# Vermatrix加扰节目切台时间优化分析

在Youfone项目中遇到了Vermatrix加扰节目切台较慢的问题，节目切换需要3秒以上的时间。经过一段时间的分析和优化，将切台时间优化了1秒左右。下面就结合Youfone项目在具体分析一下LibPlayer播放Vermatrix加扰流的过程及优化点。

## 播放流程

一个完整的切台流程是指从用户按下遥控器切台键到新的节目播放出画面这个过程，可分为以下几个阶段：



## 耗时分析

结合Yonfone实际情况，可以得出各阶段耗时：

**优化前：**

05-18 09:21:14.011 D/JSLog ( 3870): MainActivity:keycode : 19

应用处理切台动作 耗时1.1s

05-18 09:21:15.181 W/WebKit ( 3870): setSingleMedia

05-18 09:21:15.187 I/HybridOTTPlayerService( 3777): HybridOTTPlayer : SetSingleMedia mediaType:1 mediaURL:32

05-18 09:21:15.194 I/MediaPlayerService( 2978): [15] stop

05-18 09:21:15.651 I/amplayer( 2978): [player\_exit:exit]pid=14

MediaPlayer停止当前播放 耗时0.5s

05-18 09:21:15.651 I/HybridOTTPlayerService(3777): HybridOTTPlayer : MediaPlayer setDataSource:vrwc:http://149.5.14.70/PLTV/88888888/224/3221226164/index.m3u8

05-18 09:21:15.718 V/AmlogicPlayer( 2978): setDataSource url=vrwc:http://149.5.14.70/PLTV/88888888/224/3221226164/index.m3u8

05-18 09:21:15.720 I/M3uSession( 2978): Open baseUrl :http://149.5.14.70/PLTV/88888888/224/3221226164/index.m3u8

05-18 09:21:15.761 I/M3uSession( 2978): start fetch segment file,url:http://149.5.14.130:80/PLTV/88888888/224/3221226164/1495092039-3-149509204. hls.ts,seq:149509204, first=149509204

05-18 09:21:16.411 I/amffmpeg( 2978): new blocks426 len 320352

下载320k数据 耗时0.7s

05-18 09:21:16.412 D/VCASCommunication( 2978): Decrypt-> dataSize 320352 keyfileURL https://149.5.14.29/CAB/keyfile?s=8315&r=10099&t=DTV&p=1495068190 iv0x0 0x0 !!!

05-18 09:21:16.813 I/amffmpeg( 2978): [vmx\_vrwc\_read] after crypt :c->inbuffer0xeced1008 c->in\_readptr 0xecf1f368, c->in\_writeptr 0xecf53a12

解扰 耗时0.4s

05-18 09:21:16.814 I/amffmpeg( 2978): Format mpegts probed with size=2048 and score=100

05-18 09:21:16.842 I/amffmpeg( 2978): [av\_find\_stream\_info]return

解析音视频格式

05-18 09:21:16.842 I/amplayer( 2978): [video\_codec\_type\_convert]VIDEO\_DEC\_FORMAT\_H264(0x1c)

05-18 09:21:17.018 I/amffmpeg( 2978): new blocks459 len 345168

05-18 09:21:17.018 D/VCASCommunication( 2978): Decrypt-> dataSize 345168 keyfileURL https://149.5.14.29/CAB/keyfile?s=8315&r=10099&t=DTV&p=1495068190 iv0xcc 0xc3 !!!

05-18 09:21:17.204 I/amffmpeg( 2978): [vmx\_vrwc\_read] after crypt :c->inbuffer0xeced1008 c->in\_readptr 0xecf737b8, c->in\_writeptr 0xecf9096c

05-18 09:21:17.204 I/amplayer( 2978): pid[15]::decoder prepare

视频解码器初始化 耗时0.3s

05-18 09:21:17.324 I/amadec ( 2978): audiodec\_init!

05-18 09:21:18.188 I/amadec ( 2978): android out start

音频解码器初始化 耗时0.8s

05-18 09:21:18.301 I/amadec ( 2978): enable slowsync repeate.

05-18 09:21:18.301 I/amadec ( 2978): audio pts start from 0x6adfda53

起播完成 共耗时4.2s

**优化后：**

06-07 12:15:02.109 D/JSLog ( 4435): MainActivity:keycode : 19

应用处理切台动作 耗时1.2s

06-07 12:15:03.199 W/WebKit ( 4435): setSingleMedia

06-07 12:15:03.200 I/HybridOTTPlayerService( 3751): HybridOTTPlayer : SetSingleMedia mediaType:1 mediaURL:31

06-07 12:15:03.210 I/MediaPlayerService( 3017): [25] stop

06-07 12:15:03.418 I/amplayer( 3017): [player\_exit:exit]pid=24

MediaPlayer停止当前播放 耗时0.2s

06-07 12:15:03.418 I/HybridOTTPlayerService( 3751): HybridOTTPlayer : MediaPlayer setDataSource:vrwc:http://149.5.14.70/PLTV/88888888/224/3221226207/index.m3u8

06-07 12:15:03.463 V/AmlogicPlayer( 3017): setDataSource url=vrwc:http://149.5.14.70/PLTV/88888888/224/3221226207/index.m3u8

06-07 12:15:03.466 I/M3uSession( 3017): Open baseUrl :http://149.5.14.70/PLTV/88888888/224/3221226207/index.m3u8

06-07 12:15:03.518 I/M3uSession( 3017): start fetch segment file,url:http://149.5.14.133:80/PLTV/88888888/224/3221226207/1496830474-3-249471745. hls.ts,seq:249471745, first=249471745

06-07 12:15:03.830 I/amffmpeg( 3017): new blocks221 len 166192

下载160k数据 耗时0.3s

06-07 12:15:03.830 D/VCASCommunication( 3017): Decrypt-> dataSize 166192 keyfileURL https://149.5.14.29/CAB/keyfile?s=8315&r=10756&t=DTV&p=1496830200 iv0x0 0x0 !!!

06-07 12:15:04.215 I/amffmpeg( 3017): [vmx\_vrwc\_read] after crypt :c->inbuffer0xeccdb008 c->in\_readptr 0xecd03938, c->in\_writeptr 0xecd37e48

解扰 耗时0.4s

06-07 12:15:04.217 I/amffmpeg( 3017): Format mpegts probed with size=2048 and score=100

06-07 12:15:04.218 I/amffmpeg( 3017): new blocks284 len 213568

06-07 12:15:04.218 D/VCASCommunication( 3017): Decrypt-> dataSize 213568 keyfileURL https://149.5.14.29/CAB/keyfile?s=8315&r=10756&t=DTV&p=1496830200 iv0xb7 0xf0 !!!

06-07 12:15:04.250 I/amffmpeg( 3017): [av\_find\_stream\_info]return

解析音视频格式

06-07 12:15:04.250 I/amplayer( 3017): [video\_codec\_type\_convert]VIDEO\_DEC\_FORMAT\_H264(0x1c)

06-07 12:15:04.400 I/amplayer( 3017): pid[25]::decoder prepare

视频解码器初始化 耗时0.2s

06-07 12:15:04.400 I/amffmpeg( 3017): [vmx\_vrwc\_read] after crypt :c->inbuffer0xeccdb008 c->in\_readptr 0xecd37b78, c->in\_writeptr 0xecd4f1d8

06-07 12:15:04.404 I/amadec ( 3017): audiodec\_init!

06-07 12:15:04.446 I/amadec ( 3017): android out start

音频解码器初始化 耗时0.04s

06-07 12:15:04.565 I/amadec ( 3017): enable slowsync repeate.

06-07 12:15:04.565 I/amadec ( 3017): audio pts start from 0xd88c5c48

起播完成 共耗时2.5s

## 优化方案

通过以上数据对比，可以看到优化主要有以下几点：

1. **停止播放时间优化**

原来停止播放需0.5s到0.7s，主要原因是由于stop命令与read接口为同步指令，必须等待read接口返回后stop命令才能执行。而read接口又需缓冲一定数据并完成解扰后才会返回。

现在采用异步命令通知的模式，不必等待read接口下载足够缓冲并解扰后返回才能stop，可通知read接口立刻返回，将停止播放时间优化到0.2左右。

具体修改：

在crypto\_vmx\_vrwcimpl.c中添加.url\_setcmd接口，用于立即中断read接口。

static int vmx\_vrwcimpl\_setcmd(URLContext \*h, uint32\_t cmd, uint32\_t flag, void \*info)

{

VRWCCryptoContext \*c = h->priv\_data;

URLContext\* hd = (URLContext\*)(c->hd);

if (1 == cmd) {

c->isReadStart = 0;

}

return 0;

}

1. **音视频格式快速解析优化**

原来mpeg-ts解析器需缓冲256K以上数据才会对数据包头进行解复用，解扰器缓冲为320K，下载器需下载320K以上的数据才能开始解扰，然后mpeg-ts解析器才开始接复用，这个过程需耗时1.1s左右。

现在将mpeg-ts解析器缓冲数据改为128k，解扰缓冲修改为160K，下载器下载160K的数据就可以解扰继而解复用。该过程耗时优化到0.7s左右。

具体修改：

在crypto\_vmx\_vrwcimpl.c 修改解扰缓冲

#define VMX\_DECRYPT\_SIZE (VMXBLOCKSIZE \* 200)

在mpegts.c 修改解析头缓冲

接口 mpegts\_read\_header：

//handle\_packets(ts, 256\*1024 / ts->raw\_packet\_size);

handle\_packets(ts, 128\*1024 / ts->raw\_packet\_size);

1. **音视频慢同步优化**

原来音视频需等待PTS同步，然后才可以开始播放。

现在改用慢同步模式，视频解析完成了直接开始播放，不必等待音频数据同步。

在player\_para.c 修改慢同步

接口player\_startsync\_set：

//set\_sysfs\_int(slowsync\_repeate\_path, 0);

set\_sysfs\_int(slowsync\_repeate\_path, 1);

1. **下载解扰优化**

如优化方案2所提到的，现在将mpeg-ts解析器缓冲数据改为128k，解扰缓冲修改为160K，可以减少起播所需的解扰次数，配合方案3，减少了起播所需的下载数据量和解扰次数，对起播速度起到了显著的优化。

## 总结

本次优化的几个关键点分别为停止播放耗时、下载耗时和音视频同步耗时。其中下载耗时本质原因是服务器限速导致，停止播放耗时受Verimatrix4.1解扰慢影响较大，解码器初始化及同步慢很大程度也是受网速和解扰的影响。通过下载解扰异步、修改解扰缓冲、数据解析缓冲等参数，一定程度上缓解了网速和解扰慢带来的影响。

通过这次对起播时间的优化，对LibPlayer的起播流程有了更清晰的了解，也学习到了很多调试手段，对以后的工作有非常大的帮助。