

昵称: 步孤天
园龄: 8年3个月
粉丝: 19
关注: 35
[+加关注](#)

昵称: 步孤天
园龄: 8年3个月
粉丝: 19
关注: 35
+加关注

<		2023年8月							>	
日	一	二	三	四	五	六				
30	31	1	2	3	4	5				
6	7	8	9	10	11	12				
13	14	15	16	17	18	19				
20	21	22	23	24	25	26				
27	28	29	30	31	1	2				
3	4	5	6	7	8	9				

最新随笔

1. 六七个甲子纳音表中的五行是怎么算出来的
2. 打开看30GB的磁盘满了而du -sh看磁盘空间却只有6GB
3. centos7 + mariadb安装在线评估系统
4. 如何去掉Linux vim安装中的“M”
5. 如何从超过10Gsql语文本中分离出需要的部分
6. golang如何打印变量类型, golang list如何把元素转换为可用类型
7. 数据库文件导入到数据库“MySQL server has gone away”
8. 如何在Linux上通过rsync命令把包中follow的2个磁盘保存成文件
9. Windows用VNC远程连接到麒麟OS(银河麒麟)
10. 如何把数据库字段导出成报表

我的标签

互联网的内容的审核标准(1)

积分与排名

积分 - 447617
排名 - 1502

随笔分类

- android(1)
- bash shell(6)
- ceph(24)
- golang(2)
- k8s(5)
- Linux C+++(33)
- Linux中常用的工具(10)
- macbook(3)
- mongodb(3)
- python(10)
- sql(8)
- 安装及使用(18)
- 并发相关(12)
- 测试(6)
- 分布式(19)
- 工具(14)
- 机器学习(14)
- 基本功(38)
- 开发基础(80)
- 猫奇(15)
- 流媒体(10)
- 爬虫(4)
- 项目管理工具(14)
- 小程序(19)
- 信息安全(4)
- 运营及各种环境搭建(63)

阅读排行榜

1. Linux查看物理CPU个数、核数、逻辑CPU个数(173375)
2. CRC校验码原理、实例、手动计算(86721)
3. Linux下使用automake、autoconf生成configure文件(41100)
4. nginx支持HTTP2的配置过程(3263)
5. 127.0.0.1、0.0.0.0和本机IP地址的区别和使用(32100)

评论排行榜

1. 阿里云感染挖矿木马的脚本(4)
2. 在Linux用libcurl.a在链接的时候出错(4)
3. libcurl异步访问示例(2)
4. 一个TCP报文最多传输多少字节(即MSS)(2)
5. nodejs如何获取微信支付统一下单的返回的数据值(2)

推荐排行榜

1. Linux查看物理CPU个数、核数、逻辑CPU个数(11)
2. CRC校验码原理、实例、手动计算(8)
3. Linux下使用automake、autoconf生成configure文件(4)
4. gitee支持的开源许可证(2)
5. libtorrent资料整理(2)

最新评论

如何让ffmpeg-flv支持H265(HEVC)

目次

- [背景](#)
- [参考网址](#)
- [具体操作](#)
 - [1. 下载代码](#)
 - [2. 修改ffmpeg](#)
 - [3. 修改nginx-http-flv-module](#)
 - [4. nginx-http-flv测试结果](#)
 - [rtmp](#)
 - [flv](#)
 - [hls](#)

背景

Adobe不支持flash了,rtmp之前是为flash设计的,所以协议也不更新了作为FLV相关技术的所有者, Adobe估计也没动力更新规范来支持H265;

外加H265标准的一些专利所有者的授权方案被指抢钱，更加打击了很多软件厂商的积极性。

但我们这些升斗小民对h265和rtmp、flv还有需求，只能自己动手解决这个问题了。

参考网址

<https://blog.csdn.net/smallhuiju/article/details/81703434>
<https://github.com/ensonz/nginx-http-flv-module/>
<https://blog.csdn.net/ybn187/article/details/88351874>

还有下面这个，就不用看了，虽然他是原作者，但没写解决方案
<https://blog.csdn.net/winshining/article/details/101225551>

具体操作

1. 下载代码:

```
git clone https://gitee.com/mirrors/ffmpeg.git
```

2.修改ffmpeg

```

root@ll ffmpeg# git status
# On branch master
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
# modified:   libavformat/flv.h
# modified:   libavformat/flvdec.c
# modified:   libavformat/flvenc.c
#
#
no changes added to commit (use "git add" and/or "git commit -a")
[root@ll ffmpeg]# git diff
diff --git a/libavformat/flv.h b/libavformat/flv.h
index 3571b90..e776555 100644
--- a/libavformat/flv.h
+++ b/libavformat/flv.h
@@ -110,6 +110,7 @@ enum {
     FLV_CODECID_H264 = 7,
     FLV_CODECID_REALH263 = 8,
     FLV_CODECID_MPEG4 = 9,
     FLV_CODECID_HEVC = 12,
 };

enum {
diff --git a/libavformat/flvdec.c b/libavformat/flvdec.c
index e4b40a1..da5555f 100644
--- a/libavformat/flvdec.c
+++ b/libavformat/flvdec.c
@@ -36,6 +36,7 @@
@@ -36,6 +36,7 @@
#include "internal.h"
#include "avio_internal.h"
#include "flv.h"
#include "hevc.h"

#define VALIDATE_INDEX_TS_THRESH 2500

@@ -320,6 +321,8 @@ static int flv_same_video_codec(AVCodecParameters *vpar, int flags)
    return vpar->codec_id == AV_CODEC_ID_VP6A;
case FLV_CODECID_H264:
    return vpar->codec_id == AV_CODEC_ID_H264;
+ case FLV_CODECID_HEVC:
+     return vpar->codec_id == AV_CODEC_ID_HEVC;
default:
    return vpar->codec_tag == flv_codec_id;
}

@@ -369,6 +372,11 @@ static int flv_video_codec(AVFormatContext *s, AVStream *vstre
    par->codec_id = AV_CODEC_ID_MPEG4;
    ret = 3;
    break;
+ case FLV_CODECID_HEVC:
+     par->codec_id = AV_CODEC_ID_HEVC;
+     vstream->need_parsing = AVSTREAM_PARSE_NONE;
+     ret = 3; // not 4, reading packet type will consume one byte
+     break;
default:
    avpriv_request_sample(s, "Video codec (0x%02x)", flv_codec_id);
    par->codec_tag = flv_codec_id;

@@ -1239,7 +1247,8 @@ retry_duration:

if (st->codecpar->codec_id == AV_CODEC_ID_AAC ||
    st->codecpar->codec_id == AV_CODEC_ID_H264 ||
-   st->codecpar->codec_id == AV_CODEC_ID_MPEG4) {
+   st->codecpar->codec_id == AV_CODEC_ID_MPEG4 ||
+   st->codecpar->codec_id == AV_CODEC_ID_MPEG4 ||
+   st->codecpar->codec_id == AV_CODEC_ID_HEVC) {
    int type = avio_r8(s->pb);
    size--;
}

@@ -1248,7 +1257,9 @@ retry_duration:
    goto leave;
}

if (st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_
+ if (st->codecpar->codec_id == AV_CODEC_ID_H264 ||
+   st->codecpar->codec_id == AV_CODEC_ID_H264 ||
+   st->codecpar->codec_id == AV_CODEC_ID_MPEG4 ||
+   st->codecpar->codec_id == AV_CODEC_ID_HEVC) {
    // sign extension
    pts32_t cts = (avio_rb24(s->pb) & 0xffff0000) ^ 0xffff0000;
    int64_t sat_satadd64(dts, cts);

@@ -1264,7 +1275,8 @@ retry_duration:
}

if (type == 0 && (!st->codecpar->extradata || st->codecpar->codec_id == AV_
-   st->codecpar->codec_id == AV_CODEC_ID_H264)) {

```

1. Re:VMWare虚拟机出现问题
这啥玩意?
--规格严格-功夫到家
2. Re:Linux下使用automake, autoconf生成configure文件
mark,two win.
感谢分享, 跟着做果然做出来了.
--王林森
3. Re:Linux shell如何用法正则表达式匹配分組数据
博主可以试试grep加参数-Eo, 想要的参数, 在正则表达式用小括号括起来 echo `a,aa,aaa`|grep -Eo `(a+)*` a aa aaa...
--givedrug
4. Re:CRC校验码原理、实例、手动计算
有那个例子,立刻懂了.
--袁良旋
5. Re:libcurl异步访问示例
@我呆了一个月了 如果没有调用curl_global_init, curl_easy_init会自动执行此操作。只不过curl_easy_cleanup没有对应的对象, 不安全, 具体程序具体分析吧。...
--步孤天
Copyright © 2023 步孤天
Powered by .NET 7.0 on Kubernetes

```
st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_CODEC_ID_HEVC } (
AVDictionaryEntry *t;

    if (st->codecpar->extradata) {
diff --git a/libavformat/flvenc.c b/libavformat/flvenc.c
index 35bf7ac..0ed3254 100644
--- a/libavformat/flvenc.c
+++ b/libavformat/flvenc.c
@@ -34,6 +34,7 @@
#include "libavutil/opt.h"
#include "libavcodec/put_bits.h"
#include "libavcodec/aacenc.h"
+include "hevc.h"

static const AVCodecTag flv_video_codec_ids[] = {
@@ -46,6 +47,7 @@ static const AVCodecTag flv_video_codec_ids[] = {
    { AV_CODEC_ID_VP6,    FLV_CODECID_VP6 },
    { AV_CODEC_ID_VP6A,   FLV_CODECID_VP6A },
    { AV_CODEC_ID_H264,   FLV_CODECID_H264 },
+    { AV_CODEC_ID_HEVC,  FLV_CODECID_HEVC },
+    { AV_CODEC_ID_NONE,  0 }
};

@@ -491,7 +493,7 @@ static void flv_write_codec_header(AVFormatContext* s, AVCodecParameters* par, i
    FLVContext *flv = s->priv_data;

    if (par->codec_id == AV_CODEC_ID_AAC || par->codec_id == AV_CODEC_ID_H264
-        || par->codec_id == AV_CODEC_ID_MPEG4) {
+        || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
        int64_t pos;
        avio_w8(pb,
@@ -537,7 +539,11 @@ static void flv_write_codec_header(AVFormatContext* s, AVCodecParameters* par, i
        avio_w8(pb, par->codec_tag | FLV_FRAME_KEY); // flags
        avio_w8(pb, 0); // AVC sequence header
        avio_wb24(pb, 0); // composition time
-        ff_isom_write_avcc(pb, par->extradata, par->extradata_size);
+        if (par->codec_id == AV_CODEC_ID_HEVC) {
+            ff_isom_write_hvcc(pb, par->extradata, par->extradata_size, 0);
+        } else {
+            ff_isom_write_avcc(pb, par->extradata, par->extradata_size);
+        }
        data_size = avio_tell(pb) - pos;
        avio_seek(pb, -data_size - 10, SEEK_CUR);
@@ -844,7 +850,7 @@ end:
    AVCodecParameters *par = s->streams[i]->codecpar;
    FLVStreamContext *sc = s->streams[i]->priv_data;
    if (par->codec_type == AVMEDIA_TYPE_VIDEO &&
-        (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4))
+        (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC))
        put_avc_eos_tag(pb, sc->last_ts);
    }
@@ -893,15 +899,15 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
    }

    if (par->codec_id == AV_CODEC_ID_VP6F || par->codec_id == AV_CODEC_ID_VP6A ||
-        par->codec_id == AV_CODEC_ID_VP6 || par->codec_id == AV_CODEC_ID_AAC)
+        par->codec_id == AV_CODEC_ID_VP6 || par->codec_id == AV_CODEC_ID_AAC){
        flags_size = 2;
-    } else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4)
+    } else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
        flags_size = 5;
-    } else
+    } else {
        flags_size = 1;
+    }
+ }

+ /*****
+ //这里要注意了,注释掉的这段代码, James-0那篇博客上说是为了解决"服务端nginx并没有添加HEVC下发flv的header头部"的问题,
+ //但是如果把下面代码片段连成一段,用264对接nginx-http-flv时, 虽然用vlc可以播放出来, 但用flv.js根本无法播放出来. 这条路不通. 不建议大家用
+ //至于nginx怎么改的问题, 下次再说
+ if (par->codec_id == AV_CODEC_ID_AAC || par->codec_id == AV_CODEC_ID_H264
+     || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
+     int side_size = 0;
+     uint8_t *side = av_packet_get_side_data(pkt, AV_PKT_DATA_NEW_EXTRADATA, &side_size);
+     if (side && side_size > 0 && (side_size != par->extradata_size || memcmp(side, par->extradata, side_size))) {
+         av_free(par->extradata);
+         par->extradata = av_mallocz(side_size + AV_INPUT_BUFFER_PADDING_SIZE);
+         if (!par->extradata) {
+             par->extradata_size = 0;
+             return AVERROR(ENOMEM);
+         }
+         memcpy(par->extradata, side, side_size);
+         par->extradata_size = side_size;
+         flv_write_codec_header(s, par, pkt->pts);
+     } else {
+         flv_write_codec_header(s, par, pkt->pts);
+     }
+ }
+ ****/
+ if (par->codec_id == AV_CODEC_ID_AAC || par->codec_id == AV_CODEC_ID_H264
+     || par->codec_id == AV_CODEC_ID_MPEG4) {
+     || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
        buffer_size_t side_size;
        uint8_t *side = av_packet_get_side_data(pkt, AV_PKT_DATA_NEW_EXTRADATA, &side_size);
        if (side && side_size > 0 && (side_size != par->extradata_size || memcmp(side, par->extradata, side_size))) {
@@ -966,6 +993,10 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
        if (par->extradata_size > 0 && *(uint8_t*)par->extradata != 1)
            if ((ret = ff_avc_parse_nal_units_buf(pkt->data, &data, &size)) < 0)
                return ret;
+    } else if (par->codec_id == AV_CODEC_ID_HEVC) {
+        if (par->extradata_size > 0 && *(uint8_t*)par->extradata != 1)
+            if ((ret = ff_hevc_annexb2mp4_buf(pkt->data, &data, &size, 0, NULL)) < 0)
+                return ret;
+    } else if (par->codec_id == AV_CODEC_ID_AAC && pkt->size > 2 &&
+        (AV_RB16(pkt->data) & 0xffff0) == 0xffff0) {
        if (!s->streams[pkt->stream_index]->nb_frames) {
@@ -1036,9 +1067,9 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
        else
            avio_w8(pb, ((FFALIGN(par->width, 16) - par->width) << 4) |
                (FFALIGN(par->height, 16) - par->height));
-    } else if (par->codec_id == AV_CODEC_ID_AAC)
+    } else if (par->codec_id == AV_CODEC_ID_AAC) {
+        avio_w8(pb, 1); // AAC raw
-    } else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4) {
+    } else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
        avio_w8(pb, 1); // AVC NALU
        avio_wb24(pb, pkt->pts - pkt->pts);
    }
}
[root@lh ffmpeg]#
```

至此ffmpeg的修改算是完成了, 实际测试过程中发现确实有效.

(1)将h265的码流用hvc封装, 然后再把封装好的视频转回mpeglts完全可以正常播放.

(2)使用ffmpeg直接推流到nginx-http-flv, 然后再从rtmp串流中转回mpeglts也完全可以正常播放, 没有问题.

3.修改nginx-http-flv-module

同样的nginx-http-flv-module也需要修改代码

修改后的参见: <https://github.com/ensonz/nginx-http-flv-module/>

我对比了一下, 实际上ensonz只改了下面的两个文件:

ngx_rtmp_codec_module.c

ngx_rtmp_codec_module.h

这两个文件我已经压缩放在这里了: https://files.cnblogs.com/files/bugutian/ngx_rtmp_codec_module.zip

4.nginx-http-flv测试结果

按照以上方法改动之后, 我自测试的结果如下:

rtmp

输出的rtmp流是h265编码的, 使用修改后的ffmpeg将rtmp重新转换成mpeglts后可以使用vlc播放;

flv

输出的flv流是h265编码的, 使用修改后的ffmpeg将rtmp重新转换成mpeglts后可以使用vlc播放, 但flv.js无法播放;

hls

不能生成文件.无法使用.

把代码中NGX_RTMP_VIDEO_H264的部分放宽对NGX_RTMP_VIDEO_H265的限制,也只能保证hls可以生成文件,但是m3u8串流里面找不到frame,不可用。

最后衷心感谢对nginx-http-flv、ffmpeg开源项目做出贡献的开发者。

转载请注明来源: <https://www.cnblogs.com/bugutian/>

分类: [流媒体](#), [Linux C++](#)

好文置顶

关注我

收藏该文



博主太
酷啦 - 19 关注 - 35

→ 30334

- 上一篇: [验证c++11里面的future是否是新开一个线程实现的异步](#)
- 下一篇: [FNET程序实例](#)

0

推荐

0

反对

posted on 2021-03-16 20:06 [步狐天](#) 阅读(913) 评论(0) [编辑](#) [收藏](#) [举报](#)

弹尽粮绝，会员数园：会员上线，命悬一线

[刷新评论](#) [刷新页面](#) [返回顶部](#)

[登录后才能查看或发表评论, 立即 登录 或者 逛逛 博客园首页](#)

【推荐】腾讯2023全球数字生态大会——智变加速，产业焕新，立即预约直播

【推荐】领取免费阿里云ECS试用资源，快速部署Java环境，领取小礼品

【推荐】阿里云-云服务器省钱攻略：五种权益，限时发放，不容错过

【推荐】天翼云818全民上云季，爆款云主机2核2G三个月仅47.4元

【推荐】SQL专家云：SQL Server 数据库可视化、智能化运维平台

编辑推荐:

- [CSS 也能实现碰撞检测?](#)
- [WPF 实现 Element UI 风格的日期时间选择器](#)
- [微服务14：微服务治理之重试](#)
- [领域驱动设计\(DDD\)：从基础代码探讨高内聚低耦合的演进](#)
- [MediatR 和 FluentValidation 实现 CQRS 应用程序的数据验证](#)

C-阿里云

[免费领取阿里云云服务器ECS, 部署Java Web环境赢大奖](#)



阅读排行:

- [WPF实现Element UI风格的日期时间选择器](#)
- [解放生产力orm开发更新下应该这么处理求求你别再再用UpdateById了](#)
- [震惊！CSS 也能实现碰撞检测?](#)
- [Avalonia 实现聊天消息渲染、图文混排（支持Windows、Linux、信创国产OS）](#)
- [.NET Evolve 数据库版本管理工具](#)