

直播流媒体协议 技术分享

梁延彬

- 一、直播业务流媒体播放的场景

- 1、直播流的播放

- 2、直播回放的播放

主播结束直播



服务端开始转录



生成回放视频文件

二、遇到的问题

问题：回放视频加载慢，起播时间达到**20 ~ 30s**，用户等待时间长

原因：发现**mp4**格式在播放流媒体方面存在天然瓶颈，很难提升。

解决方案：研发通过技术调研，发现**HLS**协议加载更快。将转录视频格式由**mp4**切换为**HLS**协议的**m3u8**格式，更换后，起播时间减少到**3 ~ 5s**，用户体验得到很大提升。

接下来的内容：通过流媒体协议的分析对比，对直播业务的播放优化进行理论支持

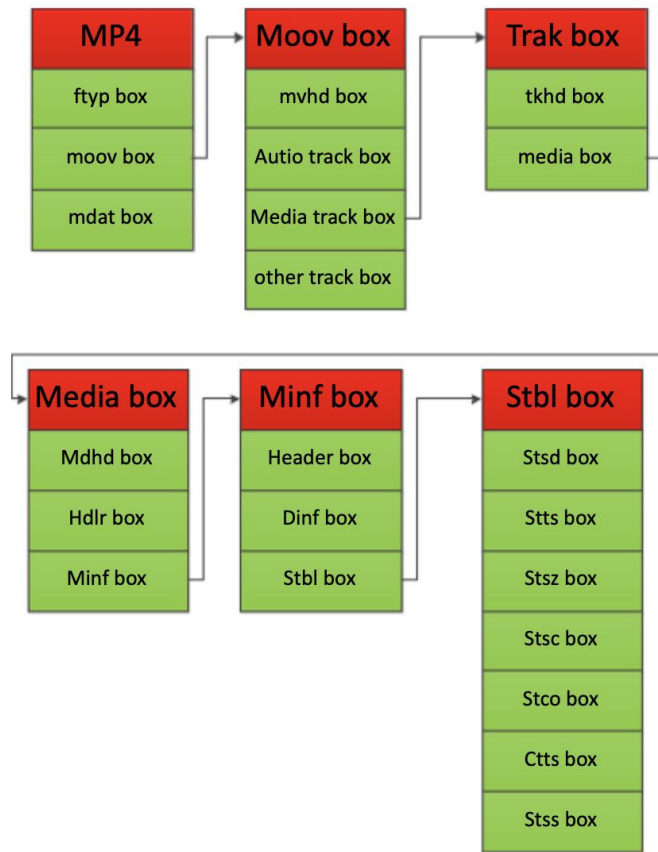
● 三、点播场景：支持边下载边播放的流媒体协议

视频协议	来源	视频流格式	特点	通讯协议	时延	带宽占用情况	项目使用	支持的播放器
HLS	苹果公司	TS和M3U8	<ul style="list-style-type: none">•对视频进行切片，按切片播放，缓存小、起播快；•拖动时间轴到任意时间播放时，可以快速定位到对应的切片进行播放，响应快。	HTTP	直播：5 ~ 20s 点播：<5s	小 一个切片	直播回放	Google ExoPlayer ijkplayer
DASH	国际标准		<ul style="list-style-type: none">•自适应比特率流技术，类似HLS	HTTP		小		Google ExoPlayer
MP4	MPEG-4 标准第 14 部分		<p>头文件较大，边下边缓存，起播相对HLS和DASH慢；</p> <p>拖动时间轴播放时，需要一定的时间缓存；</p> <p>市场上大多数的浏览器客户端均能够播放，播放成功率高。</p>	HTTP		大 拖动时间轴播放时，仍然需要下载整个头文件，耗费流量大；因流量占用较大， 建议用在短视频播放的场景。		原生 MediaPlayer Google ExoPlayer ijkplayer

● 四、mp4和HLS对比

1、mp4为什么起播慢

- **ftyp**: 一个mp4文件有且只有一个 ftyp box , 在MP4文件的开始位置, 描述的文件的版本、兼容协议。
- **moov**: 一般跟随在ftyp之后, 有且只有一个。其不包含具体媒体数据, 但包含本文件中所有媒体数据的宏观描述信息。
- (视频创建时间、视频修改时间、播放速率、视频时长、音量大小、视频宽高、字幕语言、声道、视频/音频帧位置 等)
- **mdat (Media Data Box)** : MP4文件的媒体数据存放在这里。mdat中的数据帧依次存放, 每个帧的位置、时间、长度都由moov中的信息指定。
- mdat Box 基本上占据了视频大小的 **95%** 以上, 得益于 mp4 边下边播的效果, 浏览器获取到了部分 mdat box, 就可以进行播放。



box 类型						说明
ftyp						file type, 文件类型
moov						metadata container, 存放媒体信息
	mvhd					movie header, 文件的总体信息, 如时长, 创建时间等
	trak					track container, 存放音频、视频流信息的容器
		tkhd				track header, track的总体信息, 如时长, 宽高等
		mdia				track media information container
			mdhd			media header, 存放TimeScale, trak需要TimeScale换算真实时间
			hdlr			handler, 指定trak类型是video/audio/hint
			minf			media information container
				stbl		sample table box 包含样本序号/时间/文件位置映射的信息
					stsd	sample descriptions
					stts	decoding time to sample, DTS-sample序号的映射表
					ctts	composition time to sample CTS(创作时间)-DTS对应sample序号的映射表
					stsc	sample-to-chunk, sample和chunk的映射表
					stsz/stz2	sample size, 每个sample的大小
					stss	sync sample table, 关键帧列表
					stco/co64	chunk offset, 每个chunk的文件偏移
mdat						media data container, 具体的媒体数据

● 四、mp4和HLS

1、mp4为什么加载慢

结论：普通MP4文件播放时，**ftyp**与**moov box**需同时加载完成后，并下载部分**mdat box**的帧数据后，才能开始播放。

结构不规范的mp4文件，**moov box**在文件最后，这种情况整个视频文件下载完成后才能播放！

mp4info-v1.7 [stavy.sun@hotmail.com]

mp4info | mp4tool |

audio:	mp4a	total_time:	132586--[132000]
smprate:	44100	duration:	132586
channel:	2	timescale:	1000
bitrate:	64017	maxaudiosize:	546
		audiosamplenum:	2855
video:	avc1	maxvideosize:	110742
width:	1280	videosamplenum:	3965
height:	720		
bitrate:	1536469		
frames:	3965		
fps:	30.000151		

ROOT

- ftyp
- moov
 - mvhd
 - trak
 - tkhd
 - edts
 - elst
 - mdia
 - mdhd
 - hdlr
 - minf
 - vmhd
 - dinf
 - dref
 - url
 - stbl
 - stsd
 - avc1
 - avcC
 - stts
 - stss
 - ctts
 - stsc
 - stsz
 - stco
- trak
 - udta
 - meta
 - hdlr
 - ilst
 - ? oo
 - data
- free
- mdat

● 四、mp4和HLS

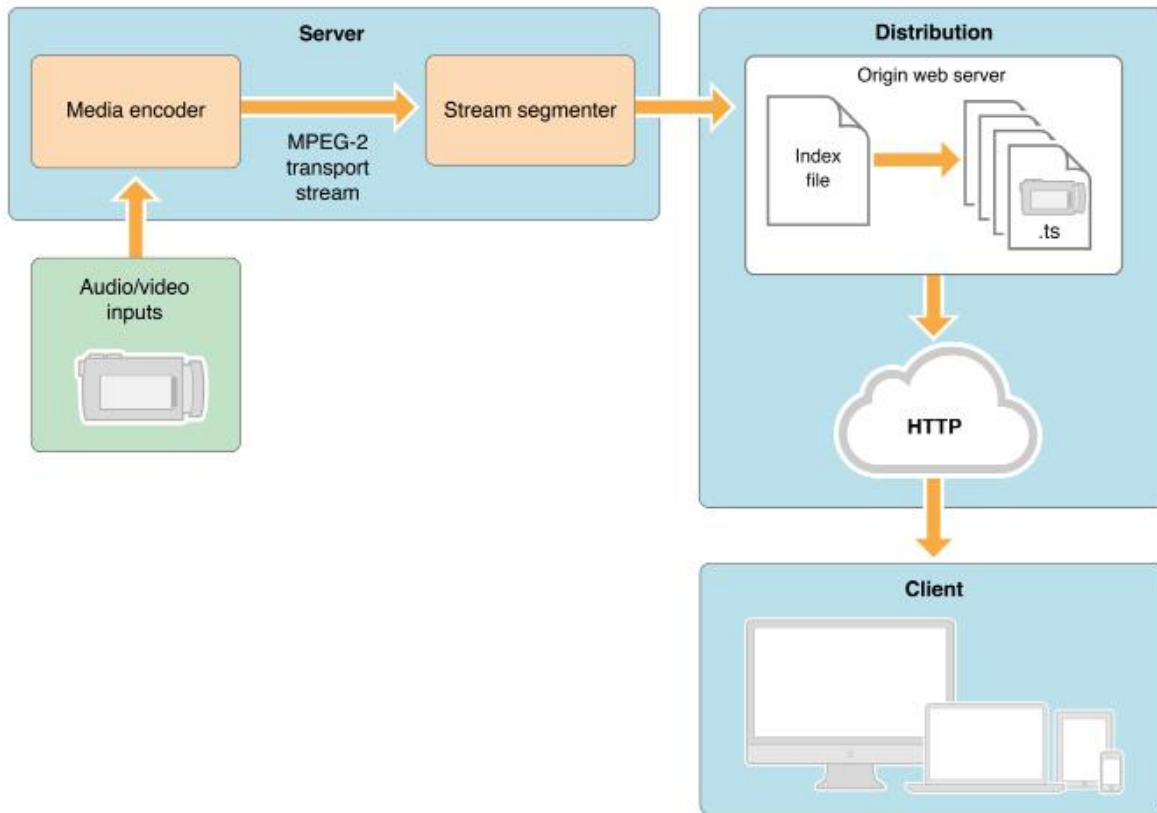
HLS为什么加载快

• **HLS 组成:** HTTP、M3U8、TS (mp4)

- 1、**HTTP** 是传输协议
- 2、**M3U8** 是索引文件
- 3、**TS** 是音视频的媒体信息。

• **服务端制作步骤:**

- 1、**Media encoder**转码模块将视频源中的视频数据转码到目标编码格式 (**H264**) 的视频数据
- 2、**stream segmenter**切片模块将视频切片, 切片的结果就是**index file** (m3u8) 和**ts**文件



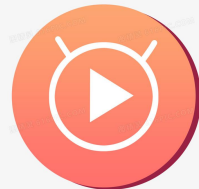
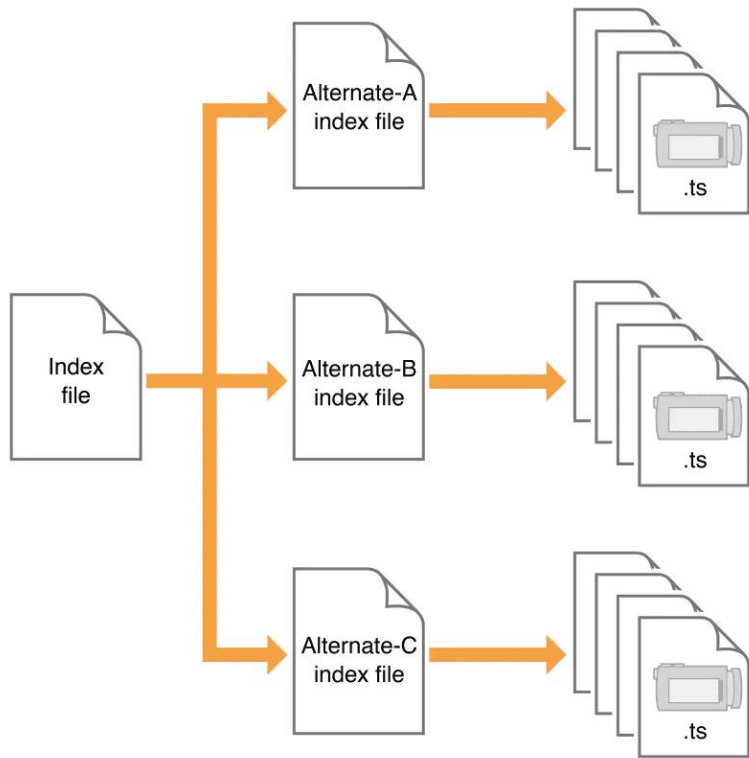
● 四、mp4和HLS

对比 HLS为什么加载快

● 客户端播放步骤:

- 1、下载一级Index file, 它里面记录了二级索引文件 (Alternate-A、Alternate-B、Alternate-C) 的地址。
- 2、下载二级索引文件, 二级索引文件中又记录了TS文件的下载地址。
- 3、按顺序下载TS视频文件并连续播放。

结论: HLS下载完一个切片 ts 或 mp4文件就可以开始播放





● 五、支持直播流播放流媒体协议

视频协议	来源	视频流格式	特点	通讯协议	时延	带宽占用情况	项目使用	支持的播放器
RTSP	Real Networks 和 Netscape2	TS、MP4	<ul style="list-style-type: none">•实时流传输协议•但技术实现复杂	UDP/TCP	几百毫秒			Ijkplayer ExoPlayer
RTMP	Adobe	FLV、F4V	<ul style="list-style-type: none">•实时消息传输协议•把数据信息拆分成一个个小信息块 (Chunk) 传输	TCP	<3s	小一个信息块	直播拉流	Ijkplayer ExoPlayer
HLS	苹果公司	TS和M3U8	<ul style="list-style-type: none">•对视频进行切片，按切片播放，缓存小、起播快；•拖动到任意时间播放时，可快速定位到对应的切片进行播放，响应快。•延时主要是服务器切片时间，客户端下载索引和切片视频的时间•直播中索引是动态更新的，客户端要重复下载•点播因为已经完成所有视频切片，可以一次获取所有索引，延时更小	HTTP	直播： 5 ~ 20s 点播： <5s	小一个切片	直播回放	ExoPlayer Ijkplayer
LHLS	尚未被加入标准协议	TS和M3U8	<ul style="list-style-type: none">•HLS 用于直播的民间优化版本•切片流程优化：服务器提前下发切片中和下一个即将切片的地址	HTTP	直播： 5 ~ 10s	小一个切片		



● 五、支持直播流播放流媒体协议

视频协议	来源	视频流格式	特点	通讯协议	时延	带宽占用情况	项目使用	支持的播放器
DASH	国际标准		•自适应比特率流技术，类似HLS	HTTP		小		ExoPlayer ijkplayer
Smooth Streaming	微软公司		•是IIS的媒体服务扩展，用于支持基于HTTP的自适应流。	HTTP		小		
HDS	Adobe		•通过HTTP向Adobe Flash Player或Adobe AIR提供多比特率内容。	HTTP				



六、流媒体播放器介绍

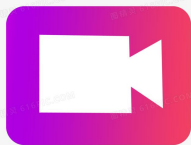
1、ijkplayer

- 项目地址: <https://github.com/Bilibili/ijkplayer>
- 出品方: Bilibili
- 介 绍

ijkplayer 是Bilibili发布的基于 **FFplay** 的轻量级 **Android/iOS** 视频播放器。

- 实现了跨平台功能, **API** 易于集成;
- 编译配置可裁剪, 方便控制安装包大小;
- 支持硬件加速解码, 更加省电;
- 提供 **Android** 平台下应用弹幕集成的解决方案





六、流媒体播放器介绍: ExoPlayer

项目地址: <https://github.com/google/ExoPlayer>

出品方: Google

优点:

- 1、支持HTTP动态自适应流媒体 (**DASH**) 和**SmoothStreaming** (在MediaPlayer上都不支持), 并支持**HLS**协议和**TS**流的播放。
- 2、不同版本**兼容性好**, 不会由于不同设备和Android版本间的变化而出现问题, 更加的稳定;
- 3、是独立的库, **体积小**、升级方便;
- 4、支持**自定义扩展**, 支持FFMPEG扩展;
- 5、支持播放列表功能, 使得音视频可以无缝播放、支持**剪辑**和**合并播放**功能;
- 6、在Android 4.4 (API级别19) 及更高版本上支持Widevine通用加密;
- 7、支持快速集成; 8、支持字幕; 9、支持媒体下载。

缺点:

对于某些设备上的纯音频播放, ExoPlayer可能比MediaPlayer消耗更多的电量。

