify
- WP Optimize
- Async Javascript
- Imagify
- MegaOptim Image Optimizer
- ShortPixel Image Optimizer

WordCamp Skopje

Scaling

1. Tips

- Cache when possible (inside and outside the code)
- Keep wp_options table under control because of the autoloading (option_value is of type LONGTEXT or in numbers 4GB, be careful!).
- Keep the plugins at minimum. If you have programming experience – check the plugin code before installing it or enable WP_DEBUG to see if the newly installed plugin triggers any errors.
- Use database indexes to speed up your SQL queries where possible.
- Get appropriate server (check CPU, RAM, etc), do not use shared hosting If your site has a lot of traffic.
- Consider load balancing

wp_options table

Name	Type
option_id	bigint(20)
option_name	varchar(191)
option_value	longtext
autoload	varchar(20)

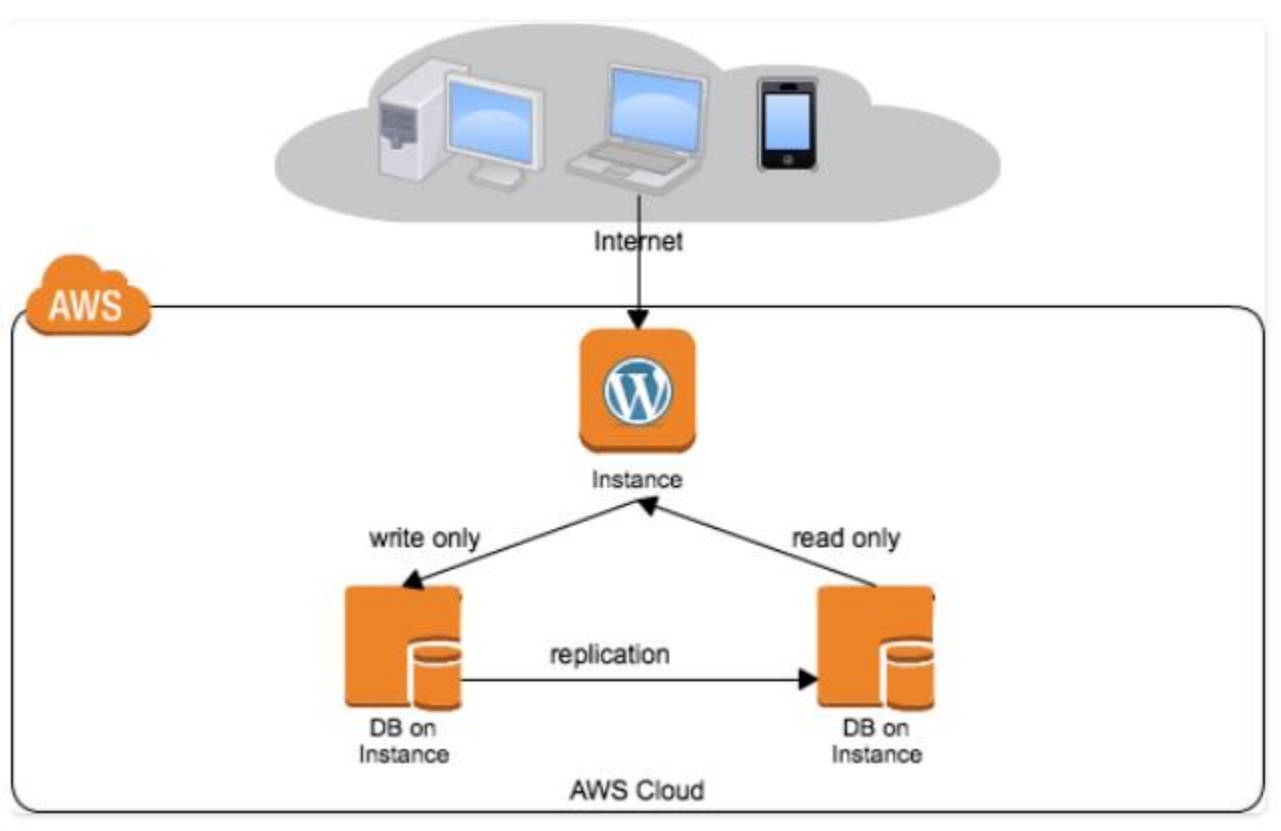
option_id	option_name	option_value	autoload
1	siteurl	http://wordcampskopje.test	yes
2	home	http://wordcampskopje.test	yes
3	blogname	Caching, Optimization, Scaling	yes
4	blogdescription	Just another WordPress site	yes
5	users_can_register	0	yes

WordCamp Skopje

Scaling

2. Loadbalancing & database replication (with HyperDB)

https://codex.wordpress.org/HyperDB
https://codex.wordpress.org/Class_Reference/wpdb



Hyper DB

- Used in WordPress.com
- Designed by automattic
- Installed via hyperdb.php dropin
- Replaces wpdb default class

WordCamp Skopje

Thank you for your attention!

Questions?

WordCamp Skopje