Drupal and Solr
Hello

I’m Alexandru Badiu
Hello

I’m Alexandru Badiu
I come from the land of vampires
We’re going to talk about Solr
So what is Solr?

Solr is an enterprise search server
It is based on Lucene
It is an Apache Software Foundation project
It has some cool features
Why would I use it?

It is 4 - 5 times faster than the standard Drupal search
Your database will be happier
Can deliver better search results (ever used site:drupal.org in Google?)
Has replication and distributed search (for that really big content website)
Select company: CNet, Netflix, Internet Archive, Digg
Some cool features:
  - Facets
  - More options when searching
  - Geographical search
Using it: the easy way

Use the Apache Solr module

http://drupal.org/project/apachesolr

Very easy to install

You can use Tomcat, Jetty or Resin

I recommend Jetty if you have a choice
Using it: the not so easy way

Build your own app
To do that we’ll learn about Solr
Since it’s a BoF let’s get interactive
Make sure you have Java (preferably 1.5)
Download zips from http://voidberg.org/drupalcon/ from and unpack
Cd to solr/example
java -jar start.jar
Go to http://localhost:8983/solr/
If it works you just installed Solr
Let’s use it
http://localhost:8983/solr/admin and search for *:* 
This is how you query Solr
Solr concepts

You have a collection of documents
Every document has fields
You define these fields in a schema
Lots of options here
  - Data types
  - Analyzers
  - Tokenizers
  - Dynamic fields
  - Field copy
Searching

Uses the Lucene query syntax

Lots of options when searching:

- everything: keyword
- in a specific field: fieldname:keyword
- phrases: fieldname:"keyword1 keyword2"
- wildcards: key?ord and keywo*
- fuzzy: keyword~
- proximity:“keyword1 keyword2”~
- range: created:[* TO 20030101] or created:{20020101 TO NOW}
- operators:AND, -, +, NOT
Searching

You can group boolean queries

To sort add ‘sort’ to the query

sort=field1 asc, field2 desc

Pagination: ‘start’ and ‘rows’

To specify what field are to be retrieved use fl

fl=*,score
How do I add stuff?

Take a look in the exampledocs directory

Take a look at post.sh

Programatically: hook_user, hook_nodeapi, hook_update_index

Generate a document

```php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, SOLR_URL.SOLR_PATH."/update");
curl_setopt($ch, CURLOPT_POST, TRUE);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-type:text/xml; charset=utf-8'));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);
$result = curl_exec($ch);
```

$xml can also be `<commit />, <optimize /> or <delete><query>id:123</query></delete>`

Querying is similar

`wt=json&json.nl=map
indent=on`
Facets

What are facets?
You are not limited to a specific order

Solr has built in support

facet=true

facet.field=field

facet.query=field:value

facet.sort=true

facet.mincount=x
Request handlers

What are request handlers?
You’ll probably use standard, dismax, spellchecker and morelikethis
Solr has others
You can have more instances of a handler with different configurations
You can write your own too
Request handlers - Dismax

qt=dismax
Boost documents based on your interest
qf=title^2 body^1 tag^0.5
bq=cms:Drupal^2
No wildcard
To get all documents use q.alt=***
Request handlers - MoreLikeThis and SpellChecker

MoreLikeThis returns documents who are similar with the ones you specify

SpellChecker... spell checks
Caching

We all know what caching is
Caching

We all know what caching is
And now for some useless trivia
Caching

Solr has a lot of caches
Some you can influence easily, some not

filterCache
fq=query
queryResultCache
documentCache

Auto warming
Explicit warming
Caching

#!/usr/bin/python
import urllib, time, sys

def query(url):
    print url
    u = urllib.urlopen(url)
    data = u.read()
    u.close()

config = {
    'imoostiri': 'articol_data, articol_tag',
    'rez': 'tipans,judet, zona, oras, pmp, status, dcomp, pstart, ncam, stot, tipv, limba, stip, tag,',
    'ci': 'type, added, tag,',
}

warm = 'q=text:[a%20TO%20z]'

url = 'http://solrurl/%s/select?%s&fl=*&wt=python'
furl = 'http://solrurl/%s/select?%s&wt=python&facet=true&facet.field=%s&facet.zeros=true&rows=0&facet.limit=-1'

for dir, facets in config.iteritems():
    surl = url % (dir, warm)
    query(surl)

    facets = facets.split(',,')
    for facet in facets:
        facet = facet.strip()
        if facet != '':
            start = time.time()
            surl = furl % (dir, warm, facet)
            query(surl)
Geolocation and Solr

No real solution out of the box
The easy way: lat:[l1 to l2] and lon:[lo1 to lo2]
LocalSolr - port of LocalLucene
radius parameter
http://localhost:8983/localcinema/
Geolocation and Solr

C-Squares

satellite swath footprint

administrative / topographic region

predicted species distribution
Geolocation and Solr

C-Squares

Latitude 38.8894 and longitude -77.0356

* 0.0005-degree square  7307:487:380:383:495:2
* 0.001-degree square  7307:487:380:383:495
* 0.005-degree square  7307:487:380:383:4
* 0.01-degree square  7307:487:380:383
* 0.05-degree square  7307:487:380:3
* 0.1-degree square  7307:487:380
* 0.5-degree square  7307:487:3
* 1-degree square  7307:487
* 5-degree square  7307:4
* 10-degree square  7307
Geolocation and Solr

C-Squares
Add a field, csquare, allowed to have multiple values
Index all sizes of the C-Square
Convert your position to a C-Square using a “radius”
Do a search like csquare:mypos
You’ll get all documents in that square
Also there’s GeoHash
Resources

http://lucene.apache.org/solr
http://lucene.apache.org/java/docs/queryparsersyntax.html
http://wiki.apache.org/solr/
http://geohash.org/
http://sourceforge.net/projects/locallucene/
Thank you

Alexandru Badiu

http://www.voidberg.org

i@voidberg.org