Apache-Solr

Laura Ikic
Michael Zauner

March 2022

1 About Solr

Apache Solr is an open-sourced search engine that originated in 2004. The early stages of the software, then known as "Solar", were created by Yonik Seeley at CNET Networks and were handed over to to the Apache foundation in 2006.

Although the software was not created by Apache itself it bases on software created by the Apache foundation. This connection is also reflected in the name Solr which is an acronym for "Search on Lucene and Resiri". Lucene being a java library used for fast indexing and Resiri a provider for servlets. As the acronym suggests Solr is a java based search engine that provides fast indexing. But not only does Solr provide highly efficient search indexing but also real time file integration which can be sorted into dynamic clusters.

The first distribution that was published under Apache was released in 2007 and was already integrated high-traffic websites. Later in 2010 the two projects Lucene and Solr merged.

2 Functionality

Apache Solr is a **CP - Consistency & Partition Tolerance** search-engine. It has a NoSQL document database with support of transactions. Data can be committed with a *softCommit* which insults in that the data is only visible in the application at runtime and when a *hardCommit* is executed the changes will be written to the disk. These values can be changed in the *solrconfig.xml* file.
In Solr everything is a document images, json, xml, pdf, ...
Solr can run on multiple *Nodes* (different ports of course).

Nodes: runs a single instance of Solr on a specific port.

Collection: is the gathering of document which belong together (e.g: users, locations,...)

Sharding: is a technique to provide scalability to systems. If the data become too big to fit on a single logical entity, it will be splitted into multiple entities called **shards**. When the user requests something from a collection, Solr searches on all shards belonging to this collection. Resulting in merged results.

Replication: is used for providing partition tolerance. The data from each shard will be stored on every replica. Solr can still work when one or more replica nodes are not available as long as there are other replicas running on the particular shard. It follows the master-slave replications algorithm. One of the replicas is assigned as the leader and when this leader detects a modification of the data it will synchronously share the data to its corresponding slaves.

Features

- *full-text search*: here Solr shines great, it can index literally every type of document and provides a lot of querying potential (including wildcards, fuzzy search, ...). Instagram e.g. uses Solr to provide there geolocation api
- *high traffic robust*: Solr is designed to scale up really high. With its design of *sharding* and *replication* the system can handle a lot of requests and it also designed to stay alive even when some shards fall out.
- *real time indexing*: indexes are built nearly at real time due to *Apache Lucene* indexing capabilities.

3 Integration

Solr can be integrated into any system that supports HTTP communication as well as JSON and XML, which are the two response types. Additionally there are also several client libraries for common programming languages such as Java, PHP, Python and many more.

Solr functions as a build in search application for content management systems and enterprise content management systems (alos include a timeline). In some cases the search engine is also being bundled into products which are marketed as big data products. This is done by companies like Clouder, Hortonworks and MapR.

Popular representatives:

- Meta (Instagram uses Solr to power it's geo-search API)
- Cisco (core for it's social network search platform)

- Ebay (search for German sites)
- Netflix (site search feature)
- Nasa (Enterprise Search component, NEBULA cloud computing platform)

4 Communication with Solr

Solr uses Rest-like HTTP communication. When requesting data the response is received as XML or JSON file. Requests can be executed via shell commands. The following list contains the most common commands which can be executed insides solr's bin directory.

Without SlideR:

- ./solr start -e schemaless
- ./solr create -c **collectionName** -d **configName** -n **configDirectory** configs -shards **nrOfShards** -replicationFactor **nrOfReplicasPerShard**
- curl http://localhost:8983/solr/**collectionName**/select?hl=trueq=**searchPhrase**
- curl http://localhost:8983/solr/**collectionName**/select?hl=trueq=**searchPhrase**~**distinctVal**
- ./solr stop -all
- ./post -c **collectionName path**

With SlideR:

- ./SildeR start
- ./SlideR create -n **collectionName** -s **nrOfShards** -r **nrOfReplicasPerShard**
- ./Slider add -c **collectionName** -p **fullPath**
- ./SlideR search -c **collectionName** -q **searchPhrase** -d **distinctVal**

5 Pros/Cons

Solr deals really great with a large number of documents. Most used case is to search and filter a large collection of some data.

Pros

- high scalability

- advanced searching
- easy to setup

Cons

- index replication can be slow

6 Configuration

Solr is OS independent, it can be installed on Linux, MacOS and Windows. But our **SlideR** application is written for Linux.

Requirements Solr

- Java 1.8 or higher

Requirements SlideR

- python3
- python3-venv
- python3-pip
- python module *typer*

For python3 + venv & pip:

```
# sudo apt-get install python3 python3-venv python3-pip
```

An aditional python module *typer* needs to be installed too. This module simplifies the interaction on the command line e.g: –help for displaying the arguments and options for a command of our application

Install typer via pip:

```
# pip3 install typer
```

Now Apache Solr needs to be downloaded.
Download: `https://solr.apache.org/downloads.html`

After download, extract the zip file to the location of preferences. A environment variable needs to be set.

set the environment variable SOLR_HOME:

```
# export SOLR_HOME="PATH_TO_SOLR/solr-8.11.1/bin"
```

But this is just for your current terminal session. If you want it to be permanent you need to place this command in your shell profile file e.b: .zshrc, .profile, ... and execute the to be active.

"reload" your profile (here with zsh shell):

```
# source .zshrc
```

Now you are ready to go :)