Real Time Search and Analytics on Big Data - Installing Solr

Introduction

This exercise will guide you through creating a Solr index on your local machine.

Prerequisites

- Linux based environment or Mac OS X. For Windows users you can use Cygwin.

Tips

- If you are running Windows, I would suggest you set up an Ubuntu 12.04 LTS (http://www.ubuntu.com/download/desktop) virtual machine with VirtualBox (https://www.virtualbox.org/wiki/Downloads).

Installation

Download and Unpack the Solr tarball

You can download Solr 4.0 from the Apache website: http://lucene.apache.org/solr/downloads.html

I have also included the download in this module's directory as well in case internet speeds are poor. Either way, you should end up with a file named apache-solr-4.0.0.tgz.

Now, unpack the tarball with the following command.

```
tar -xvzf apache-solr-4.0.0.tgz
```

Make a Copy of the Example Directory

In the apache-solr-4.0.0 directory you will see a bunch of directories, including "dist" and "example". The "dist" directory includes the jar and war files needed to install Solr in a container such as Tomcat, while the "example" directory has everything you need to run this tutorial or to use as a basis for starting a new Solr development project.

```
cd apache-solr-4.0.0
ls
CHANGES.txt LICENSE.txt NOTICE.txt  README.txt  contrib     dist        docs        example     licenses
```

Now, make a copy of the example so we can play around with it. If we ever break the example, we can always revert back to our original copy.

```
cp -r example testing-example
ls
CHANGES.txt LICENSE.txt NOTICE.txt  README.txt  contrib     dist        docs        example     licenses    testing-example
```

Start Solr!

To start the Solr server, go to the example directory that you created using the bash or some other Unix command shell and run the start jar.

```
java -jar start.jar
```

Solr will output quite a few log messages to the console which detail every phase of Solr startup, including all the plug-in components of Solr. Finally, after a few seconds, Solr will display a message similar to the following.

```
INFO: SolrDispatchFilter.init() done
2013-01-07 11:11:07.011:INFO:oejs.AbstractConnector:Started SocketConnector@0.0.0.0:8983
```

This indicates that the Solr server (running inside an embedded Jetty server) is now ready to receive and process requests on TCP port 8983.

Verify Solr is Up and Running

Is Solr really running? To find out, enter this URL in a web browser:

```
http://localhost:8983/solr/admin/ping
```

Solr will respond with an XML response:

```
<response>
    <lst name="responseHeader">
        <int name="status">0</int>
        <int name="QTime">3</int>
        <lst name="params">
            <str name="df">text</str>
            <str name="echoParams">all</str>
            <str name="rows">10</str>
            <str name="echoParams">all</str>
            <str name="q">solrpingquery</str>
            <str name="distrib">false</str>
        </lst>
    </lst>
    <str name="status">OK</str>
</response>
```

The "status" of "OK" indicates that Solr is indeed running.

So, the Solr server is up and running, but it has no data. Even so, we can still try to execute a query:

```
http://localhost:8983/solr/select?q=*:*&indent=yes
```

Note that "*:*" is a special query syntax that implies all documents. Since no documents have been added to this example server, Solr indicates this with a response which has a count of zero:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <lst name="responseHeader"> <int name="status">0</int> <int name="QTime">0</int>
        <lst name="params">
            <str name="indent">yes</str>
            <str name="q">*:*</str>
        </lst>
    </lst>
    <result name="response" numFound="0" start="0"> </result>
</response>
```

The "&indent=yes" parameter simply indicates that the response XML text should be formatted with white space to make it human readable. The default is to exclude un-necessary white space to reduce the response size.

Indexing Documents

The easiest way to add some test documents is with Solr's Simple Post Tool, which is called post.jar and lives in the exampledocs subdirectory. Solr can accept documents in a variety of formats, but the most basic are:

- XML
- JSON
- CSV

The exampledocs directory has examples of all three. Navigate to the exampledocs directory.

```
cd exampledocs/
ls
books.csv       hd.xml          manufacturers.xml   monitor.xml     post.jar        solr.xml        vidcard.xml
books.json      ipod_other.xml      mem.xml         monitor2.xml        post.sh         test_utf8.sh
gb18030-example.xml ipod_video.xml      money.xml       mp500.xml       sd500.xml       utf8-example.xml
```

To add documents in the Solr XML format to Solr, use the Simple Post Tool on your bash command line.

```
java -jar post.jar *.xml
```

The console output will look like:

```
SimplePostTool version 1.5
Posting files to base url http://localhost:8983/solr/update using content-type application/xml..
POSTing file gb18030-example.xml
POSTing file hd.xml
POSTing file ipod_other.xml
POSTing file ipod_video.xml
POSTing file manufacturers.xml
POSTing file mem.xml
POSTing file money.xml
POSTing file monitor.xml
POSTing file monitor2.xml
POSTing file mp500.xml
POSTing file sd500.xml
POSTing file solr.xml
POSTing file utf8-example.xml
POSTing file vidcard.xml
14 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/update..
```

Querying the Index

Executing Solr queries Now that a bunch of documents have been added to the Solr index, we can execute queries against Solr either from the browser or from bash using curl commands. Retrying the query we used before we added any documents:

```
http://localhost:8983/solr/select?q=*:*&indent=yes
```

Which responds with this response (shortened):

```
<response>
<lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">3</int>
    <lst name="params">
        <str name="indent">yes</str>
        <str name="q">*:*</str>
    </lst>
</lst>
<result name="response" numFound="32" start="0">
    <doc>
        <str name="id">GB18030TEST</str>
        <str name="name">Test with some GB18030 encoded characters</str>
        <arr name="features">
            <str>No accents here</str>
            <str>这是一个功能</str>
            <str>This is a feature (translated)</str>
            <str>这份文件是很有光泽</str>
            <str>This document is very shiny (translated)</str>
        </arr>
        <float name="price">0.0</float>
        <str name="price_c">0,USD</str>
        <bool name="inStock">true</bool>
        <long name="_version_">1423533206481666048</long>
    </doc>
    <doc>
        <str name="id">SP2514N</str>
        <str name="name">
        Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133
        </str>
        <str name="manu">Samsung Electronics Co. Ltd.</str>
        <str name="manu_id_s">samsung</str>
        <arr name="cat">
            <str>electronics</str>
            <str>hard drive</str>
        </arr>
        <arr name="features">
            <str>7200RPM, 8MB cache, IDE Ultra ATA-133</str>
            <str>
            NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB) motor
            </str>
        </arr>
        <float name="price">92.0</float>
        <str name="price_c">92,USD</str>
        <int name="popularity">6</int>
        <bool name="inStock">true</bool>
        <date name="manufacturedate_dt">2006-02-13T15:26:37Z</date>
        <str name="store">35.0752,-97.032</str>
        <long name="_version_">1423533206605398016</long>
    </doc>
...
```

We'll get deeper into indexing and querying documents in Solr in the future modules. But for now, congratulations on installing Solr!

Additional Resources

Apache Solr has a very good tutorial in which you install Solr as well as index and query data.

- http://lucene.apache.org/solr/4_0_0/tutorial.html