FFMPEG 读取rtsp源视频

蒋亮

2019-9-20

背景

从一个json文件中读取rtsp视频源，以关键帧或视频的形式保存下来

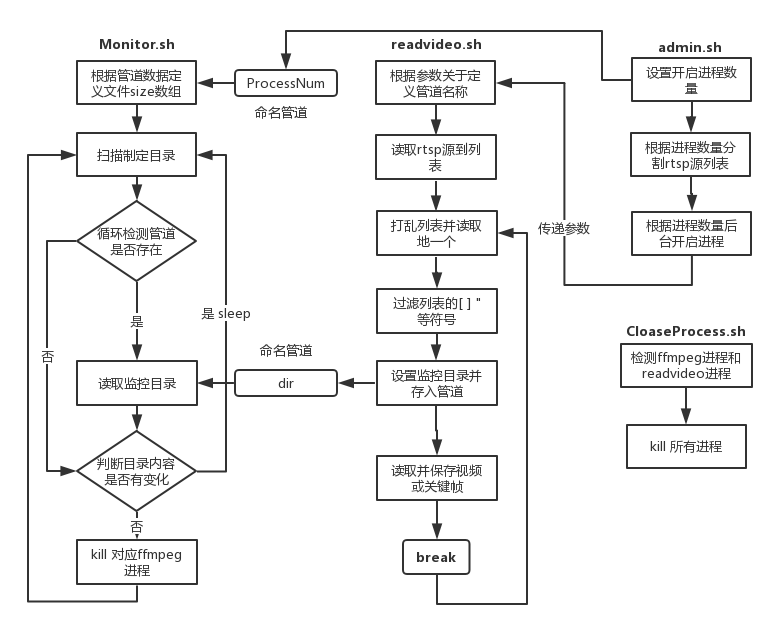
大纲流程图

Monitor.sh: 作为一个监控进程，防止ffmpeg读取视频过程卡在中间过程，导致进程一直处于阻塞状态，这种状态一般会出现在ffmpeg读取到一半，网络断开。

Readvideo.sh: 作为读取视频流并保存关键帧或视频的进程， 反复读取rtsp 的json 列表，并随机打乱顺序，每次只读取第一条视频源里面的视频流。

Admin.sh: 作为住进程在后台开启多个readvideo.sh进程，达到多线程目的

CloseProcess.sh: 关闭后台的ffmpeg进程和readvideo进程



流程详解：

Readvideo.sh:

1. 根据参数定义管道
2. cat 读取 rtsp.json 文件根据keys提取并输出重定向到tmp文件
3. awk命令 对keys打乱顺序
4. 读取地一个其中一个key，去掉引号，并过滤掉 ‘[’或’]’
5. 根据key获取rtsp
6. 定义文件保存路径 ./年月日小时/rtsp的rmd5字符串/分秒.mp4(jpg)
7. 保存视频/图片
8. Break
9. 重复步骤4
10. 重复步骤3

备注：

关于break的解释：break的作用是保证内层循环只读取打乱后的rtsp的第一个，如果去掉break，则是每次打乱rtsp之后都将所有的rtsp读取完

Monitor.sh:

1. 根据进程数定义文件大小数组，存储每次的变化大小
2. 循环检测命名管道文件是否存在，若存在则读取里面的内容，读取之后删除管道文件(此处采用命名管道的非阻塞模式，但是每次读取管道内容之后不会删除内容，导致后面)
3. 扫描目标文件夹的大小size是否变化
4. 如果没有变化，找到对应的ffmpeg进程，并kill
5. 将新的文件夹size值赋值给旧的文件夹size值、
6. Sleep 并重复步骤2

Admin.sh:

1. 设置后台开启进程数
2. 按照进程数，将原始的rtsp文件内容等分(如果等分不了，则放到最后一个子rtsp.json文件)
3. 循环后台开启多个readvideo.sh进程 并 输入索引编号

CloseProcess.sh

略

参数详解：

Readvideo.sh

        #define the file path

        Dir=$(date '+%y%m%d%H')

        ms=$(date '+%M%S')

        #use md5 map the rtsp to a unique string

        rtspmd5=$(echo $y | md5sum |cut -d" " -f1)

        #Dir="../$Dir/$rtspmd5/$ms"

        Dir="$Dir/$rtspmd5/$ms"

定义命名规则，根据shell获取系统时间，如当前时间为2019.9.22 ， 下午15.30.22,此时获取到的rtsp 为:”rtsp://admin:data2010@192.168.3.201:554/h264/ch33/main/av\_stream” 为了让同一个rtsp源的视频或关键帧保存至同一个文件夹，则需要唯一标识符，于是采用md5将rtsp映射为唯一的32位字符串

19092315/9be0c8efe7e7ed6a911e1b44dcaa6025/3022/ 递归建立这个文件夹，并通过管道通知Monitor.sh进程监控该文件夹状况

ffmpeg -stimeout 5000000 -i $prot  -vf select='eq(pict\_type\,I)'  -vframes 120  "$Dir/video.mp4"

-stimeout 5000000: 设置链接超时选项5s,这个选项对rtsp协议有用，对rtmp需要使用 -timeout 5000000 选项

-i $port: -i 表示input file/video stream

-vf: 表示滤镜，select='eq(pict\_type\,I)’ 表示关键帧选取，I这个参数则表示关键针

-vframes 120 表示保存120帧的视频，也可以替换成 -t 100 表示保存100秒的视频

$Dir/video.mp4: 19092315/9be0c8efe7e7ed6a911e1b44dcaa6025/3022/video.mp4

ffmpeg -stimeout 5000000 -i $y -vf select='eq(pict\_type\,I)\*not(mod(n\,2))' -vsync 0 -vframes 120 "$Dir/img\_%5d.jpg”

-vf select=’’eq(pict\_type\,I)\*not(mod(n\,2))“: 表示每隔两帧保存一帧关键帧

-vsync 0 不增加或减少Frame，减少粘滞帧

Img\_%5d.jpg: 以img\_00001.jpg img\_00002.jpg 方式快速命名保存一系列图片

Monitor.sh

filesizenew=$(ls -l "${Dirnew[$i]}/" | awk '{ print $5 }')

ls -l 获取文件夹下所有的文件的详细信息，其中包含各个文件的详细大小信息

adimn.sh

for (( i=0; i<$ProcessNum; i++ ))

do

    {

        ./readvideo.sh $i

    }&

done

wait

根据设置进程数量，开启相应个数的进程