更新

[Ffmpeg 进行hls切片的几个参数解析](#_2017/1/4)

[支持m3u8的播放器](#_2017/1/4_1)

一、安装ffmpeg以及相关库

1. 安装依赖包

yum install autoconf automake bzip2 cmake freetype-devel gcc gcc-c++ git libtool make mercurial nasm pkgconfig zlib-devel

执行时没有发现错误。

2．安装yasm， ffmpeg编译中为了提高编译速度，使用了汇编指令，于是需要使用这个工具。

mkdir hls\_ffmpeg\_source

cd hls\_ffmpeg\_source

wget <http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz>

tar xvf yasm-1.2.0.tar.gz

cd yasm-1.2.0

./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin"

make

make install

echo

这里注意，如果从git上克隆源代码，得到的yasm版本会比较低，导致后面安装x264会报错，Found no assembler Minimum version is yasm-1.2.0,所以这里安装1.2.0版本的yasm（测试时安装1.3.0的yasm也会报同样的错）。

3．安装h.x264库

cd hls\_ffmpeg\_source

git clone --depth 1 git://git.videolan.org/x264

cd x264

PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" --enable-static

make

make install

echo

执行没有发现错误。

4.安装x265库

cd hls\_ffmpeg\_source

hg clone https://bitbucket.org/multicoreware/x265

cd hls\_ffmpeg\_source/x265/build/linux

cmake -G "Unix Makefiles" -DCMAKE\_INSTALL\_PREFIX="$HOME/ffmpeg\_build" -DENABLE\_SHARED:bool=off ../../source

make

make install

echo

在执行make时卡主不动，先跳过。

网上评价说x265比x264的压缩率要高，解码难度要高，之后再安装。

5.安装libfdk\_aac

cd hls\_ffmpeg\_source

git clone --depth 1 git://git.code.sf.net/p/opencore-amr/fdk-aac

cd fdk-aac

autoreconf -fiv

./configure --prefix="$HOME/ffmpeg\_build" --disable-shared

make

make install

echo

没有发现错误。

6．安装libmp3lame

cd hls\_ffmpeg\_source

curl -L -O http://downloads.sourceforge.net/project/lame/lame/3.99/lame-3.99.5.tar.gz

tar xzvf lame-3.99.5.tar.gz

cd lame-3.99.5

./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/bin" --disable-shared --enable-nasm

make

make install

echo

没有发现错误。

7.安装libopus

cd hls\_ffmpeg\_source

git clone http://git.opus-codec.org/opus.git

cd opus

autoreconf -fiv

PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg\_build" --disable-shared

make

make install

echo

在make和make install中有error，不过还没有发现问题在哪里或者说影响到什么，先略过。

8.安装libogg

cd hls\_ffmpeg\_source

curl -O http://downloads.xiph.org/releases/ogg/libogg-1.3.2.tar.gz

tar xzvf libogg-1.3.2.tar.gz

cd libogg-1.3.2

./configure --prefix="$HOME/ffmpeg\_build" --disable-shared

make

make install

echo

没有发现问题。

9．安装libvorbis

cd hls\_ffmpeg\_source

curl -O http://downloads.xiph.org/releases/vorbis/libvorbis-1.3.4.tar.gz

tar xzvf libvorbis-1.3.4.tar.gz

cd libvorbis-1.3.4

./configure --prefix="$HOME/ffmpeg\_build" --with-ogg="$HOME/ffmpeg\_build" --disable-shared

make

make install

没有发现问题

10．安装libvpx

cd hls\_ffmpeg\_source

git clone https://github.com/webmproject/libvpx cd libvpx

./configure --prefix="$HOME/ffmpeg\_build" --disable-examples

make

make install

原本的 git clone --depth 1 <https://chromium.googlesource.com/webm/libvpx.git> 因为网络的问题不能正常下载，所以换源从git下载。

Make 与 make install 过程中报错，

test/partial\_idct\_test.cc: In function ‘int16\_t<unnamed>::MaxSupportedCoeff(void (\*)(const tran\_low\_t\*, uint8\_t\*, int, int))’:

应该是测试例子有问题，暂时不管，看看会影响到哪里。

11．安装ffmpeg

cd hls\_ffmpeg\_source

curl -O http://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2

tar xjvf ffmpeg-snapshot.tar.bz2

cd ffmpeg

PKG\_CONFIG\_PATH="$HOME/ffmpeg\_build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg\_build" --extra-cflags="-I$HOME/ffmpeg\_build/include" --extra-ldflags="-L$HOME/ffmpeg\_build/lib -ldl" --bindir="$HOME/bin" --pkg-config-flags="--static" --enable-gpl --enable-nonfree --enable-libfdk\_aac --enable-libfreetype --enable-libmp3lame --enable-libopus --enable-libvorbis --enable-libvpx --enable-libx264

make

make install

hash -r

因为没有安装x265,所以去掉 --enable-libx265

如果报错：

opus not found using pkg-config

要先安装 libopus-devel

Vim /etc/yum.repos.d/linuxtech.repo

Insert

[linuxtech]

name=LinuxTECH

baseurl=http://pkgrepo.linuxtech.net/el6/release/

enabled=1

gpgcheck=1

gpgkey=http://pkgrepo.linuxtech.net/el6/release/RPM-GPG-KEY-LinuxTECH.NET

esc

:wq

yum install libopus-devel

在安装过程中会有一些warning，略过。

二、安装Nginx相关模块

1.环境准备

yum install –y pcre pcre-devel

yum install –y zlib zlib-devel

没有发现问题

2.下载nginx及rtmp模块

wget http://nginx.org/download/nginx-1.6.2.tar.gz

tar xzvf nginx\_1.6.2.tar.gz

git clone git://github.com/arut/nginx-rtmp-module.git

后面编译有报错，SSL modules require the OpenSSL library，所以要先安装openssl

yum -y install openssl openssl-devel

3. 编译nginx并附加rtmp模块

./configure --prefix=/usr/local/nginx --add-module=../nginx-rtmp-module --with-http\_stub\_status\_module

make

make install

安装完成后，nginx位于/usr/local/nginx/sbin目录下，配置文件nginx.conf在/usr/local/nginx/conf目录下

++++++++测试nginx是否安装正确+++++++++++

#cd /usr/local/nginx

#./sbin/nginx -c ./conf/nginx.conf

打开网页http://localhost，如果显示Welcome表示安装成功。

除了缺少openssl以外没有发现错误。

三、测试

1．测试hls功能。

用ffmpeg进行切片（高版本的ffmpeg可以直接进行切片，不用安装segmenter）

提前拷贝test1.mp4到/home/limi/Video 下

ffmpeg -i input\_file\_path -c:v libx264 -c:a aac -strict -2 -f hls -hls\_list\_size 0 -hls\_time 5 output\_file\_path

ffmpeg -i test1.mp4 -c:v libx264 -c:a aac -f hls hls/test.m3u8

切片以及m3u8文件将会放在output\_file\_path下。

## 2017/1/4

Ffmpeg切片参数解析

(1).ffmpeg转化成HLS时附带的指令

-hls\_time n: 设置每片的长度，默认值为2。单位为秒

-hls\_list\_size n:设置播放列表保存的最多条目，设置为0会保存所有片信息，默认值为5

-hls\_wrap n:设置多少片之后开始覆盖，如果设置为0则不会覆盖，默认值为0.这个选项能够避免在磁盘上存储过多的片，而且能够限制写入磁盘的最多的片的数量

-hls\_start\_number n:设置播放列表中sequence number的值为number，默认值为0

(2).对ffmpeg切片指令的使用

ffmpeg -i output.mp4 -c:v libx264 -c:a aac -strict -2 -f hls -hls\_list\_size 0 -hls\_time 5 output1.m3u8

vim m3u8test.html

<html>

<head>

<title>HTTP Live Streaming Example</title>

</head>

<body>

<video

src="/hls/test.m3u8"

height="300" width="400"

>

</video>

</body>

</html>

因为没有使用额外的播放器，所以只能在apple自带的Safari浏览器下观看。

访问<http://yourip/testWeb/m3u8test.html> 即可播放。

### 2017/1/4

找到一个支持m3u8的播放器，mediaelement

在官网 <http://www.mediaelementjs.com/> 下载压缩包之后，解压到/usr/local/nginx/html下,将mediaelementplayer-hls.html复制一份为hls-test.html,修改其中的video标签中src地址 为 /hls2/test.m3u8,保存后用浏览器访问 yourip/web/demo/hls-test.html 即可看到点播内容。

将mediaelementplayer-hls.html再复制一份为live-test.html，修改其中的video标签中src地址为 /hls/mystream.m3u8,保存后使用ffmpeg进行推流模拟直播（或者使用open broadcast software 进行摄像头采集，因为我的显卡有问题所以没有使用）

ffmpeg -re -i test1.mp4 -vcodec libx264 -acodec aac -f flv rtmp://127.0.0.1/hls/mystream

等一会，在服务器中生成了stream.m3u8文件之后，使用浏览器访问yourip/web/demo/live-test.html 就可以看到直播。

目前存在的问题：

在观看过程中偶尔有卡顿的情况，刷新后才能重新开始观看。

在服务器端推流结束后，客户端播放没有及时停止，要等到播放结束才会停止。

2.测试rtmp功能。

在/usr/local/nginx/nginx.conf 中添加

rtmp{

server{

listen 1935;

chunk\_size 4000;

#live

application rtmp\_test{

live on;

}

}

}

重启nginx服务器。

使用ffmpeg进行推流

ffmpeg -re -i test1.mp4 -vcodec libx264 -acodec aac -f flv rtmp://127.0.0.1/rtmp\_test/test

其中rtmp://127.0.0.1/rtmp\_test/test 是推流的地址，test1.mp4是源文件。

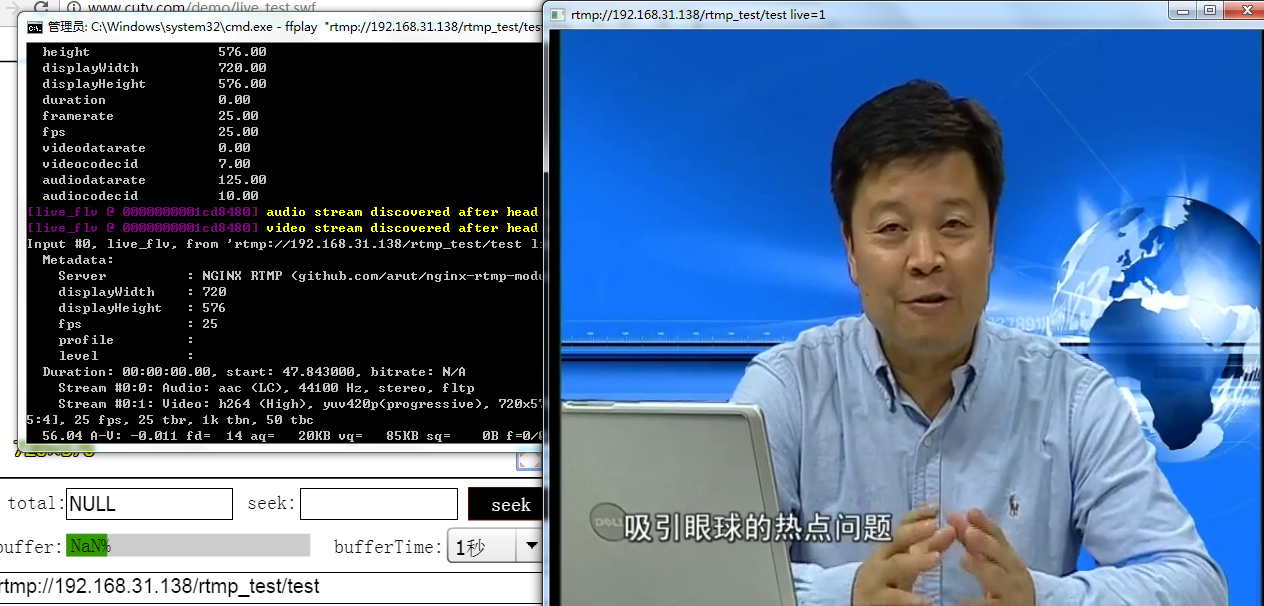
在<http://www.cutv.com/demo/live_test.swf> (打开后空白的话，需要刷新一下)，输入推流的地址即可访问。



使用ffplay也可以播放

在cmd 下 输入 ffplay “rtmp://192.168.31.138/rtmp\_test/test live=1”

即可播放。（已安装好并将ffplay放在系统变量里）



3.在服务器内部进行转换（将rtmp流转换成hls）

修改nginx.conf文件

在rtmp-server 模块中加入

application hls {

# live on; 开启实时

live on;

#hls on ;开启hls

hls on;

#hls\_path;ts文件存放路径

hls\_path /usr/local/nginx/html/hls;

# hls\_fragment ；每个ts文件包含5s的视频内容

hls\_fragment 5s;

}

在http模块中额外增加一个server

server {

listen 8080;

location /hls {

# Serve HLS fragments

types {

application/vnd.apple.mpegurl m3u8;

video/mp2t ts;

}

root html;

expires -1;

}

}

在hls文件夹中新建文件

Vim player.html

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**
5. **<title>**HLS Player**</title>**
6. **</head>**
7. **<body>**
8. **<video**  height="720" width="1280" controls**>**
9. **<source** src="/hls/mystream.m3u8" type="application/vnd.apple.mpegurl" **/>**
10. **<p** class="warning"**>**Your browser does not support HTML5 video.**</p>**
11. **</video>**
12. **</body>**
13. **</html>**

重启nginx服务器。

使用ffmpeg进行推流

ffmpeg -re -i test1.mp4 -vcodec libx264 -acodec aac -f flv rtmp://127.0.0.1/hls/mystream

使用Safari浏览器访问 192.168.31.138/hls/player.html

或者使用ffplay，注意这里的m3u8文件是实时生成的，所以ffplay进行播放时不会及时的开始播放，要等文件生成后才可以。

ffplay -report -v warning http://192.168.31.138/hls/mystream.m3u8



在推流期间可以看到ts文件以及m3u8文件，推流结束之后文件消失。

4.查看nginx 的状态

在http-server 中添加

location /stat {

rtmp\_stat all;

rtmp\_stat\_stylesheet stat.xsl;

}

location /stat.xsl {

root /path/to/nginx-rtmp-module/;

}

location /control {

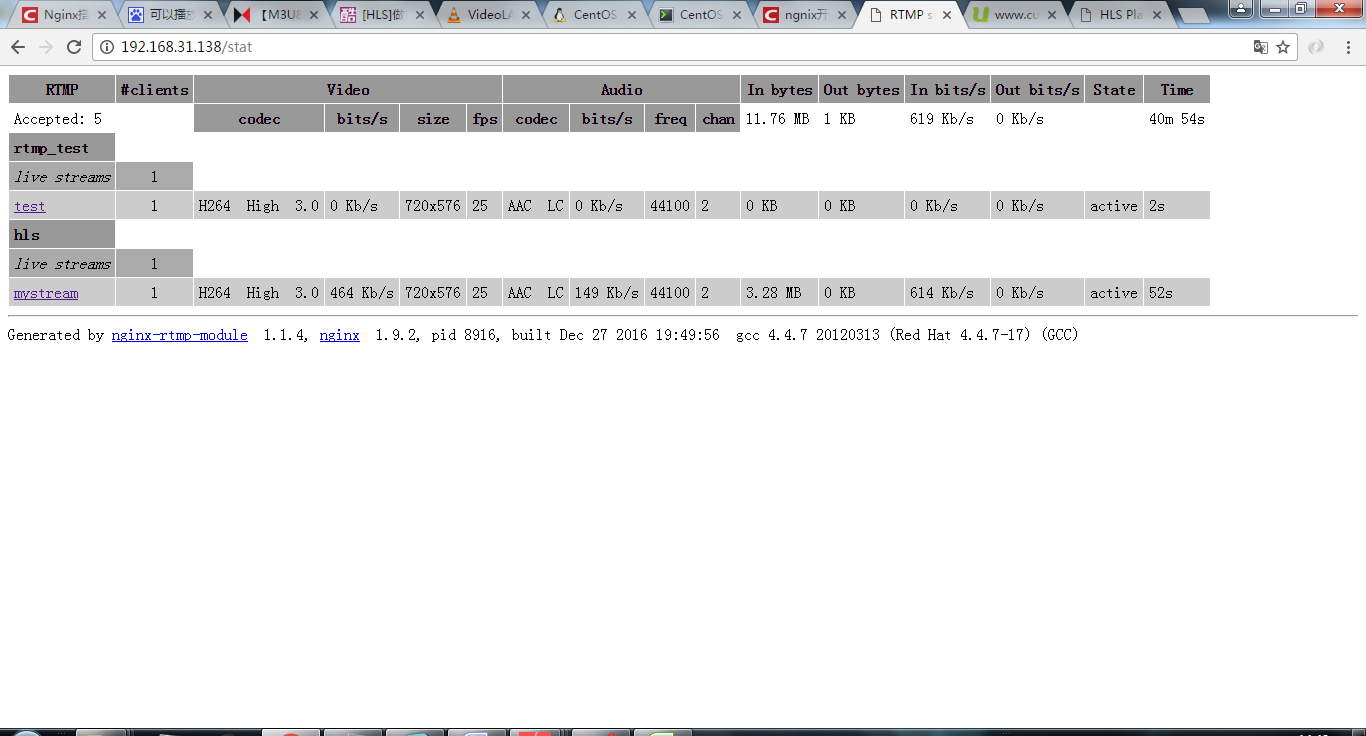
rtmp\_control all;

}

其中 root 是nginx-rtmp-module 存放的位置，加载配置文件启动之后，访问

<http://yourip:80/stat>

此处的80是server配置中所监听的端口，访问后可以看到



可以看到当前的流数量，以及名字等信息。

参考文档：

<http://blog.csdn.net/chinabinlang/article/details/45646821> FFMpeg处理RTMP流的总结

<http://www.linuxidc.com/Linux/2015-01/111182.htm> 使用Nginx+FFmpeg搭建HLS直播转码服务器

<http://blog.csdn.net/cjsafty/article/details/9108587>基于nginx的hls直播系统

<http://blog.csdn.net/tao_627/article/details/22271559> 使用Nginx-rtmp-module搭建hls直播

<http://trac.ffmpeg.org/wiki/CompilationGuide/Centos> Compile FFmpeg on CentOS

<http://ffmpeg-users.933282.n4.nabble.com/ERROR-opus-not-found-using-pkg-config-td4677265.html#a4677298> ERROR: opus not found using pkg-config

<http://www.jianshu.com/p/e1395a896f36> 使用ffmpeg对视频进行TS切片 几个参数的意义