

## 34 | 如何使用Nginx搭建最简单的直播服务器？

2019-10-01 李超

从0打造音视频直播系统

[进入课程 >](#)



讲述：李超

时长 15:55 大小 14.59M



在前面三篇文章中，我们介绍了传统直播系统架构、HLS 协议、RTMP 协议相关的知识，那今天我们就来具体实操一下，根据前面所学到的知识搭建出一套最简单的音视频直播系统。

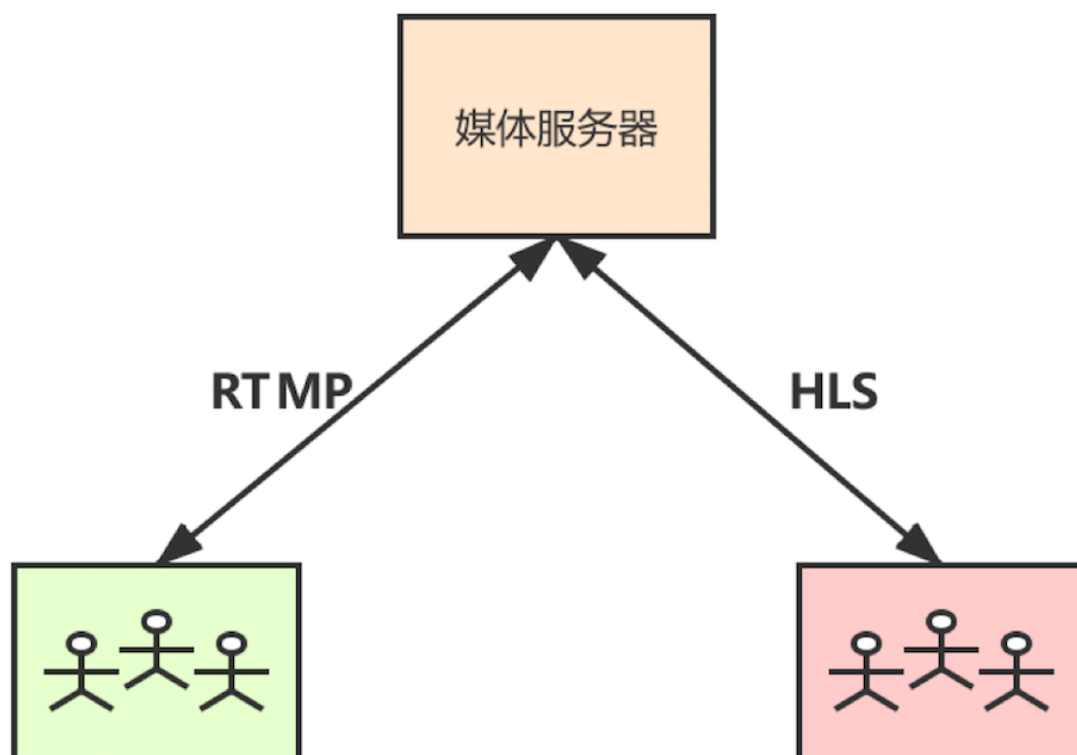
今天我们要搭建的这套直播系统相较于 [《31 | 一对多直播系统 RTMP/HLS，你该选哪个？》](#) 一文中介绍的直播系统要简单得多。该系统不包括客户端、没有 CDN 分发，只包括最基本的推流、转发及拉流功能。虽然它简单了一点，但麻雀虽小五脏俱全，通过这样一个实战操作，我们就可以将前面讲解的理论与实际结合到一起了。

当然，作为一个直播系统来说，客户端是必不可少的。但由于时间和篇幅的原因，我们只能借助一些现成的或者开源的客户端对我们的直播系统进行测试了，所以客户端的界面可能会简陋一些。也正因为如此，我才没有将它们算作咱们这个直播实验平台之中。

实际上，我们完全可以以这个直播系统实验平台为原型，逐步地将一些功能添加进去，这样很快就可以构建出一套商业可用的传统直播系统了。

## 直播系统架构

在正式开始实战之前，我们先来简要介绍一下这个直播系统的架构，如下图所示：



最简单的直播系统

这个直播架构非常简单，由两部分组成，即**媒体服务器**和**客户端**。

媒体服务器有两个功能：

推流功能，可以让客户端通过 RTMP 协议将音视频流推送到媒体服务器上；

拉流功能，可以让客户端从媒体服务器上拉取 RTMP/HLS 流。

实际上，这个架构与我们前面介绍的传统直播架构相比是有变化的，减少了信令服务器，同时将 CDN 网络变成了一台流媒体服务器。但理解了整个直播架构的原理后，我们就可以快速地将这个简单的直播架构恢复成一个正式的、可商用的直播系统。

那对于我们这个简化的直播系统来说，如何实现架构中的媒体服务器呢？

这里我们使用了目前最流行的 Nginx 来实现它。之所以选用 Nginx 主要出于以下两方面的原因：

**Nginx 的性能是很优越。**在众多的 Web 服务器中，Nginx 之所以能够脱颖而出，就是因为它性能优越。阅读过 Nginx 代码的同学应该知道，Nginx 在异步 IO 事件处理上的优化几乎做到了出神入化的地步。其架构设计也极其巧妙，不光如此，它将所有使用到的库都重新再造，对 CPU、内存的优化都做到了极致。


**Nginx 有现成的实现 RTMP 推拉流的模块。**只需要在 Nginx 安装一个 RTMP 处理模块，它就立马变成了我们想要的流媒体服务器了，操作非常简单。

接下来我们就具体实操一下，看看如何实现上面所说的流媒体服务器。

## 搭建流媒体服务端


在搭建直播平台之前，我们首先要有一台 Linux/Mac 系统作为 RTMP 流媒体服务器的宿主机，然后再按下列步骤来搭建实验环境。

从[Nginx](http://nginx.org)官方网站上下载最新发布的 Nginx 源码[nginx-1.17.4](http://nginx.org/download/nginx-1.17.4)，地址如下：  
<http://nginx.org/download/nginx-1.17.4.tar.gz>。我们可以使用 wget 命令将其下载下来，命令如下：

 复制代码

```
1 wget -c http://nginx.org/download/nginx-1.17.4.tar.gz
```

通过上面的命令就可以将 Nginx 源码下载下来了。当然，我们还需要将它进行解压缩。命令如下：

 复制代码

```
1 tar -zxvf nginx-1.17.4.tar.gz
```

要想搭建 RTMP 流媒体服务器，除了需要 Nginx 之外，还需要另外两个库，即 Nginx RTMP Module 和 OpenSSL。

## 1. RTMP Module

Nginx RTMP Module 是 Nginx 的一个插件。它的功能非常强大：

支持 RTMP/HLS/MPEG-DASH 协议的音视频直播；


支持 FLV/MP4 文件的视频点播功能；

支持以推拉流方式进行数据流的中继；

支持将音视频流录制成多个 FLV 文件；

.....

Nginx RTMP Module 模块的源码地址为：<https://github.com/arut/nginx-rtmp-module.git>。你可以使用下面的命令进行下载，存放的位置最好与 Nginx 源码目录并行。这样就可以将 Nginx RTMP Module 模块下载下来了。

 复制代码


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

## 2. OpenSSL

OpenSSL 的功能和作用我们这里就不多讲了，在第一个模块的 [《22 | 如何保证数据传输的安全（下）？》](#)一文中我们已经对它做过介绍了，记不清的同学可以再去回顾一下。


对于我们的 RTMP 流媒体服务器来说，也需要 OpenSSL 的支持。因此，我们需要将 OpenSSL 的源码一并下载下来，以便在编译 Nginx 和 Nginx RTMP Module 时可以顺利编译通过。

OpenSSL 的源码地址为：<https://www.openssl.org/source/openssl-1.1.1.tar.gz>（有可能需要 VPN）。我们仍然使用 wget 对其进行下载，命令如下：

 复制代码

```
1 wget -c https://www.openssl.org/source/openssl-1.1.1.tar.gz --no-check-certificate
```

通过上面的命令就可以将 OpenSSL 源码下载下来了。同样，我们也需要将下载后的 OpenSSL 进行解压缩，命令如下：


 复制代码

```
1 tar -zxvf openssl-1.1.1.tar.gz
```

### 3. 编译 OpenSSL & Nginx

通过上面的描述，我们就将所有需要的源码都准备好了，下面我们开始编译 Nginx。在编译 Nginx 之前，我们首先要将 OpenSSL 库编译并安装好，其编译安装过程有如下两个步骤。

第一步，生在 Makefile 文件。

 复制代码


```
1 cd openssl-1.1.1
2 ./config
```

第二步，编译并安装 OpenSSL 库。

 复制代码

```
1 make && sudo make install
```

经过上面的步骤就将 OpenSSL 库安装好了。下面我们开始编译带 RTMP Module 功能的 Nginx，只需要执行下面的命令即可，命令执行完成后，就会生成 Nginx 的 Makefile 文件。

 复制代码

```
1 cd nginx-1.17.4
2
3 ./configure --prefix=/usr/local/nginx --add-module=../nginx-rtmp-module --with-http_ssl_
```

下面我们对这里的参数做个介绍。

**prefix:** 指定将编译好的 Nginx 安装到哪个目录下。


**add-module:** 指明在生成 Nginx Makefile 的同时，也将 nginx-rtmp-module 模块的编译命令添加到 Makefile 中。

**http\_ssl\_module:** 指定 Nginx 服务器支持 SSL 功能。

**with-debug:** 输出 debug 信息，由于我们要做的是实验环境，所以输出点信息是没问题的。

**需要注意的是，在编译 Nginx 时可能还需要其他基础库，我们只需要根据执行 configure 命令时提示的信息将它们安装到宿主机上就好了。**

Nginx 的 Makefile 生成好后，我们就可以进行编译与安装了。具体命令如下：

 复制代码

```
1 make && sudo make install
```


这里需要强调的是，上面命令中的 && 符号表示前面的 Make 命令执行成功之后，才可以执行后面的命令；sudo 是指以 root 身份执行后面的 make install 命令。之所以要以 root 身份安装编译好文件是因为我们安装的目录只有 root 才有权限访问。

当上面的命令执行完成后，编译好的 Nginx 将被安装到 /usr/local/nginx 目录下。此时，我们距离成功只差最后一步了。

## 4. 配置 Nginx

上面将 Nginx 安装好后，我们还需要对它进行配置，以实现 RTMP 流媒体服务器。Nginx 的配置文件在 /usr/local/nginx/conf/ 目录下，配置文件为 nginx.conf。在

nginx.conf 文件中增加以下配置信息:

 复制代码

```
1  ...
2  events {
3      ...
4  }
5
6  #RTMP 服务
7  rtmp {
8      server{
9          # 指定服务端口
10         listen 1935;      //RTMP 协议使用的默认端口
11         chunk_size 4000; //RTMP 分块大小
12
13         # 指定 RTMP 流应用
14         application live // 推送地址
15         {
16             live on;      // 打开直播流
17             allow play all;
18         }
19
20         # 指定 HLS 流应用
21         application hls {
22             live on;      // 打开直播流
23             hls on;       // 打开 HLS
24             hls_path /tmp/hls;
25         }
26     }
27 }
28
29 http {
30     ...
31     location /hls {
32         # Serve HLS fragments
33         types {
34             application/vnd.apple.mpegurl m3u8;
35             video/mp2t ts;
36         }
37         root /tmp;
38         add_header Cache-Control no-cache;
39     }
40     ...
41 }
42 ...
```

我们来对上面的配置信息做一下简单的描述，整体上可以将这个配置文件分成三大块：




events, 用来指明 Nginx 中最大的 worker 线程的数量;

rtmp, 用来指明 RTMP 协议服务;

http, 用来指明 HTTP/HTTPS 协议服务。


其中最关键的是 RTMP 协议的配置。我们可以看到, 在 rtmp 域中定义了一个 server, 表明该 Nginx 可以提供 RTMP 服务, 而在 server 中又包括了两个 application, 一个用于 RTMP 协议直播, 另一个用于 HLS 协议的直播。

通过上面的配置, 我们就将 RTMP 流媒体服务器配置好了。接下来我们可以通过下面的命令将配置好的流媒体服务器启动起来提供服务了:

 复制代码

```
1 /usr/local/nginx/sbin/nginx
```

至此, 我们的 RTMP 流媒体服务器就算搭建好了。我们可以在 Linux 系统下执行下面的命令来查看 1935 端口是否已经打开:

 复制代码


```
1 netstat -ntpl | grep 1935
```

## 音视频共享与观看

RTMP 流媒体服务器配置好后, 下面我们来看一下该如何向 RTMP 流媒体服务器推流, 又如何从该流媒体服务器上拉 RTMP 流或 HLS 流。

### 1. 音视频共享

向 RTMP 服务器推流有好几种方式, 最简单是可以使用 FFmpeg 工具向 RTMP 流媒体服务器推流, 具体的命令如下:

 复制代码

```
1 ffmpeg -re -i xxx.mp4 -c copy -f flv rtmp://IP/live/stream
```

---



通过这个命令就可以将一个多媒体文件推送给 RTMP 流媒体服务器了。下面我们来解释一下这命令中各参数的含义：

-re, 代表按视频的帧率发送数据, 否则 FFmpeg 会按最高的速率发送数据;

-i, 表示输入文件;

-c copy, 表示不对输入文件进行重新编码;

-f flv, 表示推流时按 RTMP 协议推流;

rtmp://....., 表示推流的地址。

另外, 关于推流地址 `rtmp://IP/live/stream` 有三点需要你注意。

第一点, 推流时一定要使用 `rtmp://` 开头, 而不是 `http://`, 有很多刚入门的同学在这块总容易出错。

第二点, IP 后面的子路径 `live` 是在 `Nginx.conf` 中配置的 `application` 名字。所以, 在 `Nginx` 配置文件中你写的是什麼名字, 这里就写什麼名字。

第三点, `live` 子路径后面的 `test` 是可以任意填写的流名。你可以把它当作一个房间号来看待, 在拉流时你要指定同样的名字才可以看到该节目。

## 2. 观看

RTMP 的推流工具介绍完后, 现在我们再来看看如何观看推送的音视频流。你可以使用下面这几个工具来观看“节目”。

Flash 客户端。你可以在 IE 浏览器上使用下面的 Flash 客户端进行观看, 地址为:  
<http://bbs.chinaffmpeg.com/1.swf>。需要注意的是, 目前最新的 Chrome 浏览器已经不支持 Flash 客户端了。

VLC。在 PC 机上可以使用 VLC 播放器观看, 操作步骤是点击右侧的 `openmedia->网络->输入` `rtmp://IP/live/test`。

FFplay。它是由 FFmpeg 实现的播放器, 可以说是目前世界上最牛的播放器了。使用 FFplay 的具体命令如下: `ffplay rtmp://host/live/test`。

当然拉流也是一样的，你也可以利用 librtmp 库自己实现一个观看端。

## 小结

在本文我们全面介绍了如何通过 Nginx 搭建一套最简单的 RTMP/HLS 流媒体服务器。这套流媒体服务器功能还是很齐全的，基本上涵盖了传统流媒体服务器方方面面的知识。

我们还介绍了如何搭建流媒体服务器的原型，以及多种推流、拉流的测试工具。这些测试工具都极具价值，并且大多数都是开源项目，如果你能力比较强的话，可以直接阅读它们的代码。理解它们的代码实现，对于你实现自己的直播推 / 拉流工具有非常大的益处。

另一方面，虽然可以通过 Nginx 实现传统直播系统的原型，但如果你想将它转化为商业用途的话，那还有非常远的距离。因为商用的直播系统需要对流媒体服务器做各种性能优化，对它的宿主机的操作系统做优化，对它的传输网络做优化，等等。除此之外，一个商业的直播系统是由多个子系统构成的，如日志系统、计费系统、监控告警、质量检测系统、云存储系统、工单系统等。

通过本文你可以知道直播原理与实际应用之间的距离并不遥远。直播的原理搞清楚后，你只需要几个小时就可以将传统直播流媒体服务器的实验环境搭建出来。但应用与商用之间却还有一条鸿沟，这条鸿沟是由无数的细节构成的，而这些细节除了音视频直播的知识外，还有算法、操作系统、网络协议等方面的知识。所以说，知识是相通的，道路是曲折的。

## 思考时间

那除了 Nginx 之外，你还知道其他开源的 RTMP 流媒体服务器吗？

欢迎在留言区与我分享你的想法，也欢迎你在留言区记录你的思考过程。感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给更多的朋友。

---

# 从 0 打造音视频直播系统

手把手教你打造实时互动音视频直播系统

李超

新东方音视频直播技术专家  
前沪江音视频架构师



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 33 | FLV：适合录制的多媒体格式

下一篇 35 | 如何构建云端一对多直播系统？

## 精选留言 (1)

 写留言



tech2ipo

2019-10-01

开源RTMP流媒体服务器：SRS，github地址：<https://github.com/ossrs/srs>

作者回复：SRS 服务还是不错的！

 1

 1