Homework1 海量视频存储

- **5130309717**
- **朱文豪**

主要步骤

- 搭建组播服务器
- 搭建组播客户端
- 将接受到的视频流文件上传到HDFS中

文件结构

- 根目录下包含所有编译出来的class(基于MultiThread)， 并打包成 **Multicast.jar**，可以直接运行。
- Source Code 中是源代码，其中，**MulticastSocketServer.java** 是组播服务端，使用UDP协议分发数据流。**MulticastSocketClient.java** 是组播客户端，接收数据并缓存到本地。**FileCopyToHDFS.java** 将文件上传到HDFS文件系统，并删除缓存文件。 **MulticastConfig.conf** 是配置文件，用来定义组播IP地址，端口，输入文件地址，缓存文件地址，目标文件地址，超时时间参数，传输Buffer大小。 **MultiThread.java** 打包了所有函数，通过三线程模拟整个

过程，同样需要读取配置文件运行。

使用说明

1. 修改配置文件 MulticastConfig.conf

- 组播IP地址 **Inet_Address** : 范围从 *224.0.0.0* 到 *239.255.255.255*
- 组播端口 **Port**
- 输入文件地址 **Input_File_Address**
- 缓存文件地址 **Output_File_Address** : 默认为当前文件夹
- 目标文件地址 **HDFS_File_Address** ： 比如 *hdfs://localhost:xxxx/mydocument/*.* 端口取决于Hadoop的配置文件 core-site.xml
- 缓冲大小 **Buffer_Size** ： 决定组播传输速率，最大值为65507
- 超时时间 **Timeout_Milliseconds** ： 客户端超时退出参数。

2. 启动 hadoop

```
$ /sbin/start-all.sh
```

3. 运行 Multicast.jar

```
$ /hadoop jar Multicast.jar MultiThread
```

或者

```
$ /javac -cp $HADOOP_CLASSPATH MultiThread.java
$ /tar -cvf Multicast.jar *.class
$ /hadoop jar Multicast.jar MultiThread
```

4. 查看已上传的视频并与原始文件比较

```
$ /hadoop fs -ls /yourdocument/
$ /hadoop fs -copyToLocal /yourdocument/copy.avi  /home/me/
$ md5sum copy.avi
$ md5sum origin.avi
```

md5应该是相同的，说明文件被完整的从服务器端发送到客户端并上传到hdfs中了。

---

5. 使用VLC作为视频流服务端

使用VLC作为服务端也可以，但是会有比较大的问题，具体见下文。
步骤是：
注释 MultiThread.java 中的Server线程 `Server.start();` 并重新编译。
VLC设置中选择Stream, 使用UDP协议，并确保端口和IP与配置文件中相同。 先启动 MultiThread, **并在 Timeout 之前启动 VLC**， 便可观测到客户端开始接受到VLC的数据并在本地存储。

其他

> **1.为什么不使用 VLC Media Player 生成视频流？**
> 主要有以几个原因。首先，使用VLC生成视频流并基于UDP组播，数据传输的速度非常慢，这可能是基于安全性/可靠性的考虑？一个几百MB的视频要传输非常久。其次，我在实验的时候，用VLC作为视频源，客户端丢包非常严重，而自己实现的服务端传的数据却能完整接收，这可能和VLC的编码有关系？一直没有找到好的解决方案。
> 因此，重新实现了一个简单的组播服务端，java读取本地文件转换成数据流进行组播。但是没有视频转码的功能，因此原始文件和目标文件的编码格式应该是一致的。同时将缓冲区开到最大，从而可以高速完成数据传输和接受，并且保证了文件的一致性。实测局域网内多客户端也可以正常执行。
> **2.为什么不使用命令行启动传参数？**
> 因为参数比较较多，尤其是当搭配VLC测试的时候也会涉及到IP和端口，修改也比较频繁，每次命令行打参数比较麻烦，所以采取配置启动。

> **3.为什么Buffer最大为 65507 bytes？**
> 请参考 UDP协议. 在 IPV4 下为 (65,535 − 8 byte UDP header − 20 byte IP header) = 65,507 bytes

---

如果遇到无法执行程序或者有什么其他的问题，请联系我^_^

Tel : 13524957036

Email : weehowe.z@gmail.com

©2015 weehowe-z　　Fork me on Github