

TURBOCHARGING WP_QUERY

Matthew Boynes / @senyob / alleyinteractive.com



An aerial, black and white photograph of a city street intersection. Tall, multi-story buildings line the streets, with many windows visible. The street is filled with cars and trucks, some moving in traffic. A crosswalk with white stripes is visible in the foreground. The overall scene is a dense urban environment.

alleyinteractive

- We are a multi-platform dev agency
- WordPress.com VIP partner
- Hiring!

DISCLAIMER

Note: explained that this is a developer-focused topic

WHET THE APPETITE

Note: Opened an example site with 500k posts, ran a normal WP_Query in ~11 seconds and a turbocharged one in .02 seconds.



BEFORE

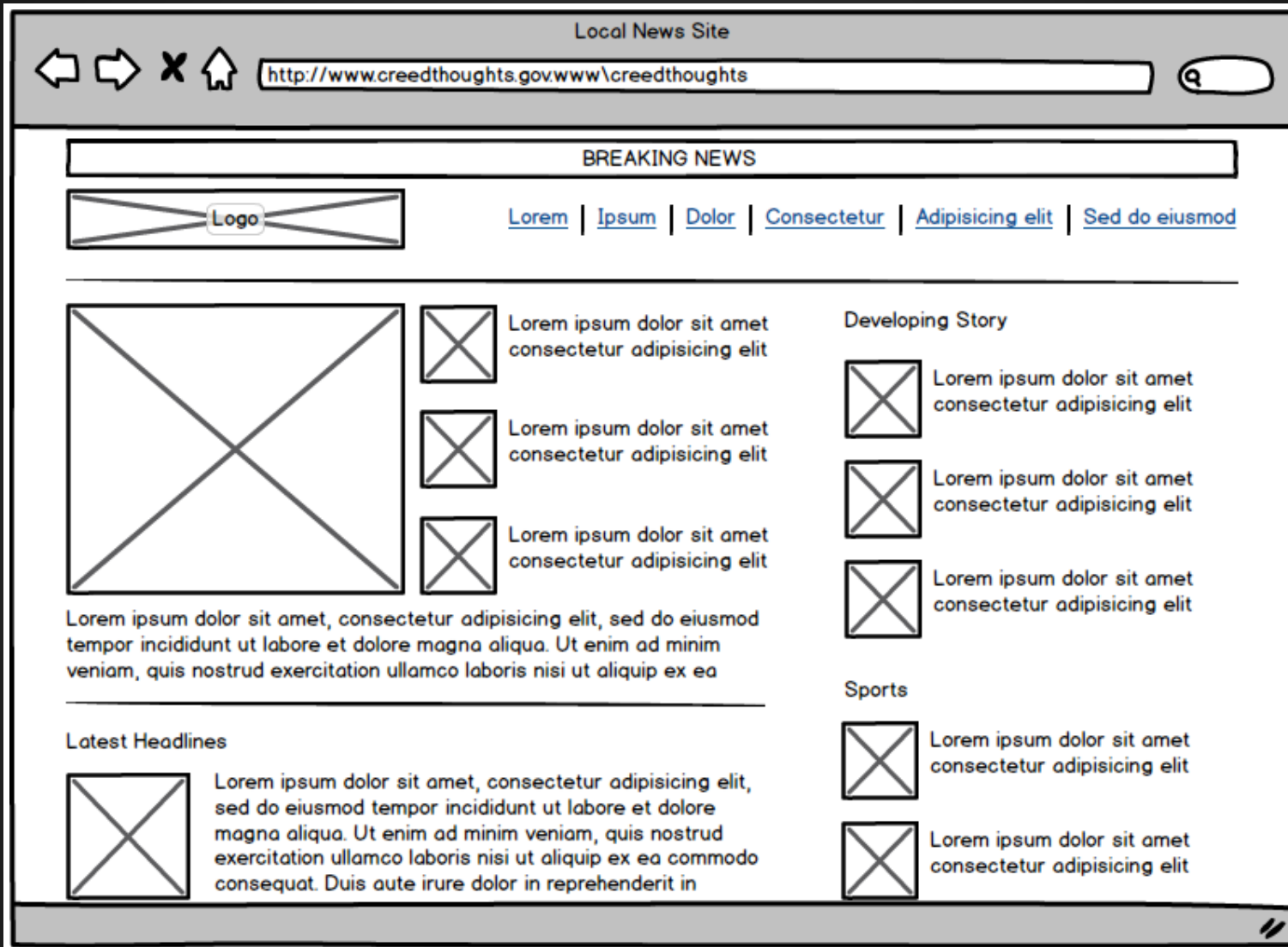


AFTER

**actual results may vary*

LET'S GO BACK TO THE BEGINNING...

- Working on a theme for a large news media company
- Theme to be used by many news sites
- Needed flexibility for complex regions of the site



LET'S GO BACK TO THE BEGINNING...

- A UI for WP_Query
- What could possibly go wrong?

PERFORMANCE CONSIDERATIONS

- Not all queries are created equal.

PERFORMANCE CONSIDERATIONS

```
get_posts( array(  
    'post__in' => array(  
        123, 456, 789  
    )  
) );
```

!=

```
get_posts( array(  
    'post__not_in' => $sticker_headlines,  
    'tax_query' => array(  
        array(  
            'field' => 'slug',  
            'taxonomy' => 'category',  
            'terms' => array( 'news', 'events', 'exclusive' ),  
            'relation' => 'OR'  
        ),  
        array(  
            'field' => 'slug',  
            'taxonomy' => 'category',  
            'terms' => array( 'sponsored', 'breaking' ),  
            'operator' => 'NOT IN',  
        ),  
        array(  
            'field' => 'slug',  
            'taxonomy' => 'post_tag',  
            'terms' => array( 'local' ),  
            'relation' => 'AND',  
        ),  
    ),  
    'meta_query' => array(  
        array(  
            'key' => 'featured',  
            'value' => '1',  
        ),  
        array(  
            'key' => '_thumbnail_id',  
            'value' => '0',  
            'compare' => '>',  
            'type' => 'NUMERIC'  
        ),  
        array(  
            'key' => 'sponsored',  
            'value' => '0',  
        ),  
    ),  
    'meta_key' => 'location',  
    'orderby' => 'meta_value',  
    'order' => 'ASC',  
) );
```

THE ROOT OF THE PROBLEM

- Flexibility
- MySQL and our data schema (e.g. wp_postmeta)

THE SOLUTION

- Ghost! </kidding>
- Complement MySQL with another technology something better at *searching*

THE SOLUTION

- "Searching" goes beyond keywords
- Structured search
- e.g. Searching for a flight
Cities, Dates, # Passengers, Layovers, Airline, ...
- WP_Query
Categories, Tags, Authors, Meta, Dates, ...

ELASTICSEARCH OVERVIEW

- What is Elasticsearch?
- Built for search
- Breaks data down into tokens

ELASTICSEARCH OVERVIEW

WordPress → word
press
wordpress

ELASTICSEARCH OVERVIEW

2014-09-06 → year: 2014
month: 9
day: 6
weekday: "sat"

ELASTICSEARCH AND WORDPRESS

- Built in Java, but has a JSON REST API
- API is simple enough to use without plugins
- Plugins can assist with indexing data
- SearchPress

POWERING WP_QUERY WITH ES

- Support native WP_Query arguments
- Build as a plugin

THE TRANSPLANT: OVERVIEW

- Started with unit tests (thanks, core!)
- ES is supplemental; MySQL is canonical
- Get post IDs from ES, get the post data from MySQL

THE TRANSPLANT: TAKE 1

Extend WP_Query

```
$query = new ES_WP_Query( array(
    'post_type' => 'event',
    'tax_query' => array( ... ),
) );

// ... or...
$posts = es_get_posts( array(
    'post_type' => 'event',
    'tax_query' => array( ... ),
) );
```

THE TRANSPLANT: TAKE 1

- Requires manual intervention in your code
- No way to automatically integrate it (e.g. based on args in `pre_get_posts`)
- If the plugin gets deactivated, everything breaks

THE TRANSPLANT: TAKE 2

Add a query var for WP_Query

```
$query = new WP_Query( array(
    'es' => true,
    'post_type' => 'event',
    'tax_query' => array( ... ),
) );

// ... or...
$posts = get_posts( array(
    'es' => true,
    'post_type' => 'event',
    'tax_query' => array( ... ),
) );
```

THE TRANSPLANT: TAKE 2

Add a query var for WP_Query

Any problems? Only 44...

```
pre_get_posts, posts_search, posts_search_orderby, posts_where, posts_join,  
    comment_feed_join, comment_feed_where, comment_feed_groupby,  
comment_feed_orderby, comment_feed_limits, posts_where_paged, posts_groupby,  
posts_join_paged, posts_orderby, posts_distinct, post_limits, posts_fields,  
posts_clauses, posts_selection, posts_where_request, posts_groupby_request,  
posts_join_request, posts_orderby_request, posts_distinct_request,  
posts_fields_request, post_limits_request, posts_clauses_request,  
posts_request, split_the_query, posts_request_ids, posts_results,  
    comment_feed_join, comment_feed_where, comment_feed_groupby,  
comment_feed_orderby, comment_feed_limits, the_preview, the_posts,  
found_posts_query, found_posts, wp_search_stopwords, get_meta_sql,  
    date_query_valid_columns, get_date_sql
```

THE TRANSPLANT: TAKE 2

Add a query var for WP_Query

- Most of those hooks are very specific to MySQL
- Lots of room for conflict

THE TRANSPLANT: CONCLUSION

- Class extension is safer
- Query var has more potential, but risky
- Decided to keep them both available
- WP_Query is too MySQL-focused

OTHER CHALLENGES

- Lack of standard ES library required lots of flexibility
- Had to re-map ES several times, e.g. add date tokens. All for the better.

CONSIDERATIONS

- The more complex the query, the better the results
- The bigger the site, the better the results
- Have to maintain another service

CONCLUSION AND REFLECTION

- Not that difficult to integrate the flavor-of-the-month data technologies
- The future: maybe a database-agnostic WP_Query method?
- Is it unreasonable to have multiple storage engines?
- Adding ES as a data layer could open up new doors for WordPress
 - Example: Jetpack now uses ES to power Related Posts

THE END

github.com/alleyinteractive/es-wp-query
github.com/alleyinteractive/searchpress

Want to do cool stuff like this?
We're hiring.

info@alleyinteractive.com

#turbotalk

alleyinteractive