16-08 nginx-hls-多码率测试环境搭建

1 nginx-rtmp开源项目

2 环境搭建

  安装依赖库

  安装nginx-http-flv-module

  安装ffmpeg

  安装nginx

3 配置文件

4 测试

5 Nginx的RTMP直播模块开发参数说明和配置范例

  直播RTMP模块中一些主要的配置选项

  在直播时的一些配置项

  HLS协议进行m3u8实时直播

  录制直播视频以便回放重播

  使用HTTP动态自适应不同带宽的视频

  用于视频点播的配置

  拉流转播到其他平台

  直播状态的消息和状态

  对直播的访问权限控制

  其他RTMP的配置选项

  下面是一个NGINX-RTMP直播配置的范例

腾讯课堂 《FFmpeg/WebRTC/RTMP音视频流媒体高级开发》 https://ke.qq.com/course/468797?tuin=137bb271

1 nginx-rtmp开源项目

项目地址：https://github.com/winshining/nginx-http-flv-module

- nginx-http-flv-module的其他功能与nginx-rtmp-module的对比：

功能	nginx-http-flv-module	nginx-rtmp-module	备注
HTTP-FLV (播放)	√	x	支持HTTPS-FLV和chunked回复
GOP缓存	√	x	
虚拟主机	√	x	
省略`listen`配置	√	见备注	配置中必须有一个`listen`
纯音频支持	√	见备注	`wait_video`或`wait_key`开启后无法工作
`reuseport`支持	√	x	
定时打印访问记录	√	x	
JSON风格的stat	√	x	
stat中包含录制详情	√	x	

2 环境搭建

Linux主机版本：ubuntu 16.04

安装依赖库

下载相关的依赖库

```
sudo apt-get update
#安装依赖：gcc、g++依赖库
sudo apt-get install build-essential libtool
#安装 pcre依赖库（http://www.pcre.org/）
sudo apt-get install libpcre3 libpcre3-dev
#安装 zlib依赖库（http://www.zlib.net）
sudo apt-get install zlib1g-dev
#安装ssl依赖库
sudo apt-get install openssl
```

安装nginx-http-flv-module

git clone https://github.com/winshining/nginx-http-flv-module.git

该源码下载后的路径是 /home/ubuntu/0voice/media/nginx-rtmp/nginx-http-flv-module，在编译nginx的时候需要用到

安装ffmpeg

参考 ubuntu ffmpeg环境搭建

安装nginx

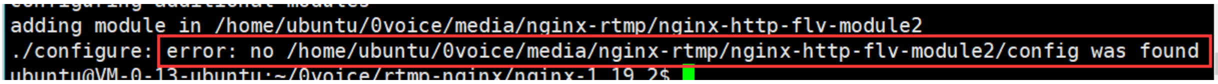
--prefix=/usr/local/rtmp-nginx 指定nginx安装目录
--add-module=/home/ubuntu/0voice/media/nginx-rtmp/**nginx-http-flv-module**

```
注意安装的时候要带ssl
#下载nginx 1.19.2版本
wget http://nginx.org/download/nginx-1.19.2.tar.gz
tar xvzf nginx-1.19.2.tar.gz
cd nginx-1.19.2/
# 配置，一定要支持https
./configure --prefix=/usr/local/rtmp-nginx --with-http_ssl_module --add-module=/home/ubuntu/0voice/media/nginx-rtmp/nginx-http-flv-module
# 编译
make
#安装
sudo make install
最终安装到目录：/usr/local/rtmp-nginx/nginx 主要是避免和原有的nginx有冲突
```

最终安装到目录：/usr/local/rtmp-nginx/nginx 主要是避免和原有的nginx有冲突

启动：sudo /usr/local/rtmp-nginx/nginx/sbin/nginx
停止：sudo /usr/local/nginx/rtmp-nginx/sbin/nginx -s stop
重新加载配置文件：sudo /usr/local/rtmp-nginx/nginx/sbin/nginx -s reload

如果nginx-http-flv-module路径不对会报错，比如下面所示。



```
configuring additional modules
adding module in /home/ubuntu/0voice/media/nginx-rtmp/nginx-http-flv-module2
./configure: error: no /home/ubuntu/0voice/media/nginx-rtmp/nginx-http-flv-module2/config was found
ubuntu@VM-0-13-ubuntu:~/0voice/rtmp-nginx/nginx-1.19.2$
```

3 配置文件

```
daemon off;
# 如果开启off对应的ts文件不并删除
# master_process off;
user root;

error_log /tmp/error.log debug;
events{
    worker_connections 1024;
}


rtmp{
    server {
        listen 1935;
        chunk_size 4000;

        #live
        application live {
            live on;

                exec /usr/bin/ffmpeg -i rtmp://localhost/live/$name
                    -c:a copy  -c:v libx264 -b:v 300K -g 30 -f flv rtmp://localhost/hls/$name_hi
                    -c:a copy  -c:v libx264 -b:v 200K -g 30 -s 462x254 -f flv rtmp://localhost/hls/$name_mid
                    -c:a copy  -c:v libx264 -b:v 100K -g 30 -s 230x128 -f flv rtmp://localhost/hls/$name_low;
        }

        application hls {
            live on;
            hls on;
            hls_path /tmp/hls;
            hls_nested on;
            hls_fragment 2s;
            hls_playlist_length 6s;

            hls_variant _hi  BANDWIDTH=350000;
        hls_variant _mid BANDWIDTH=250000;
        hls_variant _low BANDWIDTH=150000;

        }
    }
}

#HTTP
http{
    server {
        listen 8081;

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

        #hls
        location /hls {
            types {
                application/vnd.apple.mpegusr m3u8;
                video/mp2t ts;
            }
            #root /tmp;
        alias /tmp/hls;
            add_header Cache-Control no-cache;
        }
    }
}
```

4 测试

模拟丢包

```
1. 延迟设置
//延迟 300ms ± 100ms
sudo tc qdisc add dev eth0 root netem delay 300ms 100ms
sudo tc qdisc del dev eth0 root netem delay 300ms 100ms

2. 丢包
//丢包率 5%
sudo tc qdisc add dev eth0 root netem loss 30%
sudo tc qdisc del dev eth0 root netem loss 5%
```

推流：
ffmpeg -re -i time.flv -vcodec copy -acodec copy -f flv -y
rtmp://111.229.231.225/live/livestream
RTMP流地址为：rtmp://111.229.231.225/live/livestream
HLS流地址为： http://111.229.231.225:8081/hls/livestream.m3u8
http://111.229.231.225:8081/hls/livestream_hi/index.m3u8

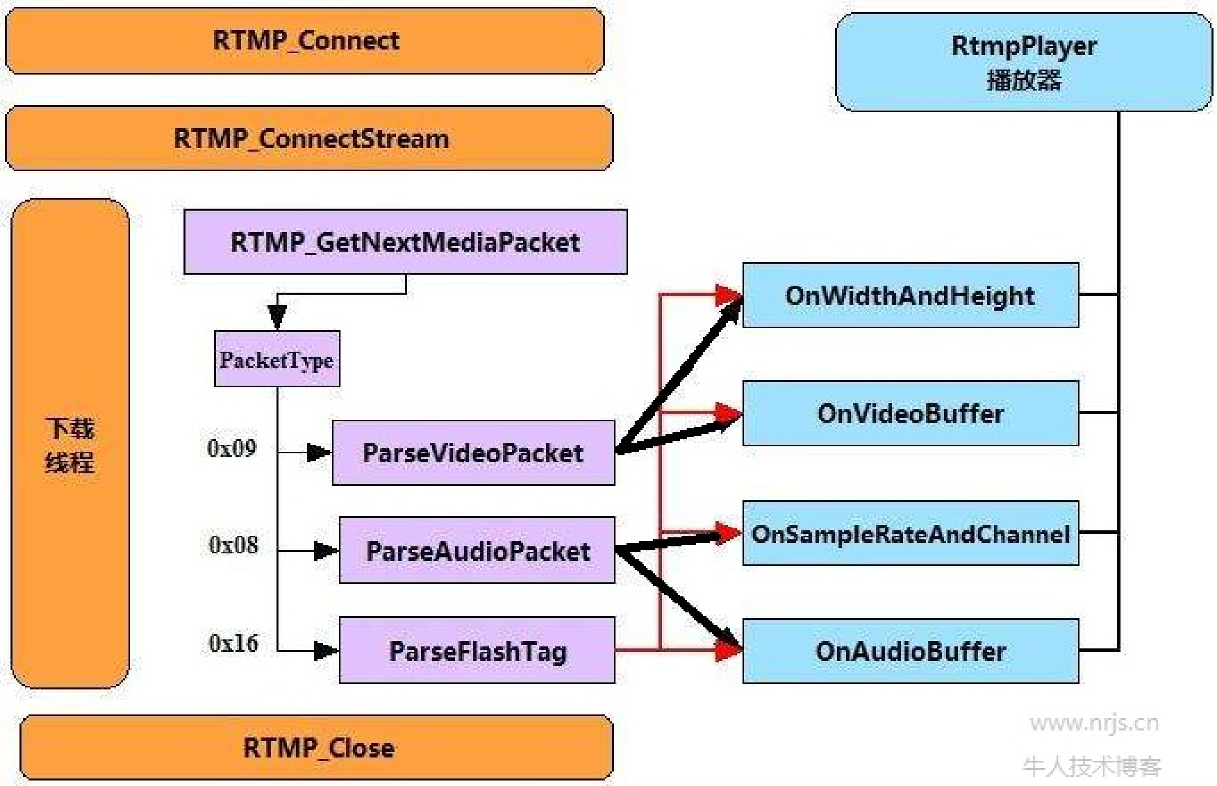
使用 ffplay和vlc进行播放测试

5 Nginx的RTMP直播模块开发参数说明和配置范例

参考：

【官方文档】Nginx模块Nginx-Rtmp-Module学习笔记（二）HLS 指令详解
https://www.cnblogs.com/tinywan/p/5981197.html

现在直播也是非常流行的一种应用场景，各种应用和H5页面上也经常用到直播。目前直播技术以RTMP方式是非常成熟的，虽然有10秒左右的延迟。但是这完全不影响直播的效果和稳定性。这次我们就以Nginx下的RTMP直播模块来讲一下直播的应用级别开发。安装这种类型的RTMP模块，需要重新编译NGINX服务。并且下载RTMP安装包。这里的安装和编译在另一篇牛人技术文章中说过了，在此就不多介绍了。下面我们来说说RTMP模块的一些命令参数和配置方法。



直播RTMP模块中一些主要的配置选项

rtmp	
server	NGINX中的服务块
listen	监听端口
application	用于在NGINX配置文件中创建同一个程序块
timeout	连接过期时间
ping	测试数据包
ping_timeout	测试数据包超时时间
max_streams	最大流数量
ack_window	
chunk_size	

max_queue

max_message

buflen

out_queue

out_cork

在直播时的一些配置项

live

meta

interleave

wait_key

wait_video

publish_notify

drop_idle_publisher

sync

play_restart

idle_streams

HLS协议进行m3u8实时直播

hls

hls_path

hls_fragment

hls_playlist_length

hls_sync

hls_continuous

hls_nested

hls_base_url

hls_cleanup

hls_fragment_naming

hls_fragment_naming_granularity

hls_fragment_slicing

hls_variant

hls_type

hls_keys

hls_key_path

hls_key_url

hls_fragments_per_key

录制直播视频以便回放重播

record

record_path

record_suffix

record_unique

record_append

record_lock

record_max_size

record_max_frames

record_interval

recorder

record_notify

使用HTTP动态自适应不同带宽的视频

dash

dash_path

dash_fragment

dash_playlist_length

dash_nested

dash_cleanup

MPEG-DASH

用于视频点播的配置

play

play_temp_path

play_local_path

拉流转播到其他平台

pull

push

push_reconnect

session_relay

直播状态的消息和状态

on_connect

on_play

on_publish

on_done

on_play_done

on_publish_done

on_record_done

on_update

notify_update_timeout

notify_update_strict

notify_relay_redirect

notify_method

对直播的访问权限控制

allow

deny

Exec 一簇函数（进程往往要调用一种exec函数以执行另一个程序）

exec_push

exec_pull

exec

exec_options

exec_static

exec_kill_signal

respawn

respawn_timeout

exec_publish

exec_play

exec_play_done

exec_publish_done

exec_record_done

其他RTMP的配置选项

access_log 访问日志

log_format

max_connections 连接数Limits 限制

rtmp_stat 数据统计

rtmp_stat_stylesheet

Multi-worker live streaming 多线程直播流

rtmp_auto_push

rtmp_auto_push_reconnect

rtmp_socket_dir

rtmp_control 直播Control 控制模块

下面是一个NGINX-RTMP直播配置的范例

```
worker_processes  1;

events {
    worker_connections  1024;
}

#这里开始是牛人技术测试直播的配置信息
rtmp {
        server {
                listen 1935;
                chunk_size 4096;
                application hls {
                      live on;
                      hls on;
                      hls_path /byDATA/NginxRtmpNRJS/webroot/tt/hls;
                      hls_fragment 5s;
                }

                #用来给115频道
                application ANuid115 {
                      live on;
                      hls on;
                      hls_path /byDATA/NginxRtmp/webroot/ANuid115;
                      hls_fragment 5s;
                }



                #用来测试
                application ANuid901 {
                      live on;
                      hls on;                          #实时回访
                      wait_key on;                     #保护TS切片
                      hls_nested on;                   #每个流都自动创建一个文件夹
                      hls_fragment 5s;                 #每个ts文件为5s的样子
                      hls_fragment_naming system;#使用系统时间戳命名ts文件
                      hls_playlist_length 10800s;      #保存m3u8列表长度时间，默认是30
秒，可考虑三小时10800秒
                      hls_cleanup on;            #是否删除列表中已经没有的媒体块TS文件，
默认是开启
                      hls_continuous on;         #连续模式
                      hls_path /byDATA/NginxRtmp/webroot/live/record/ANuid901; #媒体
块ts的位置
                }

                application ANuid902 {
                      live on;
```

```
                     hls on;
                     hls_path /byDATA/NginxRtmp/webroot/tt/ANuid902;
                     hls_fragment 5s;
              }

              application ANuid903 {
                     live on;
                     hls on;
                     hls_path /byDATA/NginxRtmp/webroot/tt/ANuid903;
                     hls_fragment 5s;
              }
       }
}


http {
    include       mime.types;
    default_type  application/octet-stream;
    #access_log  logs/access.log  main;
    sendfile        on;
    #tcp_nopush     on;
    #keepalive_timeout  0;
    keepalive_timeout  65;

#byAdd
#include /byDATA/NginxRtmp/conf/*.conf; 此处可以添加自定义配置文件目录
    #gzip  on;
    server {
        listen       80;
        server_name  localhost;
        #charset koi8-r;
        #access_log  logs/host.access.log  main;
        location / {
                     #byAdd
                     #root   /byDATA/NginxRtmp/webroot/tt;
            root   html;
            index  index.html index.htm;
        }
        #error_page  404              /404.html;
        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}
        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}
        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
    }
       #byAdd
       server {
        listen       80;
        server_name  flow.320023.com;
```

```
        location /ANuid901 {
            types {
                application/vnd.apple.mpegurl m3u8;
                video/mp2t ts;
            }
            root /byDATA/NginxRtmp/webroot/live/record;
            add_header Cache-Control no-cache;
    }
        #推流状态查看 http://flow.320023.com/stat
        location /stat {
            rtmp_stat all;
            rtmp_stat_stylesheet stat.xsl;
        }
        location /stat.xsl {
            root /byDATA/NginxRtmp/Program/nginx-rtmp-module-master/;
        }

        #rewrite /ANuid901/(.*).m3u8$ /ANuid901/$1/index.m3u8 last;        #重写用于兼容阿里云m3u8命名格式
        #rewrite /ANuid901/(.*).ts$ /ANuid901/StreamName/$1.ts last;        #重写让上面m3u8能找到ts文件
  }
  # another virtual host using mix of IP-, name-, and port-based configuration
  #
  #server {
  #    listen       8000;
  #    listen       somename:8080;
  #    server_name  somename  alias  another.alias;
  #    location / {
  #        root   html;
  #        index  index.html index.htm;
  #    }
  #}
  # HTTPS server
  #
```

```
    #server {
    #    listen       443 ssl;
    #    server_name  localhost;
    #    ssl_certificate      cert.pem;
    #    ssl_certificate_key  cert.key;
    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;
    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;
    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
}
```