

What's for dinner: Drupal + Apache Solr Search

Alejandro Garza

Who am I?

- Co-maintainer of some Solr-related modules
- Working with Drupal since 4.7
- Support Engineer at Acquia specializing in Search/Solr issues

- <https://www.drupal.org/u/janusman>

Agenda

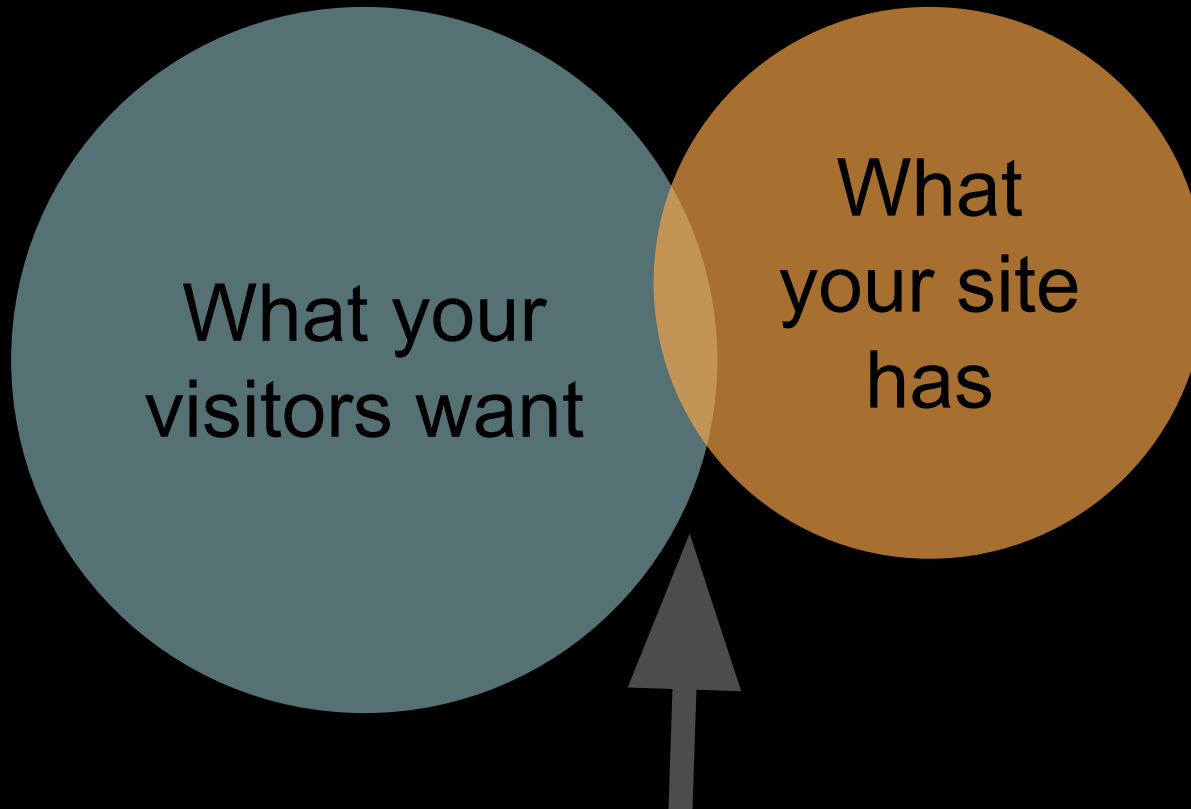
- Search [in general] is hard
- Let's experiment with Solr
- Drupal and Solr examples

Show of hands?

- What are **you** interested in?

Search = hard

Search = hard



Are they finding items in
your site?

- **Search = UX/IA**
- Is your content fielded/categorized?
 - Does your navigation + search use this?
- Are you exposing tasty data to Google?
- Do you have metrics of how your site is doing?

Google vs. site search

Google

- #1 traffic generator?
- General matching
- Targetted?
- “Secret sauce”

Site search

- Control what is matched
- Targetted
- What’s your secret sauce?
 - Use your metadata!
 - Know thy customer!

The Solr Secret Sauce

- Processing
 - tokenizing
 - synonyms, stopwords
 - stemming & other linguistic processing
 - <https://cwiki.apache.org/confluence/display/solr/Filter+Descriptions>
- Complex querying/boosting
 - per-field matching a query
 - phrase vs non-phrase matches
 - function queries

Query for "mexico books".

Search

Block provided by solr_devel.module and placed in the "content" region.

Filter by document type:

- Book (612)
- DVD (42)
- VHS (11)
- Manuscript (3)
- Projected medium (2)
- Periodical (1)

Filter by genre:

- Diversity resources, Latin and Hispanic American (169)
- Exhibitions (123)
- Diversity resources, African American (7)
- Videorecordings (53)
- Nonfiction films (34)
- Documentary films (33)
- Pictorial works (27)
- Catalogs (22)
- Biography (20)
- Feature films (19)
- Fiction films (16)
- Drama (14)

Solr query debugger

This explains how the query terms are processed and what fields they are searched in.

```

... (Array, 1 element)
  apachesolr@acquia_search_server_1 (Array, 2 elements)
    debug (Object) stdClass
      rawquerystring (String, 39 characters) mexico books "mexico books"~1000000^8.0
      querystring (String, 39 characters) mexico books "mexico books"~1000000^8.0
      parsedquery (String, 596 characters) +((DisjunctionMaxQuery((content:mexico^40.0 | t...
      parsedquery_toString (String, 512 characters) +(((content:mexico^40.0 | tm_vid_6_names:mexico...
      explain (Object) stdClass
        ih71re/node/148 (String, 5322 characters) 1.4323006 = (MATCH) sum of: 1.4323006 = (MAT...
        1.4323006 = (MATCH) sum of:
          1.4323006 = (MATCH) sum of:
            0.012575535 = (MATCH) max plus 0.01 times others of:
              1.4173794E-4 = (MATCH) weight(content:mexico^40.0 in 147), product
              8.4713567E-4 = queryWeight(content:mexico^40.0), product of:
                40.0 = boost
                1.0303891 = idf(docFreq=712, maxDocs=735)
                2.0553782E-5 = queryNorm
              0.16731434 = (MATCH) fieldWeight(content:mexico in 147), product
              1.7320508 = tf(termFreq(content:mexico)=3)
              1.0303891 = idf(docFreq=712, maxDocs=735)
              0.09375 = fieldNorm(field=content, doc=147)
            0.012568638 = (MATCH) weight(tm_vid_6_names:mexico^840.0 in 147), p
            0.024055477 = queryWeight(tm_vid_6_names:mexico^840.0), product of
              840.0 = boost
              1.3932946 = idf(docFreq=495, maxDocs=735)
              2.0553782E-5 = queryNorm
  
```

This explains how each individual item shown in the search results got ranked according to the current query.

Solr Explain output from solr_devel.module. DON'T PANIC!

- More!
 - Highlighting
 - Spellchecker
 - Elevation (a.k.a “Best Bets”)
 - “More like this”
 - Text extraction from PDF, Office docs, etc.

Solr Experimenting

Install Solr locally in 5 minutes

- Get Java
- Download Solr
- Add .txt and .xml config files from Drupal module to Solr's conf/ folder
 - Extra files available elsewhere
- Run it!
 - `java -jar start.jar`
 - `http://localhost:8983/solr/`

Solr basics

- Comparison to MySQL
 - Mysql Rows => Solr “Document”
 - Columns => “Fields”
 - Single ‘schema’, but is dynamic
 - For `text_*` definition
 - `text_1`, `text_blah`, `text_foo` all share same properties.
 - `schema.xml` defines fields

- Use it via HTTP

- HTTP GET, POST, etc.

- GET /solr/select?q=red+shoes

- POST /solr/update

- <delete><query>*:*</query></delete>

- Logging

- access, errors, etc.

Fields and their behavior

- Defined in schema.xml
 - Docs:
<http://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters>
- GUI for testing field behavior:
<http://localhost:8983/solr/admin/analysis.jsp>

Let's try stuff!

- Use our local Solr:
<http://localhost:8983/solr/>
- Recommended Chrome app:
 - Advanced REST Client

Some sample requests

Get the first 10 rows back from WHATEVER is in the index

[http://localhost:8983/solr/select/?start=0&rows=10&q=*:*](http://localhost:8983/solr/select/?start=0&rows=10&q=*:)

q=[keywords you want to search for]

&q=label:mexico+label:houses #works without qf (see below)

&q=mexico+houses #REQUIRES qf=.. (see below)

Tell Solr where it should look for the keywords given in q=

&qf=[fieldname]^[boost value]

&qf=content^40.0 #Makes matches in 'content' field mean "more"

&qf=content^40&qf=label^21.0 #You can specify multiple fields/boosts

```
# Specify fields to return:
  &fl=[fieldname,fieldname...]

# Turn on spellchecker:
  &spellcheck=true

# Add facets
  &facet=true
  &facet.field=[field1]
  &facet.field=[field2]

# Boost some things over other others.
  &bq=[fieldname:fieldvalue]^21.0

  &bq=bundle:article^21.0    #Example: make articles score
higher than others
```

Understanding processing

<http://localhost:8983/solr/admin/analysis.jsp>

Try analyzing these phrases:

- CamelCase
- Hello-world
- `www.example.com/one/two/three`
- work works working worked
- 2MB HD
- the thing and a thing with a thing
- ¡Amigo! El niño está loóking for a pixima 2 MegaPixel camera

Synonyms, protwords can help!

Align your content with what's in **their heads**.

- parking lots or parking garages??
- parks != parking
 - On noes! Stemming makes them equal! #FAIL
- Are they überspellers?
 - asymmetrical
 - epidemiology

Drupal + Solr + You ...

= ♥ ?

The Drupal Side

- 2 modules provide Drupal-Solr connection:
 - apachesolr.module
 - search_api_solr.module
- Many settings
 - Include/exclude items from index.
 - Boost items by their node type, date created, sticky, number of comments and other metadata.
 - Control boosting by text matches (per field!)

- Many (many) add-on modules
 - autocomplete
 - text extraction from PDFs/other
 - query modifiers (“more cowbell”)
 - add/alter data to be indexed
 - solr configuration generators
 - debugging

Demo Drupal + Solr

Common things that go wrong

1. Can't search or index!
 - a. Drupal not properly connected to Solr
 - b. Drupal barfs during indexing
 - c. Solr returns data that makes Drupal barf
 - d. Solr down. Check the logs!

2. Results are wrong!

- a. Item missing from results
- b. .. or, “zombie” items showing
- c. Items from another site?

Usual culprits:

- Sharing single Solr index across sites
- Mismatched expectations
 - a query matches or scores differently than what you thought.

Thank you!