

Flume-0.9.4和Hbase-0.96整合(2)

这几天由于项目的需要，需要将Flume收集到的日志插入到Hbase中，有人说，这不很简单么？Flume里面自带了Hbase

sink，可以直接调用啊，还用说么？是

的，我在本博客的[《Flume-1.4.0和Hbase-0.96.0整合》](#)

文章中就提到如何用Flume和Hbase整合，从文章中就看出整个过程不太复杂，直接做相应的配置就行了。

那么为什么今天还要特意提一下Flume-0.9.4和Hbase-0.96整合？这是因为Flume-0.9.4和Hbase-0.96整合比Flume-1.4.0和Hbase-0.96整合麻烦多了！不是随便几个配置就能搞定的，里面涉及到修改Flume和Hadoop的源码！

先看下我公司的Hadoop、Hbase、Flume等的配置吧。2013年10月末，公司的Hadoop升级到2.2.0，Hbase升级到0.96,Zookeeper升级到3.4.5，但是Flume版本由于各种原因没有升级，还是用Flume-0.9.4，而Flume-0.9.4源码是基于Hadoop-0.20.2-CDH3B4、Hbase-0.90.1-cdh3u0开发的，Hadoop-0.20.2-CDH3B4和现在的Hadoop-2.2.0完全不一样的设计，而且直接用Hadoop-0.20.2-CDH3B4会使得Flume-0.9.4不能和Hbase-0.96.0通信，都不能通信了，何谈整合！但是经过几天的奋战，我们终于通过修改Flume和Hadoop的部分源码达到了Flume-0.9.4和Hbase-0.96整合，今天就分享一下我们是怎么修改的。

考虑到篇幅的问题，已经将本文拆分为二，为你带来的不便深感抱歉！下面是两篇文章的链接：

[《Flume-0.9.4和Hbase-0.96整合\(1\)》](#)、[《Flume-0.9.4和Hbase-0.96整合\(2\)》](#)

2、修改flume-core\src\main\java\org\apache\hadoop\io\FlushingSequenceFileWriter.java和RawSequenceFileWriter.java两个java类

因为在步骤一中我们用新版本的Hadoop替换了旧版本的Hadoop，而新版本Hadoop中的org.apache.hadoop.io.SequenceFile.Writer类和旧版本的org.apache.hadoop.io.SequenceFile.Writer类有些不一样。所以导致了FlushingSequenceFileWriter.java和RawSequenceFileWriter.java两个java类出现了部分的错误，解决方法如下：

(1)、需要修改Hadoop-2.2.0源码中的hadoop-2.2.0-src\hadoop-common-project\hadoop-common\src\main\java\org\apache\hadoop\io\SequenceFile.java类，在Writer类里面添加默认的构造函数：

```
Writer(){
    this.compress = CompressionType.NONE;
}
```

然后重新编译hadoop-common-project工程，将编译后的hadoop-common-2.2.0.jar替换之前的hadoop-common-2.2.0.jar

(2)、修改FlushingSequenceFileWriter.java和RawSequenceFileWriter.java
这两个类中有错误，请用新版本Hadoop的相应API替换掉旧版本Hadoop的API，具体怎么修改，这就不说了，如有需要的同学，可以邮件联系我（wyphao.2007@163.com）

(3)、修改com.cloudera.flume.handlers.seqfile中的SequenceFileOutputFormat类修改如下：

```
this(SequenceFile.getCompressionType(FlumeConfiguration.get()),  
      new DefaultCodec());
```

修改为

```
this(SequenceFile.getDefaultCompressionType(FlumeConfiguration.get()),  
      new DefaultCodec());
```

```
CompressionType compressionType = SequenceFile.getCompressionType(conf);
```

修改为

```
CompressionType compressionType = SequenceFile.getDefaultCompressionType(conf);
```

3、重新编译Flume源码

重新编译Flume源码（如何

编译Flume源码？请参见本博客的[《Flume-0.9.4源码编译及一些编译出错解决方法》](#)

），并用编译之后的flume-core-0.9.4-cdh3u3.jar替换\${FLUME_HOME}/lib中的flume-core-0.9.4-cdh3u3.jar类。删掉\${FLUME_HOME}/lib/hadoop-core-0.20.2-cdh3u3.jar等有关Hadoop旧版本的包。

4、修改\${FLUME_HOME}/bin/flume启动脚本

仔细分析\${FLUME_HOME}/bin/flume脚本，你会发现如下代码：

```
# put hadoop conf dir in classpath to include Hadoop  
# core-site.xml/hdfs-site.xml  
if [ -n "${HADOOP_CONF_DIR}" ]; then  
    CLASSPATH="${CLASSPATH}:${HADOOP_CONF_DIR}"  
elif [ -n "${HADOOP_HOME}" ]; then  
    CLASSPATH="${CLASSPATH}:${HADOOP_HOME}/conf"  
elif [ -e "/usr/lib/hadoop/conf" ]; then  
    # if neither is present see if the CDH dir exists  
    CLASSPATH="${CLASSPATH}:/usr/lib/hadoop/conf";  
    HADOOP_HOME="/usr/lib/hadoop"  
fi # otherwise give up  
  
# try to load the hadoop core jars  
HADOOP_CORE_FOUND=false
```

```
while true; do
  if [ -n "$HADOOP_HOME" ]; then
    HADCOREJARS=`find ${HADOOP_HOME}/hadoop-core*.jar || \
      find ${HADOOP_HOME}/lib/hadoop-core*.jar || true`
    if [ -n "$HADCOREJARS" ]; then
      HADOOP_CORE_FOUND=true
      CLASSPATH="$CLASSPATH:${HADCOREJARS}"
      break;
    fi
  fi

  HADCOREJARS=`find ./lib/hadoop-core*.jar 2> /dev/null || true`
  if [ -n "$HADCOREJARS" ]; then
    # if this is the dev environment then hadoop jar will
    # get added as part of ./lib (below)
    break
  fi

  # core jars may be missing, we'll check for this below
  break
done
```

你会发现，这是Flume加载Hadoop旧版本的依赖包，在新版本的Hadoop根本就没有\${HADOOP_HOME}/conf等文件夹，所以会出现Flume不能加载对新版本Hadoop的依赖。这里教你用最简单的方法来实现对新版本的Hbase和Hadoop的依赖，在\${FLUME_HOME}/bin/flume脚本里面加入下面的CLASSPATH依赖：

```
CLASSPATH="/home/q/hbase/hbase-0.96.0-hadoop2/lib/*"
```

请注意hbase-0.96.0-hadoop2里面对hadoop的依赖，hbase-0.96.0-hadoop2里面对Hadoop的依赖包是2.1.0，用上面编译好的hadoop-common-2.2.0.jar替换\${HBASE_HOME}/lib里面的hadoop-common-2.1.0.jar

5、如何和Hbase-0.96整合

在flume-src\plugins\flume-plugin-hbasesink\src\main\java里面的添加自己的类（当然你完全可以自己创建一个新的maven工程）。如果需要和Hbase整合，必须继承EventSink.Base类，重写里面的方法（可以参照flume-src\plugins\flume-plugin-hbasesink\src\main\java\com\cloudera\flume\hbase\Attr2HBaseEventSink.java），写完之后需要重新编译flume-src\plugins\flume-plugin-hbasesink底下的类，打包成jar文件。然后将你写好的Hbase sink注册到Flume中，关于如何注册，请参见本博客的[《Flume-0.9.4配置Hbase sink》](#)。

6、结束

经过上面几步的配置，你的Flume-0.9.4就可以和Hbase-0.96整合了，祝你成功。

考虑到篇幅的问题，已经将本文拆分为二，为你带来的不便深感抱歉！下面是两篇文章的链接：

[《Flume-0.9.4和Hbase-0.96整合\(1\)》](#)、[《Flume-0.9.4和Hbase-0.96整合\(2\)》](#)

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ()