

基于flume-ng 1.4.0的TailSource程序开发

写在前面的话，最近发现有很多网站转载我博客的文章，这个我都不介意的，但是这些网站转载我博客都将文章的出处去掉了，直接变成自己的文章了！！我强烈谴责他们，鄙视那些转载文章去掉出处的人！所以为了防止这些，我以后发表文章的时候，将会在文章里面加入一些回复之后才可见的内容！！请大家不要介意，本博客Hadoop相关的文章都是原创，每篇文章都花了很长时间编写，这些都是无偿的，所以转载本文章请标明出处：

本篇博客地址：[《基于flume-ng 1.4.0的TailSource程序开发》](#)： /archives/1034

博客地址：[过往记忆](#)：

用过Flume 0.9.4的用户应该知道，在它里面自带了一个TailSource以及TailDirSource，这个Source是负责读取一个文件，并一行一行的发送到sink端，而在flume-ng 1.4.0里面没有自带TailSource，更别说TailDirSource了，虽然我们可以在flume-ng 1.4.0里面用exec的tail -F来一行一行的发送文件，但是还是不太好。而且公司最近flume需要升级，之前的程序用到了TailSource和TailDirSource，没办法只能再次开发了。在flume-ng里面开发一个source需要遵循一定的规则，我们可以通过以下两种方法来开发自己的source：

```
package org.apache.flume;
```

```
import org.apache.flume.conf.Configurable;  
import org.apache.flume.source.AbstractSource;
```

```
/**
```

```
 * User: wyp
```

```
 * /archives/1034
```

```
 * 版权所有，翻版不究
```

```
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干活
```

```
 * Date: 14-5-20
```

```
 * Time: 下午11:15
```

```
 */
```

```
public class TailSource extends AbstractSource implements EventDrivenSource,
```

```
    Configurable {
```

```
    @Override
```

```
    public void configure(Context context) {
```

```
    }
```

```
    @Override
```

```
    public synchronized void start() {
```

```
    }
```

```
    @Override
```

```
public synchronized void stop() {  
  
}  
}
```

或者

```
package org.apache.flume;  
  
import org.apache.flume.conf.Configurable;  
import org.apache.flume.source.AbstractSource;  
  
/**  
 * User: wyp  
 * /archives/1034  
 * 版权所有，翻版必究  
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干活  
 * Date: 14-5-20  
 * Time: 下午11:15  
 */  
public class TailSource extends AbstractSource implements Configurable,  
        PollableSource {  
    @Override  
    public void configure(Context context) {  
  
    }  
  
    @Override  
    public Status process() throws EventDeliveryException {  
        return null;  
    }  
  
    @Override  
    public synchronized void start() {  
  
    }  
  
    @Override  
    public synchronized void stop() {  
  
    }  
}
```

这两个类虽然继承关系不一样，但是都可以实现。下面的方法中多了一个process()方法，这个方法在PollableSourceRunner类中会被一个线程不断的调用。

关于如何具体开发TailSource，这个我就不介绍了。下面主要介绍一下如何使用我这个TailSource，在本文的后面会附上TailSource插件的下载地址，大家如有需要可以自行下载使用。TailSource支持waitTime、file、startFromEnd以及offset四个参数的配置。

1. waitTime：指的是隔多久读一次文件，程序是按照一块一块的读取文件内容到缓冲区，然后一行一行的解析；可以不配置，默认1秒读一次。
2. file：指的是需要读取的文件绝对路径，一定需要配置；
3. startFromEnd：是否从文件结尾处读取，默认为false；
4. offset：从offset偏移量开始读取，可以不配置，默认是从偏移量为0开始读。

目前本版本的TailSource不支持正则表达式的配置。如你需要使用，可以按照下面的配置进行：

```
agent.sources = r1
agent.sources.r1.type = org.apache.flume.TailSource
agent.sources.r1.channels = channel
agent.sources.r1.waitTime = 100
agent.sources.r1.file = /export1/spark/apache-flume-1.4.0-bin/test
agent.sources.r1.startFromEnd = false
agent.sources.r1.offset = 0
```

然后就可以监听到test文件的变化，并放到channel里面。下面是TailSource的下载地址

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: [【】（）](#)