



# Introduction to Apache Solr

Christos Manios

Thessaloniki Java Meetup  
2015-10-16

# 2

## Contents

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



3

# WHAT IS SOLR

(and why we care so much about it!)

# 4

## WHAT IS SOLR

- ▶ A search engine
- ▶ A REST API
- ▶ Built on Lucene
- ▶ Open Source
- ▶ Blazing-fast
- ▶ Scalable
- ▶ Fault tolerant

# 5

## WHY SOLR

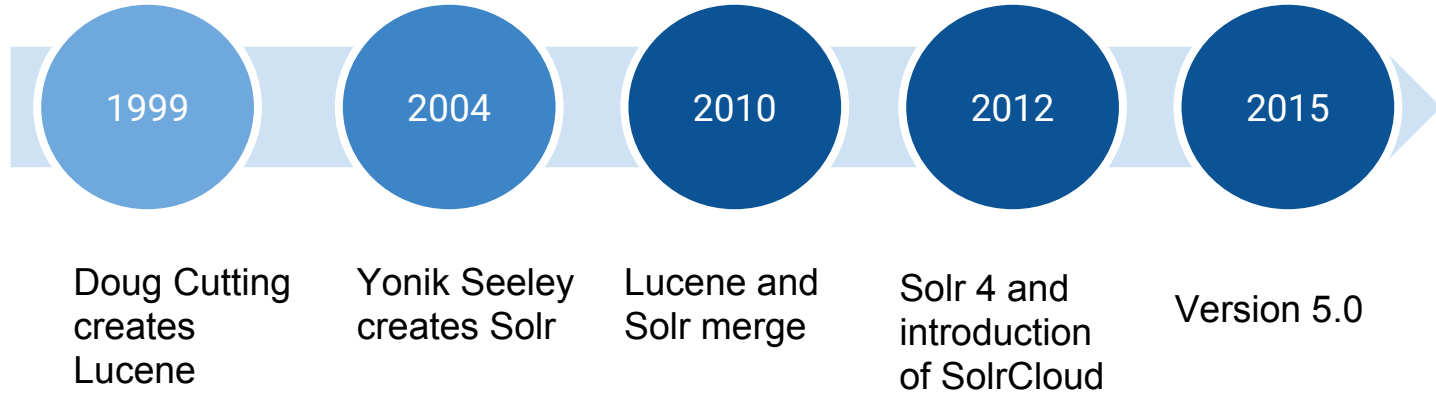
- ▶ Text search faster than RDBMS
  - ▶ Solr knows about languages
  - ▶ Specific features:
    - ▷ Highlighting
    - ▷ Faceting
    - ▷ Scoring/Boost
- and many more !!

# 6

## SOLR TIMELINE

*Lucene*

Solr 



# 7

## WHO USES SOLR

LinkedIn  
DuckDuckGo  
IBM Websphere  
Commerce  
AT&T  
Apple  
eBay  
MTV Networks  
Magento

O.T.S.  
Instagram  
Nasa  
Netflix  
Disney  
Buy.com  
Adobe  
SAP Hybris  
Bloomberg

and many more!

8



**Does Solr fit in our  
application?**

**“Well... it depends!”**





9

# SOLR ARCHITECTURE

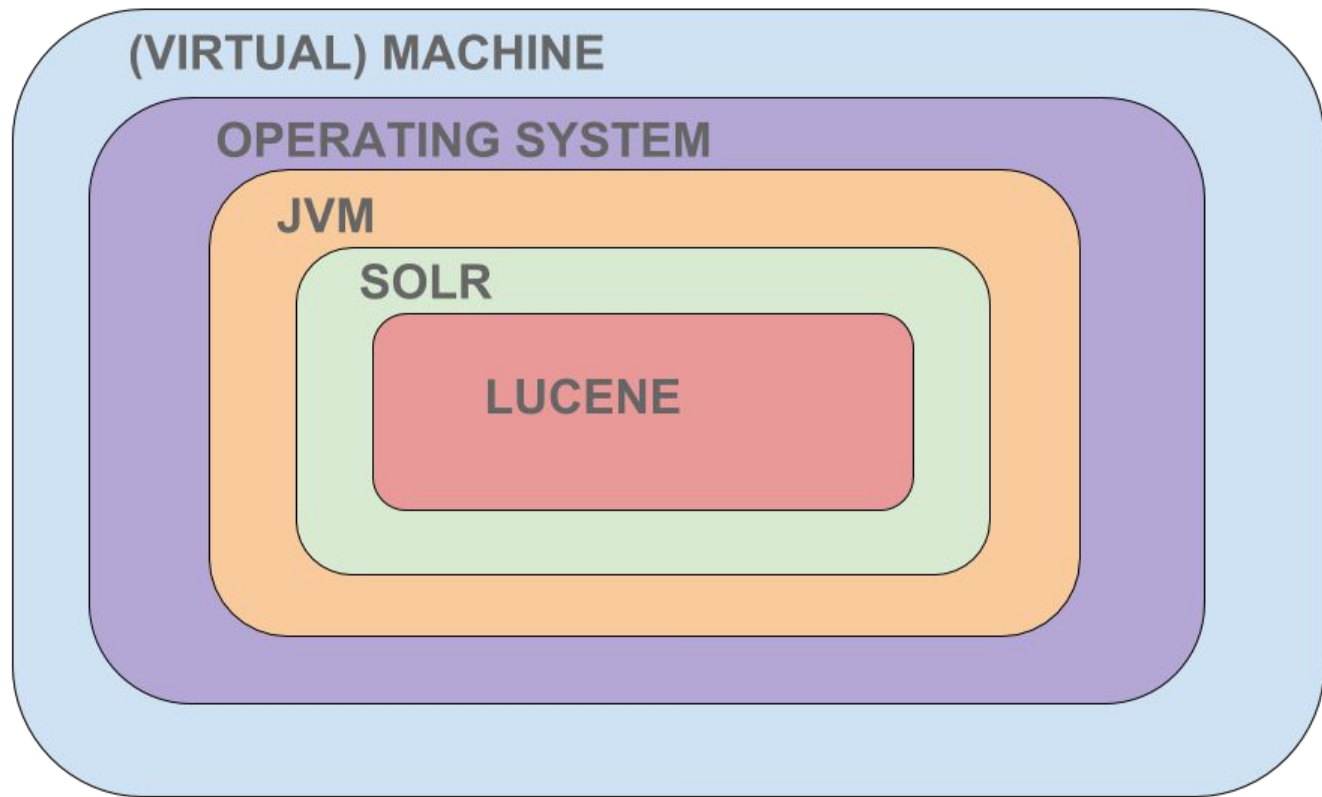
# 10

## BASIC CONCEPTS

- ▶ Standalone application server (Jetty powered)
- ▶ Document oriented
- ▶ Schema (less)
- ▶ Not ACID (document atomicity)

# 11

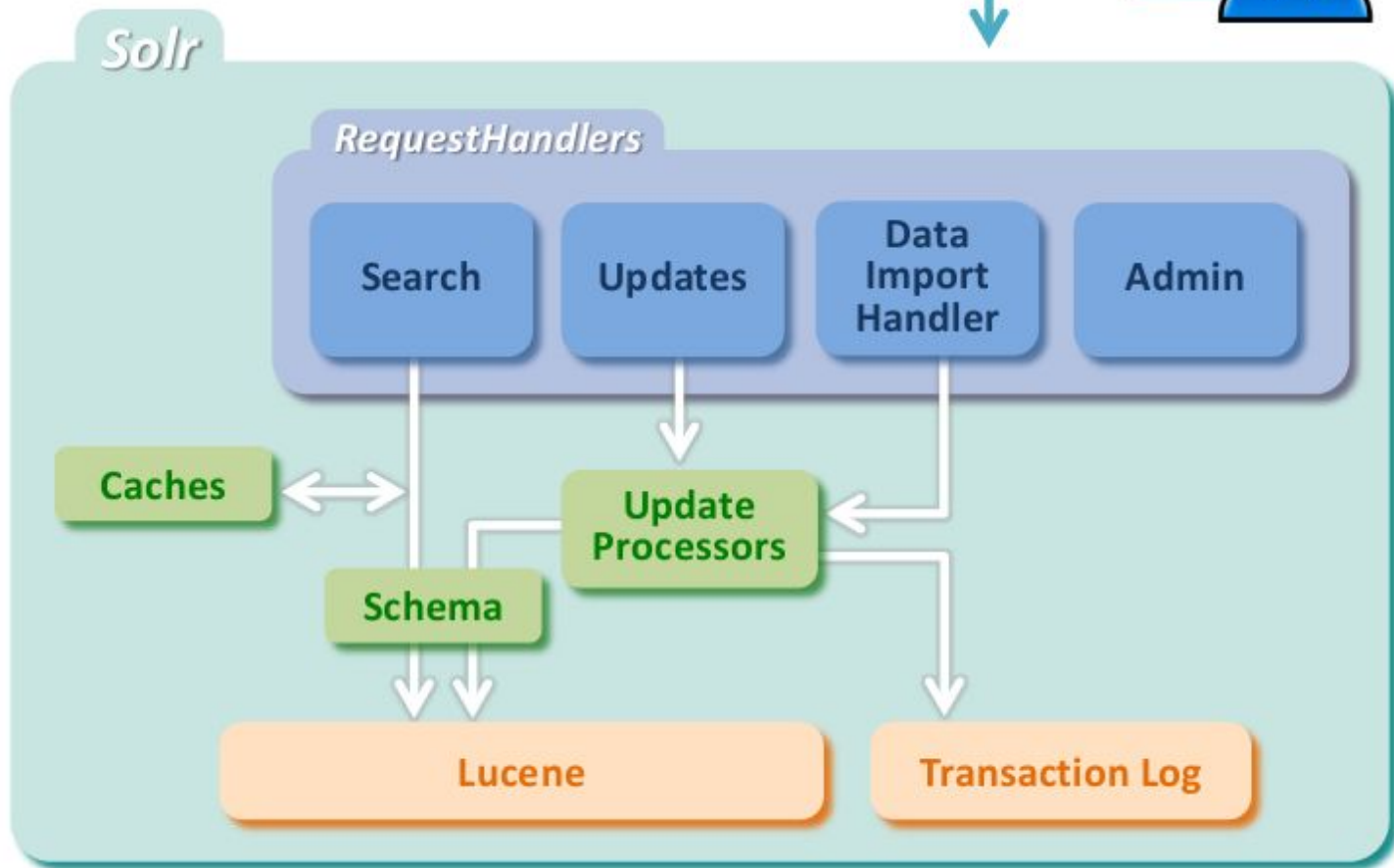
## ARCHITECTURE



# 12

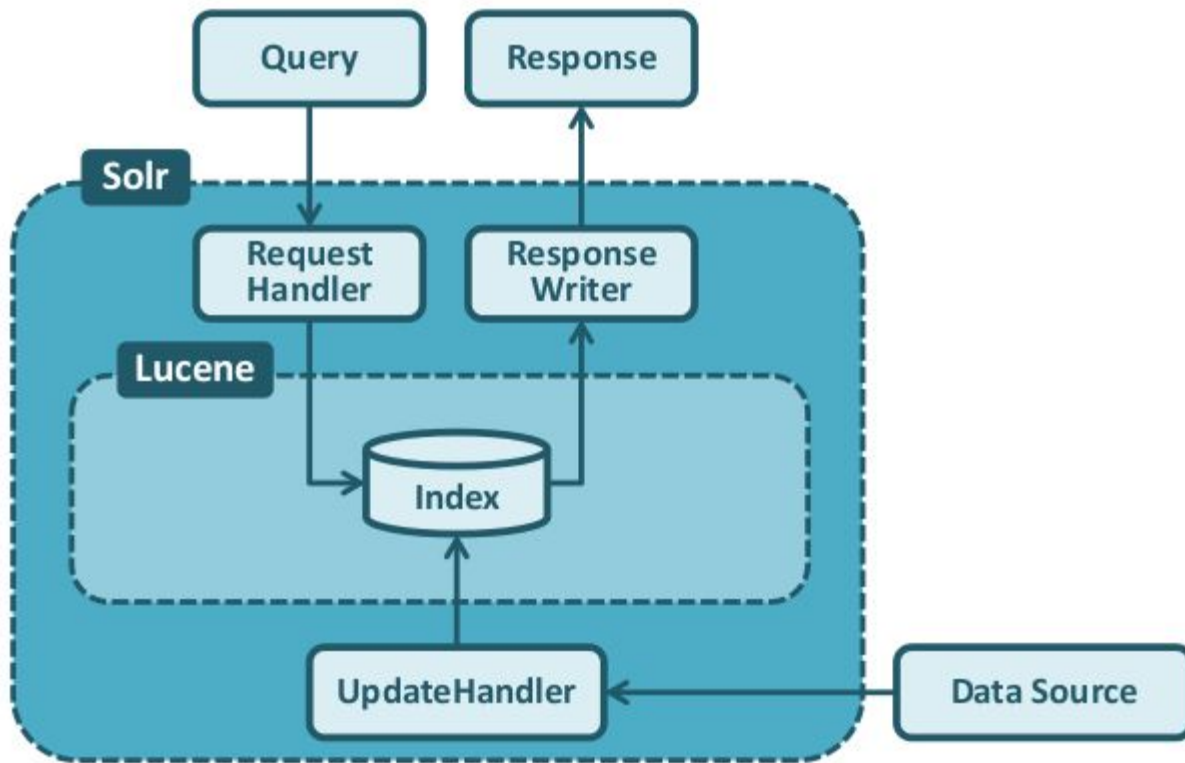
## ARCHITECTURE (2)

## Solr architecture



# 13

## ARCHITECTURE (3)



# 14

## ARCHITECTURE (4)

# 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

Schema

Config

### Update Processors

Signature

Logging

Indexing

Query  
Parsing

Analysis

High-  
lighting

Extracting  
Request  
Handler  
(PDF/WORD)

Apache Tika

Data Import  
Handler  
(SQL/RSS)

Index  
Replication

Faceting

Filtering

Search

Caching

Core Search  
IndexReader/Searcher

Apache Lucene  
Text Analysis

Indexing  
IndexWriter

# 15

## SOLR CONCEPTS AND TERMINOLOGY

- ▶ Node
- ▶ Core
- ▶ Schema
- ▶ ConfigSet
- ▶ SolrCloud
  - ▷ Collection
  - ▷ Shard
  - ▷ Zookeeper

# 16

## SCHEMA

- ▶ Every Solr core has a schema
- ▶ Defined in schema.xml
- ▶ Contains:
  - ▷ Fields
  - ▷ Data types
  - ▷ Analysers



# 17

## MANAGED SCHEMA

- ▶ Solr supports schemaless mode
- ▶ Not recommended for production
- ▶ Performance implications

# 18

## FIELD TYPES

- ▶ int, float, long, double
- ▶ date
- ▶ string
- ▶ text (multilingual \*\* )
- ▶ location

# 19

## COMMON FIELD ATTRIBUTES

- ▶ indexed
- ▶ stored
- ▶ type
- ▶ multivalued

Example:

```
<field name="id" type="string" indexed="true"  
stored="true" required="true" multiValued="  
false" />
```

# 20

## DYNAMIC FIELDS

- ▶ Fields not explicitly defined in schema
- ▶ Field names must match a pattern
- ▶ Field names prefixed or suffixed with a wildcard
- ▶ Make schema dynamic

```
<dynamicField name="*_i" type="tint"  
indexed="true" stored="true"/>
```

21

**INDEXING/  
SEARCHING /  
UPDATING /  
DELETING**

# 22

## INDEXING

You can index one or more documents using:

- ▶ bin/post command
- ▶ REST api
- ▶ SolrJ or other libraries
- ▶ DataImportHandler

# 23

## INDEXING (2)

### REST API example:

```
curl http://localhost:8983/solr/my_collection/update  
-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>'
```

# 24

## INDEXING MULTIPLE DOCS (JSON)

### REST API example:

```
curl -X POST -H 'Content-Type: application/json' 'http://localhost:8983/solr/my_collection/update' --data-binary '
```

```
[  
  {  
    "id": "1",  
    "title": "Doc 1"  
  },  
  {  
    "id": "2",  
    "title": "Doc 2"  
  }  
]
```



# 25

## SEARCHING

### REST API example:

```
curl http://192.168.1.2:8983/solr/javameetup/select?q=%3A*  
&sort=creatorName_txtel_diav+desc  
&rows=10  
&fl=id  
&wt=json  
&indent=true
```

# 26

## SEARCHING: QUERY PARSERS

Solr has the following query parsers:

- ▶ Standard (lucene)
- ▶ Dismax
- ▶ **Edismax**

# 27

## SEARCHING: RANGE QUERIES

- ▶ 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}`

# 28

## SEARCHING: DATE QUERIES

- ▶ Date format: 2015-10-16T19:19:59Z
- ▶ Dates are stored in UTC.
- ▶ Date math
  - ▶ NOW
  - ▶ NOW/YEAR
  - ▶ NOW/HOUR
  - ▶ NOW/MONTH
  - ▶ NOW/SECOND

# 29

## SEARCHING: OTHER QUERIES

- ▶ Boolean queries:
  - ▷ `+this -that`
  - ▷ `this AND that`
- ▶ Field queries:
  - ▷ `title: Bob SquarePants`
  - ▷ `company: Nickelodeon`

# 30

## SEARCHING: OTHER QUERIES (2)

- ▶ Phrase/proximity queries:
  - ▷ `"Sheldon Couper"` matches only Sheldon Couper
  - ▷ `"Sheldon Couper"~1` matches Sheldon Lee Couper
- ▶ Multi-term queries:
  - ▷ `title:Ιωάννης Μακρυγιάννης`
  - ▷ `title:(Ιωάννης Μακρυγιάννης)`
- ▶ Combine them:
  - ▷ `+this -title:that +price:[* TO 100] -name:"Sheldon Couper"`

# 31

## SEARCHING: FUZZY & WILDCARD QUERIES

- ▶ 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

# 32

## SEARCHING: RELEVANCY

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.)



# 33

## SEARCHING: RELEVANCY EXAMPLE

- ▶ Find all people with name “Κώστας” and return politicians first:
  - ▶ `q=name:”Κώστας” +occupation:Politician~100`

# 34

## SEARCHING: FILTER QUERIES

- ▶ 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 *]`

# 35

## SEARCHING: SORTING

- ▶ Solr can sort by
  - ▷ Score
  - ▷ A value in a field
  - ▷ A function
- ▶ In ascending or descending order
- ▶ Multiple fields:

```
&sort=name asc,age desc
```

# 36

## SEARCHING: FACETS

Size (Men's)

[see all](#)

▼ Regular

XS

S

M

L

XL

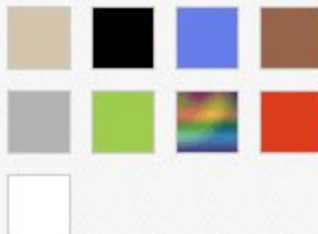
2XL

3XL

► Big & Tall

Color

[see all](#)



Material

[see all](#)

- ☐ 100% Cotton (20,531)
- ☐ 100% Wool (1,118)
- ☐ Cotton Blend (19,067)
- ☐ Denim (5,749)
- ☐ Leather (12,478)
- ☐ Polyester (43,395)
- ☐ Satin (8,992)



Hot Fashion Men Jacket Suit Slim Fit Ve

EUR **10.53** Buy It Now

☐ ☒ + more options



Top Design Men's Slim Button Dress Su

EUR **9.22** to EUR **9.87** Buy It Now

# 37

## SEARCHING: HIGHLIGHTING

The screenshot shows the GitHub search interface for the query 'jet'. The search bar contains 'jet'. The results section shows 'We've found 4,288 repository results'. The 'FACETS' section is highlighted with an orange box and an arrow pointing to it from the word 'FACETS' in orange. The 'FACETS' section includes a table of search filters: Repositories (4,288), Code (3,572,524), Issues (8,993), and Users (2,420). Below this is a 'Languages' section with a table of programming languages and their counts. The 'HIGHLIGHTING' section is highlighted with an orange box and an arrow pointing to it from the word 'HIGHLIGHTING' in orange. The 'HIGHLIGHTING' section shows the first two search results: 'NeXTs/ Jets.js' and 'Dopi/ JetKernel'. The text 'Jets.js' and 'JetKernel' are highlighted in yellow. The text 'Jêt' in the second result is also highlighted in yellow.

Pull requests Issues Gist

Search **FACETS** jet

We've found 4,288 repository results

**FACETS**

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

**HIGHLIGHTING**

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

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

chef-cookbooks/ **jetty**

# 38

## UPDATE DOCUMENT EXAMPLE

- ▶ 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.

# 39

## DELETE DOCUMENTS

- ▶ 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>
```

# 40

SEARCH  
SPEED?





# 41

## SEARCH SPEED PARAMETERS

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:

# SLOW



42

# 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/
```

# 44

## RUN SOLR

```
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)

# 45

## 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
```

The header features a light blue background with a repeating pattern of various business-related icons. These icons include a price tag, a magnifying glass, a smartphone, a document, a gear, a pie chart, an envelope, a speech bubble, a target, a thumbs up, a lightbulb, a clock, a checkmark, and a line graph. The number '46' is prominently displayed in a large, bold, blue font on the left side of the header.

46

# SOLR RESOURCES

# 47

## RESOURCES

- ▶ Official Lucene page:
  - ▷ <http://lucene.apache.org>
- ▶ Official Solr page:
  - ▷ <http://lucene.apache.org/solr>

# 48

## RESOURCES (2)

Solr [official resources page](#)  
provides links to:

- ▶ Tutorials
- ▶ Release documentation
- ▶ Reference guide
- ▶ Mailing lists



# 49

## SOLR INTEGRATION

Solr is integrated with multiple languages via libraries:

- ▶ Java ([solrj](#), [spring-data-solr](#))
- ▶ Python
- ▶ PHP
- ▶ .NET
- ▶ Go

for a full list see [here](#).

# 50

SOLR  
INTEGRATIO  
N (2)

Solr can be combined with big data software such as:

- ▶ Apache Hadoop
- ▶ Apache Cassandra
- ▶ Apache Spark
- ▶ Apache Mahout

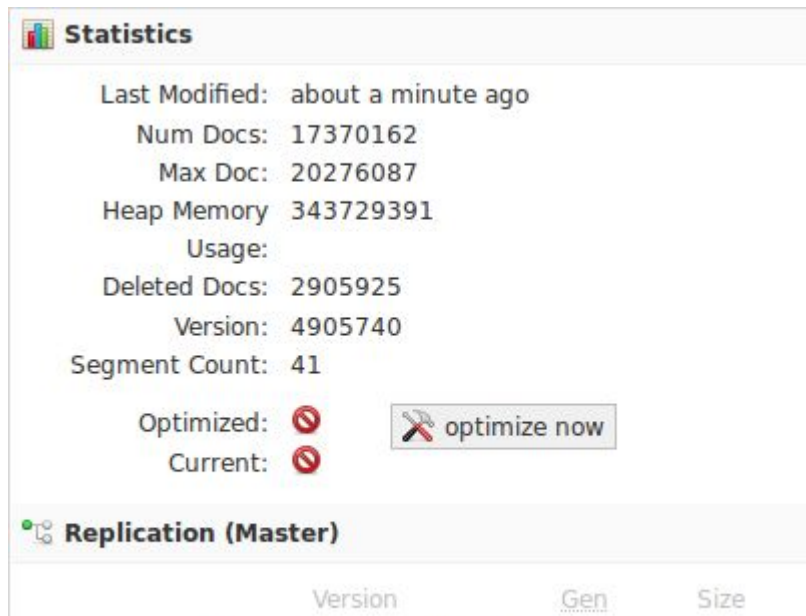
51

**SolrCloud**




# 52

## INDEX SIZE

- Be constantly aware of your index size:



The screenshot shows the Elasticsearch management interface. The top section, titled "Statistics", displays various metrics for the index. Below this, the "Replication (Master)" section shows the status of the index's segments, with columns for Version, Gen, and Size.

Statistics		
Last Modified:	about a minute ago	
Num Docs:	17370162	
Max Doc:	20276087	
Heap Memory	343729391	
Usage:		
Deleted Docs:	2905925	
Version:	4905740	
Segment Count:	41	
Optimized:		
Current:		
 <a href="#">optimize now</a>		

Replication (Master)		
Version	Gen	Size

# 53

# 2,100,000,000

maximum number of documents per core or shard

For more, consider SolrCloud solution!

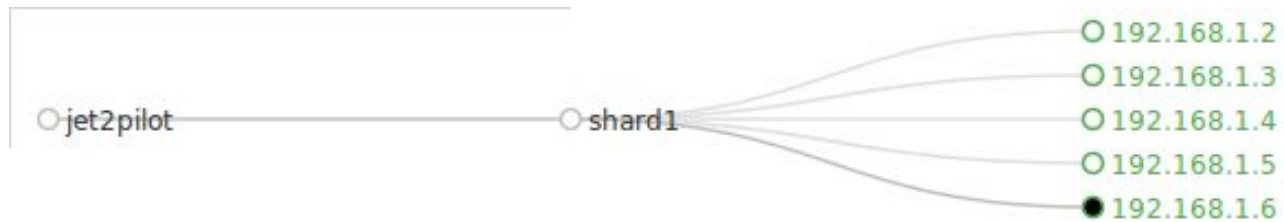
# 54

## SOLRCLOUD CHARACTERISTI CS

- ▶ Distributed search
- ▶ Sharding
- ▶ Fault tolerance
- ▶ High availability
- ▶ Apache Zookeeper coordinates:
  - ▷ shard leader election
  - ▷ updates distribution to shard leaders

# 55

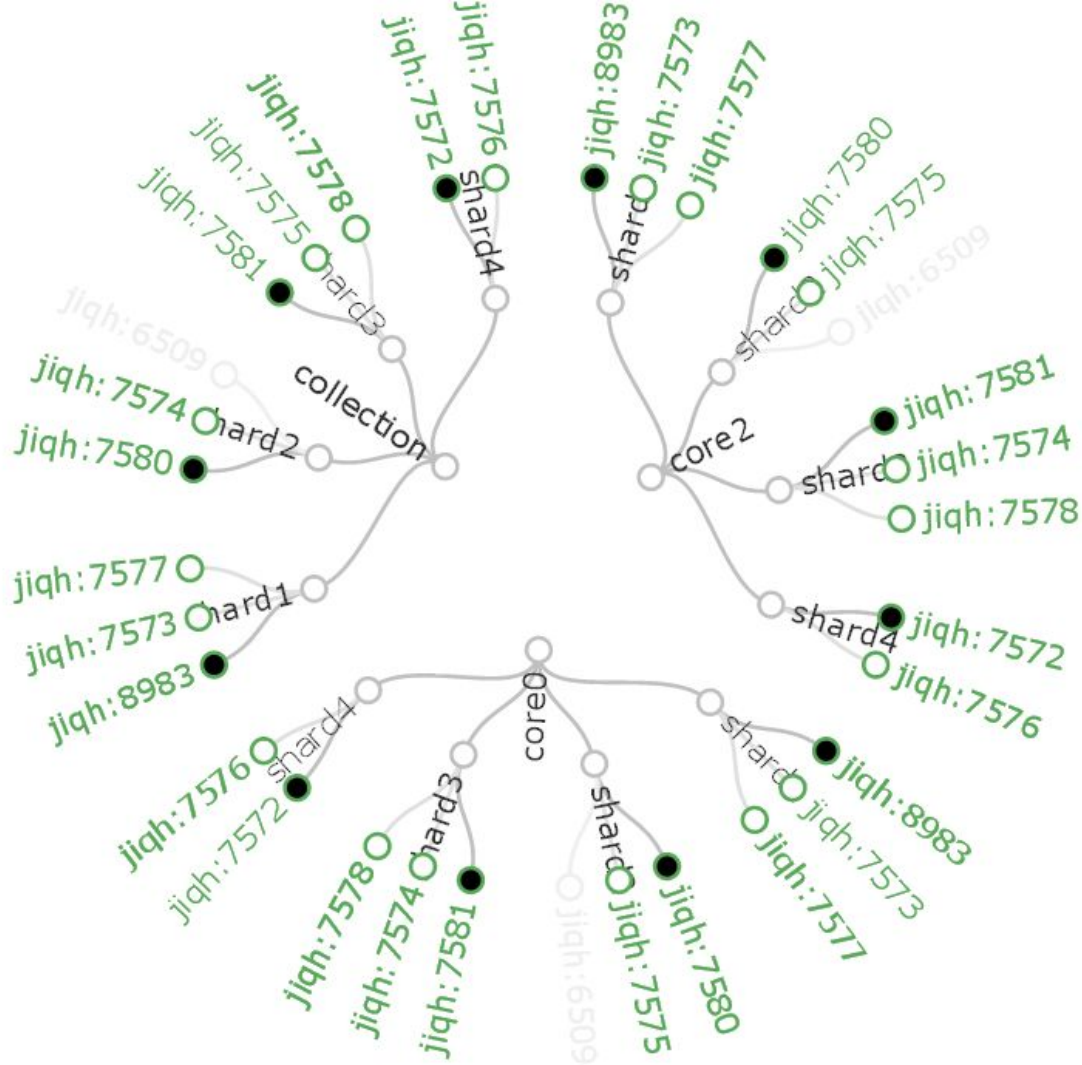
## SOLRCLOUD ADMIN PAGE



Collection with one shard

# 56

## SOLRCLOUD ADMIN PAGE (2)







# 57

## Questions?

About me:

- ▶ <https://manios.org>
- ▶ <https://github.com/manios>