

流媒体服务器安装部署

前言

随着直播行业和5G物联网的快速发展，我们正在进入一个移动互联网的时代，越来越多的应用、产品走进我们的生活，改变着生活的方式。今天我们以直播为例，用ffmpeg搭建流媒体服务器，往后可以举一反三，切换播放源即可实现直播效果。

搭建步骤

1. ffmpeg简介
 2. ffmpeg组件
 - 命令行工具
 - ffmpeg
 - ffplay
 - libavcodec
 - libavformat
 3. 需要安装的软件和插件
 - yasm
 - x264
 - ffmpeg
 - nginx
 4. 工作原理
 5. 视频流测试
 6. 多条流测试
-

一、ffmpeg简介

FFmpeg是一个自由软件，可以运行音频和视频多种格式的录影、转换、流功能，包含了 `libavcodec` 这是一个用于多个项目中音频和视频的解码器库，以及 `libavformat`（一个音频与视频格式转换库）

- ffmpeg的官网地址是: <https://www.ffmpeg.org/>
 - ffmpeg的Github项目地址是: <https://github.com/FFmpeg/FFmpeg>
-

二、ffmpeg组件

FFmpeg项目由以下几部分组成:

- FFmpeg视频文件转换命令行工具,也支持经过实时电视卡抓取和编码成视频文件;
- `ffserver`: 基于HTTP、RTSP用于实时广播的多媒体服务器.也支持时间平移;
- `ffplay`: 用SDL和FFmpeg库开发的一个简单的媒体播放器;
- `libavcodec`: 一个包含了所有FFmpeg音视频编解码器的库。为了保证最优性能和高可复用性，大多数编解码器从头开发的;
- `libavformat`: 一个包含了所有的普通音视格式的解析器和产生器的库。———### **三、需要安装的软件和插件**
- 1.yasm 下载地址: `http://yasm.tortall.net/Download.html`
- `yasm`是一个完全重写的NASM汇编工具，最新的`x264`，要求`yasm1.2`以上

```
[root@tingyan home]# tar -zxvf yasm-1.3.0.tar.gz
[root@tingyan home]# cd yasm-1.3.0
[root@tingyan yasm-1.3.0]# ./configure --prefix=/home/yasm
[root@tingyan yasm]# make && make install
```

- 2.x264

下载地址: <http://download.videolan.org/x264/snapshots/>

x264 是一个编码器, 它参照的标准是H.264

由于是手动安装的 yasm, 安装的时候, 可能会报 yasm 找不到, 需在 `/etc/profile`

```
export PATH=$PATH:/home/yasm/bin
```

```
export PKG_CONFIG_PATH=/home/x264/lib/pkgconfig/
```

之后

```
source /etc/profile
```

编译安装:

```
[root@tingyan home]# tar jxvf x264.tar.bz2
[root@tingyan home]# cd x264-snapshot-20120718-2245-stable
[root@tingyan x264-snapshot-20120718-2245-stable]# ./configure --prefix=/home/x264 -
--enable-shared --enable-static --enable-yasm --disable-asm --disable-avs
[root@tingyan x264-snapshot-20120718-2245-stable]# make && make install
```

- 3.ffmpeg

下载地址: <http://www.ffmpeg.org/download.html>

```
[root@tingyan home]# tar -zxvf ffmpeg-3.2.tar.gz
[root@tingyan home]# cd ffmpeg-3.2
```

```
[root@tingyan ffmpeg-3.2]# ./configure --prefix=/home/ffmpeg --enable-shared --enable-yasm --enable-libx264 --enable-gpl --enable-pthreads --extra-cflags=-I/home/x264/include --extra-ldflags=-L/home/x264/lib
```

```
[root@tingyan ffmpeg-3.2]# make && make install
```

编译完成后，修改环境变量，`vim /etc/profile` 添加以下字段：

```
FFMPEG=/home/ffmpeg
X264=/home/x264
YASM=/home/yasm
export FFMPEG X264 YASM
export PATH=$PATH:$FFMPEG/bin:$X264/bin:$YASM/bin
export LD_LIBRARY_PATH=$YASM/lib:$X264/lib:$FFMPEG/lib:$LD_LIBRARY_PATH
```

- 4.测试

进入 `/home/ffmpeg/bin` 目录，执行 `./ffmpeg -v`，如果 `没有提示命令无法识别或找不到` 则安装成功；

可能出现的异常信息：

```
ffmpeg: error while loading shared libraries: libavdevice.so.57: cannot open shared object file: No such file or directory
```

因为ffmpeg未找到相关的依赖包，可通过以下方法解决：

`find / -name libavdevice.so.57` 查找该依赖包的目录地址，一般是在 `ffmpeg` 的 `lib` 目录下。编辑 `vim /etc/ld.so.conf`，将上面查找到的目录添加到最后一行，保存退出，再 `ldconfig` 使配置生效

- 5.nginx

nginx下载地址：<http://nginx.org/download/>

nginx-rtmp-module下载地址：<https://github.com/arut/nginx-rtmp-module>

`nginx-rtmp-module` 解压到 `home` 目录下的软件，注意修改目录名

```
[root@ting home]# wget http://nginx.org/download/nginx-1.14.1.tar.gz
```

```
[root@ting home]# tar -zxvf nginx-1.14.1.tar.gz
```

```
[root@ting home]# cd nginx-1.14.1
```

```
[root@ting nginx-1.14.1]# ./configure --prefix=/home/nginx --add-module=../nginx-rtmp-module/ --with-http_ssl_module
```

```
[root@ting nginx-1.14.1]# make && make install
```

- 6.修改nginx配置文件

```
[root@ting nginx]# vim conf/nginx.conf
```

加入 `rtmp` 模块（`rtmp{}`内容和`http{}`内容同级）

```
rtmp {
    server {
        listen 9000; #指定服务端口
        chunk_size 4000;

        #rtmp 直播流配置
        application live {
            live on;
            allow play all;
        }

        #hls 直播流配置
        application hls {
            live on;
            hls on;
            hls_path /home/var/www/hls;
            hls_fragment 3s;
        }
    }
}
```

其中 `hls_path` 文件夹必须要有读写的权限

在 `http` 中添加 `hls` 配置

```
location /hls {
    types {
        application/vnd.apple.mpegurl m3u8;
        video/mp2t ts;
    }
    root /home/var/www;
    add_header Cache-Control no-cache;
    add_header Access-Control-Allow-Origin *;
}
```

完整的配置如下：

```
worker_processes 1;

events {
    worker_connections 1024;
}

rtmp {
    server {
```

```

listen 9000;
chunk_size 4000;

#rtmp 直播流配置
application rtmplive {
    live on;    #开启直播模式
    allow play all;
}

#hls 直播流配置
application hls {
    live on;
    hls on;
    hls_path /home/var/www/hls; #ts文件存放目录
    hls_fragment 3s;    #每段ts文件的时长
}
}

http {
    include mime.types;
    default_type application/octet-stream;

    sendfile on;
    keepalive_timeout 65;

    server {
        listen 9001;
        server_name localhost;

        location / {
            root html;
            index index.html index.htm;
        }

        location /hls {
            types {
                application/vnd.apple.mpegurl m3u8;
                video/mp2t ts;
            }
            root /home/var/www;
            add_header Cache-Control no-cache;
            add_header Access-Control-Allow-Origin *;
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}

```

注：需要建立 `/home/var/www/hls` 对应的目录，这个目录是用来存放推 `hls` 协议流时生成的ts文件。

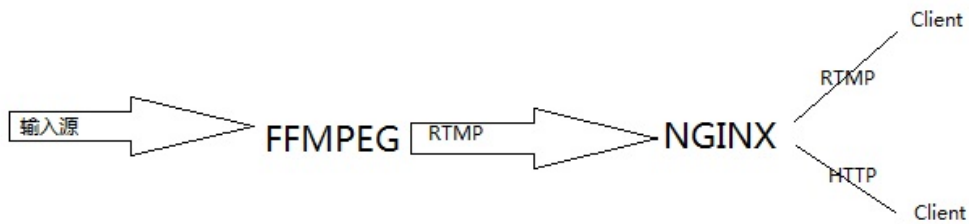
启动nginx：

```
[root@ting nginx]# /home/nginx/sbin/nginx -c /home/nginx/conf/nginx.conf
```

四、工作原理

nginx 配合 ffmpeg 做流媒体服务器的原理是：

nginx 通过 rtmp 模块提供 rtmp 服务，ffmpeg 推送一个 rtmp 流到 nginx，然后客户端通过访问 nginx 来收看实时视频流。HLS也是差不多的原理,只是最终客户端是通过HTTP协议来访问的,但是 ffmpeg 推送流仍然是 rtmp 的。



五、视频流测试

- 1.推流

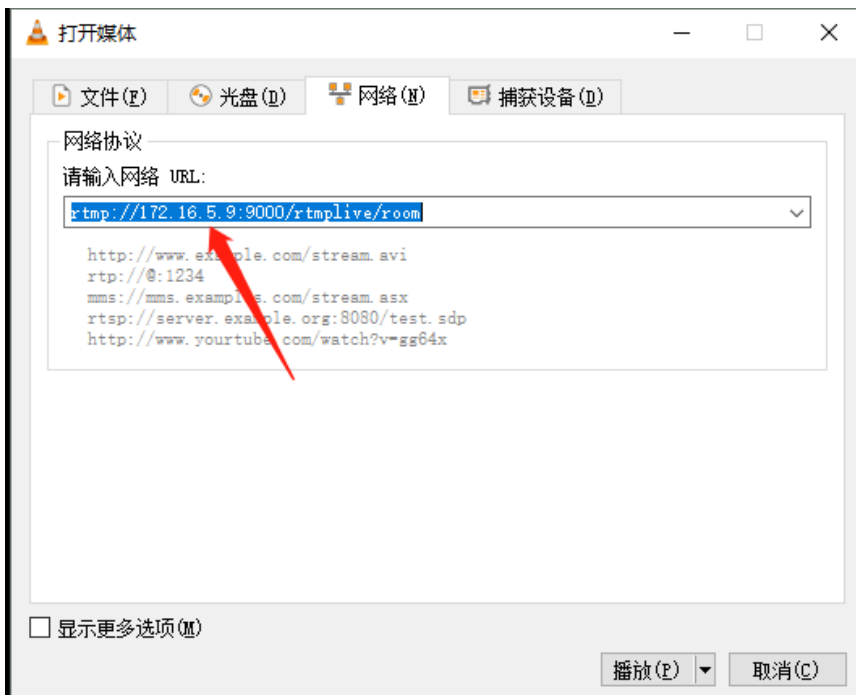
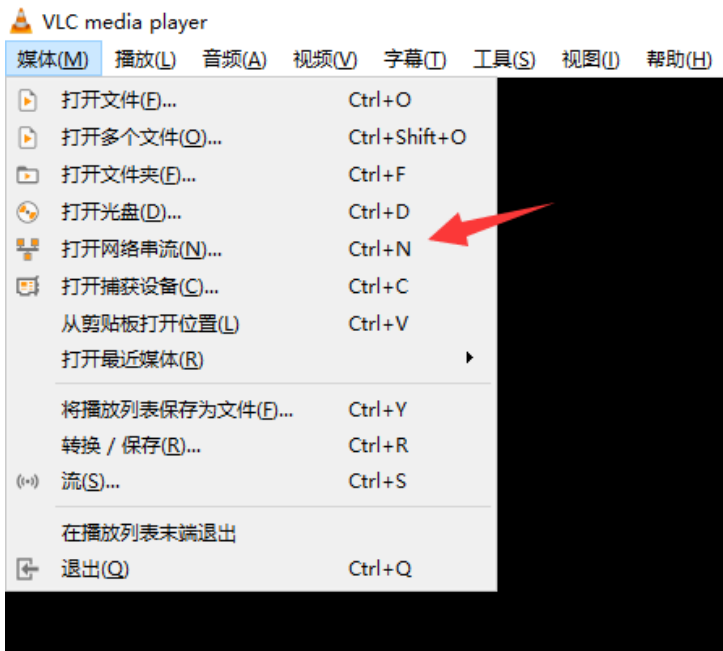
进入到 ffmpeg 的 bin 目录下：

```
./ffmpeg -re -i "/home/test.mp4" -c copy -f flv rtmp://localhost:9000/rtmplive/room
```

test.mp4 为上传到 home 目录的测试文件，room 可自定义名称

- 2.测试流

打开vlc播放器：



把 172.16.5.9 替换为测试服务器的ip地址，然后点击播放。

六、多条流测试

- 1. 一个端口多条流

多条流对应一个端口，直接修改推流命令即可：

如：推hls协议的流

```
./ffmpeg -re -i "/home/test.mp4" -c copy -f flv rtmp://localhost:9000/hls/room
```

```
./ffmpeg -re -i "/home/test.mp4" -c copy -f flv rtmp://localhost:9000/hls/room2
```

推流命令前面参数不用修改，只需更改参数最后的流标识，`room`、`room2`，每条流对应一个标识。因为是`hls`协议，所以播放的url也会相应改变：

```
http://172.16.5.9:9001/hls/room
```

`hls` 协议是对应 `http` 协议的，对应的端口号也变为了 `9001`，因为在 `http` 模块中，我们配置了 `9001` 对应 `hls`

- 2.多个端口多条流

多条流对应多条流，即每条流都有一个单独的端口作为推流和拉流来使用，现在我们以 `hls` 协议做示例，这时 `nginx` 需要修改配置，修改如下：

在 `rtmp` 模块中添加 `hls`：

```
application hls2 {
    live on;
    hls on;
    hls_path /home/var/www/hls2;
    hls_fragment 3s;
}
```

在 `http` 中添加 `server`：

```
server {
    listen 9002;
    server_name localhost;

    location / {
        root html;
        index index.html index.htm;
    }

    location /hls2 { #这里也是需要添加的字段。
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
        root /home/var/www;
        expires -1;
        add_header Cache-Control no-cache;
        add_header Access-Control-Allow-Origin *;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
```



```
    root    html;
  }
}
```

可以看出来，多了一个 9002 端口对应 hls2，其他配置基本不变，注意需要在相应的目录中新建一个 hls2 目录作为流文件缓存目录，同时需要重启nginx：

```
[root@ting nginx]# /home/nginx/sbin/nginx -c /home/nginx/conf/nginx.conf -s reload
```

推流测试：

```
./ffmpeg -re -i "/home/test.mp4" -c copy -f flv rtmp://localhost:9000/hls/room
```

```
./ffmpeg -re -i "/home/test.mp4" -c copy -f flv rtmp://localhost:9000/hls2/room
```

推流命令只需更改转发的地址即可，即 hls、hls2... hlsn

播放地址：

```
http://172.16.5.9:9001/hls/room
```

```
http://172.16.5.9:9002/hls2/room
```