**如何在UDH Search中配置中文分词**

作者：费英林

目录

[1. 安装准备 2](#_Toc420052935)

[1.1. UDH1.0 2](#_Toc420052936)

[1.2. UDH Search 2](#_Toc420052937)

[1.3. Ik Analyzer 2](#_Toc420052938)

[2. 定义分词库 3](#_Toc420052939)

[2.1. 定义分词库文件名 3](#_Toc420052940)

[2.2. 定义分词库文件 3](#_Toc420052941)

[3. 分词库文件部署 3](#_Toc420052942)

[3.1. 创建类目录 3](#_Toc420052943)

[3.2. 部署库文件到类目录 4](#_Toc420052944)

[3.3. 重启Solr服务 4](#_Toc420052945)

[4. IK类文件部署 4](#_Toc420052946)

[5. 索引创建 4](#_Toc420052947)

[6. 验证 5](#_Toc420052948)

UDH Search是基于Cloudera Search构建，与Hadoop平台高度集成，提供了用于索引创建、存储以及查询的工具和模型。由于Apache Solr对中文的分词及停词的处理效果很不好，我们需要借助第三方工具来实现这些功能。下文以Ik Analyzer（简称IK）为例，阐述如何在Hadoop平台下的搜索引擎里添加中文分词。

本文重点阐述分词工具的配置与使用，关于UDH及Search相关的内容请参考相关文档。

# 安装准备

## UDH1.0

 参见UDH1.0安装手册。

## UDH Search

 参见‘UDH Search系统构建及其应用’。

## Ik Analyzer

 下载链接：<http://code.google.com/p/ik-analyzer/downloads/list>

 版本：IK Analyzer 2012FF\_hf1.zip

# 定义分词库

 目前，IK自带的分词库有275714个汉语词条。另外，IK分词器也提供了对分词及停词词库的扩展支持。

## 定义分词库文件名

 解压缩IK Analyzer 2012FF\_hf1.zip到一临时目录，包含以下文件：

-rw-r--r-- 1 root root 1165908 10月 26 2012 IKAnalyzer2012FF\_u1.jar

-rw-r--r-- 1 root root 414 2月 14 2012 IKAnalyzer.cfg.xml

-rw-r--r-- 1 root root 17778 1月 17 2012 LICENSE.txt

-rw-r--r-- 1 root root 278 1月 19 2012 NOTICE.txt

-rw-r--r-- 1 root root 161 4月 15 2011 stopword.dic

 其中IKAnalyzer.cfg.xml是IK Analyzer的配置文件，我们需要在这个文件里指定分词库的文件名：

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">

<properties>

 <comment>IK Analyzer 扩展配置</comment>

 **<entry key="ext\_dict">delimiters.dic;</entry>**

 <entry key="ext\_stopwords">stopword.dic;</entry>

</properties>

## 定义分词库文件

 分词库文件格式是无BOM的UTF-8编码的中文文本文件，文件扩展名不限。词库中每个中文词汇独立占一行，使用\r\n的DOS方式换行。在配置文件中，用户可一次配置多个词库文件，文件名使用“;”号分隔。

# 分词库文件部署

 UDH Search安装后会生成一个Tomcat应用，我们需要将分词（停词）库文件部署在Tomcat服务器上，保证IK运行的时候可以访问到这些文件。

## 创建类目录

默认安装后，我们需要在Tomcat服务器上手工生成一个类目录：

mkdir /usr/lib/solr/webapps/solr/WEB-INF/classes/

## 部署库文件到类目录

拷贝IK配置文件和词库文件到所有的Solr服务器：

cp IKAnalyzer.cfg.xml /usr/lib/solr/webapps/solr/WEB-INF/classes/

cp delimiters.dic.dic /usr/lib/solr/webapps/solr/WEB-INF/classes/

## 重启Solr服务

库文件部署后必须重启服务器：

sudo service solr-server restart

# IK类文件部署

由于索引的创建程序是运行在Yarn平台上的，MapReduce框架需要在Hadoop的类路径里访问到IK的Jar包。我们可以通过两种方式实现类加载。一种是将Jar包放到/usr/lib/hadoop-yarn/lib/目录下，重启Yarn服务，这样Hadoop就可以从全局变量里访问到类路径。这种方式的优点是不需要在每次运行作业时都指定类路径，缺点是Jar包内容有变动时，我们需要重启Yarn服务，这对于生产环境来说是不合适的。在生产环境里，我们通常不会去重启Yarn这样的主服务。

另一种方式是在每一次作业运行时指定类路径，不需要重启服务，同时提供了更大的灵活性，可随时更改类文件。

出于一致性考虑，我们将IK的Jar包复制到Solr的类路径下，但要记住，运行在Hadoop平台上的Solr是无法自动加载这个类库的：

cp IKAnalyzer2012FF\_u1.jar /usr/lib/solr/webapps/solr/WEB-INF/lib

# 索引创建

以下以舆情分析中的activity数据为例，给出索引创建的具体命令：

# 清空历史数据

solrctl collection --deletedocs activity

# 执行Hadoop作业

 在这个命令里，我们加入了libjars这个参数，指定了IK包的路径，实现动态加载：

HADOOP\_CLASSPATH=/usr/lib/hbase/hbase-protocol.jar \

hadoop --config /etc/hadoop/conf jar /usr/lib/hbase-solr/tools/hbase-indexer-mr-\*-job.jar \

--conf /etc/hbase/conf/hbase-site.xml \

**--libjars /usr/lib/solr/webapps/solr/WEB-INF/lib/IKAnalyzer2012FF\_u1.jar** \

-D 'mapred.child.java.opts=-Xmx500m' --hbase-indexer-file $HOME/morphlines-activity.xml \

--zk-host 127.0.0.1/solr --collection activity --go-live \

--log4j /etc/hbase/conf.dist/log4j.properties

# 验证

打开Solr界面，找到相应的Shard进行分析，可看到分词效果：

