物联网智能视频服务（消费版）

常见问题

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

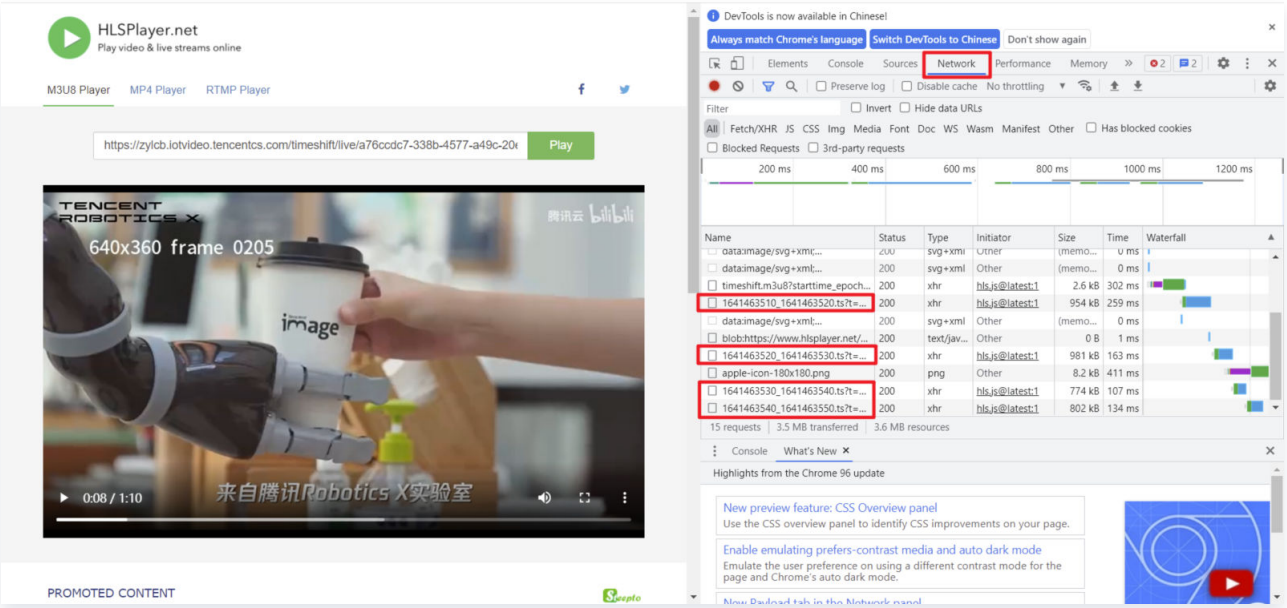
常见问题
开发相关问题
云存相关问题
云存视频下载方法
云存录像问题排查方法
云存录像播放器异常
云存录像回放时间异常
云存录像花屏
并行事件录像时长不正确
云存录像多1分钟
事件云存无事件消息
云存上传成功但播放器不播放
电脑端播放云存视频的方法
音视频传输和对讲相关问题
如何实现码率自适应
发送音视频数据返回错误问题
观看直播或者回放时画面卡顿或者花屏
观看直播或者回放时画面延时大或者黑屏
音视频传输和对讲问题排查方法
物模型相关问题
字符串相关物模型上传失败
其他问题
CPU 使用率高如何优化
常用工具
一般性问题

常见问题
开发相关问题
云存相关问题
云存视频下载方法

最近更新时间：2023-08-09 12:01:41

方法一（推荐）

1. 使用 Chrome 浏览器打开 `https://www.hlsplayer.net/` 等 m3u8 在线播放器。
2. 按【F12】打开开发者工具，输入云存的播放地址并播放。
3. 可以在 **Network** 标签页中看到若干 ts 文件，如图所示：



4. 在这些 ts 文件上右键选择 **Copy > Copy link address**。
5. 将复制的链接放入任意下载器中进行下载（Chrome 直接访问该链接也可下载）。

方法二（推荐）

自行编写 Python 脚本进行下载，这里给出简易下载脚本，仅供参考。

```
from urllib.parse import urlparse
import requests

def get_m3u8(url):
    r = requests.get(url)
    if (r.status_code != 200):
```

```
        return None
    return r.content.decode("utf-8")

def make_ts_list(url, m3u8):
    ts_list = []
    m3u8_url = urlparse(url)
    url_head = m3u8_url.scheme + '://' + m3u8_url.hostname
    m3u8_lines = m3u8.split("\n")
    for each_line in m3u8_lines:
        if not each_line.startswith('#') and each_line != '':
            ts_list.append('%s%s' %(url_head, each_line))
    return ts_list

def download_ts(ts_list):
    for each_ts in ts_list:
        ts_url = urlparse(each_ts)
        pos = ts_url.path.rfind('/') + 1
        filename = ts_url.path[pos:]
        print("download " + filename)
        r = requests.get(each_ts)
        if (r.status_code != 200):
            return
        with open(filename, "wb") as fw:
            fw.write(r.content)
    print("download finish")

def main():
    aim_url = "https://zylcb.iotvideo.tencentcs.com/timeshift/live/timeshift.m3u8"
    m3u8 = get_m3u8(aim_url)
    ts_list = make_ts_list(aim_url, m3u8)
    download_ts(ts_list)

if (__name__ == "__main__"):
    main()
```

方法三

1. Chrome 浏览器安装“网页资源嗅探器”等类似插件（此类插件众多，这里不做推荐）。
2. 打开 `https://www.hlsplayer.net/` 等 m3u8 在线播放器。
3. 输入云存的播放地址并播放，嗅探器会自动识别视频并下载。

注意事项

不要使用 ffmpeg，vlc 等工具进行下载，这类工具会进行二次封装或二次格式转换，导致原始信息丢失。

云存录像问题排查方法

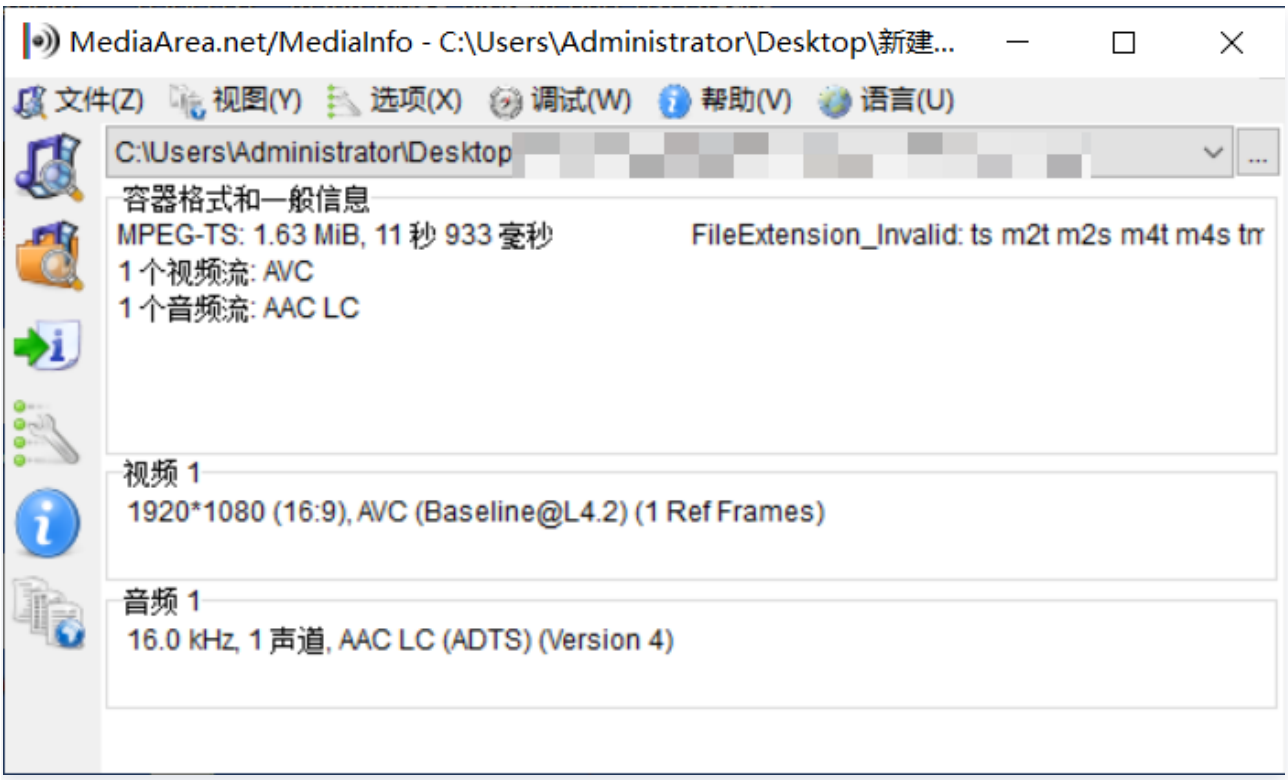
最近更新时间：2022-02-15 10:55:49

准备工作

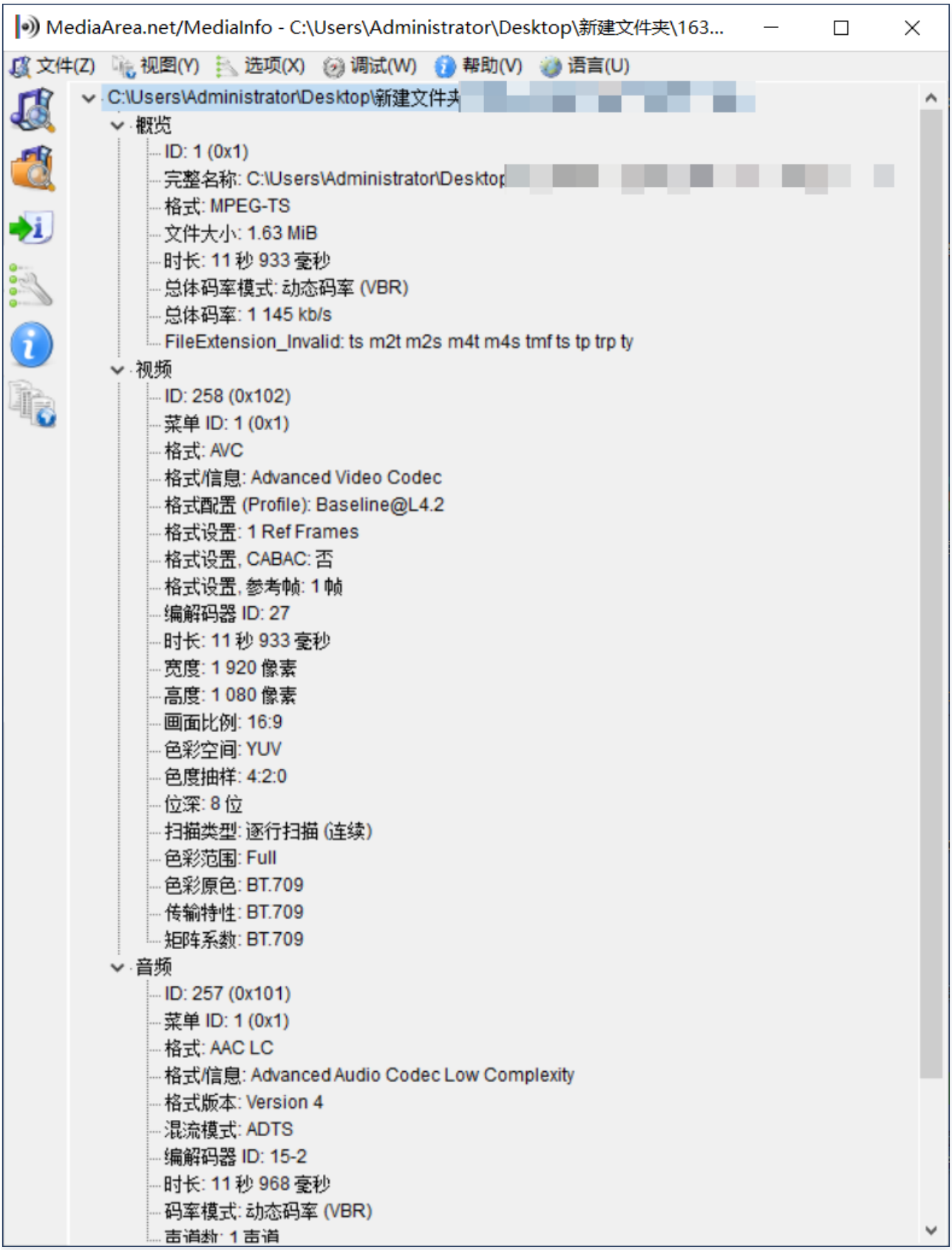
- 按前文所述方法 下载云存视频 。
- 准备相关工具软件。

云存录像基础检查

使用 MediaInfo 打开视频文件即可看到基本信息。



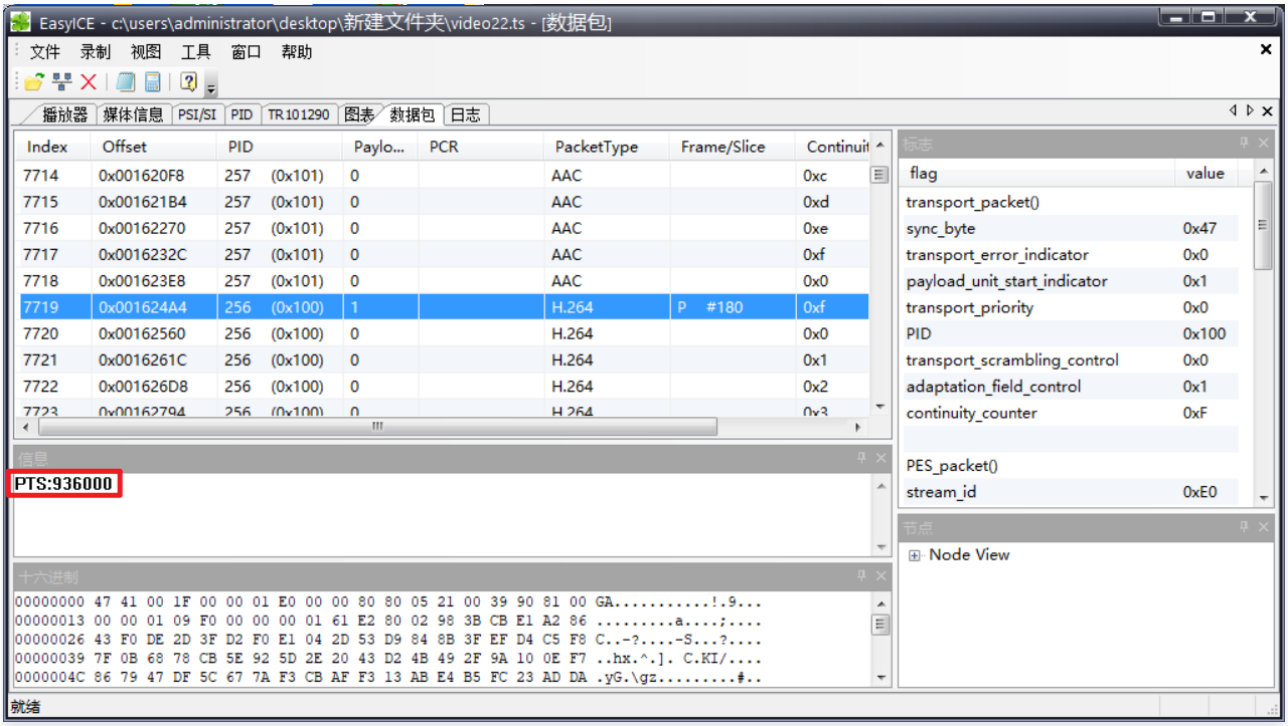
单击**视图** > **树状图** 可以看到更为详细的信息。



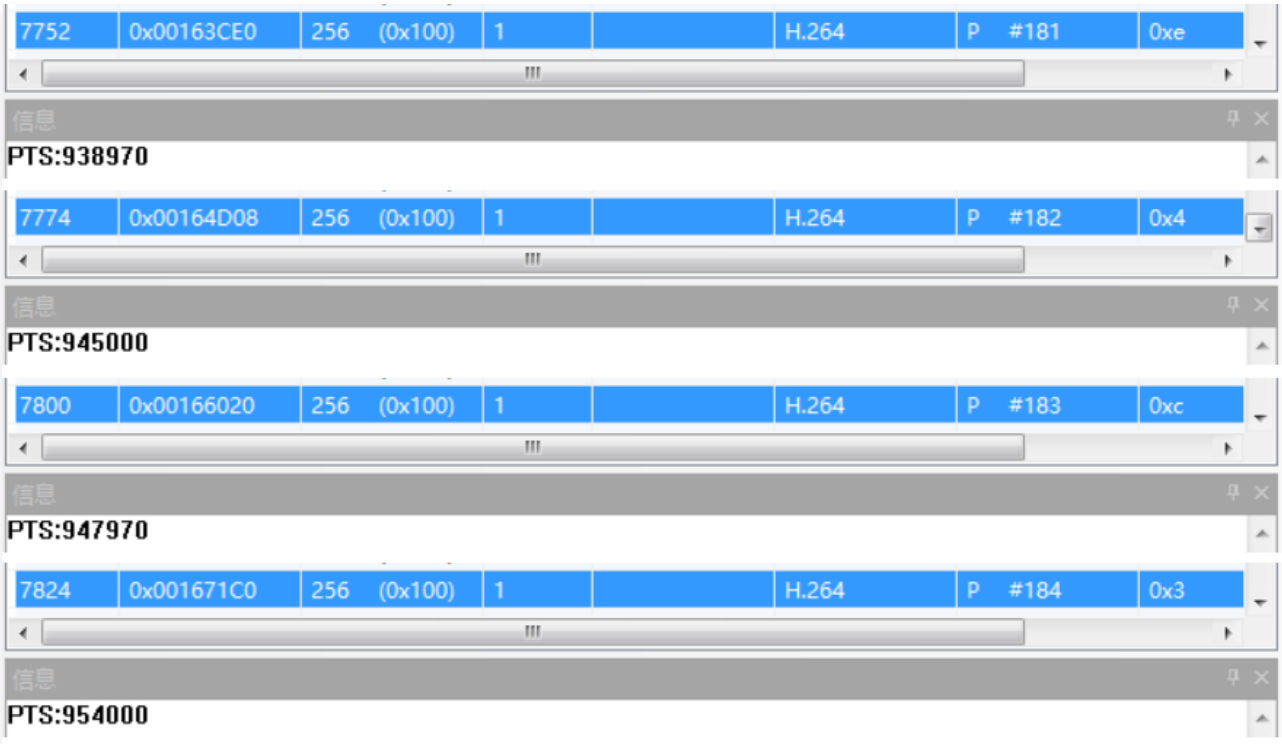
通过这些信息可以对视频做基本检查，例如视频分辨率是否正确、帧率是否正常、音频流数据是否缺失等。

云存录像时间异常问题

使用 EasyICE 打开视频，单击 **数据包** 标签页，找到任意几个连续的视频帧，计算它们的 pts 间隔是否正确。



如下图所示，从第180帧开始几个视频帧的 PTS 分别为 936000，938970，945000，947970，954000。

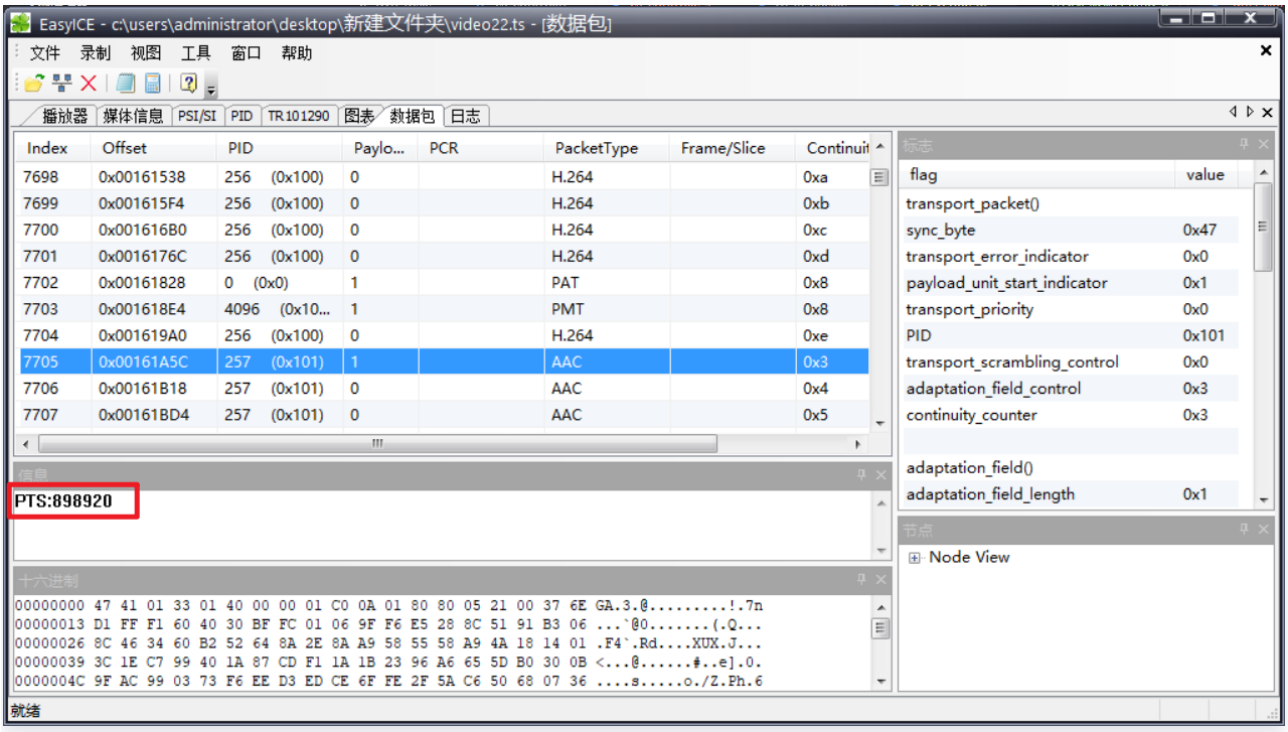


MPEG-TS 标准规定音视频要使用一个90KHz的相对时钟或绝对时钟进行同步，因此将它们换算成毫秒需要除以90，计算结果为：10400，10433，10500，10533，10600。

假设设备端的帧率为25fps，这几个时间戳虽然分布不均匀，但之间的差值基本在50ms左右，可以认为视频时间戳正常。

同样地，音频帧也可以按照这种方式检查，但需要注意，云存会将所有音频非 aac 音频转换为 aac 格式，aac 每个音频帧为1024个采样点，假设音频采样率为44.1KHz，那么每个音频帧的时长为1024/44100=0.02322s=23.22ms。

接下来检查音视频时间戳是否同步，随意找几个视频帧附近的音频，观察音频帧和视频帧的 PTS 误差是否过大。

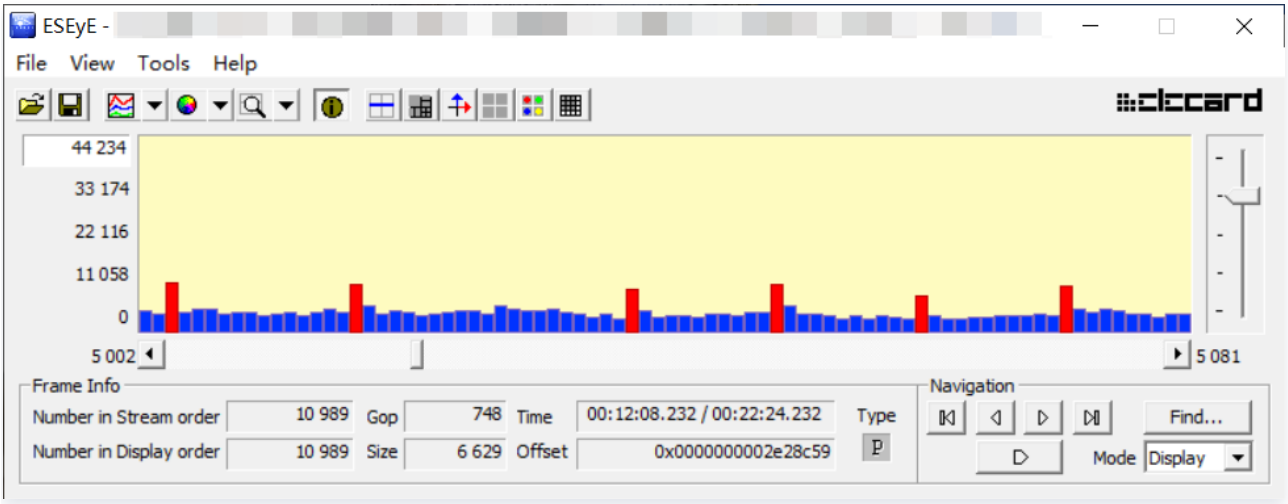


如上图所示，这是第180帧之前的一个音频帧，PTS 为 898920 即 9988ms，和视频帧 10400ms 相差412ms，这个误差有点大，但基本正常，一般来说误差在几百毫秒内都属于可接受范围。如果音视频之间的时间戳偏差过大，请在推流时检查时间戳是否正常。

云存录像花屏问题

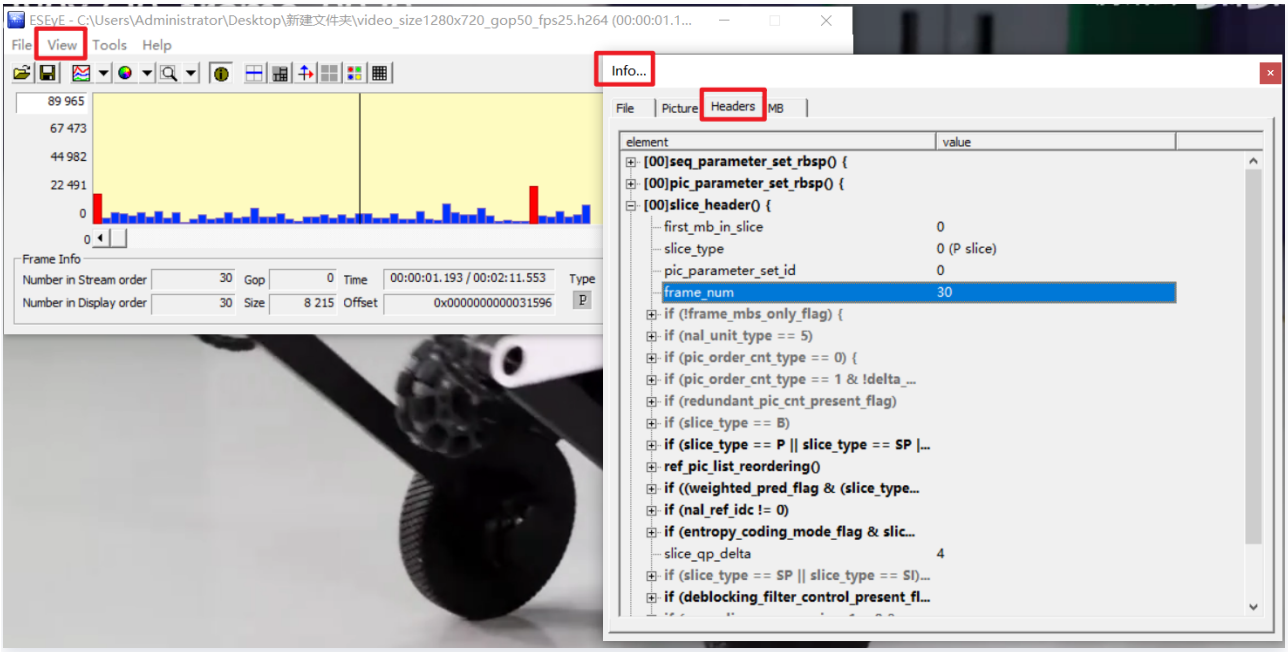
使用 Elecard StreamEye Tools 打开视频。

假设 GOP 为15帧，如图所示这里 I帧、P帧分布不均匀，明显有大量视频帧丢失，云存视频播放到这里就会花屏。



也可单击 View > Info > Headers > slice_header() 查看前后两帧的 frame_num 是否连续，如果不连续则

有丢帧。


ESEyE - C:\Users\Administrator\Desktop\新建文件夹\video_size1280x720_gop50_fps25.h264 (00:00:01.1...
File View Tools Help
89 965
67 473
44 982
22 491
0
Frame Info
Number in Stream order 30 Gop 0 Time 00:00:01.193 / 00:02:11.553 Type
Number in Display order 30 Size 8 215 Offset 0x0000000000031596 P
Info...
File Picture Headers MB
element value
[00]seq_parameter_set_rbsp() {
[00]pic_parameter_set_rbsp() {
[00]slice_header() {
first_mb_in_slice 0
slice_type 0 (P slice)
pic_parameter_set_id 0
frame_num 30
if (!frame_mbs_only_flag) {
if (nal_unit_type == 5)
if (pic_order_cnt_type == 0) {
if (pic_order_cnt_type == 1 & !delta_...
if (redundant_pic_cnt_present_flag)
if (slice_type == B)
if (slice_type == P || slice_type == SP |...
ref_pic_list_reordering()
if ((weighted_pred_flag & (slice_type...
if (nal_ref_idc != 0)
if (entropy_coding_mode_flag & slic...
slice_qp_delta 4
if (slice_type == SP || slice_type == SI)...
if (deblocking_filter_control_present_fl...

云存录像播放器异常

最近更新时间：2022-02-15 10:54:51

详细描述

云存录像播放异常，包括但不限于以下情况：

- 云存视频部分播放器能正常播放，其他播放器不能播放。
- 云存视频在线播放失败，下载后播放正常。
- 云存视频播放卡顿。
- 播放器闪退。

原因分析

目前常用的播放器功能差异较大，需要根据实际情况判断，常见原因有：

- 不支持软解码（或硬解码）H.264。
- 不支持软解码（或硬解码）H.265。
- 软解码H.265 CPU使用率过高导致卡顿（普通家用电脑（或手机）软解码高码率4K H.265视频可能非常吃力）。
- 云存视频异常，播放器解码错误导致闪退。

解决方法

建议根据播放器的实际情况进行调整，例如：

- 开启或关闭H.264软解码（硬解码）功能。
- 开启或关闭H.265软解码（硬解码）功能。
- 开启跳帧功能，优先保证画面流畅度。
- 开启纠错功能，跳过异常数据。

版权所有：腾讯云计算（北京）有限责任公司

云存录像回放时间异常

最近更新时间：2023-05-23 17:12:49

详细描述

云存录像回放时间异常，包括但不限于以下情况：

- 音视频不同步。
- 前30秒只有音频，从第30秒开始有音频和视频，音视频之间相差30秒（或类似问题）。
- 视频回放时播放速度时快时慢。
- 实际录像时长为1分钟，回放时画面只显示了一瞬间，播放器闪退。
- 实际录像时长为1分钟，播放器进度条显示时长为16小时，且画面卡住不动。
- 实际录像时长为1分钟，播放器进度条显示时长远超1分钟，且无画面或无声音或即无画面也无声音。
  部分播放器对于这类情况的处理也不同，最终实际的播放效果也不同，上述现象仅供参考。

原因分析

以上几种问题或类似问题都是音视频帧时间戳异常导致的。云存视频回放严格依赖时间戳，因此在推送音视频帧时务必保证时间戳正确。
下面进行逐个分析：

- **音视频不同步。**
  通常来说音视频帧的时间戳是这一帧采集的时刻，一般硬件编码器都带有时间戳，这个时间戳建议直接从编码器取出。如果编码器不带时间戳在手动添加时间戳时请尽量保证时间戳准确。

  手动添加时间戳常见错误是没有考虑误差积累，特别是音频有重采样、格式转换等操作，视频有改变帧率等操作更容易引入误差，导致音视频帧的时间戳误差越来越大。

  > 例如视频帧率是 30fps，每帧之间相差33.333……毫秒，整除为33毫秒，不进行误差补偿的时间戳为0,33,66,99,132,165,198，补偿后为 0,33,66,100,133,166,200。

  另一种情形是硬件编码的时间戳从初始化的那一刻开始计时，例如音频编码器在第0秒初始化，视频编码器在第3秒初始化，两个编码器的时间戳都是从0开始计时，两个编码器的时间戳虽然都从0开始计时，但因为初始化时间不同使拿到的音视频时间戳就始终相差3秒，导致音视频不同步。
- **前30秒只有音频，从第30秒开始有音频和视频，音视频之间相差30秒（或类似问题）。**
  同样属于音视频不同步问题，见上文。
- **视频回放时播放速度时快时慢。**
  为了光线不足时的画质，部分芯片在夜间或黑暗环境下会延长曝光时间来提升画面亮度，因此导致帧率降低。这种情况往往是手动计算帧率导致的。

  > 例如明亮环境下默认20fps，黑暗环境下降低为 10fps，但时间戳仍然按20fps计算。
  > 如下所示，假设在第5帧处帧率由20fps变为10fps，则：
  > - 正常时间戳 0,50,100,150,200,300,400,500,600,700

> - 异常时间戳 0,50,100,150,200,250,300,350,400,450

最终的效果就是明亮环境下画面正常，黑暗环境下画面速度为正常的2倍。如果芯片的 ISP 算法会改变帧率，建议直接从编码器获取时间戳，如需手动计算时间戳请按实际帧率计算，或者在向 SDK 推送视频帧时直接从系统获取毫秒级时间戳。

- **实际录像时长为1分钟，回放时画面只显示了一瞬间，播放器闪退。**
  音视频帧的时间戳填写错误，典型情况就是误将时间戳填写为帧序号。

  > 假设帧率为20fps：
  > - 正常时间戳 0,50,100,150,200,300,400,500,600,700
  > - 异常时间戳 0,1,2,3,4,5,6,7,8,9

  播放的效果为画面以正常速度的50倍进行快放，即1分钟的视频仅用1秒左右就播放完了，给人的感觉就是画面只显示了一瞬间，播放器闪退。
- **实际录像时长为1分钟，播放器进度条显示时长为16小时，且画面卡住不动。**
  SDK 接收的音视频帧时间戳单位是毫秒，造成这种情况是误将时间戳填成了微秒，使得1分钟的录像变为了16小时，画面其实并没有真正卡住，而是以相当于千分之一的速度慢放。
- **实际录像时长为1分钟，播放器进度条显示时长远超1分钟，且无画面或无声音或即无画面也无声音。**
  这种情况是音视频使用了不同的时间戳导致的。例如音频的时间戳从0开始计时，视频的时间戳从当前 UTC 事件开始计时，两者相差的时间非常大，使得播放器的进度条显示的时间异常以及播放异常。
  SDK 要求必须填写音视频帧的时间戳，但不对时间戳的参考时间做强制要求，用户可以根据自己的实际情况填写。例如长供电设备的时间戳可以使用精确到毫秒的 UTC 时间，断电设备的时间戳可以从0开始计时，或者也可以使用其他值同时作为音视频时间戳的基准。总之音频和视频一定要使用相同的参考时间或相同的时间源，不要各自用各自独立的时间源。

解决方法

保证音视频时间戳准确无误（手动计算时间戳时考虑误差、可变帧率等情况），保证音视频时间戳采用相同的参考时间或相同的时间源（例如编码器时钟、RTC 时钟、UTC 时钟、1毫秒 tick 时钟等）。

云存录像花屏

最近更新时间：2023-05-23 17:12:49

详细描述

云存录像回放时花屏（部分播放器可能会跳过花屏部分继续播放）。

原因分析

录像花屏说明云存上传过程中可能出现丢帧。例如由于网速原因导致云存缓存满，此时无法推送新的音视频帧进来，如果用户不做缓存则只能将这些帧丢掉，待网速恢复以后继续上传后续的音视频帧。这种情况下丢帧的地方就有可能异常，部分播放器会强制解码或者会跳过寻找下一个 I 帧继续播放。

解决方法

开启播放器的纠错功能等。
如果不希望花屏，用户可以将无法推送的视频数据暂存起来，等网络恢复以后继续发送；或丢弃无法推送的 P 帧，直到下一个 I 帧再进行推送，以此减少花屏。

版权所有：腾讯云计算（北京）有限责任公司

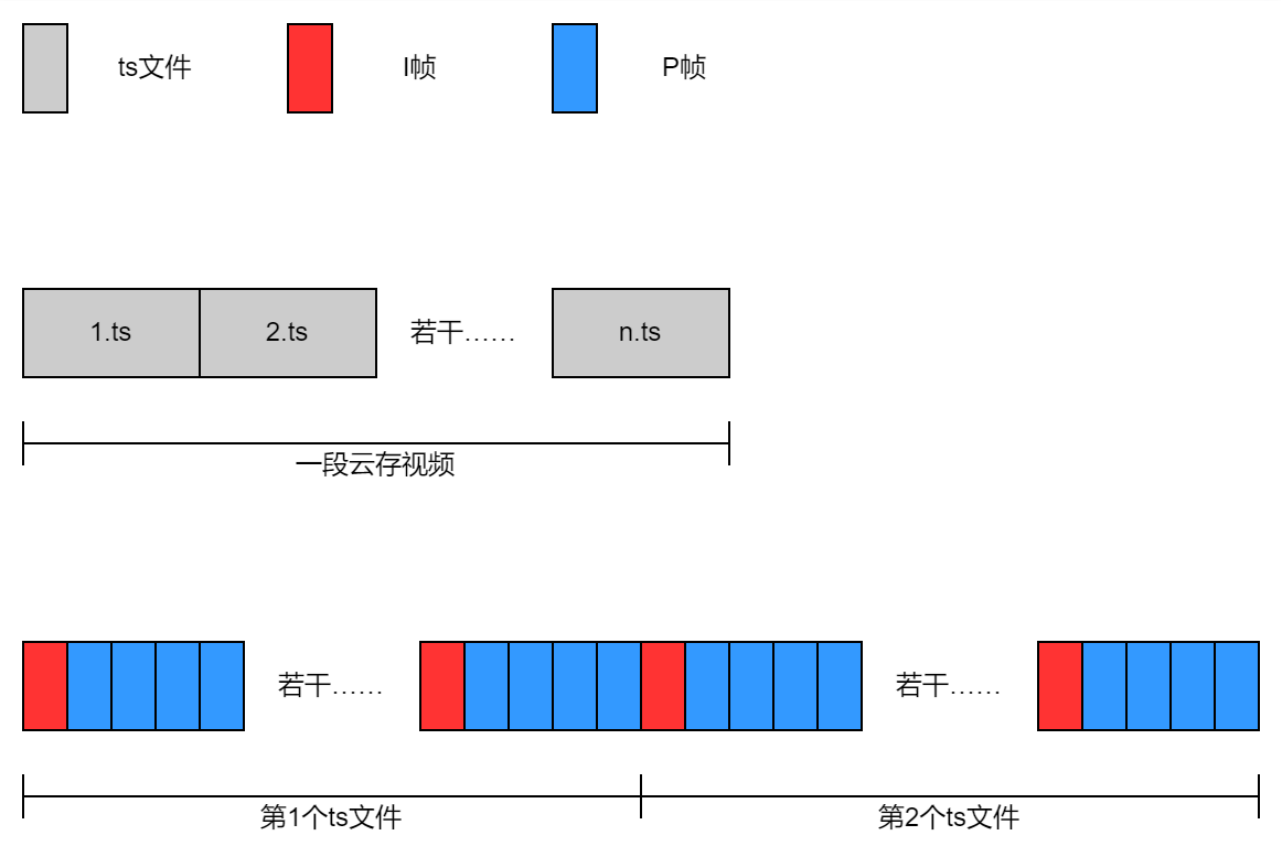
并行事件录像时长不正确

最近更新时间：2022-02-15 10:54:59

详细描述

并行事件录像时长不正确，例如第0秒触发了事件1，第15秒触发了事件2，第30秒同时结束了事件1和事件2。拉取的事件列表中事件1时长30秒，事件1视频时长30秒，事件2时长15秒，视频时长20秒。

原因分析



如图所示，出于服务器负载等原因的考虑，目前 SDK 会将每10秒的视频分割为一个 ts 文件（或 ts 分片），ts 文件会在 I帧 处进行分割以免花屏，实际长度有一定波动。

例如：

fps=20，GOP=40帧，即每2秒一个 I帧，在第10秒时正好有一个 I帧，此时会分割一个 ts 文件。

fps=20，GOP=60帧，即每3秒一个 I帧，在第10秒时没有 I帧，因此会等到下一个 I帧，即第12秒处进行分割。

上述问题服务器中实际保存了【0秒至10秒】，【11秒至20秒】，【21秒至30秒】的三个视频文件，当查找事件2的视频即15秒 – 30秒的视频时，服务器会返回起止时间在15秒到30秒范围内的所有视频分片，即【11秒至20秒】，【21秒至30秒】这两个视频分片，因此实际看到的事件2的视频时长为20秒。同理，假设在第9秒触发了事件3，在第21秒结束事件3，事件3的实际持续时间为12秒，对应的视频为【0秒至10秒】，【11秒至20秒】，【21秒至30秒】这三个视频。

解决方法

该问题不影响云存录像，如果需要视频时长精确匹配事件时长，可以通过播放器精确定位视频时间来实现。

云存录像多1分钟

最近更新时间：2022-02-15 10:55:02

详细描述

云存录像实际长度为3分钟，回放时进度条显示为4分钟，且最后1分钟无法播放。

原因分析

全时云存顾名思义是需要持续录像的，正常的全时云存使用流程是初始化以后会收到 `iv_cs_push_stream_start_cb` 回调，之后用户应当持续推送音视频数据，直到因为云存套餐到期、退出云存等原因收到 `iv_cs_push_stream_stop_cb` 回调再停止推流。
全时云存推流过程中可以调用 `iv_cs_event_start` 等接口触发事件。如果用户因特殊原因停止录像，SDK 内部会等待1分钟，如果1分钟内没有恢复就会结束录像，下次推流时恢复正常。等待的这1分钟内因为没有数据进来，导致回放时进度条显示的时间比实际录像时间长1分钟，且这1分钟无法播放。
开通了全时云存套餐却没有按 SDK 要求正确推流就会导致这种现象，就是开通了全时套餐，但没有按 `iv_cs_push_stream_start_cb` 回调的指示开始推流，而是自行按照触发事件开始推流、结束事件停止推流的方式使用，SDK 发现无视频流继续等待1分钟后结束录像于是造成了问题中的现象。

解决方法

全时云存进行持续录像，或更换事件云存套餐。

事件云存无事件消息

最近更新时间：2022-02-15 10:55:06

详细描述

事件云存触发事件后只有视频，没有图片（或没有事件消息，或其他类似情形）。

原因分析

调用 `iv_cs_event_start` 等接口时没有检查错误码继续推流就可能导致上述问题。

解决方法

调用 `iv_cs_event_start` 等接口时请检查返回值，如有异常不建议继续推流，这种情况下 SDK 无法保证数据能够正常上传。

云存上传成功但播放器不播放

最近更新时间：2022-02-15 10:55:10

详细描述

通过设备端日志发现云存已经上传成功，回放时进度条显示有时间，但无法播放。

原因分析

云存视频要求必须上传音频和视频，如果用户只上传视频数据部分播放器可以正常播放，部分播放器则不行。
不能正常播放的常见原因是播放器优先使用音频帧的时间戳做音视频同步，因此播放器会一直向后读取，直到遇到音频数据才开始播放。
无音频，不播放（ts 里面标记有音频，实际没音频数据，播放器卡顿等）。

解决方法

- 正常上传音视频数据。
- 如果不需要音频（例如无麦克风、静音、保护隐私等）建议发送用0填充的音频数据帧。
- 如设备端无法上传音频，则需要修改播放器的相关设置，以 ffplay 为例，添加 `-an` 参数禁用音频，添加 `-sync video` 参数使用视频进行同步。

电脑端播放云存视频的方法

最近更新时间：2023-08-09 12:01:42

- 使用 VLC、PotPlayer 等播放器，选择播放在线视频，输入云存链接即可播放。
- 使用 ffplay，输入命令并替换云存链接 `ffplay <replace your link address>` ，加入 `-loglevel trace` 参数可显示详细信息，一定程度上有助于排查云存视频问题。
- 使用 Chrome 浏览器打开 `https://www.hlsplayer.net/` 等 m3u8 在线播放器，输入云存的播放地址并播放。

音视频传输和对讲相关问题
如何实现码率自适应

最近更新时间：2023-05-23 17:12:49

详细描述

实际网络环境波动较大，如何实现码率自适应。

解决方法

方法	说明
方法一	`iv_avt_init` 初始化参数 **pstInitParm > congestion** 中可以设置是否启用水位告警以及告警的有高中低三挡水位值，当 p2p 内部缓存的水位到达这个值的时候会收到 `iv_avt_notify_cb` 回调。
方法二	使用过程中主动调用 `iv_avt_get_send_stream_buf` 查询当前水位值。
方法三	使用过程中主动调用 `iv_avt_get_send_stream_status` 查询当前的瞬时网速和1秒内的平均网速。

用户根据以上三种方法的查询结果自行开发并实现码率自适应。
下面给出一种实现思路，仅供参考。

- 当发现 p2p 的水线超过一定值时，降低视频码率。

  例如当水位超过低水位时将视频码率降为原来的80%。网络正常的情况下 p2p 水位值很低，2mbps码率的视频水位值一般在100KB以下。该数值仅供参考，送入体积较大的 I 帧、网络波动等都会影响水位值。
- 推流过程中每间隔一定时间（例如1秒）调用 `iv_avt_get_send_stream_status` 获取网速信息。

  由于瞬时速度的波动较大，这里建议使用1秒内的平均传输速度，设置一定长度的队列（例如长度为5，如果调用间隔比较短可以适当加长窗口），将该数值存入队列同时删除队列内最旧的一个数值，去掉一个最高值去掉一个最低值，计算平均值。算出的平均值可用于控制码率，一般而言此数值与视频码率相近，当发现平均网速低于视频码率时主动降低视频码率到一个比平均网速更低的值。

用户可结合以上方法实现或借鉴 cubic 拥塞控制算法等的思想实现自己的码率自适应策略。对于 p2p 透传数据请参考 `iv_avt_p2p_set_buf_watermark` ， `iv_avt_p2p_get_send_buf` ， `iv_avt_p2p_get_send_status` 接口，具体实现思路类似。

发送音视频数据返回错误问题

最近更新时间：2023-07-06 10:01:13

详细描述

调用 `iv_avt_send_stream` 发送音视频数据返回错误，不同的错误码的原因不同。

原因分析

几种常见的错误码原因如下：

- 错误码为-303时，表示此时内部缓存满，送入数据失败，一般是网络原因导致此时的网络速度低于数据发送速度。
- 错误码为-305时，表示此时的 `visitor, channel, video_res_type` 三个参数中至少有一个值与 `iv_avt_start_real_play_cb` 通知的值不一致。
- 错误码为-306时，表示当前码率启动推流时送入的第一个视频帧不是 IDR 帧。
- 错误码为-308时，表示送入的音视频格式与 `iv_avt_get_av_enc_info_cb` 设置的格式不一致,或者本身送入的数据帧格式有问题,导致流媒体协议封装失败。

解决方法

- 出现错误码-303时，在这个错误之前一般都会有水位报警，需要降低码率，码率控制方法参见 文档说明 。
- 出现错误码-305时，需要用户检查自己的代码参数配置是否存在问题。
- 出现错误码-306时，需要用户推送的第一个视频帧为 IDR 帧，也可以不用理会，等待编码器正常产生 IDR 帧，SDK 会把返回错误码的数据帧丢弃。
- 出现错误码-308时，首先需要用户检测 `iv_avt_get_av_enc_info_cb` 回调中设置的格式与实际数据帧是否匹配，如果匹配还出现该错误，需要用户将发生保存的数据保存下来，分析该数据的格式是否正确。

版权所有：腾讯云计算（北京）有限责任公司

观看直播或者回放时画面卡顿或者花屏

最近更新时间：2023-08-09 12:01:42

详细描述

在小程序或者 App 端，观看设备的直播时，画面卡顿、不流畅或花屏。

原因分析

卡顿的原因有很多种，需要逐一排除， 其排查方法如下：

- 将 App 或小程序端收到的音视频流保存在本地，格式一般为 flv；使用第三方播放器（推荐 PotPlayer 或者 VLC）观看本地保存的音视频数据，如果仍然出现卡顿，则是从原因1开始分析，否则从原因2开始。
- 原因1：这种卡顿一般是因为视频数据有缺失，从本地保存的音视频流中提取出H264/H265裸数据(推荐使用 ffmpeg)， 使用 elecard 分析 H264/H265 裸数据， 找到卡顿的时间点， 确认是否有丢帧(可根据 silce header 中的 frame_num 值判断)， 一般都是有丢帧的， 丢帧一般都是在设备端引起的。 在设备端查找该时间点发生的音视频数据， 调用 `iv_avt_send_stream` 时是否有错误， 或者编码器生成的数据是否有丢帧(可将编码器的 GOP 实时值打印出来判断)。
- 原因2：这种卡顿一般是网络带宽低于数据码率或者时序有问题引起的， 先判断卡顿点设备端是否有水位报警，然后使用 flv 分析工具分析数据的时序是否有问题， 如果时序没有问题则需要在设备端做码率控制。

解决方法

- 如果是设备端带宽原因引起的卡顿，需要做码率自适应。
- 如果是设备端丢帧导致的，则需要用户检查代码中的丢帧逻辑是否有问题。
- 如果是读取编码器数据有问题，则需要用户检查业务逻辑中 CPU 是否占用太高或者取数据线程优先级太低。

观看直播或者回放时画面延时大或者黑屏

最近更新时间：2023-08-09 12:01:42

详细描述

在小程序或者 App 端，观看设备的直播时，画面延时大或者黑屏。

原因分析

- 画面延时大， 一般是音视频帧的 PTS 出现异常导致的， IoT Video SDK 要求送入的音视频帧 PTS 单位必须是毫秒， 如果配置的单位不是毫秒， 则会在观看时出现异常。
- 画面延时大， 还有可能是设备端缓存的音视频数据太多， 需要用户检查业务中缓存的数据是否太多。
- 画面延时大， 还可能是播放器缓存的音视频数据太多， 多半伴有音视频 PTS 的同步问题， 需要在 App 端或者小程序端检查音视频帧 PTS 中差值是否太大。
- 黑屏一般也是 PTS 出现问题， 最常见的是 PTS 出现了回环， 不是单调递增的， 导致播放出现问题。

解决办法

- 如果是 PTS 问题， 需要设备端用户检查送入的音视频帧 PTS 配置是否有问题， 设备端 IoT Video SDK 在发送时不会做对音视频帧做同步缓存或者修改其 PTS。
- 如果是设备端缓存数据多引起的问题， 则需要用户检查自己的业务逻辑问题， 设备端 IoT Video SDK 只会因为网络延迟大的原因缓存数据， 其他则不会缓存。
- 还有一种场景是， 有些摄像头设备带有 PTZ 功能， 在转动时为了滤除马达的声音， 不发送音频而是等待设备静止后发送， 这样也会破坏 PTS 的连续性， 造成延时很大， 推荐采用发送静音帧的方式来解决该问题。

版权所有：腾讯云计算（北京）有限责任公司

音视频传输和对讲问题排查方法

最近更新时间：2022-02-15 10:55:52

准备工作

- APP 端保存接收到的视频流。
- 准备相关工具软件。

FLV 文件时序问题

使用 flvAnalyser 打开视频。

如果视频中有时序问题，则如图所示：

flvAnalyser - raw_video14-16_14-44.flv

FLV提取ES流 FLV导出信息 ES流导出信息 视图(V) 图表信息 FLV-AV1配置 工具 帮助(H)

类型	序号	偏移地址	数据大小	图像	编码格式	时间(HH:MM:SS.MS)(ms)	帧间隔(ms)	备注
(9)	28522	0x0531d991	14398	I	h264	[2 days 22:52:48.971] (255168971)	80	SPS PPS
(9)	28523	0x053211de	1908	P	h264	[2 days 22:52:49.011] (255169011)	40	
(8)	28524	0x05321961	161		aac	[2 days 22:52:49.012] (255169012)	120	
(9)	28525	0x05321a11	2642	P	h264	[2 days 22:52:49.091] (255169091)	80	
(8)	28526	0x05322472	170		aac	[2 days 22:52:49.132] (255169132)	120	
(9)	28527	0x0532252b	2813	P	h264	[2 days 22:52:49.171] (255169171)	80	
(9)	28528	0x05323037	2265	P	h264	[2 days 22:52:49.211] (255169211)	40	
(8)	28529	0x0532391f	167		aac	[2 days 22:52:49.301] (255169301)	169	
(9)	28530	0x053239d5	2983	P	h264	[2 days 22:52:49.301] (255169301)	90	
(9)	28531	0x0532458b	2959	P	h264	[2 days 22:52:49.371] (255169371)	70	
(9)	28532	0x05325129	2352	P	h264	[2 days 22:52:49.411] (255169411)	40	
(8)	28533	0x05325a68	153		aac	[2 days 22:52:49.412] (255169412)	111	
(9)	28534	0x05325b10	3049	P	h264	[2 days 22:52:49.491] (255169491)	80	
(8)	28535	0x05326708	181		aac	[2 days 22:52:49.532] (255169532)	120	
(9)	28536	0x053267cc	3214	P	h264	[2 days 22:52:49.571] (255169571)	80	
(9)	28537	0x05327469	2387	P	h264	[2 days 22:52:49.611] (255169611)	40	
(8)	28538	0x05327dcb	154		aac	[2 days 22:52:49.652] (255169652)	120	
(9)	28539	0x05327e74	3895	P	h264	[2 days 22:52:49.691] (255169691)	80	
(9)	28540	0x05328dba	4772	P	h264	[2 days 22:52:49.771] (255169771)	80	
(9)	28541	0x0532a06d	14584	I	h264	[2 days 22:52:54.633] (255174633)	4862	SPS PPS
(9)	28542	0x0532d974	3845	P	h264	[2 days 22:52:54.712] (255174712)	79	
(8)	28543	0x0532e888	175		aac	[2 days 22:52:54.753] (255174753)	5101	
(9)	28544	0x0532e946	3907	P	h264	[2 days 22:52:54.793] (255174793)	81	
(9)	28545	0x0532f898	3236	P	h264	[2 days 22:52:54.832] (255174832)	39	
(8)	28546	0x0533054b	156		aac	[2 days 22:52:54.873] (255174873)	120	
(9)	28547	0x053305f6	3463	P	h264	[2 days 22:52:54.913] (255174913)	81	
(9)	28548	0x0533138c	4036	P	h264	[2 days 22:52:54.993] (255174993)	80	
(8)	28549	0x0533235f	177		aac	[2 days 22:52:54.993] (255174993)	120	
(9)	28550	0x0533241f	3175	P	h264	[2 days 22:52:55.033] (255175033)	40	
(9)	28551	0x05333095	3741	P	h264	[2 days 22:52:55.113] (255175113)	80	
(8)	28552	0x05333f41	170		aac	[2 days 22:52:55.153] (255175153)	160	
(9)	28553	0x05333ffa	3691	P	h264	[2 days 22:52:55.192] (255175192)	79	
(9)	28554	0x05334e74	2903	P	h264	[2 days 22:52:55.232] (255175232)	40	
(8)	28555	0x053359da	160		aac	[2 days 22:52:55.273] (255175273)	120	
(9)	28556	0x05335a89	3654	P	h264	[2 days 22:52:55.312] (255175312)	80	
(9)	28557	0x053368de	3560	P	h264	[2 days 22:52:55.392] (255175392)	80	
(8)	28558	0x053376d5	173		aac	[2 days 22:52:55.393] (255175393)	120	
(9)	28559	0x05337791	2782	P	h264	[2 days 22:52:55.432] (255175432)	40	
(9)	28560	0x0533827e	3781	P	h264	[2 days 22:52:55.512] (255175512)	80	
(8)	28561	0x05339152	162		aac	[2 days 22:52:55.513] (255175513)	120	
(9)	28562	0x05339203	3714	P	h264	[2 days 22:52:55.593] (255175593)	81	
(9)	28563	0x0533a094	14560	I	h264	[2 days 22:52:55.632] (255175632)	39	SPS PPS
(8)	28564	0x0533d983	168		aac	[2 days 22:52:55.633] (255175633)	120	
(9)	28565	0x0533da3a	3896	P	h264	[2 days 22:52:55.712] (255175712)	80	
(8)	28566	0x0533e981	162		aac	[2 days 22:52:55.793] (255175793)	160	
(9)	28567	0x0533ea32	3885	P	h264	[2 days 22:52:55.792] (255175792)	80	
(9)	28568	0x0533f96e	3310	P	h264	[2 days 22:52:55.832] (255175832)	40	
(8)	28569	0x0534066b	158		aac	[2 days 22:52:55.913] (255175913)	120	

该位置PTS过大，一般是出现批量的丢帧，观看效果在此处可能出现跳帧或者卡顿，具体现象还需要导出视频的raw数据，使用elecard进一步分析

物模型相关问题
字符串相关物模型上传失败

最近更新时间：2022-02-15 10:55:31

详细描述

字符串相关的物模型， 包括属性， 行为或者事件， 上传失败但又没有错误码返回。

原因分析

最常见的原因是物模型中字符串使用 json 封装或者传入其他特殊字符， 物模型的值最终会被封装到一个大 json 消息体中， 如果传入的字符串本身就是 json 或者有特殊字符， 破坏了整个消息体的 json 完整性， 上报的消息不合法会被后台丢弃。

解决办法

如果需要使用字符串传输 json 信息， 推荐使用 base64 编码后再传输。

版权所有：腾讯云计算（北京）有限责任公司

其他问题

CPU 使用率高如何优化

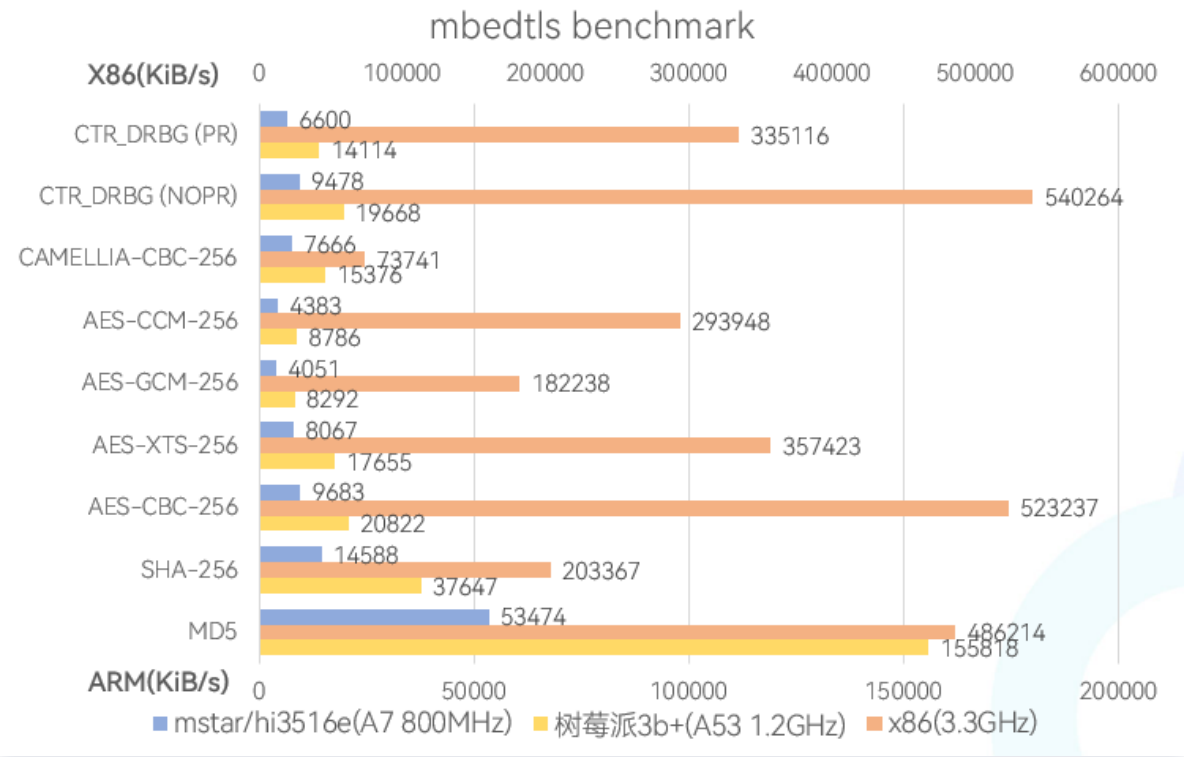
最近更新时间：2023-07-06 10:01:13

详细描述

CPU 使用率高如何优化。

原因分析

一般都是由加密算法造成的，在低端芯片上更为明显。
云存和 P2P 视频传输默认都开启加密功能，云存和 P2P 视频传输目前采用的加密算法分别为 AES-CBC-128 和 AES-CTR-128。
下面给出部分加密算法在不同平台的跑分测试结果：



可以看到 AES-CBC-256 在 Hi3516E 系列的 CPU 上加密性能约为9600KB/s（AES-CBC-128 性能略高于 AES-CBC-256），假设云存视频的码率为 2mbps，即每秒的数据量大约为 256KB，计算可得 CPU 使用率约为3%，实际使用过程中受其他业务影响 CPU 使用率可能高于估算值。
P2P 视频传输采用的 AES-CTR-128 性能和 AES-CBC-128 性能相近，假设有多个用户同时向设备端拉流观看，CPU 的使用率会成倍增长，给设备端带来较大压力。
SDK 使用的 mbedtls 版本为2.16.9，用户可以自行下载对应版本并编译进行跑分测试，方法如下：
设置环境变量并编译

```
export CC="XXXXX"
export CFLAGS="-std=c99"
make
./programs/test/benchmark
```

即为性能测试程序，在设备上运行该程序查看跑分结果并估算 CPU 使用率。

解决方法

- 用户自行适配 mbedtls 的硬件加速相关接口，并替换 SDK 内默认的 mbdetls 库。
- 关闭加密功能（不推荐）。

常用工具

最近更新时间：2022-02-15 10:55:38

1. MediaInfo
   查看音视频文件的格式信息。
   下载地址
2. EasyIce
   分析 ts 视频文件或视频流。
   下载地址
3. Elecard StreamEye Tools
   分析 h264 视频文件。
   下载地址
4. Elecard HEVC Analyzer
   分析 h265 视频文件。
   下载地址
5. flvAnalyser
   Flv 分析工具。
   下载地址
6. VLC、PotPlayer、ffplay 等视频播放器。
7. mp4box
   mp4 文件分析工具。
   下载地址
8. Bento4
   mp4 文件分析工具。
   下载地址

版权所有：腾讯云计算（北京）有限责任公司

一般性问题

最近更新时间：2021-12-23 17:28:24

物联网智能视频服务（消费版）有哪些功能?

腾讯云物联网智能视频服务（消费版），帮助客户简单、快速、低成本实现设备接入、宽带传输、云端存储、远程观看等一站式视频场景能力，提供视频直播、录像回放、云端存储、边看边录、语音对讲等音视频功能，以及丰富的 AI 算法模型实现具体场景的智能解析及应用。

物联网智能视频服务（消费版）有哪些应用场景?

物联网智能视频服务（消费版）可应用于 IPC 摄像头、智能手表、智能猫眼、智能行车记录仪等智能视频类设备，实现远程看护、商业客流统计、安全驾驶等场景应用。

物联网智能视频服务（消费版）数据如何收费?

腾讯云物联网智能视频服务（消费版）采用预付费模式，对设备接入量、云存套餐使用量及算法使用数进行计费。更多详情请参见 购买指南 。

物联网智能视频服务（消费版）如何接入?

您可以参考 快速入门 文档，完成快速接入。

在物联网智能视频服务（消费版）中，数据的安全性如何获得保障?

物联网智能视频服务（消费版）通过多种视频加密技术，实现端到端链路数据加密、设备身份认证、密码保护及管理，全方位保护您的数据隐私和设备安全。

物联网智能视频服务（消费版）是否提供手机 App?

物联网智能视频服务（消费版）仅提供 App SDK，用户可通过 App SDK 完成自有 App 开发。