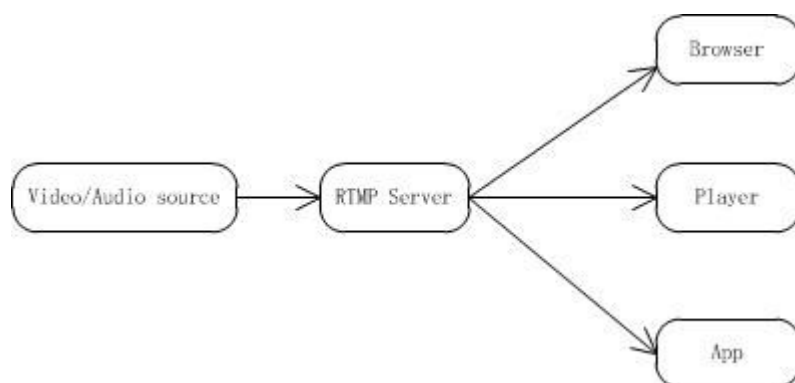


利用 RTMP 实现直播系统

By 革命者(kennymagic@foxmail.com)

一，目标:

利用开源或者免费工具实现一个直播系统；同时支持在浏览器、播放器和嵌入到 PC 应用或者移动 APP 中观看直播。



二，技术选型:

视音频源端:

[Adobe Flash Media Live Encoder 3.2](#)，可以在 windows 和 mac 安装；

如果你已有一些支持 RTMP 的采集设备，那是最好了；

或许你想在自己的应用实现，这样你就必须自己开发处理采集，编码和协议传输了(以后再表)；

RTMP Server:

[FMS](#) -- Adobe 公司出品的服务器，价格昂贵，当然是最正宗的，因为 RTMP 就是 Adobe 公司的私有协议；

[Wowza](#) -- 同样需要授权费，大概是\$55 per month/instance，效率和稳定性都还不错；

[Red5](#) -- 一个开源实现，效率和稳定性都稍微差些，由于它是 java 实现的，所以天生支持跨平台运行；

[Nginx-rtmp-module](#) -- nginx 的一个第三方模块，如果你熟悉 nginx 那是不错的选择，当然它也是免费的，不过功能就没有其他几个丰富了；

这里我选择 nginx+nginx-rtmp-module 作为服务器，这是我认为最容易上手的一种方式了(如果你把 windows 作为服务器那可能麻烦些，官方提供的 windows 二进制版本是没有 nginx-rtmp-module 的，而且 nginx 在 windows 下的性能比 linux 就差太多了)

客户端:

[VLC](#) 也可以安装其他支持 rtmp 的播放器；

[JW Media Player](#) 一个开源的 flash 视音频播放器，利用它我们可以直接在浏览器观看直播；(移动端的浏览器是不知道 flash 的)

ffmpeg/librtmp 如果你希望在自己的应用实现播放器，或者希望在移动端直接接收 RTMP 流，那就要自己开发了(以后再表)

最后选型是: [Adobe Flash Media Live Encoder 3.2](#) + [Nignx-rtmp-module](#) + [JW Media Player](#)

三, 实现

(本文将 nginx 安装到 Centos 6.5 下, IP 为 192.168.0.51)

1, 下载安装 [Adobe Flash Media Live Encoder 3.2](#)

2, 编译安装配置 nginx + nginx-rtmp-module (nginx 1.7 无法编译通过)

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

```
#tar -zxvf nginx-1.6.2.tar.gz
```

```
#git clone https://github.com/arut/nginx-rtmp-module.git (如果没有安装 git 则直接下载 zip 包)
```

```
#cd nginx-1.6.2
```

```
#./configure --add-module=../nginx-rtmp-module --with-http_ssl_module
```

```
#make
```

```
#make install
```

配置(详细查看 <https://github.com/arut/nginx-rtmp-module>), 编辑 nginx/nginx.conf, 增加 rtmp 模块:

```
rtmp {
    server {
        Listen 1935;
        chunk_size 4000;

        #可以将 mylive 改成你想要的名字
        application mylive {
            live on;
        }
    }
}
```

在 http 模块增加:

```
location /stat{
    rtmp_stat all;
    rtmp_stat_stylesheet stat.xsl;
}

location /stat.xsl{
    root html;
}
```

同时需要将 nginx-rtmp-module 源码目录下的 stat.xsl 拷贝到 nginx/html 下, 这样就可以通过网页查看服务器的 RTMP 状态了。

启动 nginx

3, 下载免费版本 JW player: <https://account.jwplayer.com/static/download/jwplayer-6.10.zip> (需要注册登录后才可以下载), 将其解压到 nginx/html 下;

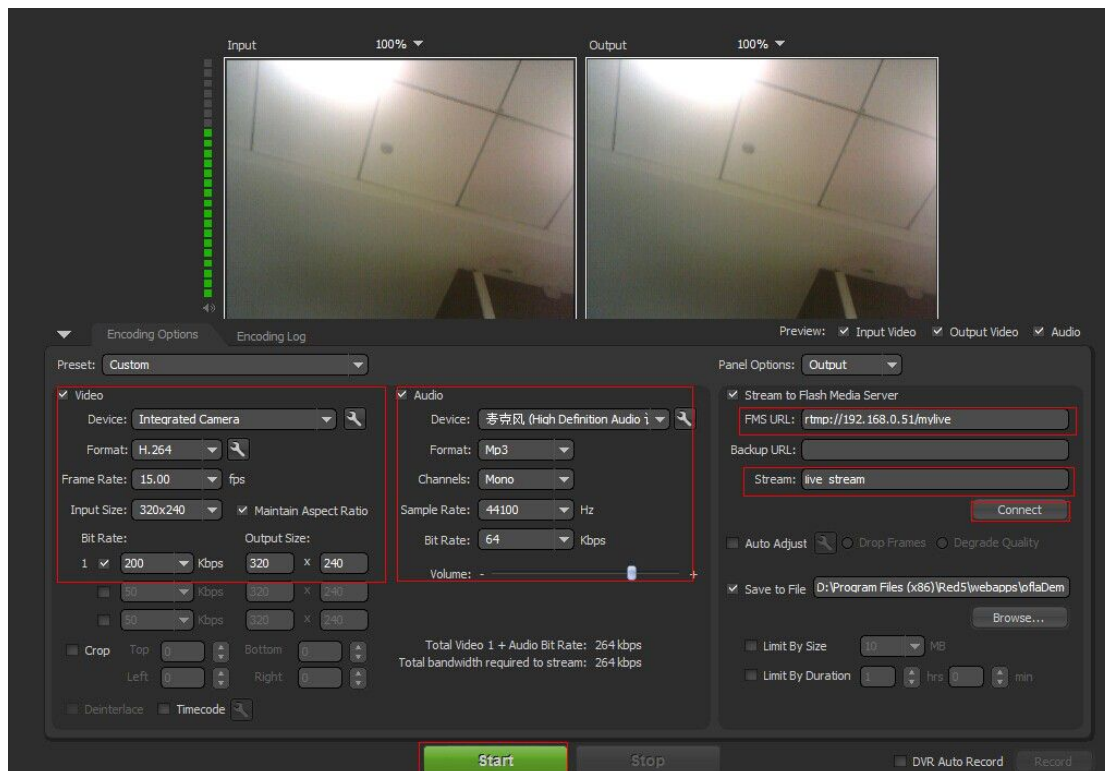
4, 编写 live.html, 保存到 nginx/html 下;

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>live-player</title>
<style type="text/css">
</style>
</head>
<body>
<script type="text/javascript" src="jwplayer/jwplayer.js"></script>
<center>
<b>RTMP 直播系统</b>
<div id="mediaspace">This text will be replaced</div>
<script type="text/javascript">
    var server = window.location.hostname;
    // mylive 对应 nginx.conf 配置项 application 的名字
    // live_stream 对应 Adobe Flash Stream Media Live Encoder 配置的 stream 名称
    var live_url = 'rtmp://' + server + '/mylive' + '/live_stream';
    jwplayer('mediaspace').setup({
        'flashplayer': 'jwplayer/jwplayer.flash.swf',
        'file': live_url,
        'controlbar': 'bottom',
        'width': '760',
        'height': '428',
        //autostart: true, //如果打开此标志, 在打开网页时会自动播放直播流
    });
</script>
</body>
</html>
```

这时候我们在浏览器输入 <http://192.168.0.51/live.html>:



5, Adobe Flash Media Live Encoder 使用



A, 在 FMS URL 填写 RTMP 服务器地址, mylive 则是在 nginx.conf 配置的 rtmp application 的名字(根据自己的配置填写);

B, 在 Stream 填写流的名字, 这个名字相当是一个文件的名字, 填写好后必须修改 live.html 的配置;

这时候我们按“Connect”, 如果没有提示错误则表示连接服务器成功了, 如果提示错误, 则要检查服务器是否运行, 是否已配置了 rtmp 模块相关属性, 尤其是 application 名字是否相对应。

C, 在 Video 栏配置视频相关属性

Device: 表示摄像头信息, 如果有多个摄像头, 则可以通过选择切换;

Format: 视频编码格式, 只支持 H264 和 VP6, 我们选择 H264 就好;

Frame Rate: 帧率, 如果越大则视频会更流畅和清晰, 但同时也会对码流有影响;

Input Size: 摄像头采集的分辨率, 我们最好是按摄像头最好的分辨率设置;

BitRate: 码流, 越大占的带宽就越多, 最好根据输出分辨率和网络来调节, 如果分辨率大, 而码流小则图像就会比较模糊, 如果在网络不好的状况可以降低码流来保证流畅度;

Output Size: 编码输出分辨率, 这个会影响码流和图像质量;

D, 在 Audio 栏配置音频相关属性

Device: 麦克风

Format: 音频编码格式, 一般有 MP3 和 AAC, 在不同平台不一样, 我在 windows 下只有 MP3, 在 mac 下 MP3 和 AAC 都有;

Channels: 声道, Mono 单声道, Stereo 双声道, 根据自己的采集设备进行选择;

Sample Rate: 声音采样率, 一般是越大采集出来的声音会越清晰, 但同时会影响到码流;

Bit Rate: 码流, 类似视频的 Bit Rate;

但我们都设置好了就可以点击“start”进行直播了, 此时我们再点击 <http://192.168.0.51/live.html> 页面的播放按钮, 在 3s 左右后就会看到直播画面;



在我的实测中(局域网和 Internet 都有测试), 一般都在 1-3s 之间的延迟;

四, 问题和解决

- 1, 在你检查了 N 遍参数都正确后, 视频一直在 loading 状态或者失败了, 那请你检查一下服务器的防火墙是否开放 1935 端口;
- 2, 视频很卡, 则可以将 video 参数 BitRate 和 Output Size, 相应调小些, 同时也可以将 Audio 的 Sample Rate 和 Bit Rate 调小;
- 3, 没有声音, 可以通过切换 audio 参数的 Format 试试;

五, 后续

通过现有的工具, 然后几行代码, 我们就可以实现一个简单的直播系统了, 但离我们的

生产环境还是有差距的，也许我们有那么几个疑问：

服务器可以支持多少并发，性能如何？

如何在移动端观看直播？

如何在自己的应用程序支持像 Adobe Media Live Encode 一样的功能？

如何在自己的应用程序直接接收 RTMP 流？