Introduction to Apache Solr

Christos Manios
2

1. What is Solr
2. Solr Architecture / Concepts
3. Install / Configure
4. Index, Query, Update, Delete data
5. Solr integration
6. Solr resources
7. SolrCloud
WHAT IS SOLR
(and why we care so much about it!)
WHAT IS SOLR

▸ A search engine
▸ A REST API
▸ Built on Lucene
▸ Open Source
▸ Blazing-fast
▸ Scalable
▸ **Fault tolerant**
Text search faster than RDBMS
Solr knows about languages
Specific features:
  ▶ Highlighting
  ▶ Faceting
  ▶ Scoring/Boost
and many more !!
SOLR TIMELINE

1999
Doug Cutting creates Lucene

2004
Yonik Seeley creates Solr

2010
Lucene and Solr merge

2012
Solr 4 and introduction of SolrCloud

2015
Version 5.0
<table>
<thead>
<tr>
<th>LinkedIn</th>
<th>O.T.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DuckDuckGo</td>
<td>Instagram</td>
</tr>
<tr>
<td>IBM Websphere</td>
<td>Nasa</td>
</tr>
<tr>
<td>Commerce</td>
<td>Netflix</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>Disney</td>
</tr>
<tr>
<td>Apple</td>
<td>Buy.com</td>
</tr>
<tr>
<td>eBay</td>
<td>Adobe</td>
</tr>
<tr>
<td>MTV Networks</td>
<td>SAP Hybris</td>
</tr>
<tr>
<td>Magento</td>
<td>Bloomberg</td>
</tr>
</tbody>
</table>

and many more!
Does Solr fit in our application?

“Well… it depends!”
SOLR ARCHITECTURE
Standalone application server (Jetty powered)
Document oriented
Schema (less)
Not ACID (document atomicity)
Lucene/Solr Architecture

Request Handlers
- /admin
- /select
- /spell

Response Writers
- XML
- Binary
- JSON

Update Handlers
- XML
- CSV
- Binary

Search Components
- Query
- Highlighting
- Spelling
- Statistics
- Faceting
- Debug
- More like this
- Clustering

Distributed Search

Config

Update Processors
- Signature
- Logging
- Indexing

Apache Tika

Extracting Request Handler (PDF/WORD)

Data Import Handler (SQL/RSS)

Index Replication

Core Search
IndexReader/Searcher

Apache Lucene
Text Analysis

Indexing
IndexWriter
SOLR CONCEPTS AND TERMINOLOGY

- Node
- Core
- Schema
- ConfigSet
- SolrCloud
  - Collection
  - Shard
  - Zookeeper
Every Solr core has a schema
Defined in schema.xml
Contains:
- Fields
- Data types
- Analysers
Solr supports schemaless mode

Not recommended for production

Performance implications
- int, float, long, double
- date
- string
- text (multilingual ** )
- location
COMMON FIELD ATTRIBUTES

- indexed
- stored
- type
- multivalued

Example:

```xml
<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
```
- Fields not explicitly defined in schema
- Field names must match a pattern
- Field names prefixed or suffixed with a wildcard
- Make schema dynamic

```xml
<dynamicField name="*_i" type="tint" indexed="true" stored="true"/>
```
You can index one or more documents using:

- bin/post command
- REST api
- SolrJ or other libraries
- DataImportHandler
REST API example:

```
-H "Content-Type: text/xml" --data-binary '  
<add>
  <doc>
    <field name="id">012ab1</field>
    <field name="authors">Patrick Eagar</field>
    <field name="subject">Sports</field>
    <field name="dd">796.35</field>
    <field name="isbn">0002166313</field>
    <field name="yearpub">1982</field>
    <field name="publisher">Collins</field>
  </doc>
</add>'
```
REST API example:

    {
        "id": "1",
        "title": "Doc 1"
    },
    {
        "id": "2",
        "title": "Doc 2"
    }
]'
REST API example:
Solr has the following query parsers:

- Standard (lucene)
- Dismax
- Edismax
Allow the selection of documents whose fields fall within a range

Ranges with [] are inclusive at both sides
- price:[0 TO 100]
- price:[0 TO *]
- price:[* TO 100]

Range queries with {} are exclusive
- price:{0 TO 100}

Can mix { and ]
- price:[0 TO 100]
- Date format: 2015-10-16T19:19:59Z

- Dates are stored in UTC.

- Date math
  - NOW
  - NOW/YEAR
  - NOW/HOUR
  - NOW/MONTH
  - NOW/SECOND
Boolean queries:
  ▪  +this -that
  ▪  this AND that

Field queries:
  ▪  title: Bob SquarePants
  ▪  company: Nickelodeon
Phrase/proximity queries:
  ▶ "Sheldon Couper" matches only Sheldon Couper
  ▶ "Sheldon Couper"~1 matches Sheldon Lee Couper

Multi-term queries:
  ▶ title:Ιωάννης Μακρυγιάνης
  ▶ title:(Ιωάννης Μακρυγιάνης)

Combine them:
  ▶ +this -title:that +price:[* TO 100] -name:"Sheldon Couper"
Sometimes we don't know exactly what you are looking:
- It starts with pro: pro*
- It ends with tion: *tion
- Not sure about a letter: j?t

Something like chris:
- chris~
- chris~0.9

Regular expression: /H.*t/ matches Hornet
Relevancy is the quality of results returned from a query, encompassing both what documents are found, and their relative ranking (the order that they are returned to the user.)
Find all people with name “Κώστας” and return politicians first:

q=name:”Κώστας” +occupation: Politician~100
Limit the possible responses to the main query
- Do not change ordering or scoring
- Can be based on any query type

Example:
&fq=category:music
&fq=price:[0 TO 100]
&fq=rating:[3 TO *]
Solr can sort by
- Score
- A value in a field
- A function
- In ascending or descending order
- Multiple fields:

```
&sort=name asc, age desc
```
Searching: Highlighting

Repositories: 4,288
Code: 3,572,524
Issues: 8,993
Users: 2,420

Languages:
- Java: 1,183
- JavaScript: 484
- C++: 269
- PHP: 199
- Shell: 143
- Ruby: 143
- Python: 130

NeXTs/Jets.js
Native CSS search engine
Updated 2 days ago

Dopi/JetKernel
Linux kernel for the Jét (S8000)
Updated on 13 Jan 2012

chef-cookbooks/jetty
Solr performs atomic (partial) updates.
- It marks the old version of the document as deleted.
- It adds the new version of the document.
- Updates are based on the unique ID.
- Not possible to update by query.
Delete documents by query (WARNING! The following deletes all docs!!)

http://192.168.1.1:8983/solr/update?
commit=true&stream.body=<delete><query>*</query></delete>
SEARCH SPEED?

40
It depends on:
1. Document size
2. Field cardinality
3. RAM assigned to JVM
4. Indexing rate (updates / sec)
5. Query rate (queries / sec)
6. Query quality

Be careful or it will become:
INSTALL / CONFIGURE SOLR
43

Install Solr

- Download from a mirror
- Unzip
- Run

```
bob@bobos-PC$ ls solr*
solr-5.3.1.zip
bob@bobos-PC$ unzip -q solr-5.3.1.zip
bob@bobos-PC$ cd solr-5.3.1/
```
bob@bobos-PC$ /opt/solr-5.3.1 $ bin/solr start -p 8983

Waiting up to 30 seconds to see Solr running on port 8983

[//]
Started Solr server on port 8983 (pid=6240). Happy searching!

(in Windows use: bin/solr.cmd)
CREATE A NEW CORE

$ bin/solr create_core -c javameetup -d basic_configs

Setup new core instance directory:
/opt/solr-5.3.1/server/solr/javameetup
Solr official resources page provides links to:

- Tutorials
- Release documentation
- Reference guide
- Mailing lists
Solr is integrated with multiple languages via libraries:

- Java (solrj, spring-data-solr)
- Python
- PHP
- .NET
- Go

for a full list see here.
Solr can be combined with big data software such as:

- Apache Hadoop
- Apache Cassandra
- Apache Spark
- Apache Mahout
Be constantly aware of your index size:
53

2,100,000,000

maximum number of documents per core or shard

For more, consider SolrCloud solution!
- Distributed search
- Sharding
- Fault tolerance
- High availability
- Apache Zookeeper coordinates:
  - shard leader election
  - updates distribution to shard leaders
Collection with one shard
Questions?

About me:
- https://manios.org
- https://github.com/manios