Solr-ruby

the best open source ***search*** engine + ruby

rubyconf 2007

Presented by: Erik Hatcher

Solr

- Search server
- Enterprise scale (100M+ documents), very fast
- Open source: Apache Software License
- Java webapp, built on Lucene
- Features: caching, replication, faceting, highlighting, spell checking, admin interface, more...
- Very active community, evolving continuously

Solr Powered

- CNET
- Internet Archive
- Netflix
- Smithsonian
- digg
- AOL: sports and music channels
- more every day...

Lucene

- Java search engine library
- Created by Doug Cutting, renowned search engine expert
- Powers Technorati, IBM OmniFind Yahoo Edition, JIRA, Krugle, Nabble, Simpy, jGuru, Monster, Wikipedia, and many many more

Lucene

- Index Structure
    - Documents
        - Fields
            - Terms
- Relevance:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} \left(\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d) \right)$$

Solr 101: add/update

POST /solr/update

```
<add>
  <doc>
    <field name="id">MA147LL/A</field>
    <field name="name">Apple 60 GB iPod with Video Playback Black</field>
    <field name="manu">Apple Computer Inc.</field>
    <field name="cat">electronics</field>
    <field name="cat">music</field>
    <field name="features">iTunes, Podcasts, Audiobooks</field>
    <field name="features">Stores up to 15,000 songs, 25,000 photos, or 150 hours of video</field>
    <field name="features">2.5-inch, 320x240 color TFT LCD display with LED backlight</field>
    <field name="features">Up to 20 hours of battery life</field>
    <field name="features">Plays AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video</field>
    <field name="features">Notes, Calendar, Phone book, Hold button, Date display, Photo wallet, Built-in games, JPEG photo playback,
Upgradeable firmware, USB 2.0 compatibility, Playback speed control, Rechargeable capability, Battery level indication</field>
    <field name="includes">earbud headphones, USB cable</field>
    <field name="weight">5.5</field>
    <field name="price">399.00</field>
    <field name="popularity">10</field>
    <field name="inStock">true</field>
  </doc>
</add>
```

Solr 101: searching

GET /solr/select?q=ipod+AND+video&indent=on

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <!-- ... -->
    <result name="response" numFound="1" start="0">
        <doc>
            <arr name="cat">
                <str>electronics</str>
                <str>music</str>
            </arr>
            <arr name="features">
                <str>iTunes, Podcasts, Audiobooks</str>
                <str>Stores up to 15,000 songs, 25,000 photos, or 150 hours of video</str>
                <str>2.5-inch, 320x240 color TFT LCD display with LED backlight</str>
                <str>Up to 20 hours of battery life</str>
                <str>Plays AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video</str>
                <str>Notes, Calendar, Phone book, Hold button, Date display, Photo wallet, Built-in games, JPEG photo playback, Upgradeable
firmware, USB 2.0 compatibility, Playback speed control, Rechargeable capability, Battery level indication</str>
            </arr>
            <str name="id">MA147LL/A</str>
            <bool name="inStock">true</bool>
            <str name="includes">earbud headphones, USB cable</str>
            <str name="manu">Apple Computer Inc.</str>
            <str name="name">Apple 60 GB iPod with Video Playback Black</str>
            <int name="popularity">10</int>
            <float name="price">399.0</float>
            <str name="sku">MA147LL/A</str>
            <date name="timestamp">2007-10-12T16:12:59.942Z</date>
            <float name="weight">5.5</float>
        </doc>
    </result>
</response>
```

Solr 101: searching

GET /solr/select?q=ipod+AND+video&indent=on&wt=ruby

```
{
  # ...
  'response'=>{'numFound'=>1,'start'=>0,'docs'=>[
  {
   'includes'=>'earbud headphones, USB cable',
   'price'=>399.0,
   'manu'=>'Apple Computer Inc.',
   'name'=>'Apple 60 GB iPod with Video Playback Black',
   'inStock'=>true,
   'popularity'=>10,
   'weight'=>5.5,
   'id'=>'MA147LL/A',
   'sku'=>'MA147LL/A',
   'timestamp'=>'2007-10-12T16:12:59.942Z',
   'cat'=>[
    'electronics',
    'music'],
   'features'=>[
    'iTunes, Podcasts, Audiobooks',
    'Stores up to 15,000 songs, 25,000 photos, or 150 hours of video',
    '2.5-inch, 320x240 color TFT LCD display with LED backlight',
    'Up to 20 hours of battery life',
    'Plays AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video',
    'Notes, Calendar, Phone book, Hold button, Date display, Photo wallet, Built-in games, JPEG photo playback, Upgradeable firmware, USB 2
}}
```

solr-ruby

- Ruby DSL
- HTTP communication to Solr

```ruby
require 'solr'

solr = Solr::Connection.new("http://localhost:8983/solr", :autocommit => :on)

solr.add(:id => "DOC_001", :name => "Solr + Ruby: two great tastes...")
solr.add(:id => "DOC_002", :name => "... that taste great together!")

results = solr.search("taste")
puts "Found #{results.total_hits}..."
results.each do |doc|
  puts doc.inspect
end
```

solr-ruby powered

- acts_as_solr
- Flare
  - Blacklight
- Collex
- ... your app!?

Mapper

- Quack: #each
- Moo: #[]

```ruby
books = Solr::Importer::DelimitedFileSource.new("books.tsv")

mapping = {
  :id => :upc,
  :title_text => :title,
  :author_text => :author,
  :publisher_facet => :publisher,
  :language_facet => :language,
  :genre_facet => Proc.new {|r| r[:genre].split('/').map {|s| s.strip}},
  :published_year_facet => Proc.new {|r| r[:published].scan(/\d\d\d\d/)[0]},
  :source_facet => "books.tsv",
}

indexer = Solr::Indexer.new(books, mapping)
indexer.index do |record, solr_document|
  # can modify solr_document here before it is indexed
  # or log progress
end
```

acts_as_solr

```
Book.find_by_solr("pragmatic").collect {|b| b.title}
=> ["Rails Recipes (Pragmatic Programmers)",
    "Programming Ruby: The Pragmatic Programmers' Guide, Second Edition",
    "Pragmatic Version Control Using Subversion",
    "Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Apps",
    "The Pragmatic Programmer: From Journeyman to Master",
    "Best of Ruby Quiz Volume One (Pragmatic Programmers)",
    "Pragmatic Unit Testing in Java with JUnit",
    "Pragmatic Version Control Using CVS"]
```

```
Book.find_with_facet("pragmatic",
                     {:facets => {:fields => [:publisher],
                                  :zeros => false,
                                  :sort=>true}})[:facets]
 => {"facet_queries"=>{},
     "facet_fields"=>{
         "publisher_facet"=>["Pragmatic Bookshelf", 4,
                             "The Pragmatic Programmers", 3,
                             "Addison-Wesley Professional", 1]}}
```

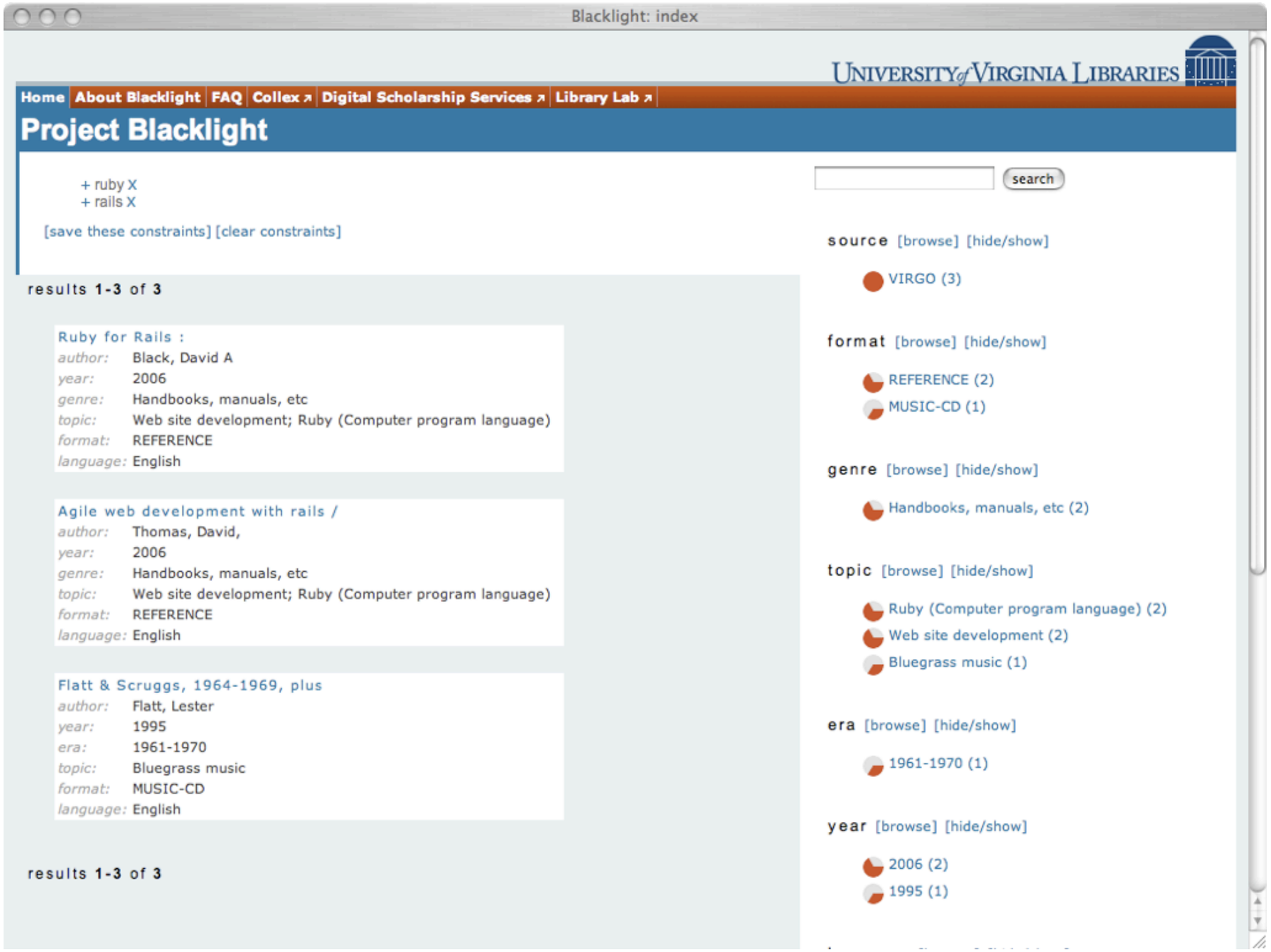
acts_as_solr_“lite”

```ruby
class Book < ActiveRecord::Base
  has_and_belongs_to_many :categories

  after_save :update_solr
  after_destroy :remove_from_solr

  SOLR_MAPPING = {
    :pk_i => :id,
    :id => Proc.new {|record| record.solr_id},
    :type_s => Proc.new {|record| record.class.name},
    :title_t => :title,
    :author_t => :author,
    :publisher_facet => :publisher,
    :year_facet => Proc.new {|record| record.published_date ? record.published_date.year : nil},
    :category_facet => Proc.new {|record| record.categories.collect {|c| c.name}}
  }

  def solr_id
    "#{self.class.name}:#{id}"
  end

  def update_solr
    solr = Solr::Connection.new("http://localhost:8982/solr")
    doc = Solr::Importer::Mapper.new(SOLR_MAPPING).map(self)
    solr.add(doc)
    solr.commit
  end

  def remove_from_solr
    solr = Solr::Connection.new("http://localhost:8982/solr")
    solr.delete(solr_id)
    solr.commit
  end

end
```

Flare

- Expose Solr through a Rails-based UI
- Constraint management
- Facet visualization
- Ajax suggest
- SIMILE integration: Timeline and Exhibit
- Status: nice for demos; needs work

“ruby” on “rails”



Future

- Solr introspection
  - schema/config savvy
  - Support upcoming query components
- Fold in acts_as_solr
- Custom Solr response writer for full data type compatibility
  - dates, ordered Hash

Help!

- DSL/API guidance
- ActiveRecord integration
- Documentation
  - Currently lame, I know. Sorry!
- Apache is new to Ruby
  - gem server?
- Flare design

Coming soon...

- Contributed: solr-ruby, acts_as_solr, and Flare recipes

Contact Info

- erik@ehatchersolutions.com
- Blog: http://code4lib.org/erikhatcher