

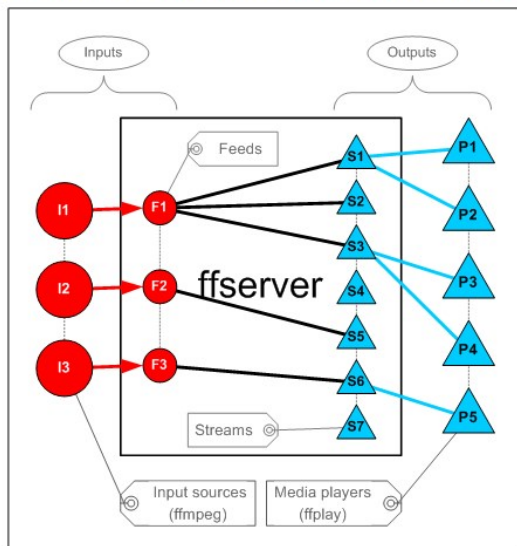
原创 王log 于 2019-08-07 20:41:05 发布 14801 收藏 18
分类专栏: ffmpeg 流媒体

ffmpeg 同时被 2 个专栏收录 0 订阅 13 篇文章 订阅专栏

1、ffserver

ffserver在ffmpeg3.4版本以后的版本被移除，能用做http/rtsp 流媒体 服务器。其架构如下图所示。

Let's take a closer look of ffserver, to better describe its possibilities. Consider the following image:



There are several elements shown on the image. Let's name them all first:

- Input sources (I)
- Feeds (F)
- Streams (S)
- Media players (P)

上图中I、F、S、P概念：

- 《1》、输入源
能将音/视频传给ffserver的外部应用，通常是ffmpeg应用。输入源会和ffserver连接然后将自己绑定到一个或多个feed上。（同一时刻一个feed只能绑定一个输入源）。一个输入源能绑定到多个feed的前提条件是改输入源能为其绑定的每个feed输出不同的流。
- 《2》feeds
feed元素是ffserver的一部分，用于将一个输入源与一个或多个stream关联。当你想将一个输入源输出为多种输出格式时推荐将一个feed与多个stream关联。（例如，同时输出full HD分辨率和为手机输出小分辨率）
- 《3》streams
stream是ffserver的一部分，是外部想要获取stream的客户端与ffserver的连接点。每一个stream可以处理多个客户端连接。
- 《4》、Media players
外部应用，一般指的是客户端。

目录

- 1、ffserver
- 2、ffserver.conf配置
- 3、ffmpeg推流
- 4、验证
- 5、参考

分类专栏

janus	7篇
语音识别	1篇
onvif	1篇
编解码	1篇
(GP)GPU	1篇
图像处理	10篇
深度学习	3篇
Makefile&shell	6篇
linux编程	7篇
流媒体	29篇
ffmpeg	13篇
C/C++	16篇
GB28181	5篇

<https://blog.csdn.net/FPGATOM>

2、ffserver.conf配置

```
1 HTTPPort 8090
2 RTSPPort 5454
3 # Address on which the server is bound. Only useful if you have
4 # several network interfaces.
5 HTTPBindAddress 192.168.1.252
6
7
8 MaxHTTPConnections 2000
9
10
11 # Access log file (uses standard Apache log file format)
12 # '-' is the standard output.
13 MaxClients 1000
14
15
16 # This the maximum amount of kbit/sec that you are prepared to
17 # consume when streaming to clients.
18 MaxBandwidth 30000
19
20
21 # Access log file (uses standard Apache log file format)
22 # '-' is the standard output.
23 CustomLog -
24
25
26 NoDaemon
27 #NoDefaults
28
29
30 #####
31 # Definition of the live feeds. Each live feed contains one video
32 # and/or audio sequence coming from an ffmpeg encoder or another
33 # ffserver. This sequence may be encoded simultaneously with several
34 # codecs at several resolutions.
35
36
37 <Feed feed1.ffm>
38 File /tmp/feed1.ffm
39 FileMaxSize 5M
40
41
42 # Only allow connections from localhost to the feed.
43 ACL allow 127.0.0.1
44
45
46 </Feed>
47
48
49 #####
50 # Now you can define each stream which will be generated from the
51 # original audio and video stream. Each format has a filename (here
52 # 'test1.mpg'). FFServer will send this stream when answering a
53 # request containing this filename.
54
55
56 #<Stream test1>
57 #Format rtp
58 #VideoCodec libx264
59 #AudioCodec aac
```



```

60 # coming from live feed 'feed1'
61 #Feed feed1.ffm
62
63
64 #File btv2.ts
65
66
67 # Format of the stream : you can choose among:
68 # mpeg      : MPEG-1 multiplexed video and audio
69 # mpegvideo  : only MPEG-1 video
70 # mp2       : MPEG-2 audio (use AudioCodec to select layer 2 and 3 codec)
71 # ogg       : Ogg format (Vorbis audio codec)
72 # rm        : RealNetworks-compatible stream. Multiplexed audio and video.
73 # ra        : RealNetworks-compatible stream. Audio only.
74 # mpjpeg    : Multipart JPEG (works with Netscape without any plugin)
75 # jpeg      : Generate a single JPEG image.
76 # mjpeg     : Generate a M-JPEG stream.
77 # asf       : ASF compatible streaming (Windows Media Player format).
78 # swf       : Macromedia Flash compatible stream
79 # avi       : AVI format (MPEG-4 video, MPEG audio sound)
80 #Format mpegts
81
82
83 #BitExact
84 #DctFastint
85 #IdctSimple
86 #videoFrameRate 25
87 #VideoSize 1920x1080
88
89
90 # Ratecontrol buffer size
91 #VideoBufferSize 4096
92
93
94 #VideoBitRate 90000
95 #VideoGopSize 25
96 #NoAudio
97 #AudioBitRate 32
98 # Number of audio channels: 1 = mono, 2 = stereo
99 #AudioChannels 1
100
101
102 # Sampling frequency for audio. When using low bitrates, you should
103 # lower this frequency to 22050 or 11025. The supported frequencies
104 # depend on the selected audio codec.
105 #AudioSampleRate 44100
106
107
108 #</Stream>
109
110
111 <Stream test1-rtsp>
112 #File "./in.264"
113 Feed feed1.ffm
114 Format rtp
115 VideoCodec libx264
116 VideoGopSize 25
117 VideoFrameRate 25
118 VideoBufferSize 80000
119 VideoSize 1920x1080
120 #VideoBitRate 90000

```

```

221 #AudioSampleRate 44100
222 PreRoll 15
223 NoDefaults
224 NoAudio
225 </Stream>
226
227 #一个输出声音，有声音时用ffmpeg是播放不出来，原因待查找
228 #只要设置为NoAudio就可以一个输入源输出两种分辨率
229 <Stream test2-rtsp>
230 #File "./in.264"
231 Feed feed1.ffmpeg
232 Format rtp
233 VideoCodec libx264
234 VideoGopSize 25
235 VideoFrameRate 25
236 VideoBufferSize 80000
237 VideoSize 352x288
238 #VideoBitRate 90000
239 AudioCodec aac
240 AudioSampleRate 44100
241 # Number of audio channels: 1 = mono, 2 = stereo
242 #AudioChannels 2
243 AudioBitRate 32
244 PreRoll 15
245 NoDefaults
246 #NoAudio
247 </Stream>
248
249 <Stream stat.html>
250 Format status
251 ACL allow localhost
252
253
254 # You can allow ranges of addresses (or single addresses)
255 #ACL ALLOW <first address> <last address>
256 ACL allow 192.168.0.0 192.168.255.255
257
258
259 </Stream>

```

运行ffserver，制定配置文件

```
1 | ffserver -f ffserver.conf
```

3、ffmpeg^Q推流

ffmpeg推流到ffserver遵循以下格式:

```
1 | ffmpeg <inputs> <feed URL>
```

其中< feed URL >的格式如下:

```
1 | http://<ffserver_ip_address_or_host_name>:<ffserver_port>/<feed_name>
```

该URL格式中的参数意义如下:

- <ffserver_ip_address_or_host_name> - using the "BindAddress" directive
- < ffserver_port > - using the "Port" directive

- < feed_name > - using the "" block

若需要接受外部IP的客户端连接，则将host_name设置为本机IP，若不需要则设置为localhost即可。原文如下。

ffmpeg and ALSA.

- The last, but not the least important, part "<http://localhost:8090/feed1.ffm>" represents the feed URL, which tells ffmpeg to connect to fserver and send it the audio + video streams for broadcast. In this example we used the hostname "localhost" which means that everything is running on our computer, but if you need to feed the live online fserver, you'll need to change the "localhost" to the real host name or IP address of your fserver computer. Also make sure that your feed name ends with ".ffm" and if it's not the case, then add "-f ffm" before your feed URL, to manually specify the output format (because ffmpeg won't be able to figure it out automatically any more), like this "-f ffm <http://localhost:8090/blah.blah>".

soon as you type the command above, you should see ffmpeg displaying some statistics about your input streams and counting output frames, which is a pretty good sign that everything works (so far).

• this example, you would need at least the following things defined in your config file (three dots "..." represent the other data that is irrelevant for this topic):

Port 8090

例子:

```
1 只在本地主机使用:
2  ffmpeg -re -i "/test.mp4" -bf 0 -g 25 -r 25 -s 1920x1080 -b:v 4M -maxrate 8M -bufsize 1800k -vcodec libx264 -a
3
4  其他IP主机也可使用:
5  ffmpeg -re -i "/test.mp4" -bf 0 -g 25 -r 25 -s 1920x1080 -b:v 4M -maxrate 8M -bufsize 1800k -vcodec libx264 -a
```

4、验证

《1》、ffplay接收流

```
1 ffplay -i rtsp://localhost:5456/test1-rtsp
2 ffplay -i rtsp://localhost:5456/test2-rtsp
```

《2》、浏览器查看状态如下

ffserver Status x +

localhost:8092/stat.html

ffserver Status

Available Streams

Path	Served Conns	bytes	Format	Bit rate kbit/s	Video kbit/s	Audio kbit/s	Codec	Feed
test1-rtsp	0	0	rtp	0	0	0	libx264	feed1.ffm
test2-rtsp	0	0	rtp	32	0	0	libx264	feed1.ffm
stat.html	1	0	-	-	-	-	-	-

Feed feed1.ffm

Stream	type	kbit/s	codec	Parameters
0	video	4000	libx264	1920x1080, q=1-1, fps=25
1	audio	32	aac	2 channel(s), 44100 Hz
2	video	4000	libx264	352x288, q=1-1, fps=25

Connection Status

Number of connections: 2 / 1000
Bandwidth in use: 0k / 30000k

#	File	IP	URL	Proto	State	Target bit/s	Actual bit/s	Bytes transferred
1	stat.html	127.0.0.1	/stat.html	HTTP/1.1	HTTP_WAIT_REQUEST	0	0	0
2	feed1.ffm(input)	127.0.0.1	/feed1.ffm	HTTP/1.1	RECEIVE_DATA	12000k	1634k	11173k

Generated at Wed Aug 7 10:49:53 2019

<https://blog.csdn.net/FPGATOM>

5、参考



王log

码龄7年

暂无认证

60

2万+

5万+

18万+



原创

周排名

总排名

访问

等级

2193

712

71

59

529

积分

粉丝

获赞

评论

收藏



私信

关注

搜博文文章



热门文章

pytorch数据加载、模型保存及加载 34608

ffmpeg+ffserver搭建rtsp服务器 14797

积分图原理及实现 6593

C语言不定参数的两种实现 6146

GB28181基本注册流程整理 5483

最新评论

新版本glib使用epoll代替poll
weixin_41582868: nready = ev_poll (fds, f
ds_size, /*timeout=*/0); debug执行时, ...

新版本glib使用epoll代替poll
王log: 后来我也发现这个demo确实有问题
, 然后仿照glib的select方式实现的g_poll...

新版本glib使用epoll代替poll

《1》、 <https://trac.ffmpeg.org/wiki/ffserver>

《2》、 <https://stackoverflow.com/questions/37403282/is-there-anyone-who-can-success-real-time-streaming-with-ffserver>

《3》、 [ffmpeg码率参数设置](#)

《4》、 [ffserve stat设置使用](#)

《5》、 [ffmpeg preset参数](#)

RTSP流媒体服务器的搭建与测试

08-16

本文主要介绍了live555搭建RTSP流媒体服务器，并采用VLC软件进行相应测试

ffmpeg-rtsp服务器

04-25

支持跨平台编译rtsp服务器自己需要跨平台编译dlib, live555, ffmpeg, ffmpeg-live555编解码发布直播, 不依赖第三放库 (nginx等) 做直播. 支持抓取本地视...

参与评论



请发表有价值的评论， 博客评论欢迎灌水，良好的社区氛围需大家一起维护。



抢沙发

评论

开发EasyRTSPServer组件时如何使用ffserver搭建一套RTS...

3-29

在我们开发RTSP视频流的分发阶段,我们也考虑过如何使用ffserver分发RTSP视频流,如何搭建RTSP服务器,除了EasyRTSPServer之外,我们探寻出了另外...

...和音频服务器_bingqingsuimeng的专栏_ffserver rtsp

3-15

1. ffserver.conf RTSPPort 8554 BindAddress 0.0.0.0 RTSPBindAddress 0.0.0.0 MaxHTTPConnections 2000 MaxClients 1000 MaxBandwidth 30000 Cu...

基于ffmpeg-live555的rtsp服务器，支持h264/h265编码，支持mp4等格式

07-08

基于ffmpeg和live555开发的rtsp服务器，支持h264/h265编码，支持mp4等格式。测试文件放在执行文件当前目录，URL格式为：rtsp://ip:port/filename

[Rtsp]RTSP对实时摄像头视频流进行转换 (FFmpeg+FFserver)

mengzhengjie的专栏 4021

http://www.cuplayer.com/player/PlayerCode/RTSP/2014/0706/1414.html 输入数据为来自摄像头的RTSP视频流，然后利用FFmpeg将其输出到FFserver...

流媒体协议——搭建RTSP服务器的多种方法——ffserver, vl...

4-27

在USB摄像头采集终端上构建rtsp流媒体服务器,将USB摄像头数据转化为rtsp,可以多个客户端访问 三种实现方式:ffserver(ffmpeg)、vlc、live555 1 ffserver...

FFserver流媒体服务器 FFMPEG推流 RTSP拉流_chouhun176...

3-21

[root@localhost ffmpeg]# cat ffserver.conf HttpPort 8090 RtspPort 554 HttpBindAddress 0.0.0.0 MaxClients 1000 MaxBandwidth 10000 NoDaemon <F...

ffserver详细介绍

码农崛起 4144

ffmpeg与ffserver的协同工作 ffmpeg和ffserver配合使用可以实现实时的流媒体服务，可以实时传输来自摄像头的的数据，客户端可以采用HTTP、RTSP、R...

【音视频】RTSP服务器搭建及FFMPEG命令推流 (3-1)

hcmonk的博客 278

最近开始使用ffmpeg做网络推流的事情，例如rtmp、rtsp推流等。这里参考以下文章先搭建rtsp服务器实现ffmpeg命令推流，之后会写代码实现推流功能。...

ffmpeg rtsp 推流_手把手搭建RTSP流媒体服务器

weixin_39968995的博客 4736

0. 引言本文主要讲解如何搭建RTSP流媒体服务器的过程，使用开源项目ZLMediaKit。通过这个开源项目，推RTSP流到服务器，然后拉流端可以拉取RTS...

移植ffmpeg至ARM平台，利用ffmpeg+ffplay搭建rtsp流媒体服务器

老谭_酸菜的博客 426

一、前言 利用FFMPEG+FFPlay搭建一个RTSP流媒体服务器，以便终端设备进行拉流。系统框架如下图所示。LCD通过UDP发送H264的裸流给车载播控...

【ffmpeg】ffmpeg推送流到rtsp服务器

心生的小鱼专栏 4203

1引 言 在项目开发过程中经常遇到接入RTSP视频流的需求。由于外网无法访问项目现场的内网服务器，无法进行测试。想着在网上找一个RTSP流进行测试...

FFMPEG和FFSERVER搭建流媒体服务器实例 (windows下测试通过)

12-23

FFMPEG和FFSERVER搭建流媒体服务器实例 (windows下测试通过)， 可以根据自己的需求修改里面的ffserver.conf文件。详情可以参考http://blog.csdn...

ffserver配置流媒体服务器

shaoyundong的专栏 2586

如果要实现一个流媒体服务器，不管是实时监控还是点播文件，ffmpeg项目无疑是一个比较理想的平台 今天下载了ffmpeg 1.1，在CentOS 6.4上安装使用...

用ffserver实现rtsp服务器的实验笔记

d9394952的博客 589

参考: <https://blog.csdn.net/hoyjam1/article/details/51281679> 建议配置文件: /etc/config/ffserver.conf Port 10535 RTSPPort 5454 BindAddress 0.0.0.0...

关于利用ffserver搭建RTSP服务的代码 Streaming with FFServer

暂时为空 9172

In this article I want to explain how to install ffserver for streaming with libx264 and mp3lame on ubuntu 12.04 with clean process. 1) Download FFMpeg...

weixin_41582868: 你好，使用g_poll和ev_poll得的输出不一样，这样正确吗？ nrea...
webrtc ice(rfc5245)记录
大家一起学编程 (python) : 六六六
janus服务器《一》--总概述
Lansonli: 博主文章写得真好，感悟颇深，感谢分享！有时间也来关注一下我的博文，...

您愿意向朋友推荐“博客详情页”吗？

    
强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

新版本glib使用epoll代替poll

lsquic使用《二》--echo服务

lsquic使用《一》--概述

2021年 13篇

2020年 15篇

2019年 24篇

2018年 22篇

ffmpeg+ffserver搭建流媒体服务器 热门推荐 chuciao4350的博客 1万+
刚开始学习组播、直播方面的东西，简单总结一下利用ffmpeg+ffserver做电视信号流媒体实时直播。 总体架构就是dvb-IP网关将电视cable信号解扰解复用...

Rtsp协议，FFmpeg推流，rtsp服务器，vlc拉流播放下载安装使用使用详细流程以及EasyDarwin服务的... 最新发布 A_awen的博客 3105
Rtsp协议，FFmpeg推流，rtsp服务器，vlc拉流播放下载安装使用使用详细流程以及EasyDarwin服务的方案

ffmpeg 之 RTSP — 陌上烟雨遥的博客 2306
一 前沿 打算花两篇文章来写RTSP，第一篇介绍RTSP 协议，第二篇介绍RTSP 是如何在FFMPEG 中实现的。RTSP 是早期的流媒体协议，协议由RealN...

FFMPEG音视频开发: 发布RTSP流(采用EasyDarwin作为流媒体服务器) DS小龙哥的专栏 1647
环境介绍 当前使用的操作系统: win10 64位 一、部署EasyDarwin服务器 据官网介绍，EasyDarwin拥有完整的源代码，可以帮助开发者更快更简单实现流...

ffserver 作为 rtsp 服务器 yangzhao0001的博客 4418
ffmpeg和ffserver配合使用可以实现实时的流媒体服务，这里实时的数据来自摄像头，如果考虑到应用程序和网络状况，这里客户端看到的画面还是会落...

基于EasyDarwin、ffmpeg实现rtsp推流，及VLC播放 MacKendy的博客 1820
文本主要是对工具等的一些实现，并不包含代码。主要是在本地，借助EasyDarwin启动rtsp服务器，用ffmpeg进行rtsp数据源的推流，最后用vlc在本地播...

“相关推荐”对你有帮助么？

 非常没帮助  没帮助  一般  有帮助  非常有帮助

©2022 CSDN 皮肤主题：技术黑板 设计师：CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

 王log 关注

0

0

0

18

¥

¥

专栏目录