Apache Solr

Sayed Anisul Hoque

Outlines

- History
- Overview
- Architecture
- Use case
- Demo

Lucene/Solr history

- Doug Cutting created Lucene in 1999.
    - Full text search engine library written entirely in Java
- Yonik Seeley created Solr in 2004 while working at CNET.
    - Enterprise search platform written in Java.
    - Web service that manages the lifecycle of documents in index.
- Became an Apache project in 2007.
- Apache Lucene and Solr projects merge in 2010.

What is Solr ?

- Java Web Server
  - HTTP interface
  - Own RESTful architecture
- Designed to manage indexes of the actual data
  - Index: Metadata useful for searching over the actual data
- NRT (Near Real-Time) index updates
- Type Flexible
  - Documents and fields
- Can be used as an embedded framework using internal API.

Why use Solr ?

- Scalable
- Ready-to-deploy
- Optimized for searching
- Text-centric
- Results sorted by relevance

[source: Solr in Action]

Who uses Solr ?

Buy.com [Rakuten Commerce LLC]
Cnet
Netflix
Disney
Apple
NASA
MTV
AWS CloudSearch
Whitehouse.gov

More: https://wiki.apache.org/solr/PublicServers

Solr Characteristics

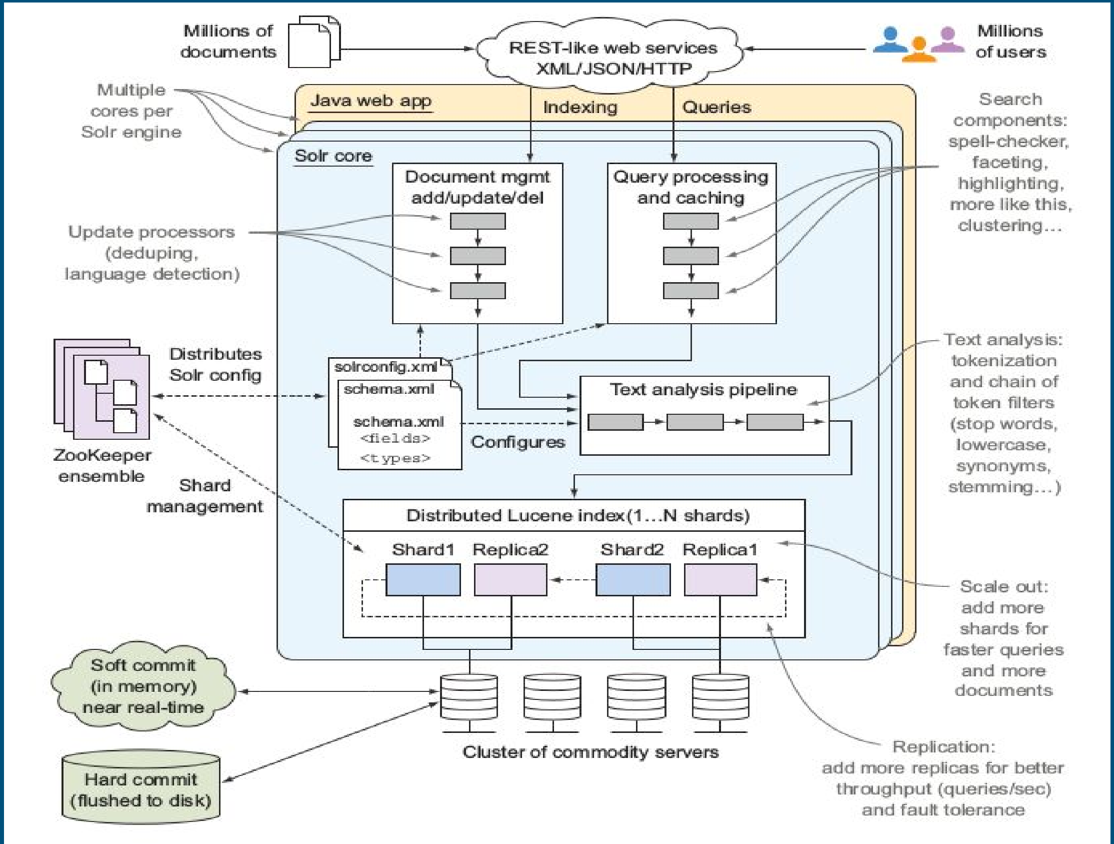
- Text-centric
- Read-dominant
- Document-oriented
- Flexible schema

[source: Solr in Action]

Solr use cases

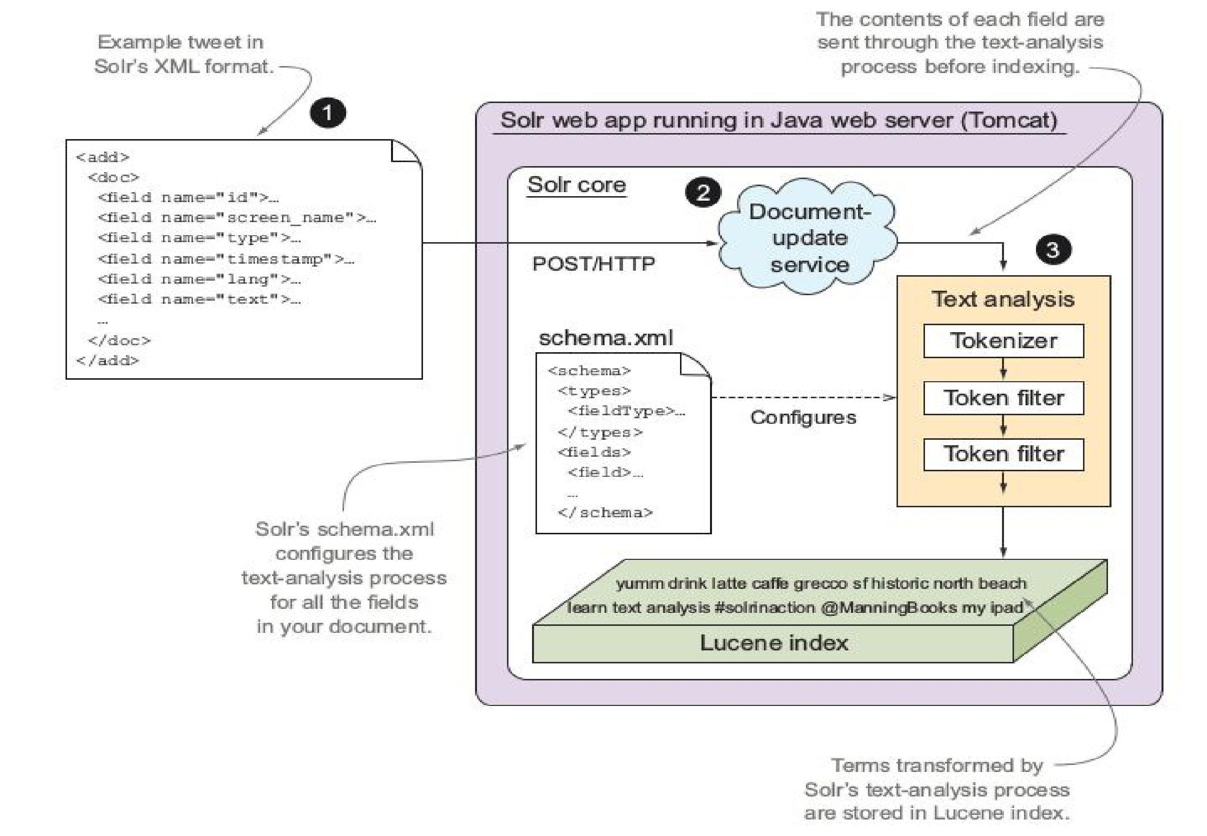
- Keyword Searching – queries of terms and boolean operators
- Ranked Retrieval – sorted by relevancy score (descending order)
- Faceting – ability to apply filter queries based on matching fields
- Result Sorting – sort the documents based on field values
- Synonyms – expand queries based on configurable definition list
- Auto-Suggestions – present list of possible query terms
- Geo-Spatial Search – locate and sort documents by distance
- Scalability – ability to break a large index into multiple shards and distribute indexing and query operations across a cluster of nodes

Solr Components


Millions of documents
REST-like web services XML/JSON/HTTP
Millions of users
Multiple cores per Solr engine
Java web app
Indexing
Queries
Search components: spell-checker, faceting, highlighting, more like this, clustering...
Solr core
Document mgmt add/update/del
Query processing and caching
Update processors (deduping, language detection)
Distributes Solr config
solrconfig.xml
schema.xml
schema.xml
<fields>
<types>
Configures
Text analysis pipeline
Text analysis: tokenization and chain of token filters (stop words, lowercase, synonyms, stemming...)
ZooKeeper ensemble
Shard management
Distributed Lucene index(1...N shards)
Shard1
Replica2
Shard2
Replica1
Scale out: add more shards for faster queries and more documents
Soft commit (in memory) near real-time
Hard commit (flushed to disk)
Cluster of commodity servers
Replication: add more replicas for better throughput (queries/sec) and fault tolerance

Source: Solr in Action

Indexing Process



Example tweet in Solr's XML format.

```
<add>
 <doc>
  <field name="id">…
  <field name="screen_name">…
  <field name="type">…
  <field name="timestamp">…
  <field name="lang">…
  <field name="text">…
  …
 </doc>
</add>
```

```
<schema>
 <types>
  <fieldType>…
 </types>
 <fields>
  <field>…
  …
 </schema>
```

The contents of each field are sent through the text-analysis process before indexing.

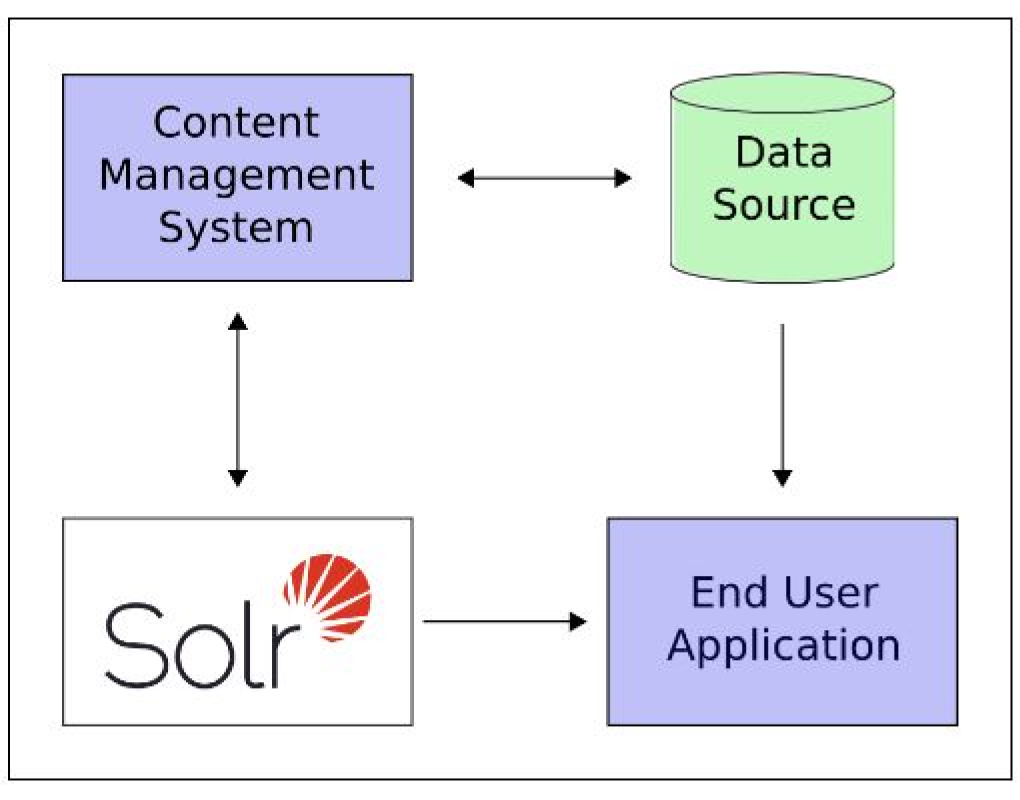
Solr's schema.xml configures the text-analysis process for all the fields in your document.

Terms transformed by Solr's text-analysis process are stored in Lucene index.

1
2
3
Solr web app running in Java web server (Tomcat)
Solr core
Document-update service
POST/HTTP
schema.xml
Configures
Text analysis
Tokenizer
Token filter
Token filter
yumm drink latte caffe grecco sf historic north beach
learn text analysis #solrinaction @ManningBooks my ipad
Lucene index

Source: Solr in Action

Solr Integrations



Source: Solr wiki

Indexing Wikipedia

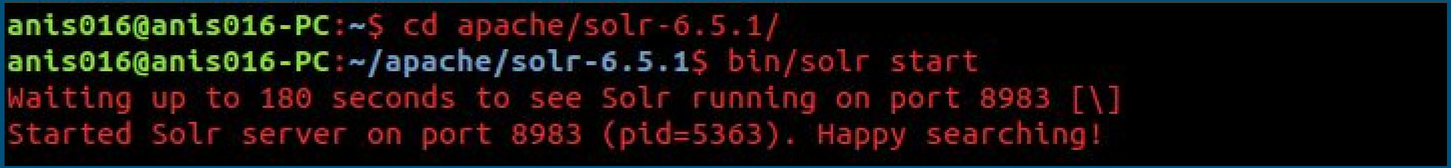
- Wikipedia XML files
  - HTTP Interface
- DataImportHandler library
  - Relational database
  - XML Files

Indexing Wikipedia

- Download Apache Solr
  - Link: lucene.apache.org/solr [version: 6.X]
  - Requirements: Java version 8 or greater

Indexing Wikipedia

- Run the solr instance
  - $ bin/solr start

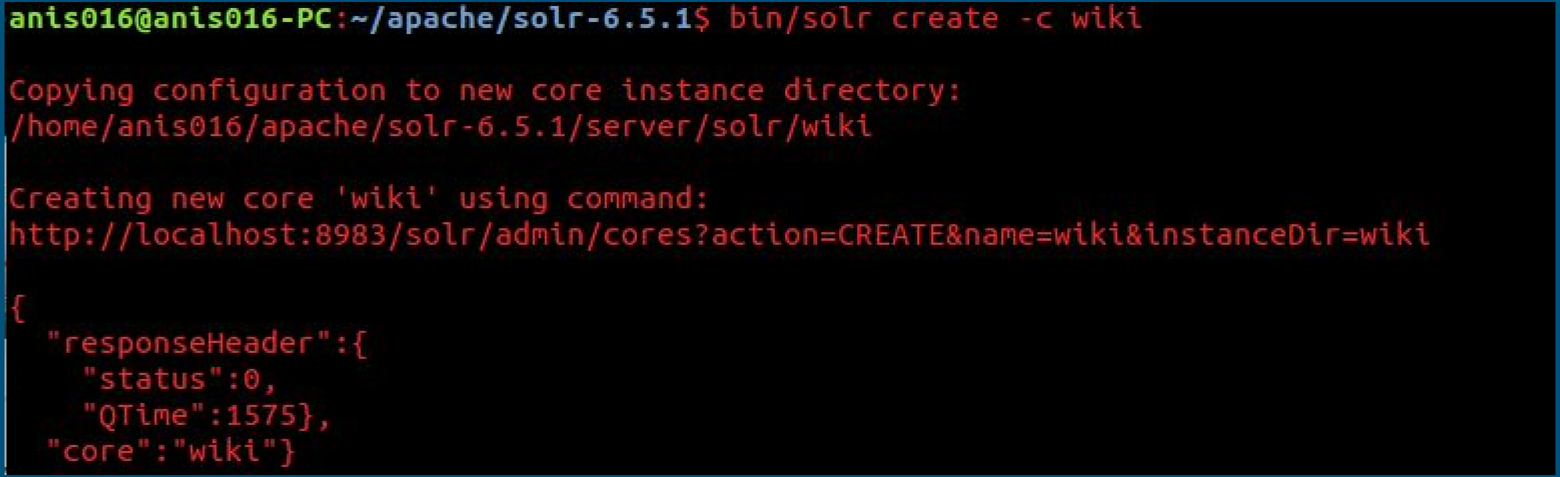


```
anis016@anis016-PC:~$ cd apache/solr-6.5.1/
anis016@anis016-PC:~/apache/solr-6.5.1$ bin/solr start
Waiting up to 180 seconds to see Solr running on port 8983 [\]
Started Solr server on port 8983 (pid=5363). Happy searching!
```

  - Browse the Solr web panel
    - localhost:8983
    - Core: Solr collection of documents.

Indexing Wikipedia

- Create a Core
  - $ bin/solr create -c <collection-name>



```
anis016@anis016-PC:~/apache/solr-6.5.1$ bin/solr create -c wiki

Copying configuration to new core instance directory:
/home/anis016/apache/solr-6.5.1/server/solr/wiki

Creating new core 'wiki' using command:
http://localhost:8983/solr/admin/cores?action=CREATE&name=wiki&instanceDir=wiki

{
  "responseHeader":{
    "status":0,
    "QTime":1575},
  "core":"wiki"}
```

Indexing Wikipedia

- Get the wikipedia dumps
  - https://dumps.wikimedia.org/enwiki/latest/
- Configure solr files
  - solrconfig.xml [Runtime Configurations]
  - data-config.xml [Data Configuration]
  - schema.xml [defines schema(content of docs)]

Indexing Wikipedia

- Restart server
  - $ bin/solr stop -all
  - $ bin/solr start
- Call RequestHandler in browser
  - localhost:8983/solr/wiki/dataimport?command=full-import
  - localhost:8983/solr/wiki/dataimport [status]
- Query in the search panel.