**Ubuntu 14.10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**

Step1. 准备ubuntu环境：

sudo apt-get update

Step2. 安装必要的libs：

sudo apt-get install build-essential libpcre3 libpcre3-dev libssl-dev

sudo apt-get install gcc make automake bzip2 unzip patch subversion libjpeg62-dev

Step3. 创建工作路径和安装yasm：

sudo apt-get install yasm(如果版本大于等于1.2.0)

su root

mkdir ~/working

cd ~/working

wget http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz tar zxvf yasm-1.2.0.tar.gz cd yasm-1.2.0 ./configure make && make install

Step4. 卸载原有的ffmpeg和x264：

apt-get remove ffmpeg x264

Step5. 安装一些MPlayer编解码库（仅限64位系统）：

wget -c <http://www2.mplayerhq.hu/MPlayer/releases/codecs/essential-amd64-20071007.tar.bz2>

tar xvjf essential-amd64-20071007.tar.bz2

mkdir /usr/local/lib/codecs

cp -Rvp essential-amd64-20071007/\* /usr/local/lib/codecs/

编辑下面文件

vim /etc/ld.so.conf

添加以下两行到上面的文件里

/usr/lib

/usr/local/lib

Step7. 安装常用的编解码库

sudo apt-get install build-essential git-core checkinstall texi2html libfaac-dev \  
libopencore-amrnb-dev libopencore-amrwb-dev libsdl1.2-dev libtheora-dev \  
libvorbis-dev libx11-dev libxfixes-dev zlib1g-dev libxvidcore-dev

Step8. 安装x264

wget <ftp://ftp.videolan.org/pub/videolan/x264/snapshots/last_stable_x264.tar.bz2>

tar xvjf last\_stable\_x264.tar.bz2

cd x264-snapshot-20150803-2245-stable/

./configure --enable-shared --enable-pic

make && make install

Step9. 安装libxpv

wget <https://github.com/webmproject/libvpx/releases/tag/v1.4.0>

tar xvjf libvpx-v1.4.0.tar.bz2

cd libvpx-v1.4.0

./configure --enable-shared --enable-pic

make && make install

Step10. 安装libmp3lame-dev

apt-get install libmp3lame-dev

Step11. 安装FFMPEG

tar xvjf ffmpeg-2.7.2.tar.bz2

cd ffmpeg-2.7.2

./configure --enable-gpl --enable-version3 --enable-shared --enable-nonfree --enable-postproc --enable-libfaac --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libtheora --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libxvid

make && make install

让动态链接库被系统共享

ldconfig

Step12. 安装mplayer and mencoder

//这里需要将ffmpeg的源码目录拷贝到mplayer目录下并命名为ffmpeg同时执行./configure

svn checkout svn:*//svn.mplayerhq.hu/mplayer/trunk mplayer*

cd mplayer/

./configure

make && make install

Step13. 安装[flvtool2](https://github.com/unnu/flvtool2)

1. udo apt-get install ruby
2. gem -v
3. gem souce -l
4. https:*//rubygems.org/*
5. gem sources --remove https:*//rubygems.org/*
6. gem sources -a https:*//ruby.taobao.org/*
7. gem sources -l
8. gem install flvtool2
9. 因为网络的原因，切换为淘宝的服务器
10. ll /usr/local/bin/flvtool2

step14. 查看一下已经安装好的音频和[视频编码器](https://www.baidu.com/s?wd=%E8%A7%86%E9%A2%91%E7%BC%96%E7%A0%81%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)

查看所有所支持的音频编码

mencoder -oac help

查看所有所支持的视频编码  
mencoder -ovc help

Step15. 安装nginx

先安装各种依赖(nginx需要pcre支持,yamdi用来为flv创建关键帧才能随意拖动)

sudo apt-get install gcc g++ libssl-dev zlib1g-dev libpcre3-dev yamdi

第一个是nginx\_mod\_h264\_streaming，让nginx支持flv/mp4流播放

wget http://h264.code-shop.com/download/nginx\_mod\_h264\_streaming-2.2.7.tar.gz

tar zxvf nginx\_mod\_h264\_streaming-2.2.7.tar.gz

注意：先要修改一下这家伙的源码，注释掉nginx\_mod\_h264\_streaming-2.2.7/src/ngx\_http\_streaming\_module.c的158到161行

/\* TODO: Win32 \*/

//if (r->zero\_in\_uri)

// {

// return NGX\_DECLINED;

// }

第二个是nginx-rtmp-module，让nginx支持rtmp/hls协议

wget -O nginx-rtmp-module.zip https://github.com/arut/nginx-rtmp-module/archive/master.zip

unzip nginx-rtmp-module.zip

下载清缓存的模块

wget -O ngx\_cache\_purge.zip https://github.com/FRiCKLE/ngx\_cache\_purge/archive/master.zip

unzip ngx\_cache\_purge.zip

下载安装nginx:

wget http://nginx.org/download/nginx-1.8.0.tar.gz

tar zxvf nginx-1.8.0.tar.gz

cd nginx-1.8.0

./configure --user=daemon --group=daemon --prefix=/usr/local/nginx/ --add-module=../nginx-rtmp-module-master --add-module=../ngx\_cache\_purge-master --add-module=../nginx\_mod\_h264\_streaming-2.2.7 --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_sub\_module --with-http\_gzip\_static\_module --with-http\_flv\_module

make && make install

**编译过程中出错：**

**make[1]: \*\*\* [objs/addon/src/mp4\_reader.o] 错误 1**

**解决方法：**

**vim objs/Makefile (修改objs/Makefile文件, 去掉其中的"-Werror"), 然后就能够正常编译了。**

Step16. 设置nginx为系统服务

sudo wget <https://raw.github.com/JasonGiedymin/nginx-init->ubuntu/master/nginx -O /etc/init.d/nginx

sudo chmod +x /etc/init.d/nginx

sudo update-rc.d nginx defaults

sudo service nginx start

sudo service nginx stop

Step17. 配置Nginx

新建如下目录

/mnt/media/video -> 存放原始视频

/mnt/media/app -> 存放转成m3u8的视频,供http访问(HLS)

/mnt/media/vod -> 存放转换后的flv和mp4视频,供http或rtmp访问

编辑/usr/local/nginx/conf/nginx.conf进行相关配置

FFMPEG HLS切片

ffmpeg**-i source.mp4 -codec:v libx264 -codec:a mp3 -map 0 -f ssegment -segment\_format mpegts -segment\_list playlist.m3u8 -segment\_time 10 out%03d.ts**

ffmpeg -re -i source.mp4 -codec:v libx264 -codec:a libfaac -map 0 -f hls  -hls\_list\_size 6 -hls\_wrap 10 -hls\_time 10 playlist.m3u8