Modernizing WordPress Search with Elasticsearch

Who Am I?

- My name is **Taylor Lovett**
- Director of Web Engineering at 10up
- Open source community member
- WordPress core contributor
- ElasticPress team member

10up is hiring!

@tlovett12

taylor.lovett@10up.com

Doesn’t WordPress have search built-in?

WordPress Search is Rudimentary

- Only searches post title, content, and excerpt.
- Relies on MySQL and thus is slow.
- Relevancy calculations are poor and overly simplistic.
- Not able to handle any advanced filtering.

What is **Elasticsearch**?

elasticsearch.

http://www.elasticsearch.org/

Elasticsearch

- Open-source search server written in Java based on a technology called Lucene (open-source search software by Apache).
- A **standalone database server** that provides a RESTful interface to accept and store data in a way that is optimized for search.
- Extremely **scalable**, **performant**, and **reliable**

Elasticsearch

- Relevant results
- Autosuggest
- Fuzzy matching
- Geographic searches
- Filterable searches
- Data weighting
- **Much more**

Get an Elasticsearch Server

- Very flexible and customizable. There is not really a “one size fits all” setup. Generally, you have two options:

- **Option 1:** Pay someone else to manage/host your Elasticsearch cluster (SaaS)

- **Option 2:** Host your own cluster

Elasticsearch SaaS

- found.no
- qbox.io
- heroku.com
- etc…

Let’s setup our own Elasticsearch cluster.

Elasticsearch Self-Hosted Cluster

- Make sure you have Java installed.
- Download Elasticsearch: http://www.elasticsearch.org/downloads/1-4-2/
- Assuming you are installing on a Linux distribution, you can easily install using DEB or RPM packages.

Configuring Your Cluster

Install Elasticsearch on it’s own box or VM!

Configuring Your Cluster (cont.)

- *How much memory is available to the heap?*
  Configure your Elasticsearch heap size (~half of available RAM, defaults to 256M min and 1G max):

  ES_HEAP_SIZE=512m in /etc/default/elasticsearch

- *Can the current process be swapped?*
  Disable process swapping for Elasticsearch:

  bootstrap.mlockall: true in config/elasticsearch.yml

  "Linux kernel tries to use as much memory as possible for file system caches and swaps out unused application memory, possibly resulting in the Elasticsearch process being swapped. Swapping is very bad for performance and for node stability." See: http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/setup-configuration.html.

Configuring Your Cluster (cont.)

- *How much memory can the current system user acquire?* Raise resources that system user can acquire:

  In /etc/security/limits.conf:
  elasticsearch - memlock unlimited

  In /etc/default/elasticsearch
  MAX_LOCKED_MEMORY=unlimited

- *How many files can the current system user have open?* Raise *nix file limit (65535 is ES default):

  In /etc/security/limits.conf:
  elasticsearch - nofile 65535

Configuring Your Cluster (cont.)

- *Can Elasticsearch execute dynamic scripts in queries?*
  Disable dynamic scripting (potentially dangerous and usually unnecessary)

  In /etc/elasticsearch/elasticsearch.yml
  script.disable_dynamic: true

Start Your Server

- Assuming you installed Elasticsearch as a service on a Linux/Unix distribution:

  sudo service elasticsearch start

Hello World!

- You should be able to view http://yourdomainorIP:9200 and see some JSON.
- If this doesn’t work, try SSH’ing into your ES server and doing a cURL to localhost:
  curl http://localhost:9200

Cluster Health

- View http://yourdomainorIP:9200/_cluster/health

  You will see something like this:

```
{
  "cluster_name":"elasticsearch",
  "status":"yellow",
  "timed_out":false,
  “number_of_nodes":1,
  "number_of_data_nodes":1,
  “active_primary_shards":5,
  "active_shards":5,
  "relocating_shards":0,
  "initializing_shards":0,
  "unassigned_shards":5
}
```

Cluster Health (cont.)

- **“active_primary_shards”**: Number of primary (non-replica) shards that are active and available.

- **shard**: Every document (post) lives on a shard. Indexes are divided into shards (5 by default). There are primary and replica shards. Replica shards help with failover. Shards are distributed across nodes evenly and help with parallel processing. A shard replica must live on a different node than the primary.

Cluster Health (cont.)

- **“cluster_name”**: Elasticsearch is meant to be a *distributed* search database. A cluster is one or more nodes.

- **Node:** A single server on your cluster. Nodes can be set up to store data or as master nodes.

- **“number_of_nodes”**: This is the number of nodes in your cluster.

Cluster Health (cont.)

- **"status"**: There are three statuses possible:

  Red: Not all primary shards are available

  Yellow: Primary shards are available but replicas are not.

  Green: Cluster is fully operational with primary and replica shards

Cluster Health (cont.)

- Cluster status is yellow because there is only one node. Replica shards need to be assigned to a different node than their primary; this is not possible in our current setup.
- It's good practice to create at least 2 nodes in our production clusters.

Security

- Elasticsearch has no concept of security, authentication, or authorization built-in.
- By default, your instance is open to the world. **Good** for development, but **bad** for production.

Private Instance

- We can easily lock down your ES instance

  In /etc/elasticsearch/elasticsearch.yml
  network.host: localhost

- *Note:* this will make it difficult to navigate your ES instance in your web browser which is useful for debugging. We can also do something like setup a reverse proxy (using Nginx) with a password protected endpoint to get around this.

Now that we have gone through Elasticsearch basics, let’s talk about integrating Elasticsearch with WordPress.

What is **ElasticPress**?

https://github.com/10up/elasticpress

ElasticPress

A free 10up WordPress plugin that powers WordPress search with Elasticsearch.

ElasticPress

- Build filterable performant queries (filter by taxonomy term, post meta key, etc.)
- Fuzzy search post title, content, excerpt, taxonomy terms, post meta, and authors
- Search across multiple blogs in a multi-site instance
- Results returned by relevancy. Relevancy calculations are highly customizable.
- Very extensible and performant

ElasticPress Requirements

- WordPress 3.7+
- A host (not WordPress.com VIP) that either gives SSH access or the ability to run WP-CLI commands.
- An instance of Elasticsearch.
- WP-CLI

Installation

- **Github:** http://github.com/10up/elasticpress
- **WordPress.org:** http://wordpress.org/plugins/elasticpress

Point to Your ES Instance

- In your **wp-config.php** file, add the following where the URL is the domain or IP of your Elasticsearch instance:

```
define( 'EP_HOST', 'http://192.168.50.4:9200' );
```

Index Your Posts

- Just activating the plugin will do nothing. We need to run a WP-CLI command:

  ```
  wp elasticpress index --setup [--network-wide]
  ```

- **--network-wide** will force indexing across all the blogs on a network of sites in multisite. This is required if you plan to do cross-site search. It's basically a shortcut so you don't have to run the command once with the **--url** parameter for each of your blogs.

What Happens Next?

- Once ElasticPress is activated and posts are indexed, it will integrate with **WP_Query** to run queries against Elasticsearch instead of MySQL.
- WP_Query integration will only happen on search queries (**“s” parameter**) or when the **ep_integrate** parameter is passed.

Query Integration

```
new WP_Query( array(
   ’s’ => ‘search terms’,
   ‘author_name’ => ‘taylor’,
   …
) );

new WP_Query( array(
   ’ep_integrate’ => ‘true’,
   ‘author_name’ => ‘taylor’,
   …
) );

new WP_Query( array(
   ‘author_name’ => ‘taylor’,
   …
) );
```

Note on Search

- We can use ElasticPress to power queries OTHER than search! This is **powerful**.

Advanced Queries

- ElasticPress uses Elasticsearch to do many cool types of queries in a performant manner. By passing special params to a WP_Query object we can:
    - Search taxonomy terms
    - Filter by taxonomy terms (unlimited dimensions)
    - Search post meta
    - Filter by post meta (unlimited dimensions)
    - Search authors
    - Filter by authors
    - Search across blogs in multisite
    - *more!*

Example Queries

```
new WP_Query( array(
   ’s’ => ‘paris france’,
   ‘sites’ => ‘all’,
) );
```

Example Queries

```
new WP_Query( array(
   ’s’ => ‘paris france’,
   ‘search_fields’ => array(
      ‘post_title’,
      ‘post_content’,
      ‘taxonomies’ => array( ‘category’ ),
   ),
) );
```

Example Queries

```
new WP_Query( array(
   ’s’ => ‘paris france’,
   ‘post_type’ => ‘page’,
   ‘author_name’ => ‘taylor’,
   ‘search_fields’ => array(
      ‘post_title’,
      ‘post_content’,
      ‘meta’ => array( ‘city_name’ ),
   ),
) );
```

Example Queries

```
new WP_Query( array(
   ’s’ => ‘paris france’,
   ‘tax_query’ => array(
      array(
         ‘taxonomy’ => ‘category’,
         ‘terms’ => array( ‘term1’, ‘term2’ ),
      ),
   ),
   ‘search_fields’ => array(
      ‘post_title’,
      ‘post_content’,
      ‘meta’ => array( ‘city_name’ ),
      ‘author_name’,
   ),
   ‘site’ => 3,
) );
```

WP_Query Integration

- The goal for ElasticPress is to make WP_Query work behind the scenes through Elasticsearch and provide extra functionality.
- This means we have to support every query parameter which isn’t the case yet. Github contains a full list of parameters WP_Query supports with ElasticPress and usage for each parameter.

Elasticsearch in Your Language

- Elasticsearch is designed to be internationalized.
- Out of the box, it will work fine with most languages.

Analysis and Analyzers

- When a document is indexed in Elasticsearch, text is analyzed, broken into terms (tokenized), and normalized with token filters.

- In normalization, strings might be lowercased and plurals stripped.

Custom Analyzers

- ElasticPress by default uses a pretty standard set of analyzers intended for the English language.
- We can easily customize our analyzers for use with other languages by filtering **ep_config_mapping** (see EP source code).
- You can read about language specific analyzers here:

  http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/analysis-lang-analyzer.html

Documentation

Full documentation with installation instructions:

https://github.com/10up/ElasticPress

Feedback and Continuing Development

- If you are using ElasticPress on a project, please let us know and give us feedback!
- Pull requests are welcome!

Questions?

@tlovett12

taylor.lovett@10up.com

taylorlovett.com