目的:使Nginx支持Rtmp协议推流，并支持HLS分发功能及FFmpeg转码多码率功能。

一、准备工作

模块:nginx-rtmp-module-master(支持rtmp协议)

下载地址:

http://nginx.org

https://github.com/arut/nginx-rtmp-module

1、安装依赖包:

#yum -y install gcc glibc glibc-devel make nasm pkgconfig lib-devel openssl-devel expat-devel gettext-devel libtool mhash.x86\_64 perl-Digest-SHA1.x86\_64 gcc-c++

2、安装git工具:

#mkdir soft-source

#cd soft-source

#wget http://www.codemonkey.org.uk/projects/git-snapshots/git/git-latest.tar.gz

#tar xzvf git-latest.tar.gz

#cd git-2013-02-04

#autoconf

#./configure

#make && make install

# git --version

git version 1.8.1.GIT

#cd ..

【错误处理】

如果 git-latest.tar.gz大小为0，请下载git-latest-tar.xz

然后xz -d git-latest.tar.xz解压为.tar

再tar xvf git-latest.tar

3、安装ffmpeg及其依赖包：

++++++++Yasm+++++++++++

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

#tar xzvf yasm-1.2.0.tar.gz

#cd yasm-1.2.0

#./configure

#make

#make install

#cd ..

++++++++x264+++++++++++

#git clone git://git.videolan.org/x264

#cd x264

#./configure --enable-shared

#make

#make install

#cd ..

错误处理：直接 git clone git://git.videolan.org/x264 可能下载不下来

换源 git clone <https://github.com/mirror/x264.git> x264,后面的x264是放到那个文件夹的意思

++++++++LAME+++++++++++

#wget 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 --enable-nasm

#make

#make install

#cd ..

++++++++libogg+++++++++++

#wget http://downloads.xiph.org/releases/ogg/libogg-1.3.0.tar.gz

#tar xzvf libogg-1.3.0.tar.gz

#cd libogg-1.3.0

#./configure

#make

#make install

#cd ..

错误：

遇到下载很慢的情况，等一会

++++++++libvorbis+++++++++++

#wget http://downloads.xiph.org/releases/vorbis/libvorbis-1.3.3.tar.gz

#tar xzvf libvorbis-1.3.3.tar.gz

#cd libvorbis-1.3.3

#./configure

#make

#make install

#cd ..

++++++++libvpx+++++++++++

#git clone http://git.chromium.org/webm/libvpx.git

#cd libvpx

#./configure --enable-shared

#make

#make install

#cd ..

镜像地址：

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

++++++++FAAD2+++++++++++

#wget http://downloads.sourceforge.net/project/faac/faad2-src/faad2-2.7/faad2-2.7.tar.gz

#tar zxvf faad2-2.7.tar.gz

#cd faad2-2.7

#./configure

#make

#make install

#cd ..

++++++++FAAC+++++++++++

#wget http://downloads.sourceforge.net/project/faac/faac-src/faac-1.28/faac-1.28.tar.gz

#tar zxvf faac-1.28.tar.gz

#cd faac-1.28

#./configure

#make

#make install

#cd ..

【错误处理】

编译FAAC-1.28时遇到错误：

mpeg4ip.h:126: error: new declaration ‘char\* strcasestr(const char\*, const char\*)’

解决方法：

从123行开始修改此文件mpeg4ip.h，到129行结束。

修改前：

#ifdef \_\_cplusplus

extern "C" {

#endif

char \*strcasestr(const char \*haystack, const char \*needle);

#ifdef \_\_cplusplus

}

#endif

修改后：

#ifdef \_\_cplusplus

extern "C++" {

#endif

const char \*strcasestr(const char \*haystack, const char \*needle);

#ifdef \_\_cplusplus

}

#endif

++++++++Xvid+++++++++++

#wget http://downloads.xvid.org/downloads/xvidcore-1.3.2.tar.gz

#tar zxvf xvidcore-1.3.2.tar.gz

#cd xvidcore/build/generic

#./configure

#make

#make install

cd ../../../

++++++++ffmpeg+++++++++++

#git clone git://source.ffmpeg.org/ffmpeg

#cd ffmpeg

#./configure --prefix=/opt/ffmpeg/ --enable-version3 --enable-libvpx --enable-libfaac --enable-libmp3lame --enable-libvorbis --enable-libx264 --enable-libxvid --enable-shared --enable-gpl --enable-postproc --enable-nonfree --enable-avfilter --enable-pthreads

#make && make install

#cd ..

【错误处理】

如果提示libvpx decoder version must be >=0.91，请从Baidu搜索一下libvpx-v1.1.0.tar.bz下载。

bzip2 -d libvpx-v1.1.0.tar.bz2

tar xvf libvpx-v1.1.0.tar.bz2

cd libvpx-v1.1.0

./configure --enable-shared --enable-vp8

make

make install

修改/etc/ld.so.conf如下:

include ld.so.conf.d/\*.conf

/lib

/lib64

/usr/lib

/usr/lib64

/usr/local/lib

/usr/local/lib64

/opt/ffmpeg/lib

#ldconfig

【说明】

动态装入器找到共享库要依靠两个文件 — /etc/ld.so.conf 和 /etc/ld.so.cache。

安装完成后，ffmpeg位于/opt/ffmpeg/bin目录下。

错误：

添加上参数 -enable--libfaac 时，总会报错，unknown option“-enable--libfaac”，暂时跳过。

参考文档:

<http://blog.csdn.net/long_zhe_/article/details/47026159> 基于HLS的流媒体方案

<https://my.oschina.net/joshuashaw/blog/516015> rtmp与hls流媒体服务器搭建：ubuntu下Nginx搭建初探与rtmp-module的添加

<http://redstarofsleep.iteye.com/blog/2123752> nginx+nginx-rtmp-module+ffmpeg搭建流媒体服务器

<http://blog.csdn.net/cdnight/article/details/39156711> centos搭建hls系统

<http://blog.csdn.net/cdnight/article/details/39157125> centos搭建hls系统 新

<http://blog.csdn.net/cdnight/article/details/39160067>【整理】HLS视频协议第一弹--centos下面配置ffmpeg，segmenter以适应hls切片需要

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

<http://www.cppblog.com/Error/archive/2014/01/14/205365.aspx> centos编译安装git最新版本遇到的问题及解决方法

<http://blog.csdn.net/mengzhengjie/article/details/51191140> CentOS6下基于Nginx搭建mp4/flv流媒体服务器(可随意拖动)并支持RTMP/HLS协议(含转码工具)

FAAC编译有问题

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

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

3.编译nginx-rtmp

./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表示安装下正确，如果没有显示，请查看一下nginx的日志。

++++++++测试RTMP+++++++++++

修改/usr/local/nginx/conf/nginx.conf的内容如下：

#debug

daemon off;

master\_process off;

error\_log ./error.log debug;

events{

worker\_connections 1024;

}

rtmp{

server {

listen 1935;

chunk\_size 4000;

#live

application myapp {

live on;

}

}

从网上下载一款RTMP推流工具，我使用的OBS（Open Broadcaster Software），开始推流rtmp://your\_ip/myapp/test，使用播放器（http://www.cutv.com/demo/live\_test.swf）查看是否正常。

测试代码：

ffmpeg -re -i rtmp://live.hkstv.hk.lxdns.com/live/hks -c:a aac -ac 1 -strict -2 -b:a 64k -c:v libx264 -f flv rtmp://192.168.31.248/myapp/test

ffmpeg -re -i /home/limi/Videos/test.mp4 -vcodec libx264 -acodec aac -f flv rtmp://localhost/myapp/test

++++++++测试HLS切片功能+++++++++++

修改/usr/local/nginx/conf/nginx.conf的内容如下：

#debug

daemon off;

master\_process off;

error\_log ./error.log debug;

events{

worker\_connections 1024;

}

rtmp{

server {

listen 1935;

chunk\_size 4000;

#live

application myapp {

live on;

hls on;

hls\_path /tmp/hls;

hls\_fragment 2s;

hls\_playlist\_length 6s;

}

}

}

#HTTP

http{

server {

listen 80;

#welcome

location / {

root html;

index index.html index.htm;

}

#hls

location /hls {

types {

application/vnd.apple.mpegusr m3u8;

video/mp2t ts;

}

root /tmp;

add\_header Cache-Control no-cache;

}

}

}

使用VLC或iPAD上的播放器进行查看 http://yourip/hls/test.m3u8。

++++++++测试FFMPEG转码功能+++++++++++

修改/usr/local/nginx/conf/nginx.conf的内容如下：

#debug

daemon off;

master\_process off;

error\_log ./error.log debug;

events{

worker\_connections 1024;

}

rtmp{

server {

listen 1935;

chunk\_size 4000;

#live

application myapp {

live on;

exec /opt/ffmpeg/bin/ffmpeg -i rtmp://localhost/myapp/$name

-c:a copy -c:v libx264 -b:v 512K -g 30 -f flv rtmp://localhost/hls/$name\_low;

}

application hls {

live on;

hls on;

hls\_path /tmp/hls;

hls\_nested on;

hls\_fragment 2s;

hls\_playlist\_length 6s;

hls\_variant \_hi BANDWIDTH=640000;

}

}

}

#HTTP

http{

server {

listen 80;

#welcome

location / {

root html;

index index.html index.htm;

}

#hls

location /hls {

types {

application/vnd.apple.mpegusr m3u8;

video/mp2t ts;

}

root /tmp;

add\_header Cache-Control no-cache;

}

}

}

使用ffmpeg转码时，

exec /opt/ffmpeg/bin/ffmpeg -i rtmp://localhost/myapp/$name

-c:a copy -c:v libx264 -b:v 512K -g 30 -f flv rtmp://localhost/hls/$name\_low;

仅对视频进行转码，音频不做处理，同时向流从myapp转推到hls， hls\_variant会生成一个多码率的m3u8文件，同时把切片文件存放到test\_low目录下，访问多码率时，直接访问http://yourip/hls/test.m3u8，根据这个m3u8中的实现的内容访问相应的码流，在本例中，实际码流URL为http://yourip/hls/test\_low/index.m3u8

参考文档

<http://blog.csdn.net/growdane/article/details/51570806> nginx配合rtmp模块搭建流服务器

<http://www.cutv.com/demo/live_test.swf> 测试地址

<http://blog.csdn.net/mengmakies/article/details/51794248?ref=myread> 一言不合你就用环信搞个直播APP

<http://blog.csdn.net/kl222/article/details/12968815> Nginx搭建hls流媒体服务器