Install SolrCloud

[roger.xialei@gmail.com](mailto:roger.xialei@gmail.com)

May 18th 2015

[1. 准备环境 2](#_Toc426715607)

[2. 下载安装 Solr 5.x.x 3](#_Toc426715608)

[2.1 解压出安装脚本 3](#_Toc426715609)

[2.2使用install\_solr\_service.sh安装 3](#_Toc426715610)

[2.3 查看状态 3](#_Toc426715611)

[3. 修改配置文件 4](#_Toc426715612)

[3.1 环境变量定义：/data/solr/solr.in.sh 4](#_Toc426715613)

[3.2 配置文件集：configsets 4](#_Toc426715614)

[4. 外部依赖包 6](#_Toc426715615)

[4.1文档解析Tika 6](#_Toc426715616)

[4.2 中文分词Ansj 6](#_Toc426715617)

[5. 启动solrcloud 8](#_Toc426715618)

[5.1 启动服务 8](#_Toc426715619)

[5.2 创建collection/shards 8](#_Toc426715620)

[5.3 停止、重启及添加节点 9](#_Toc426715621)

[6. 管理控制台 10](#_Toc426715622)

[7. 使用docker 10](#_Toc426715623)

[8. 附录 11](#_Toc426715624)

[8.1 solrconfig.xml 11](#_Toc426715625)

[8.2 schema.xml 12](#_Toc426715626)

## 准备环境

安装 Zookeeper：Refer to installing zookeeper.docx.

Collection3

Shard1

Collection2

Shard1

Collection1

Shard1

Replica3

Collection1

Shard2

Collection2

Shard2

Collection3

Shard2

Shard3

Collection1

Shard3

Collection2

Replica2

Replica1

Shard3

Collection3

## 2. 下载安装 Solr 5.x.x

### 2.1 解压出安装脚本

tar xzf solr-5.2.1.tgz solr-5.2.1/bin/install\_solr\_service.sh --strip-components=2

### 2.2使用install\_solr\_service.sh安装

./install\_solr\_service.sh solr-5.2.1.tgz -i /usr/local -d /data/solr/node1 -u solr -s solr -p 8983

Solrcloud环境下，如果要在一台服务器上运行多个solr节点：

1. 不同节点的solr home在不同的物理磁盘上，避免io竞争
2. 不同节点拥有独立的include file： solr.in.sh
3. 不同节点拥有独立的/etc/init.d/solr2作为服务脚本

./install\_solr\_service.sh solr-5.2.1.tgz -i /usr/local -d /data/solr/node2 -u solr -s solr2 -p 8984

### 2.3 查看状态

service solr status

*service solr2 status*

## 3. 修改配置文件

### 3.1 环境变量定义：/data/solr/solr.in.sh

SOLR\_JAVA\_MEM="-Xms6g –Xmx6g"

*GC\_TUNE pre-configured Java Garbage Collection settings*

SOLR\_OPTS="$SOLR\_OPTS -Dsolr.autoSoftCommit.maxTime=3000"

ZK\_HOST="192.168.0.229:2181,192.168.0.230:2181,192.168.0.231:2181/solr"

SOLR\_HOST=192.168.0.229

If using a chroot for the first time, you need to bootstrap the Solr znode tree in ZooKeeper by using the zkcli.sh script, such as:

$ /usr/local/solr/server/scripts/cloud-scripts/zkcli.sh -zkhost 192.168.0.229:2181/solr -cmd bootstrap -solrhome /data/solr/node1/data

### 3.2 配置文件集：configsets

(1). 拷贝configsets

cd /data/solr/

cp -rf /usr/local/solr/server/solr/configsets/basic\_configs/conf .

(2). 修改solrconfig.xml

<https://cwiki.apache.org/confluence/display/solr/Configuring+solrconfig.xml>

(3). 修改schema.xml

<https://cwiki.apache.org/confluence/display/solr/Format+of+solr.xml>

1. fieldType
2. field/dynamicField

(4). stopwords

lang/stopwords\_en.txt

lang/stopwords\_cn.txt

## 4. 外部依赖包

### 4.1文档解析Tika

Tika's runtime dependency of 'jhighlight' was removed as the latter was found to contain some LGPL-only code. Until that's resolved by Tika, you can download the jar yourself and place it under /usr/local/solr/contrib/extraction/lib.

<http://central.maven.org/maven2/com/uwyn/jhighlight/1.0/jhighlight-1.0.jar>

### 4.2 中文分词Ansj

<http://maven.ansj.org/org/ansj/ansj_seg/>

<http://maven.ansj.org/org/nlpcn/nlp-lang/>

<https://github.com/xialei/ansj4solr>

ansj\_seg-2.0.8.jar

*ansj\_seg-2.0.8-min.jar*

nlp-lang-1.0.jar

/usr/local/solr/server/solr-webapp/webapp/WEB-INF/lib

<!-- csf\_defined : text\_general\_cn : used to index and query Chinese text -->

<fieldType name="text\_general\_cn" class="solr.TextField" positionIncrementGap="100">

<analyzer type="index">

<tokenizer class="org.ansj.solr.ANSJTokenizerFactory" conf="ansj.conf" />

<filter class="solr.StopFilterFactory" ignoreCase="true" words="lang/stopwords\_cn.txt" />

<filter class="solr.WordDelimiterFilterFactory"

splitOnCaseChange="1"

generateWordParts="1"

generateNumberParts="1"

catenateWords="0"

catenateNumbers="0"

catenateAll="0"

preserveOriginal="1"/>

<filter class="solr.PorterStemFilterFactory"/>

<filter class="solr.RemoveDuplicatesTokenFilterFactory"/>

<filter class="solr.LowerCaseFilterFactory"/>

</analyzer>

<analyzer type="query">

<tokenizer class="org.ansj.solr.ANSJTokenizerFactory" analysisType="1" conf="ansj.conf" />

<filter class="solr.StopFilterFactory" ignoreCase="true" words="lang/stopwords\_cn.txt" />

<filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>

<filter class="solr.WordDelimiterFilterFactory"

splitOnCaseChange="1"

generateWordParts="1"

generateNumberParts="1"

catenateWords="0"

catenateNumbers="0"

catenateAll="0"

preserveOriginal="1"/>

<filter class="solr.PorterStemFilterFactory"/>

<filter class="solr.RemoveDuplicatesTokenFilterFactory"/>

<filter class="solr.LowerCaseFilterFactory"/>

</analyzer>

</fieldType>

ansj.conf

lastupdate=123  
files=dic/extDic.txt,dic/aaa.txt  
其中lastupdate 是一个数字，只要这次比上一次大就会触发更新操作（solr调用分词器时会重新加载字典），可以用时间戳。 files是用户词库文件，以英文逗号隔开。

## 5. 启动solrcloud

Node4

Node3

Node2

Node1

Shard1

replica

Shard3

Shard2

Shard1

Collection1 – company base

Collection2 -- news

Collection3 -- others

Once you have started up more Solr nodes than numShards, the nodes will create replicas for each shard, distributing them evenly across the node, as long as they all belong to the same collection.

### 5.1 启动服务

在每个节点上启动solr服务，确保zk已经chroot

$ service solr start -c

$ bin/solr healthcheck –c collection1 –z 192.168.0.229:2181

### 5.2 创建collection/shards

**命令行创建**

./solr create\_collection -help

./bin/solr create\_collection -c collection1 -d /data/solr/conf  -shards 3  -replicationFactor 2

**API创建**

<http://192.168.0.229:8983/solr/admin/collections?action=CREATE&name=collection1&numShards=3&replicationFactor=2&maxShardsPerNode=2&collection.configName=example>

更新配置文件：

/usr/local/solr/server/scripts/cloud-scripts/zkcli.sh -z 192.168.0.229:2181/solr -cmd upconfig -d /data/solr/conf -n collection1

### 5.3 停止、重启及添加节点

To stop Solr in SolrCloud mode：

$ bin/solr stop –all

To restart your SolrCloud nodes：

$ /etc/init.d/solr restart -c -p 8983 -s /data/solr/node1/data

$ /etc/init.d/solr restart -c -p 7574 -z localhost:9983 -s /data/solr/node2/data

Adding a node to an existing cluster：

$ mkdir -p /data/solr/node3/data

$ cp server/solr/solr.xml /data/solr/node3/data/

$ bin/solr start -cloud -s /data/solr/node3/data -p 8987 -z 192.168.0.229:9983

## 6. 管理控制台

<http://localhost:8983/solr/#/~cloud>

<dependency>

<groupId>org.apache.solr</groupId>

<artifactId>solr-core</artifactId>

<version>5.2.1</version>

</dependency>

<dependency>

<groupId>org.apache.solr</groupId>

<artifactId>solr-solrj</artifactId>

<version>5.2.1</version>

</dependency>

## 7. 使用docker

## 8. 附录

### 8.1 solrconfig.xml

<!-- csf : add suggester -->

<searchComponent name="suggester" class="solr.SpellCheckComponent">

<lst name="spellchecker">

<str name="name">suggester</str>

<str name="classname">org.apache.solr.spelling.suggest.Suggester</str>

<str name="lookupImpl">org.apache.solr.spelling.suggest.tst.TSTLookup</str>

<str name="field">suggest\_autocomplete</str>

<str name="buildOnCommit">true</str>

</lst>

</searchComponent>

<requestHandler name="/suggester" class="org.apache.solr.handler.component.SearchHandler">

<lst name="defaults">

<str name="spellcheck">true</str>

<str name="spellcheck.dictionary">suggester</str>

<str name="spellcheck.count">10</str>

<str name="spellcheck.onlyMorePopular">true</str>

</lst>

<arr name="components">

<str>suggester</str>

</arr>

</requestHandler>

<!--config for announce service-->

<requestHandler name="/update/extract/announce"

startup="lazy"

class="solr.extraction.ExtractingRequestHandler" >

<lst name="defaults">

<!-- All the main content goes into "text"... if you need to return

the extracted text or do highlighting, use a stored field. -->

<str name="fmap.content">text</str>

<str name="lowernames">true</str>

<str name="uprefix">announce\_</str>

<!-- capture link hrefs but ignore div attributes -->

<str name="captureAttr">true</str>

</lst>

</requestHandler>

<!--config for news service-->

<requestHandler name="/update/extract/news"

startup="lazy"

class="solr.extraction.ExtractingRequestHandler" >

<lst name="defaults">

<!-- All the main content goes into "text"... if you need to return

the extracted text or do highlighting, use a stored field. -->

<str name="fmap.content">text</str>

<str name="lowernames">true</str>

<str name="uprefix">news\_</str>

<!-- capture link hrefs but ignore div attributes -->

<str name="captureAttr">true</str>

</lst>

</requestHandler>

<!--config for research service-->

<requestHandler name="/update/extract/rpt"

startup="lazy"

class="solr.extraction.ExtractingRequestHandler" >

<lst name="defaults">

<!-- All the main content goes into "text"... if you need to return

the extracted text or do highlighting, use a stored field. -->

<str name="fmap.content">text</str>

<str name="lowernames">true</str>

<str name="uprefix">rpt\_</str>

<!-- capture link hrefs but ignore div attributes -->

<str name="captureAttr">true</str>

</lst>

</requestHandler>

### 8.2 schema.xml

<!-- csf\_defined : in case you want to split Chinese word into single character -->

<fieldType name="csf\_letter\_cn" class="solr.TextField" >

<analyzer>

<tokenizer class="solr.StandardTokenizerFactory"/>

</analyzer>

</fieldType>

<!-- csf\_defined : in case you want to split english word into single letter,

e.g. you want to split abbr word for pinyin, gsyh for industry bank -->

<fieldType name="csf\_letter" class="solr.TextField" >

<analyzer>

<tokenizer class="solr.LetterTokenizerFactory"/>

<filter class="solr.LowerCaseFilterFactory" />

</analyzer>

</fieldType>

<!-- csf\_defined : in case when you want to sort text in Chinese, but you don't need to index it,

as indexed fields will be split when analyze;

use string fieldType when you want to sort on english words. -->

<fieldType name="csf\_text\_sort" class="solr.CollationField" language="zh" strength="primary" />