



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2017

十亿级视频播放技术优化揭秘

王辉

自我介绍

- 王辉，来自腾讯，09年起从事QQ空间技术研发。
- 近期主要关注手机短视频、视频直播、AI智能硬件。
- 热爱跑步。



风起

在Facebook核心平台上，每天超过**80亿**段视频被观看，短视频也给Facebook带来了强劲的收入。

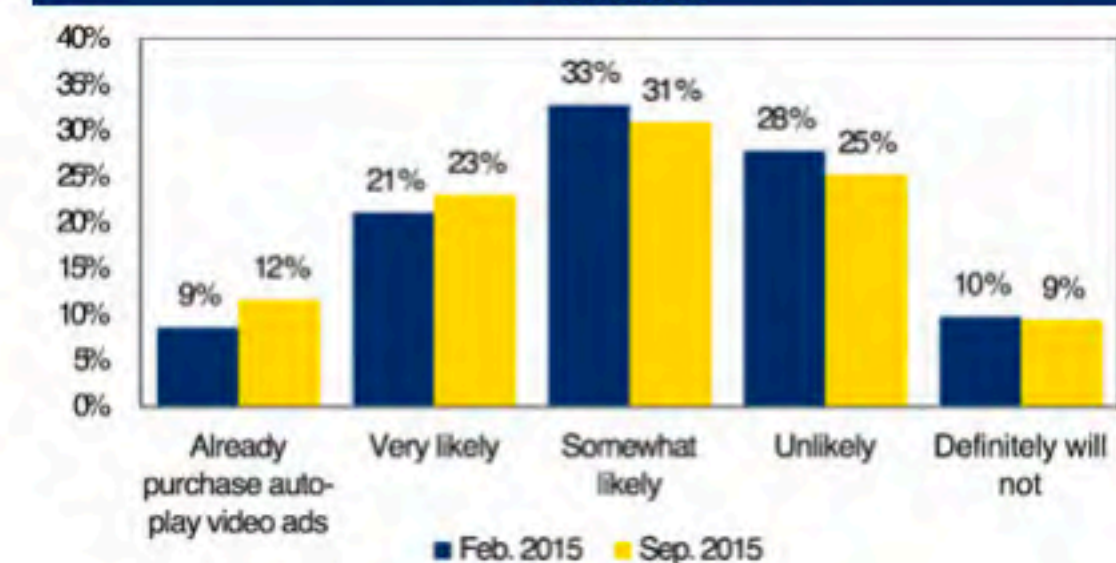
2) Autoplay Video Ads ... Potentially Well Over \$1.5B in '16 Revenue



About Autoplay Video Advertisements

- Videos have been increasingly incorporated into Facebook news feeds over the past year or two
- Video advertisements have increased in frequency in newsfeeds, replacing some static ad inventory
 - Typically run for about 15 seconds on mute, with sound coming on when clicked
- More than 8B daily videos watched on the core Facebook platform

What is the likelihood you buy Autoplay Video Ads within 6 months?

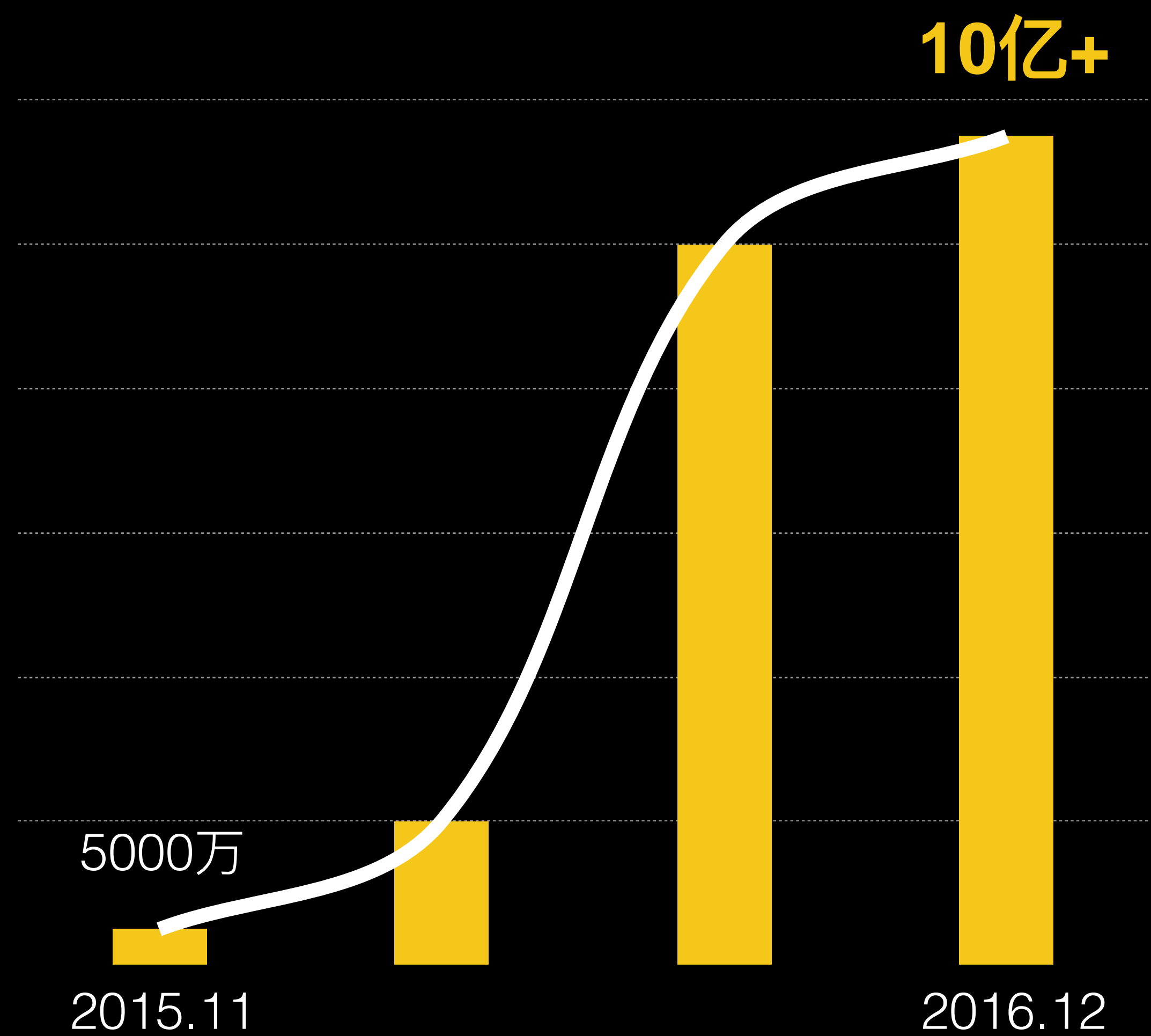


Source: Company reports, RBC Capital Markets estimates, SurveyMonkey, n=405 in September 2015

(#)

RBC Capital Markets

QQ空间数据



Agenda

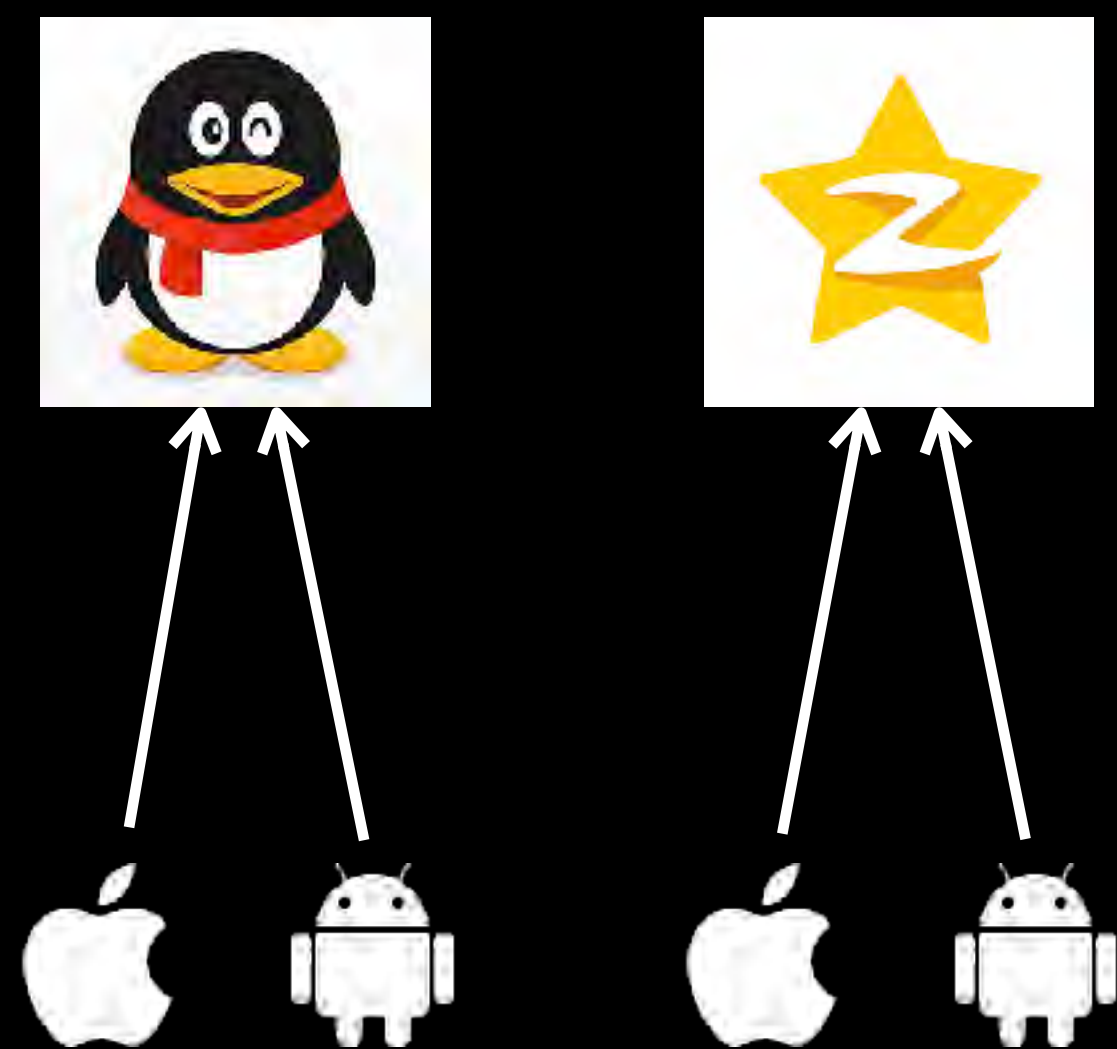


快速上线

体验统一，监控完善



项目背景



- 4个团队维护4端代码
- 4*N套代码逻辑，效率低。

项目背景



不自动播放



H5播放



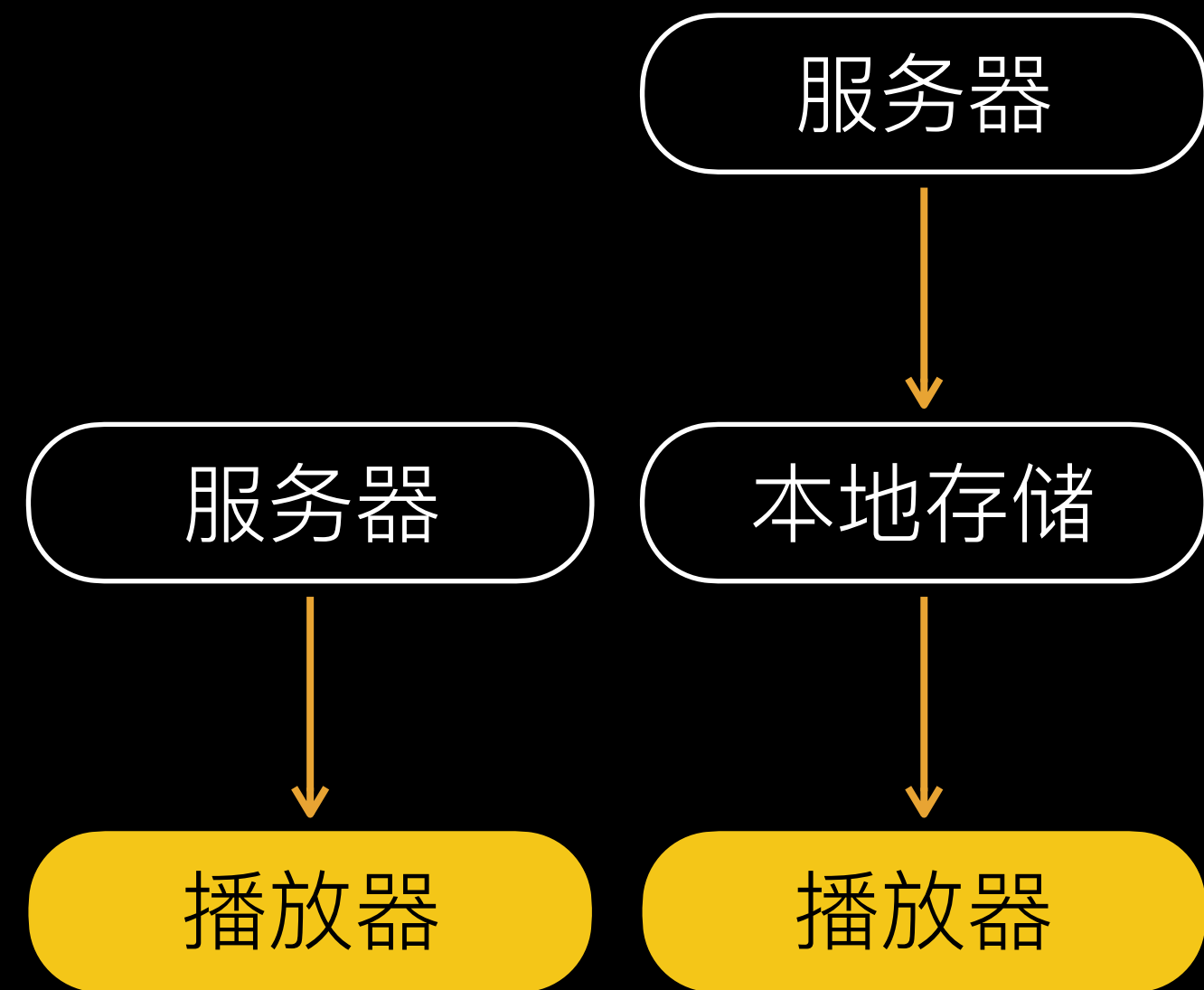
外部播放器



系统播放器

- 观看体验不一致
- 观看体验差

项目背景



短视频老架构

- 播放和业务耦合，扩展性差
- 播放成功率低、失败原因未知
- 不支持边下边播，缓冲时间长
- 流量浪费严重

核心产品要点



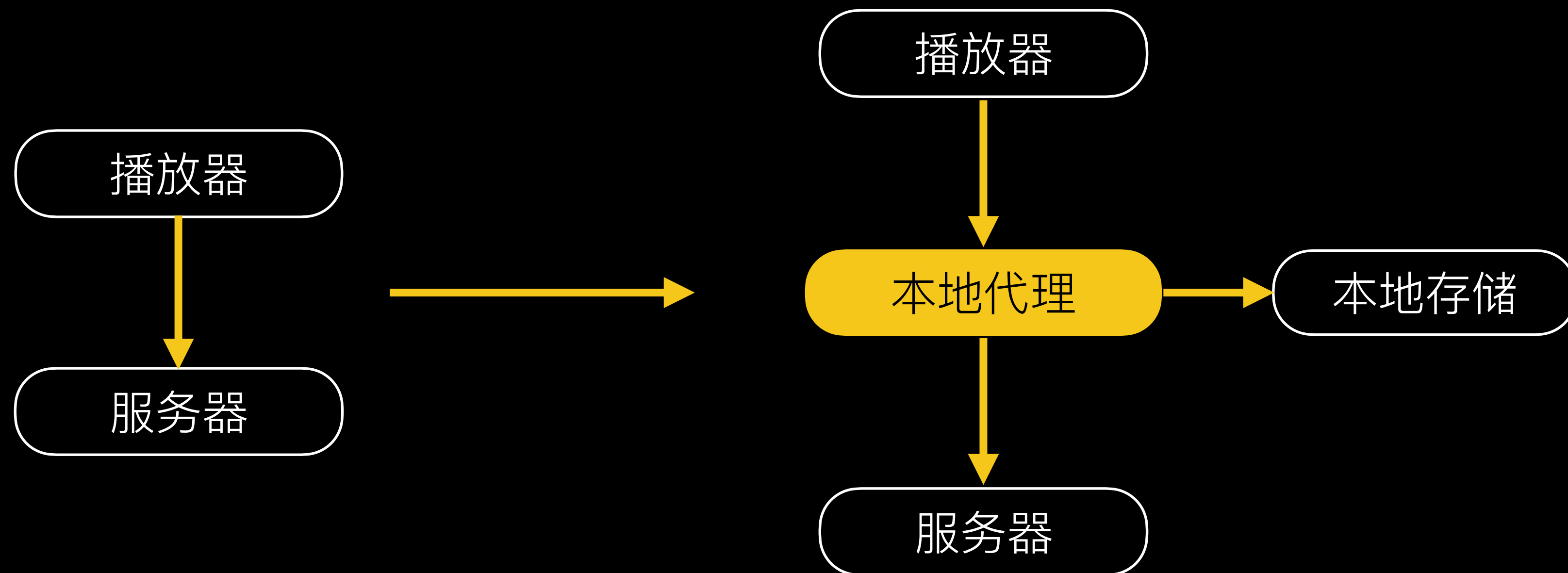
统一的播放体验为项目扫清了非常多的阻碍。

核心技术要点

- 边下边播，加快播放速度
- 流量控制，具备云控制能力，让带宽消耗可控
- 与业务合理解耦
- 模块化复用，统一播放逻辑，跨平台
- 支持第三方视频源，比如来自腾讯视频的视频源
- 监控上报，多维度监控，对业务知根知底

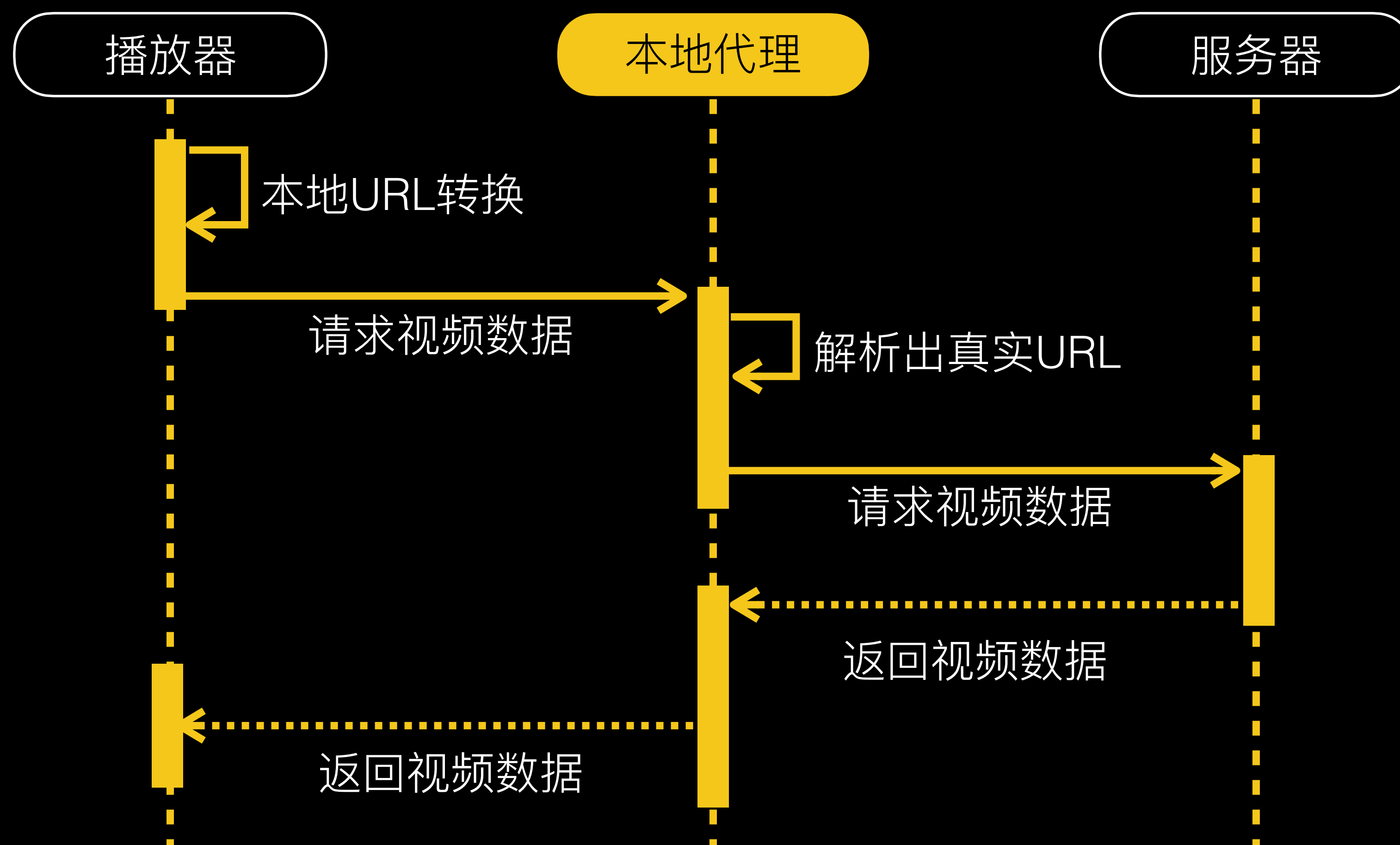
如何控制下载

核心：播放器直连转变为本地代理服务器中转



如何控制下载

播放器与服务器之间的数据往来本地代理可以完全把控。



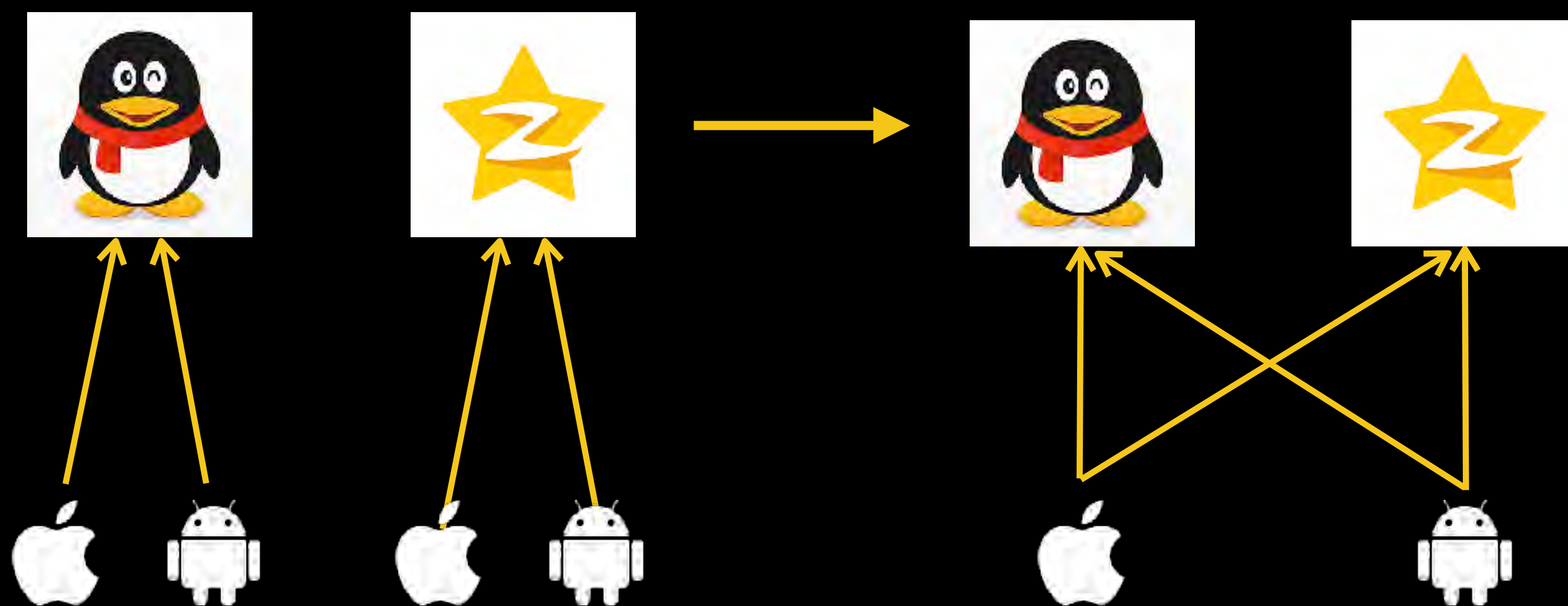
统一架构



VideoController完全与系统无关，可以放到C层做跨平台复用。



团队职能优化



- 按FT划分团队
- 一套代码复用两个APP
- 一套下载模块复用四端
- 团队更专注

监控完善

一个成熟项目必需和必备的要素。

目的：

- 问题定位
- 耗时统计
- 成功率统计
- 实时告警

问题定位——错误归类

- 传统的捞log分析的方式，效率太低。
- 错误归类，每一类错误都有唯一的错误码进行识别，方便快速定位问题。

63	-800000000	-6000	0	启动播放60秒后没有开始播放
64	-800100000	-6001	0	setDataSource IOException
65	-800200000	-6002	0	prepareAsync IllegalState Exc
66	-800300000	-6003	0	网络提示后调用prepareAsync
67	-800400000	-6004	0	启动播放其它异常
68	-199900019	1	-19	系统播放器报-19，但播放实际
69	-199901010	1	-1010	
70	-200000000	0	0	what和extra都是0，但抛了onl
71	下载错误引起的异常			
72	-499799999	-1000	-199799999	下载数据contentType不对引起
73	-499399999	-1000	-199399999	下载其它HttpDataSourceExce
74	-450019995	-1000	-150019995	无网络引起的播放失败（腾讯视
75	-450019976	-1000	-150019976	发出请求错误
76	-499199999	-1000	-199199999	无网络引起的播放失败（兜底
77	-450019997	-1000	-150019997	DNS解析失败（腾讯视频下载
78	-459799999	-1000	-159799999	
79	-499099999	-1000	-199099999	http头正确的情况下，返回非法
80	-459719983	-1000	-159719983	下载错误码-159719983，表示
81	-459619983	-1000	-159619983	http 404,资源不存在
82	-449899999	-1000	-149899999	下载错误码-149899999，服务
83	-499599999	-1000	-199599999	下载错误码-199599999，无法
84	-459019983	-1000	-159019983	28618
85	-450019995	-1000	-150019995	连接失败，无网络（腾讯视频
86	-459619983	-1000	-159619983	http 404,资源不存在
87	-498699999	-1000	-198699999	下载错误码 -198699999 Sock
88	-509299999	-1100	-199299999	防盗连超时
89	-509299914	-1100	-199299914	changekey返回较慢逻辑问题
90	-509300775	-1100	-199300775	换ckey/vkey接口调用失败
91	-508599999	-1100	-198599999	因无网络导致的防盗链失败

问题定位——上报流水

- 将一次视频播放的全过程，设置为一组流水。
- 将过程中遇到的所有的关键事件（如下载，首次缓冲，seek次数等等）定义为单条流水，流水字段中包含关键事件的各项参数（如执行数据，结果，错误码等）。

<u>host_uin</u>	<u>qua</u>	<u>network_type</u>	<u>client_time</u>	<u>time_cost</u>	<u>event_id</u>
1284878718	V1_AND_SQ...	WIFI	14840641730...	\N	video_resolution
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_download_size
1284878718	V1_AND_SQ...	WIFI	14840641730...	1	video_start_play
1284878718	V1_AND_SQ...	WIFI	14840641750...	2.028	video_first_buffer_time_consum
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_play_ret
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_play_occur_buffer_times
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_download_ret
1284878718	V1_AND_SQ...	WIFI	14840641758...	2105	video_real_start
1284878718	V1_AND_SQ...	WIFI	14840641758...	0.001	video_first_play
1284878718	V1_AND_SQ...	WIFI	14840641758...	0.0	video_update_vkey
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_download_speed
1284878718	V1_AND_SQ...	WIFI	14840641758...	0.0	video_buffer_time_consuming
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_total_time
1284878718	V1_AND_SQ...	WIFI	14840641758...	\N	video_seek_occur_times

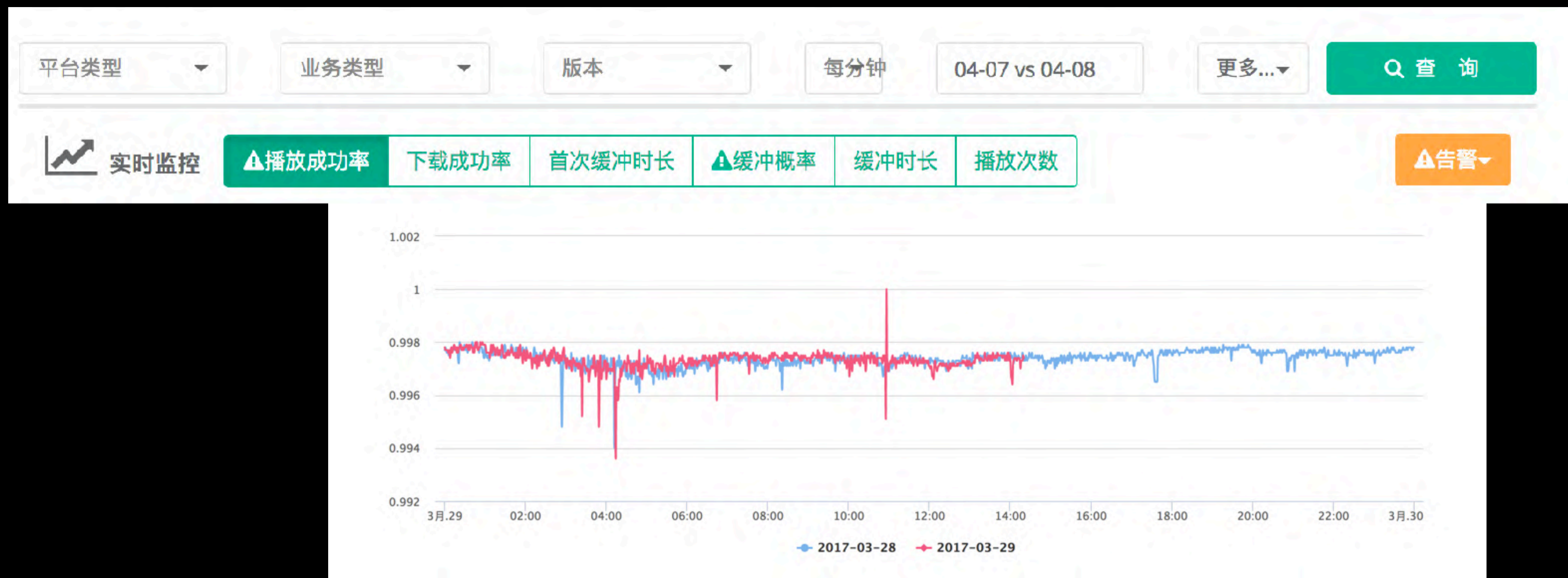
问题定位——错误统计

时间	平台	版本	错误码	错误次数	错误占比	负责人	错误描述
2017-03-28	android独立版	7.0.1.288	-800000000		66.7%		40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.0.1.288	-460019983		33.3%		腾讯视频删除或者审不通过通过通知漏过，已经让腾讯视频修改
2017-03-28	android独立版	7.1.5.288	-800000000		29.3%		40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.1.1.288	-173901003		20%		ERROR_CANNOT_CONNECT
2017-03-28	android独立版	7.1.1.288	-800000000		20%		40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.1.1.288	-203800000		20%		偶现播放器非法操作
2017-03-28	android独立版	7.1.1.288	-499599999		20%		播放错误码，对应的下载错误码-199599999，无法连接腾讯视频下载服务器
2017-03-28	android独立版	7.1.1.288	-2347383648		20%		播放器报ERROR_UNKNOWN
2017-03-28	android独立版	7.1.5.288	-2347383648		20%		播放器报ERROR_UNKNOWN
2017-03-28	android独立版	7.1.5.288	-2347379543		12%	请更新	请更新

对错误码进行分析统计，对每个版本的TOP错误可以进行针对性解决和跟进解决进展。

实时统计

对播放成功率、下载成功率、缓冲耗时、缓冲概率、播放次数都有实时的统计曲线。



实时告警

DLP告警(ROOT)

[分析 | <http://dlp.isd.com/index.php/rootdlpresult/view?dlpid=2080&mdate=2017-03-23%2001:26:00>]DLP:01:26 Android
独立版下载成功率(未恢复)PushTime:01:37
分析内容:
主调::
被调::
03/23

DLP告警(ROOT)

[分析 | <http://dlp.isd.com/index.php/rootdlpresult/view?dlpid=2071&mdate=2017-03-23%2001:26:00>]DLP:01:26 Android
独立版播放成功率(未恢复)PushTime:01:37
分析内容:
主调::
被调:::

中国联通 00:14 54%

微信 手机运维告警号

<http://dlp.isd.com/index.php/rootdlpresult/view?dlpid=2073&mdate=2017-03-23%2004:37:00>

DLP:04:37 Android结合版下载成功率(未恢复)PushTime:04:50
分析内容:
主调::
被调::
MSNG:http://moa.tencent.com/cal/?act=open&app=msng&args=args%3Ddlp_198790
PC:<http://dlp.isd.com/index.php/rootdlpresult/view?dlpid=2078&mdate=2017-03-23%2004:37:00>

DLP:04:37 Android结合版播放成功率(未恢复)PushTime:04:50
分析内容:
主调::
被调::
MSNG:http://moa.tencent.com/cal/?act=open&app=msng&args=args%3Ddlp_198791
PC:<http://dlp.isd.com/index.php/rootdlpresult/view?dlpid=2069&mdate=2017-03-23%2004:37:00>

多维分析 [统计时间: 2017-03-24 04:26 ~ 04:52] 错误码 省份 城市 平台 业务

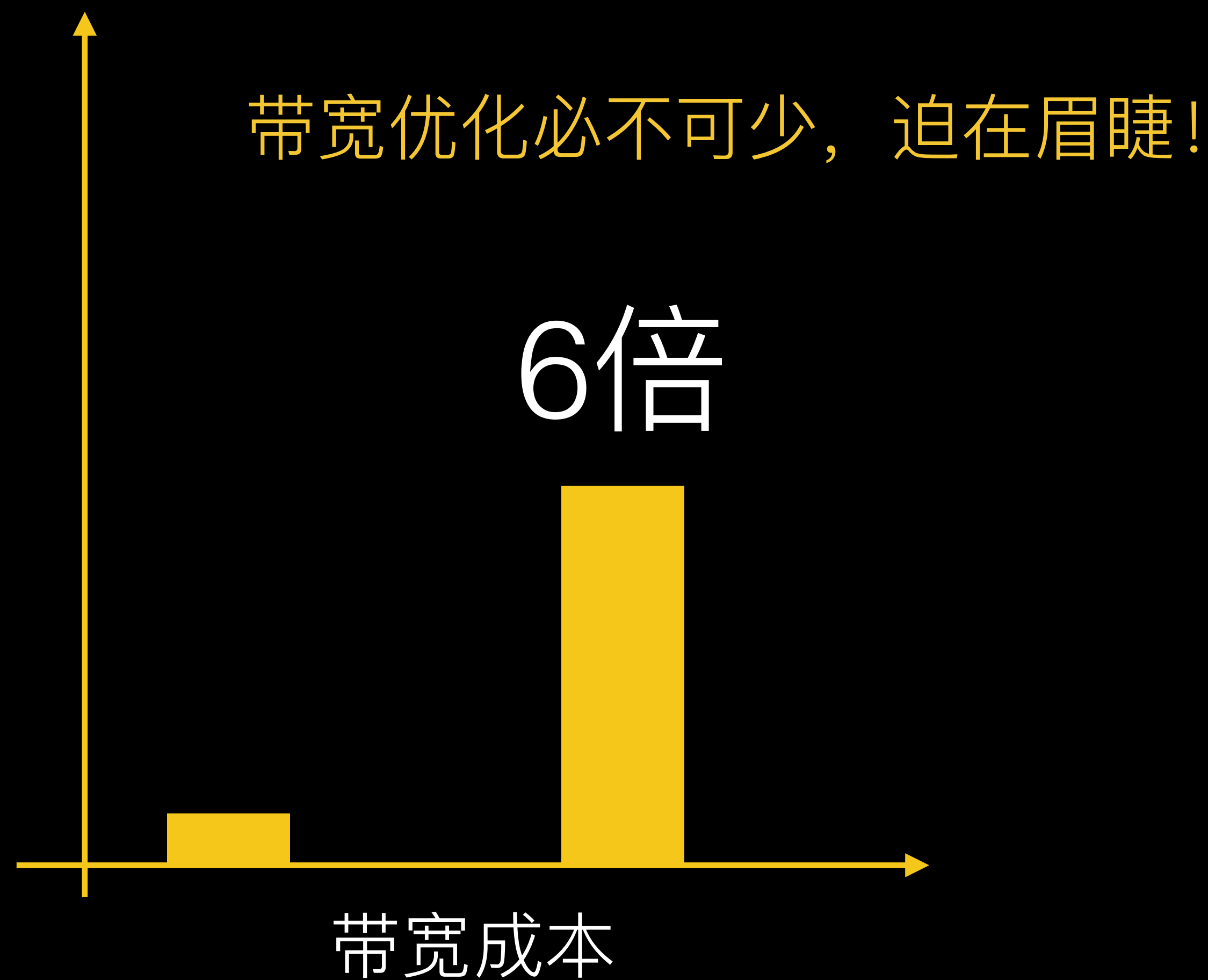
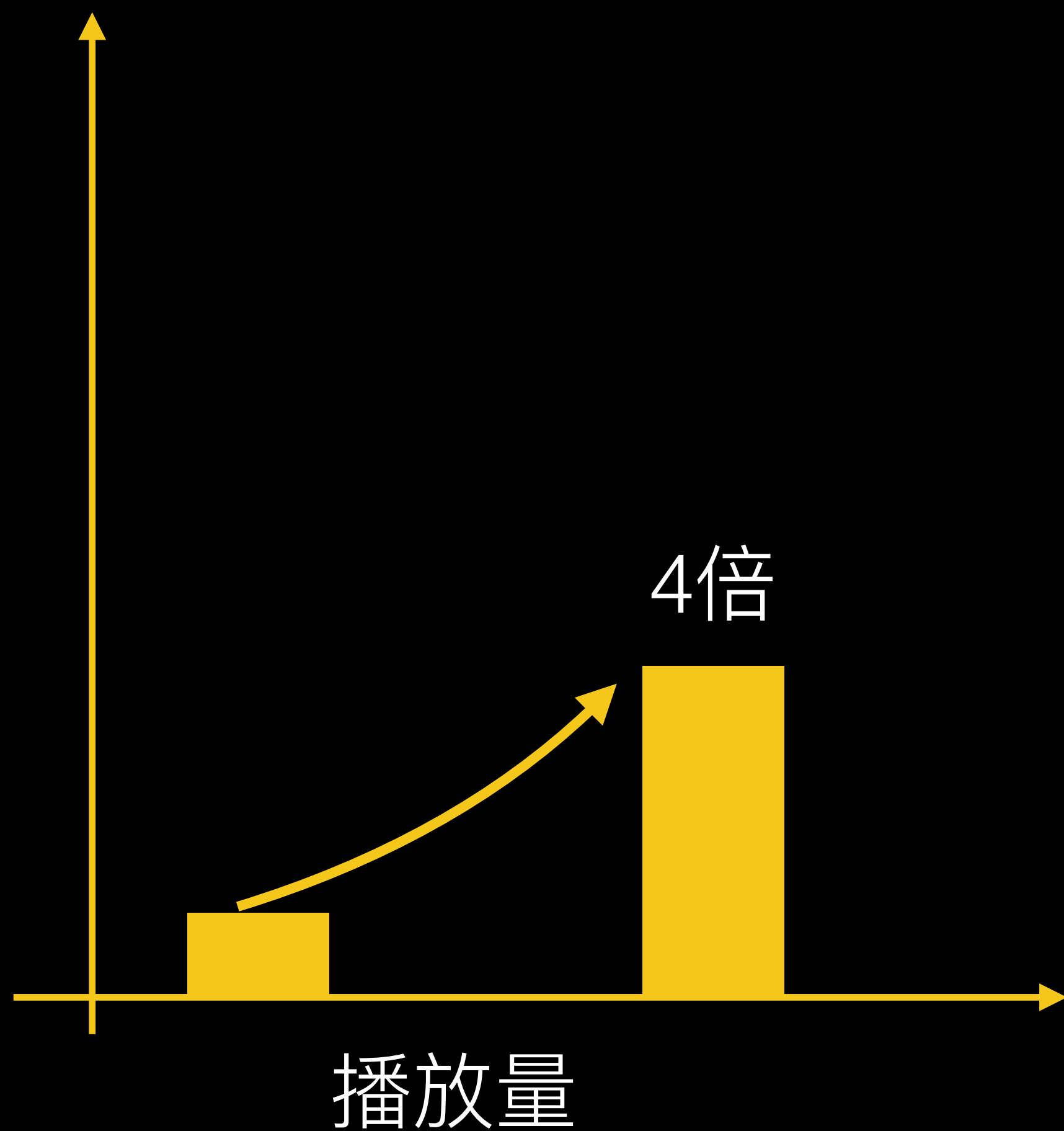
错误码	错误次数	错误占比	错误描述	平台
其它	4	64.99	null	null
-1005	1	11.32	NSURL网络中断	iPhone
-100000100	1	4.56	网络劫持, 通常...	iPhone
-800000000	1	3.6	40s内连接服务...	Android
-1005	1	3.18	NSURL网络中断	iPhone
-459729217	1	2.88	403失败换vkey...	Android
-310110004	1	2.8	msf命令字超时	iPhone
-800000000	1	2.36	40s内连接服务...	Android
-450019995	1	2.21	无网络(腾讯视...	Android
-499599999	1	2.11	播放错误码, 对...	Android

成本优化

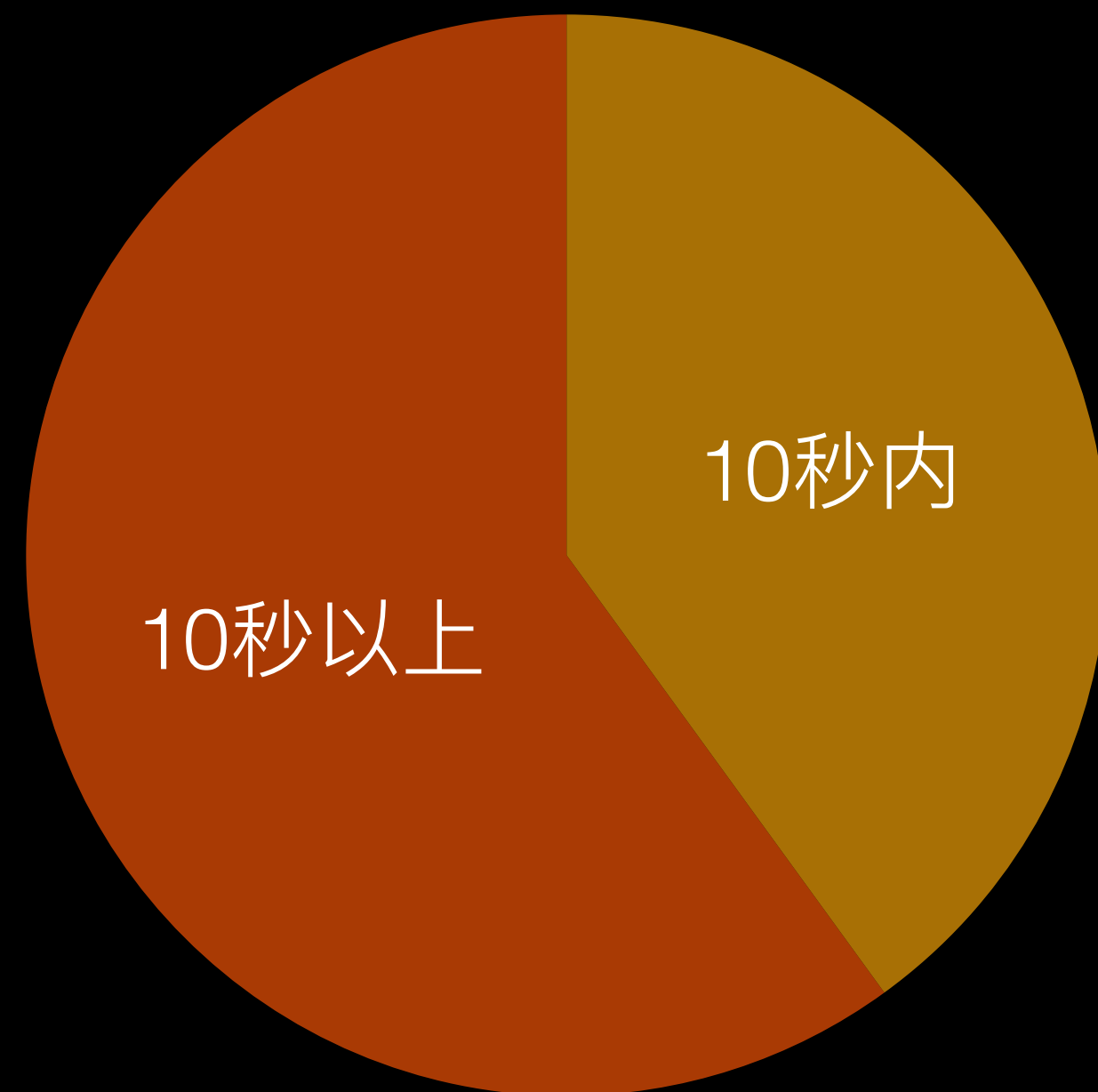
流量控制，带宽优化



上线第一个月



用户行为改变

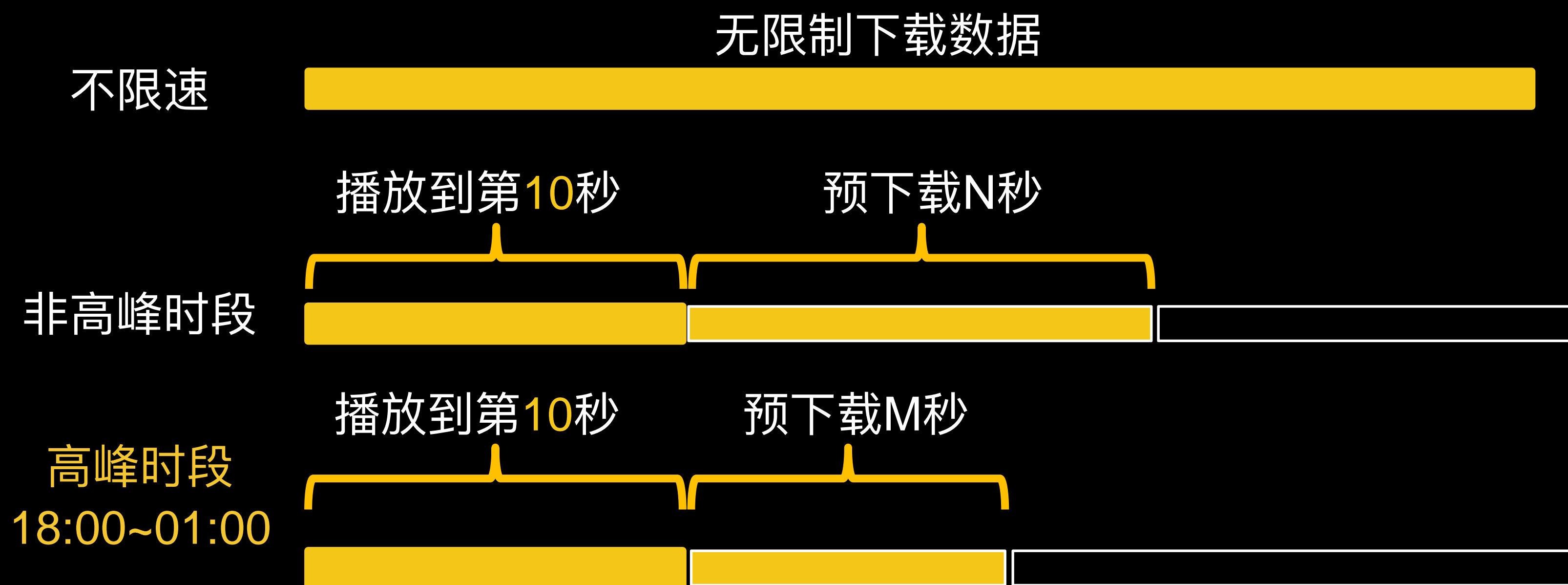


用户播放行为

- 改为自动播和边下边播之后，用户播放视频的意愿更强；
- 挑选心理，不是每个视频，用户都感兴趣；
- 没有被用户观看的视频数据都是浪费。

流量控制

保持缓冲的Buffer长度在一定的范围内，减少不必要的下载量，控制带宽成本。



1.按配置的区间多级限速，假设配置的区间是 (X,Y)，限速逻辑如下：

$< X-5$	$X-5 \sim X$	$X \sim Y-10$	$Y-10 \sim Y+5$	$> Y+5$
不限速	3倍码率限速	2倍码率限速	1倍码率限速	1/2码率限速

2.播放前10秒时，可播放时长超过10秒进行1倍码率限速。

3.码率切换，切换低一档码率。

优化效果

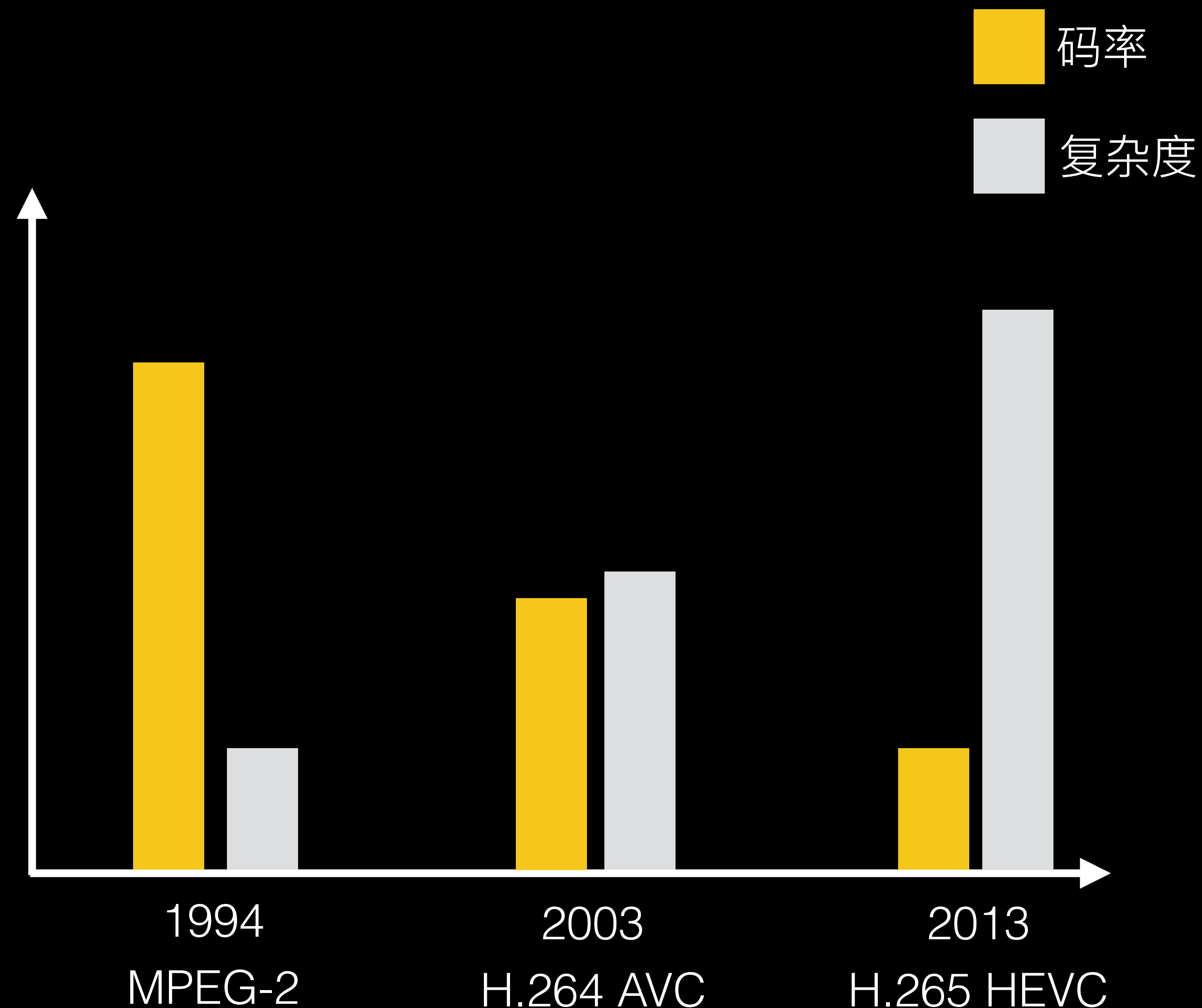
- 晚高峰期18:00~1:00在用户观看视频期间, 只会预下载N秒的数据, 在视频播放缓冲率和成本之间取一个平衡。通过这个优化策略, 带宽下降**25.4%**。

进一步优化

HEVC

H.265 - HIGH EFFICIENCY VIDEO CODING

- 压缩率提升30-50%，节省带宽
- 相比H264，解码复杂度提高4倍，更消耗系统资源
- 编码时间更长（3~7倍）



H265挑战-客户端

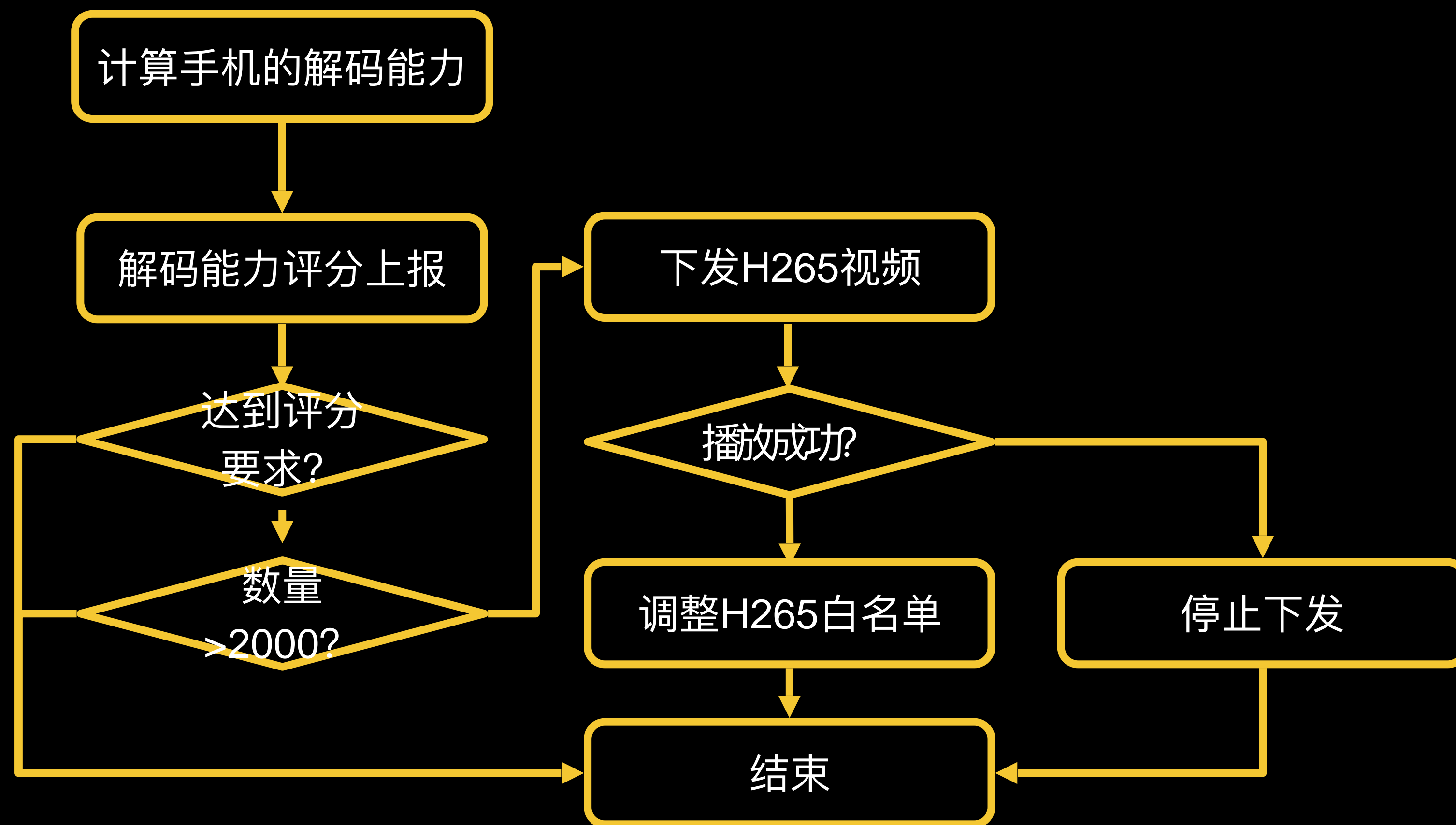
H265本质上是拿时间换空间，相比H264更消耗CPU，编解码耗时更长。

编解码	硬件支持	相对H264耗时	是否可行	问题
编码	无	3~7倍	不可行	编码耗时太长
解码	很少，兼容性差	相同	可行，软解	软解能力参差不齐，CPU占用高，在不同的视频源下可能引起卡顿，发热等问题

结论：解码可行，编码不用考虑，可以放到后台做。

H265客户端解码适配

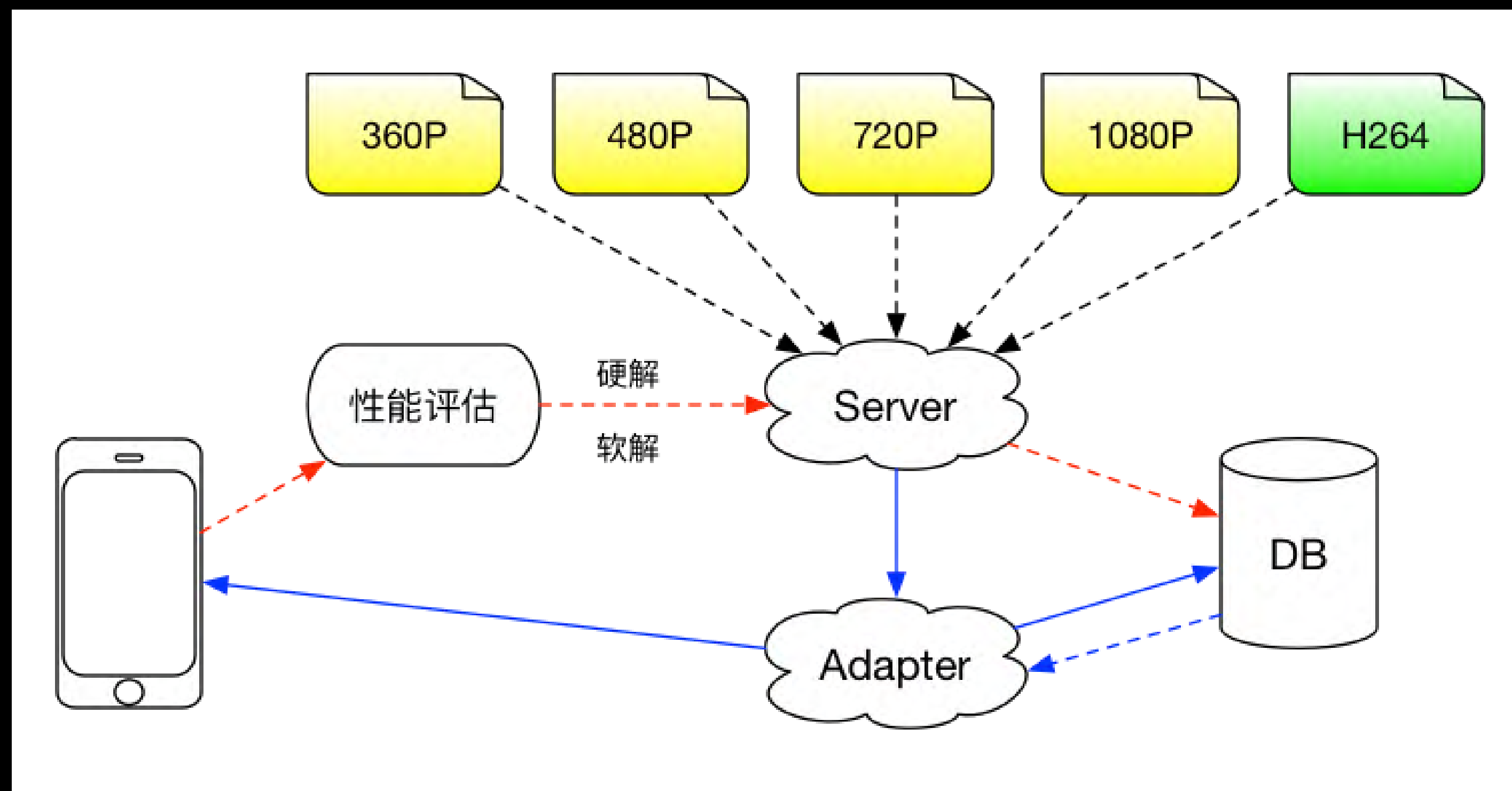
评估当前手机的解码能力。在合适时机，运行一次大规模的浮点数运算，并将评分结果上传后台服务器进行云适配。



H265客户端解码适配

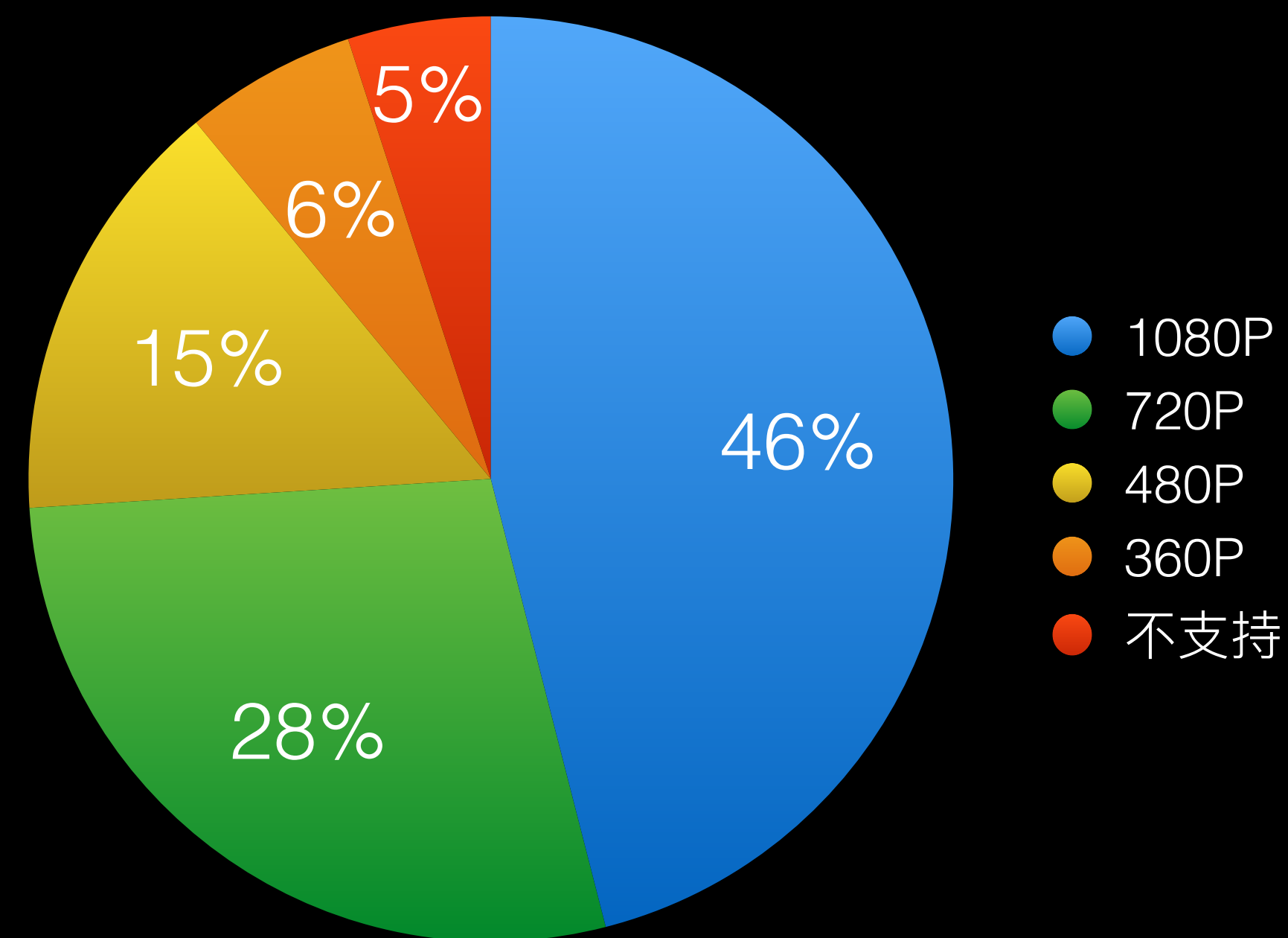
从而保证**软件解码柔性可用**，针对视频源规格按机型适配降级，保证用户视频播放体验

- 动态调整；
- 评分体系建立；
- 便于清晰度和解码方式调整；



H265客户端解码覆盖

H265外网具有软解能力的用户覆盖率达**94%**



外网用户手机H265软解能力分布

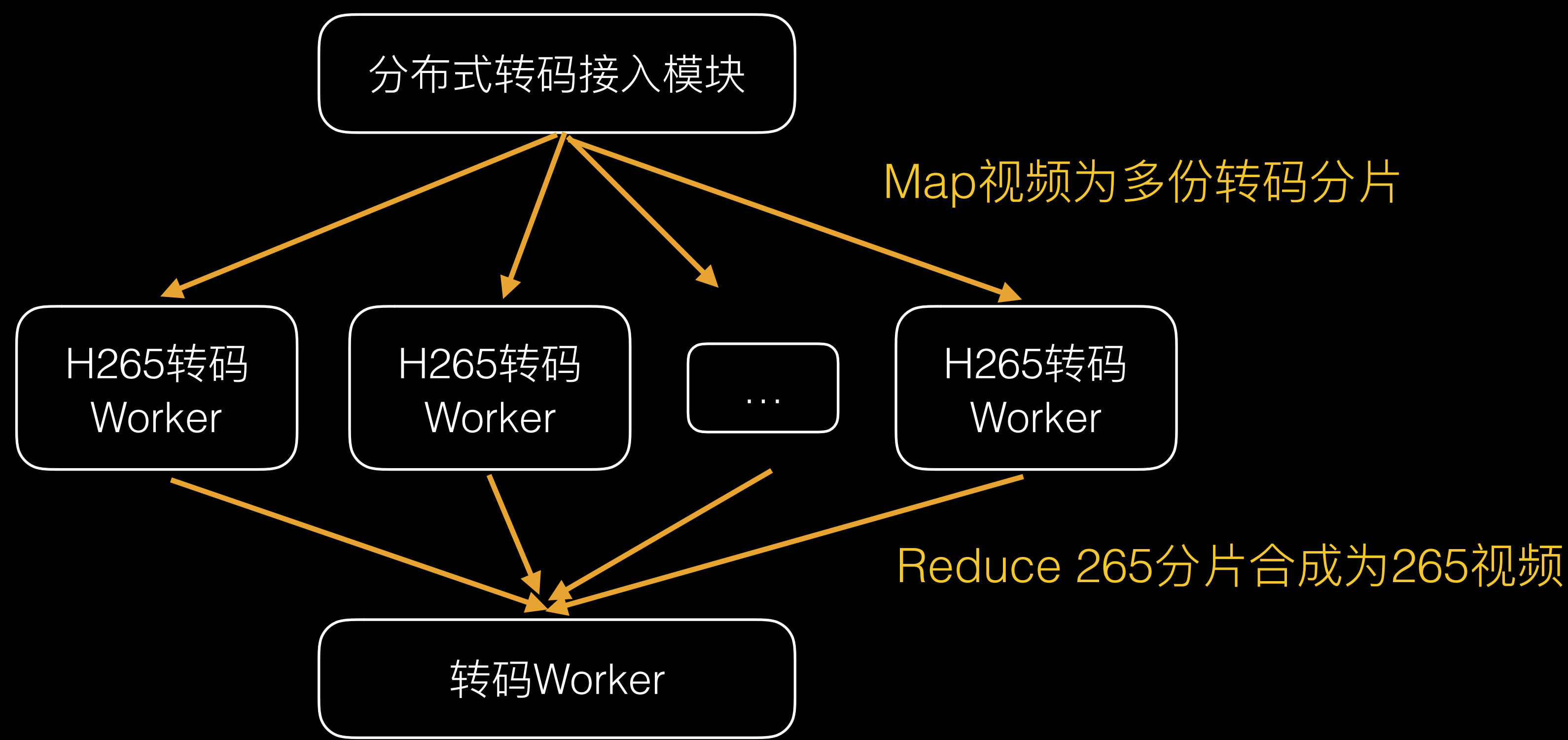
H265编码后台策略

只针对播放量位于TOP N(N视服务器资源消耗而定)的**热点视频**进行H265后台转码，即不会消耗太多服务器资源，又可以保证大部分视频流量享受优化效果。



H265编码后台优化

在质量一致的前题下，分布式分片转码10min视频，只需要2~3min即可完成；不采用分布式分片转码10min视频大约需要转码30~40min。



自研H.265编码器

- 在变换单元分割提前终止、多参考帧自适应选择、预测单元模式智能决策和码率动态质量调整的优化下自研H.265转码优化得到不错的收益
- **1M码率附近**在同等质量下，tivc编码器速度比x265@medium速度平均快8%，码率节省14%
- **在512k码率附近**，tivc编码器比开源的x265@medium速度平均快15%，码率节省18%

优化效果

- H265压缩效率比H264高出30%~50%，**节省带宽31%**左右。



体验优化-秒开优化

即点即播，丝滑顺畅



秒开体验

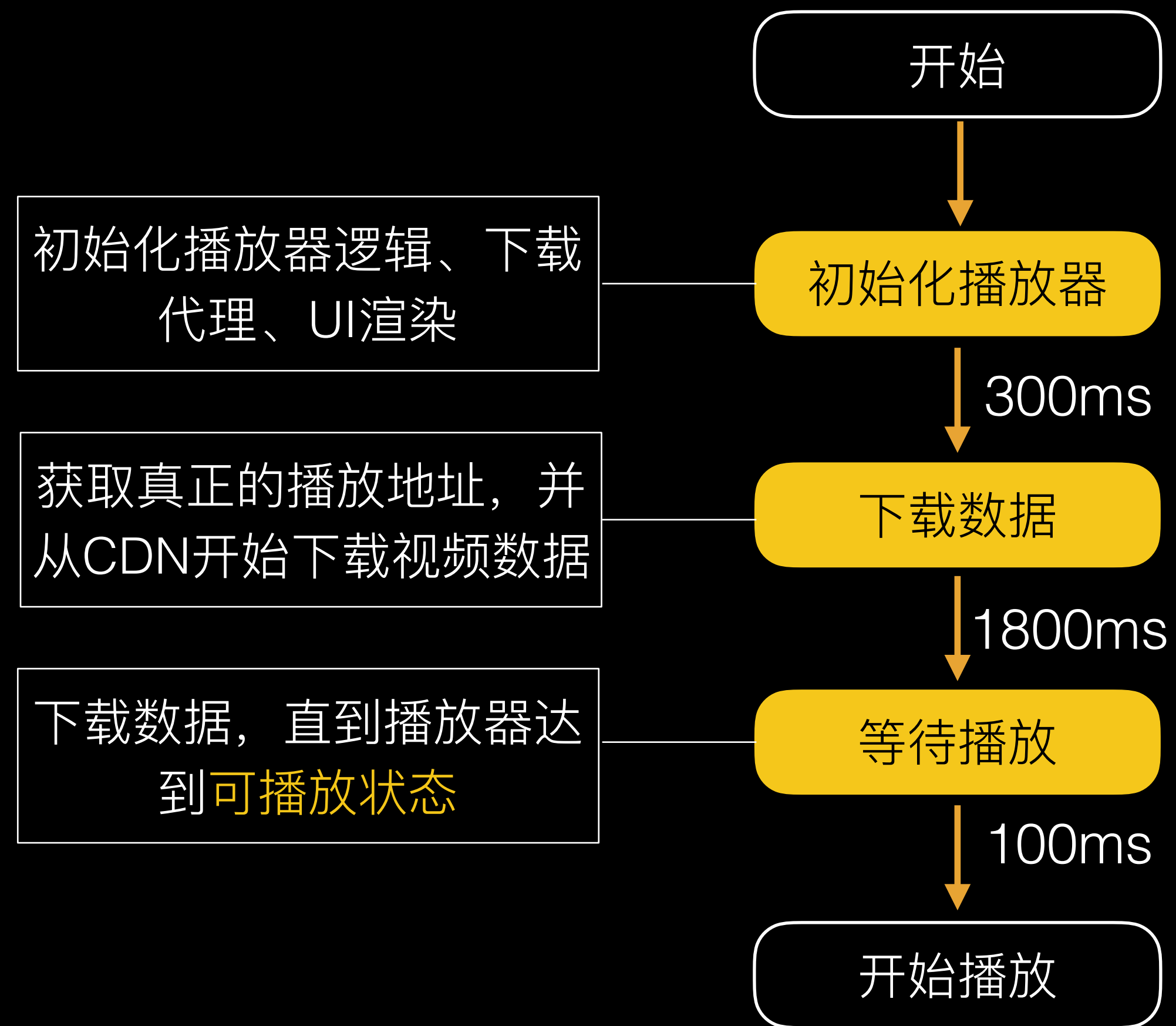
目标：视频一进入视野范围，就能开始播放，并且不产生缓冲，播放体验越好，用户播放视频的意愿就越强。



1秒以内



核心流程



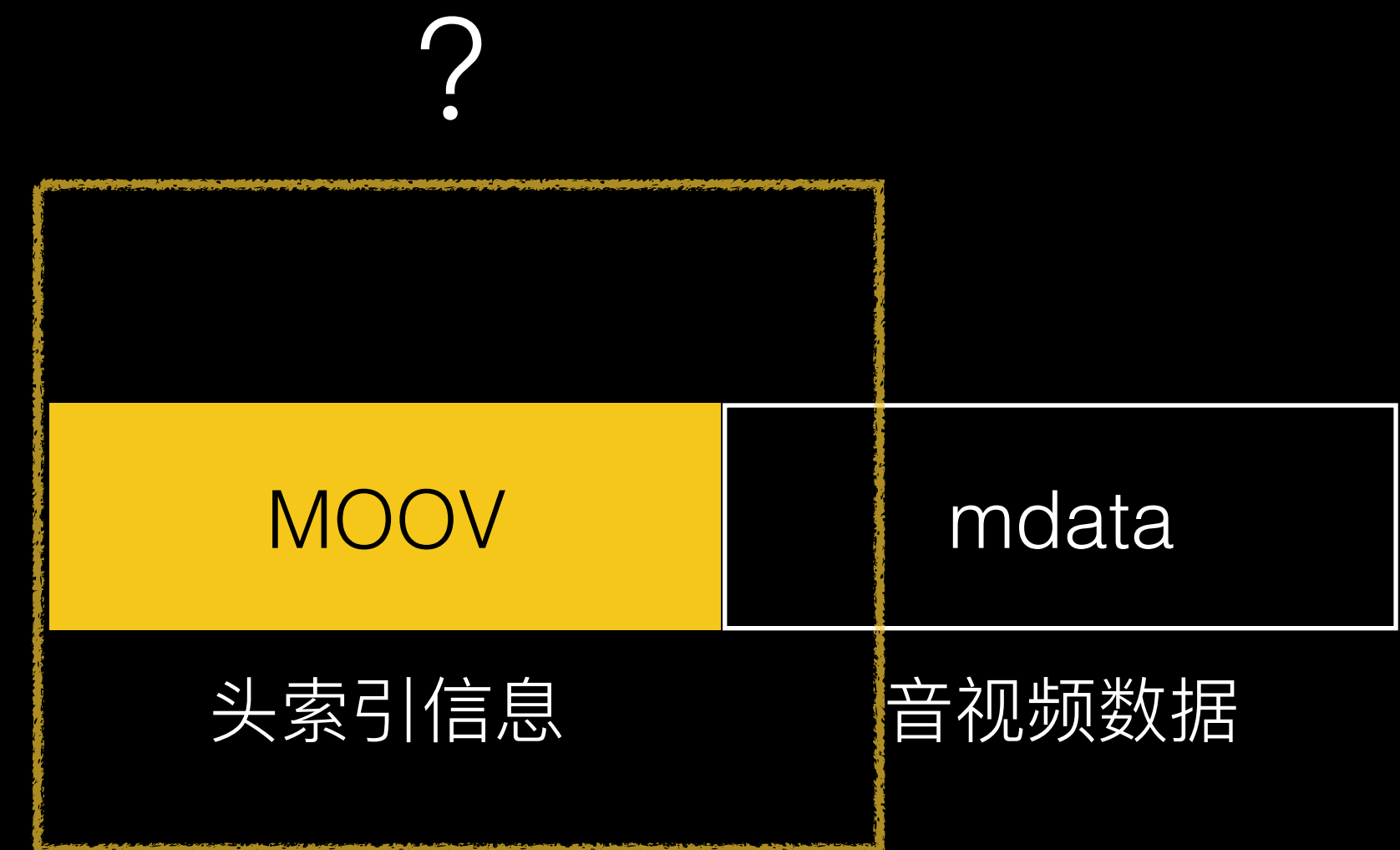
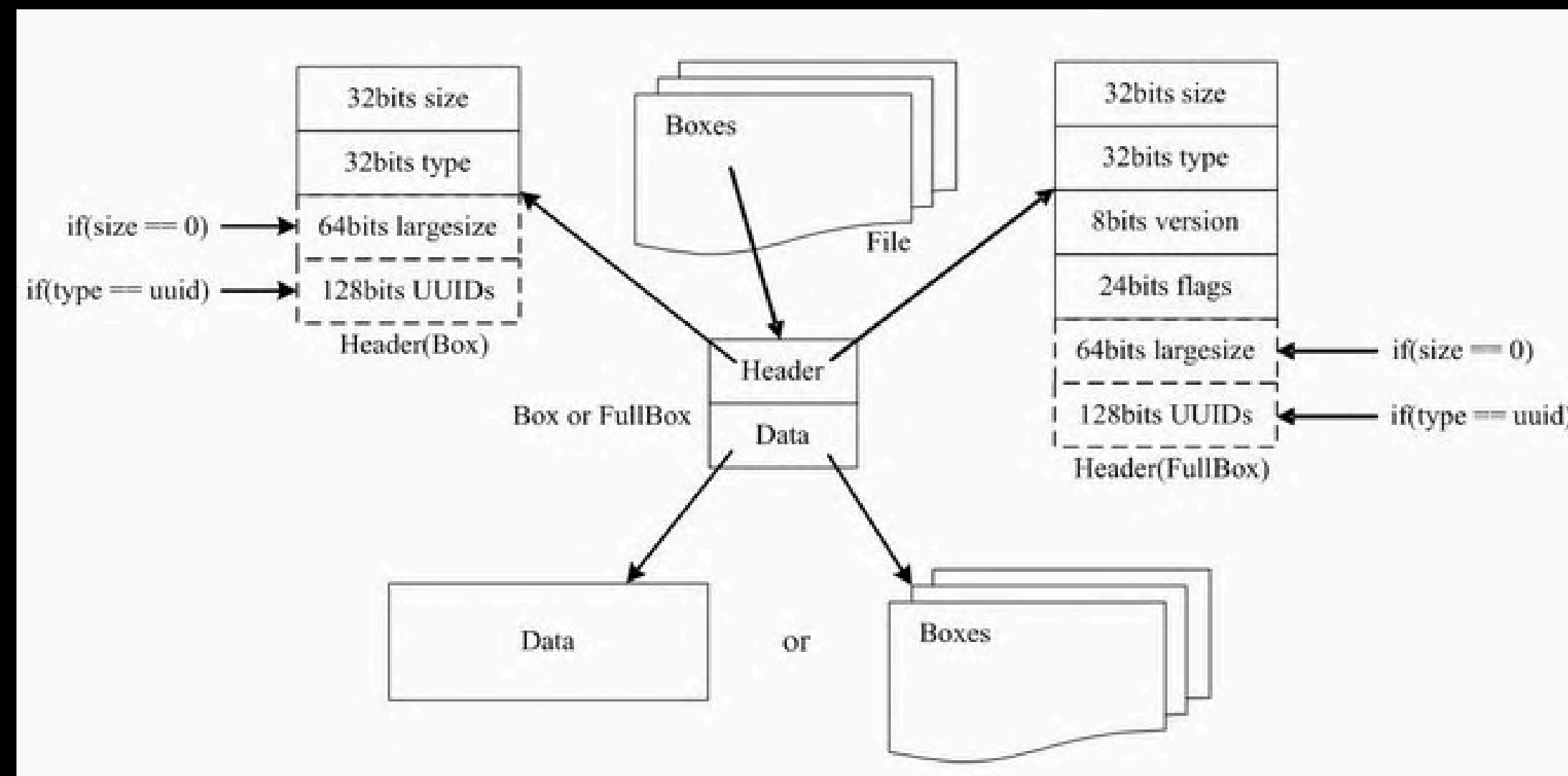
总耗时：

- 下载视频数据耗时， $\text{下载数据量} / \text{下载速度}$
- 客户端耗时

播放器需要下载**多少数据**才能播放？

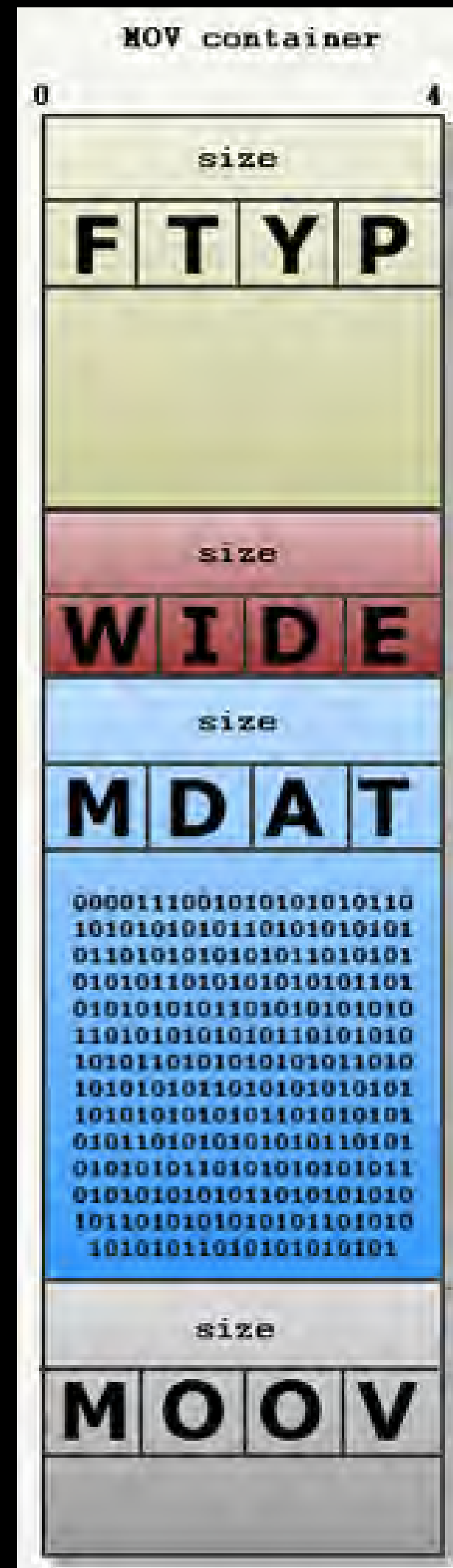
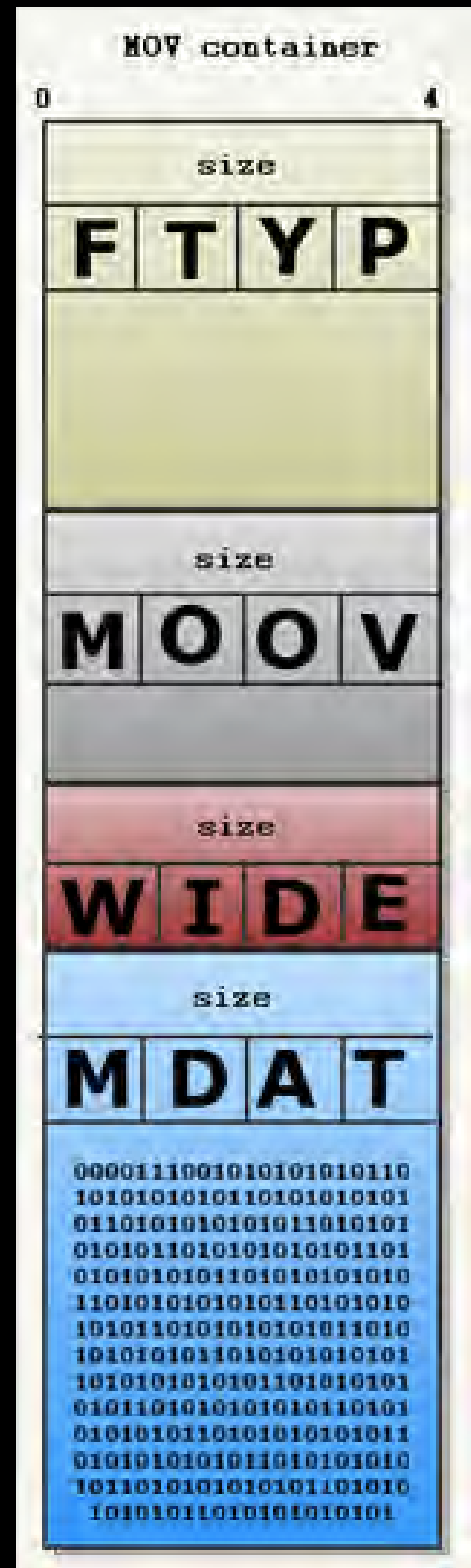
MP4容器格式

MP4是一种描述较为全面的容器格式，被认为可以在其中嵌入任何形式的数据，各种编码的视频、音频等，数据格式**灵活多变**。



MP4是由一个个“box”组成的，大box中存放小box，一级嵌套一级来存放媒体信息。

MOOV的问题



播放器	播放器行为
iOS	多一次Range预测请求
Android某些机型	需要下载全量数据
Android大部分机型	多一次Range预测请求
FFmpeg Based Player	多一次Range预测请求

MOOV

很大程度决定视频的播放时机。

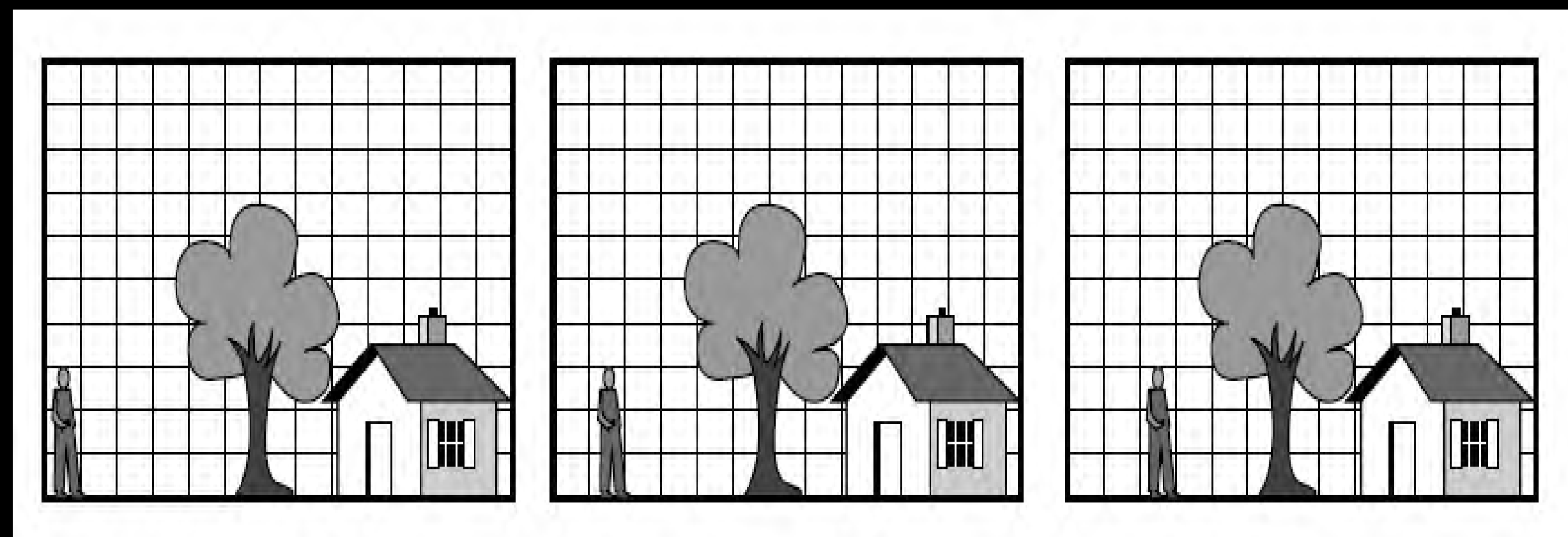
常见的方式 moov in the end of file **5%**

MOOV优化

- 后台统一做一次转码修复(有能力的情况)
- 客户端采集后做一次转码修复

MData组织方式

H.264是目前使用频率较高的一种视频编码标准。

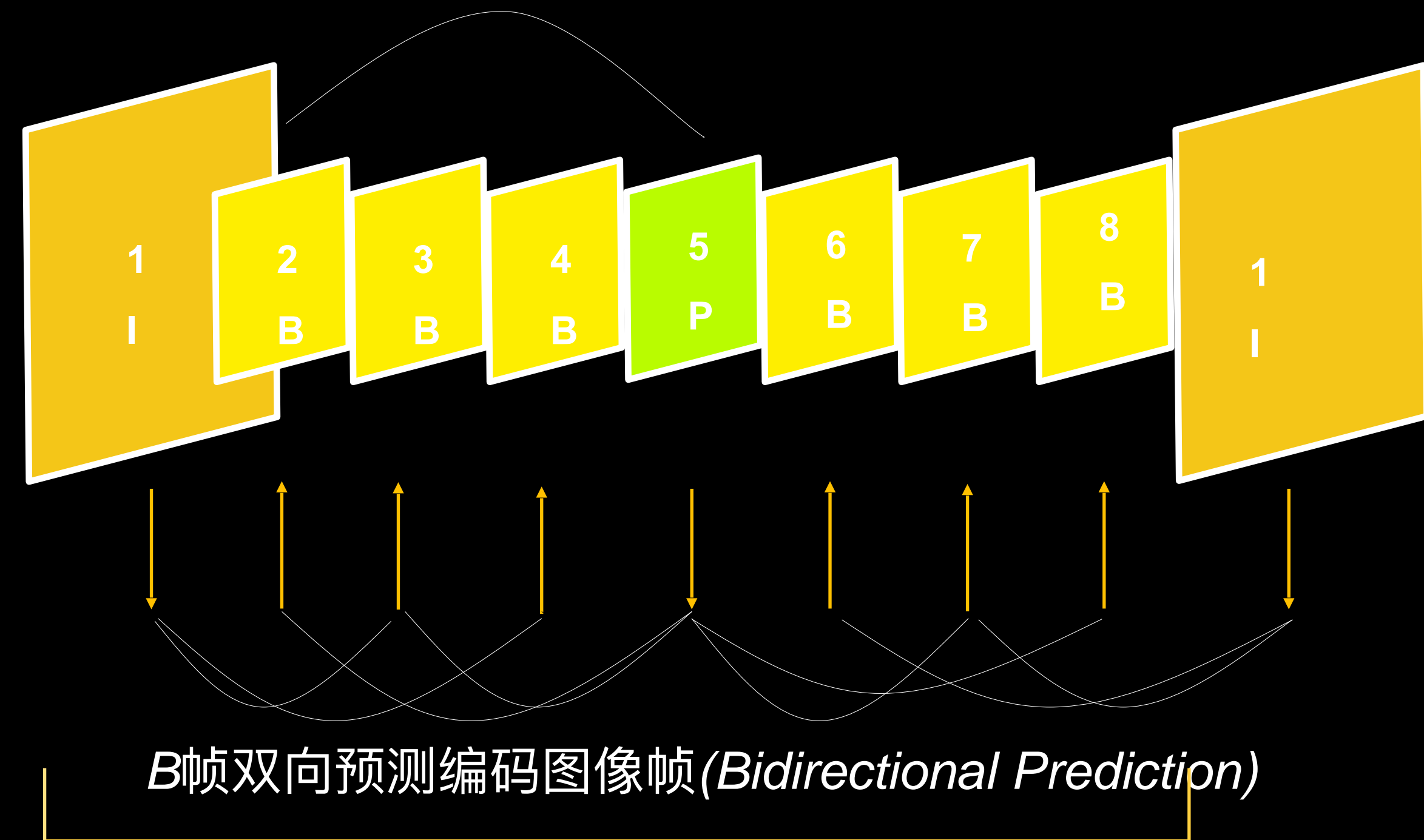


划分序列

定义帧

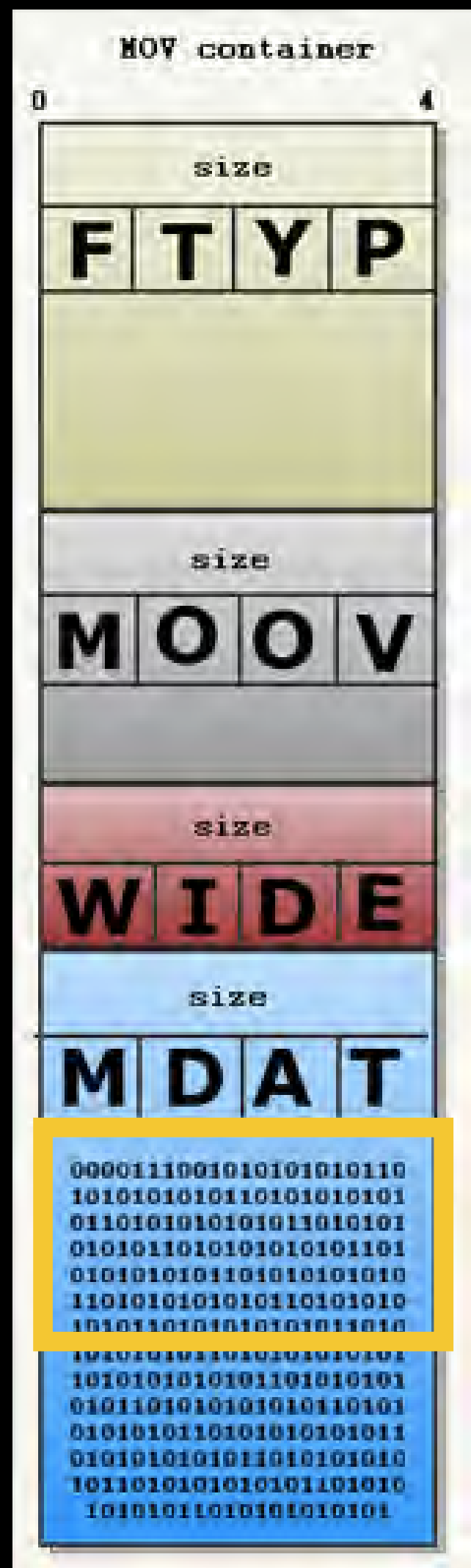
预测帧

P帧前向预测编码图像帧 (Forward Prediction)



编码序列(GOP)

播放器行为



播放器	播放器行为
iOS	一个GOP (推测)
Android 6.0及以下	5秒视频数据
Android 7.0及以上	一个GOP
FFmpeg Based Player	关键帧

需要下载多少数据才能开始播放，各播放器行为存在差异。

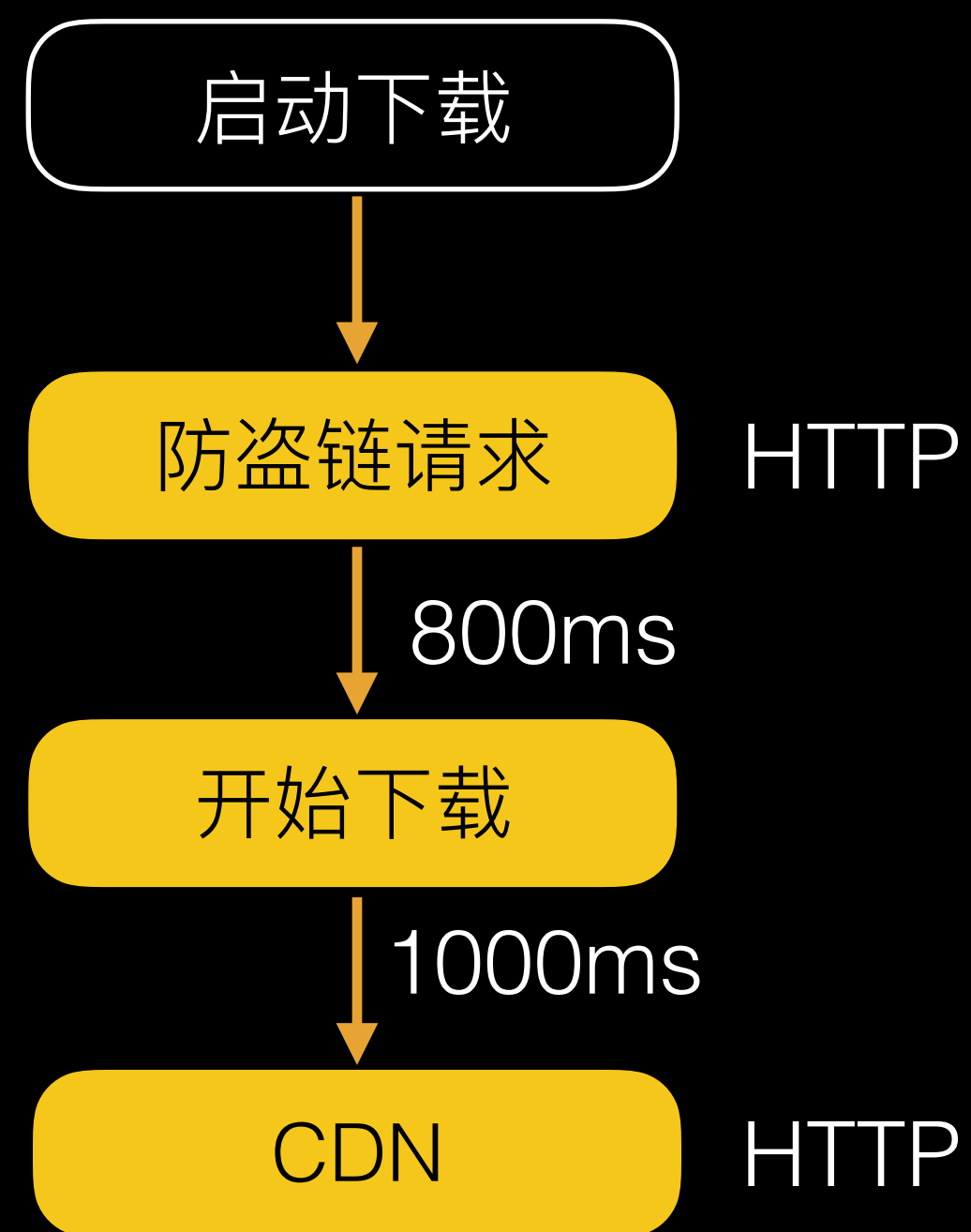
MData优化

- 采用外部开源播放器，但要关注兼容性和耗电情况

播放器	首次缓冲耗时
系统播放器	1.13s
FFmpeg Based Player	0.72s

优化下载耗时

国内的移动网络环境错综复杂，不仅要处理断网、慢速、抖动等网络本身的题，还要处理跨网、运营商劫持等国情问题。



思路

- 防盗链HTTP请求优化。
- 优化与CDN的连通性和连通速度

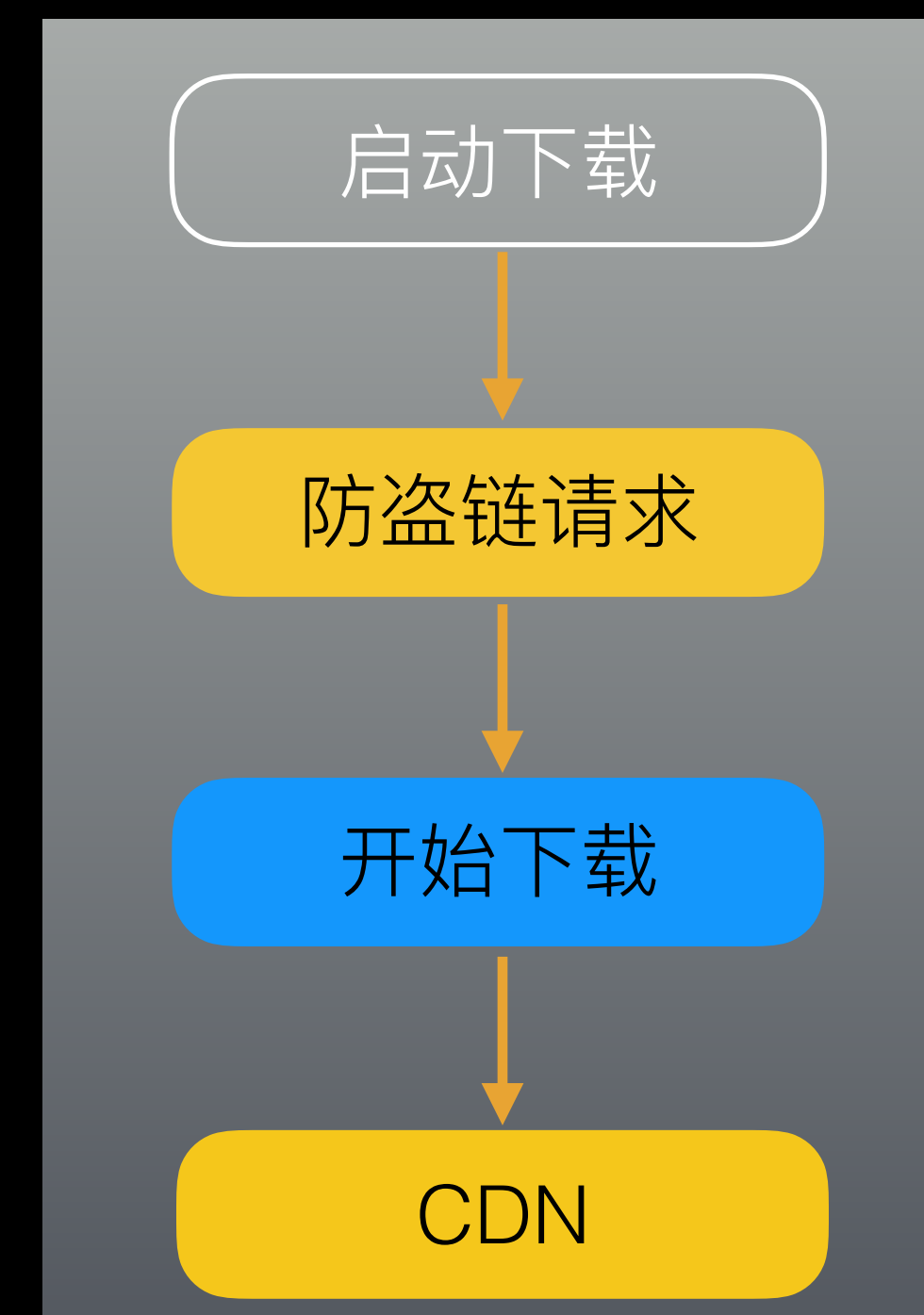
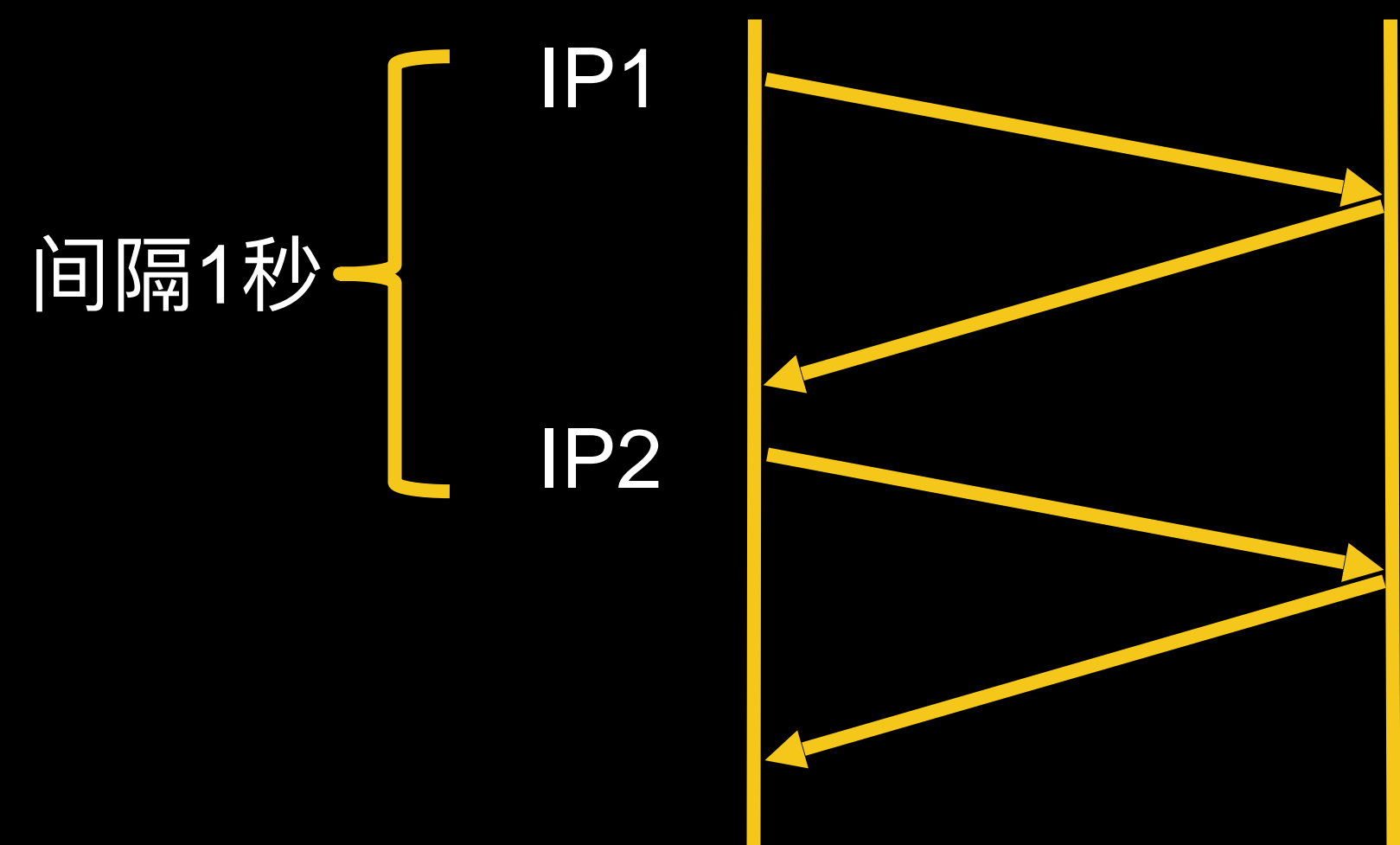
防盗链耗时优化



IP直出与竞速

IP直出：2组IP+一组域名

竞速方式：跑马策略，兼顾性能和效率



进一步思考

索引文件的大小大约是512字节/秒，按照视频600Kbps的码率和外网平均400KB/s的下载速度播放一个15s的短视频，数据加载过程的理论值是

$$(0.512*15 + 600*5/8)/400 = 0.9567s$$

随着外网码率的提升，这部分的理论数据还会变大。

使用场景分析

核心场景：

好友动态(NewsFeed)

视频推荐浮层



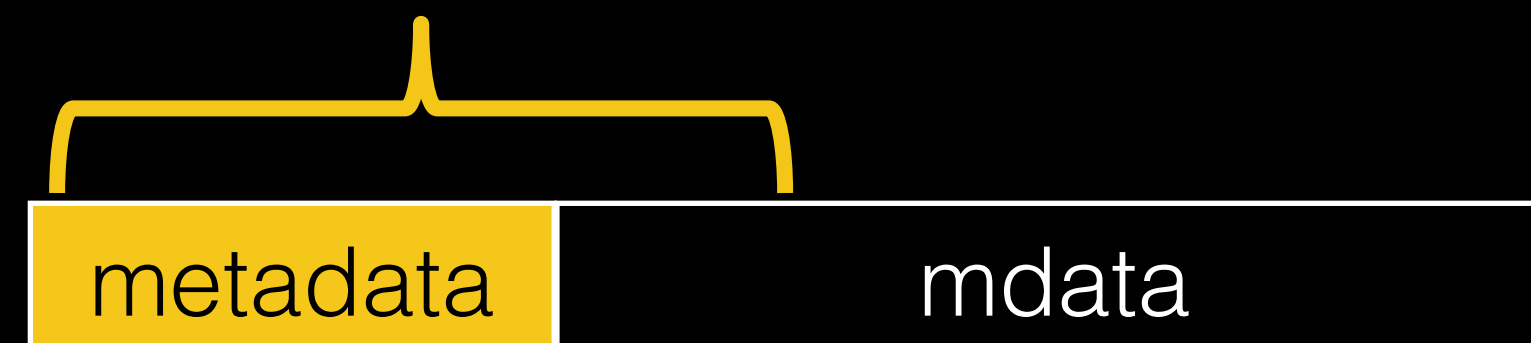
预加载是王道



预加载视频数据是减少首次缓冲耗时的最重要的办法。

但预加载不能跟当前播放的视频抢下载带宽，达到一个合适的阈值时开始下载下一个视频的数据。

$$512 * \text{视频时长} + \text{预加载时长} * \text{码率}$$



预加载时机

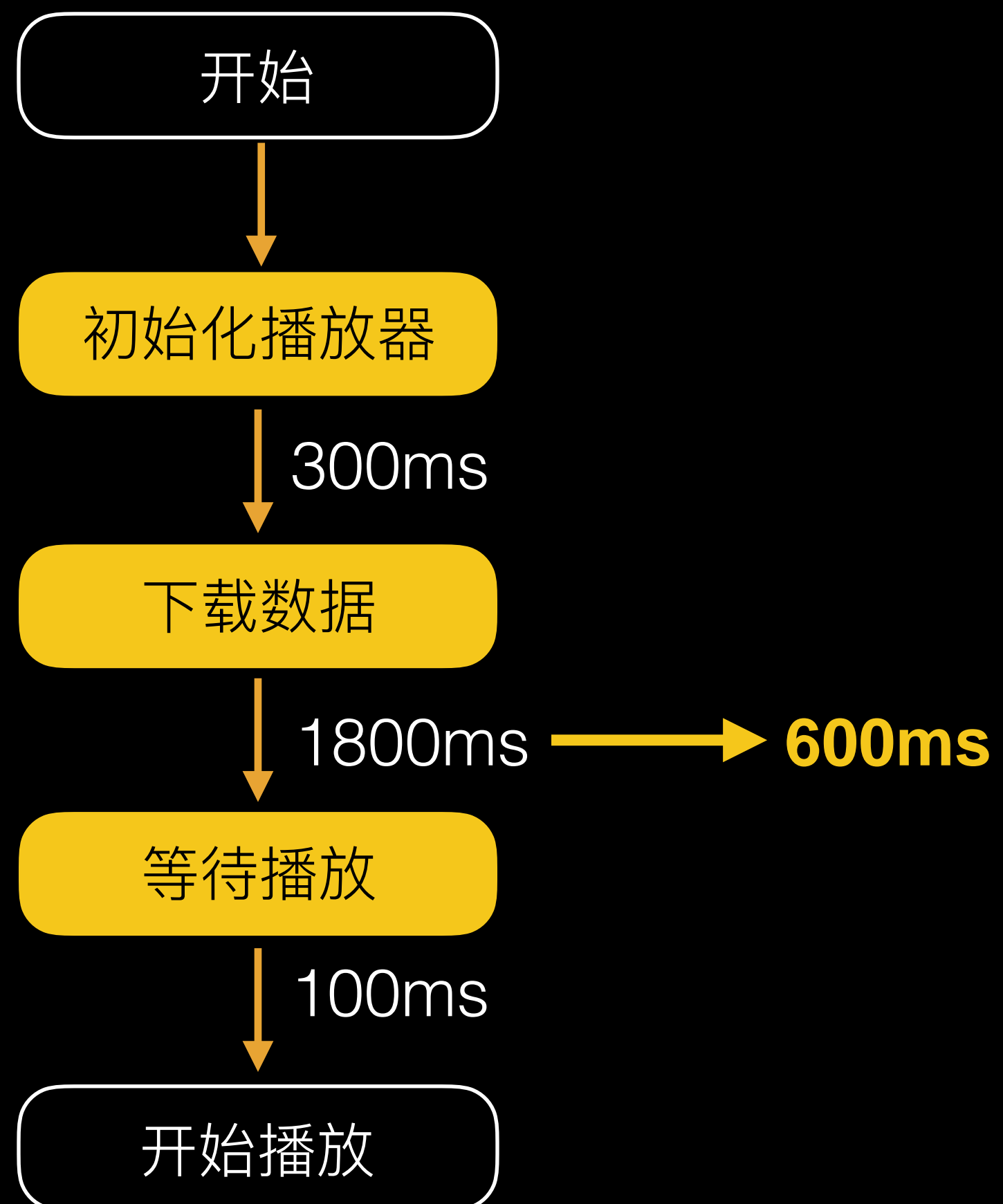


时机：优先下载图片、封面，再预下载视频数据



时机：当前视频加载时长 $>80\%$ 或者缓冲时长 $>20s$
(两个值均支持云配置)

优化效果



客户端性能优化

系统资源是有限的，特别是安卓低端机用户。

通过外网流水发现，部分缓冲慢的用户，在有视频数据缓存的情况下，读缓存，解码，渲染还是很慢，时间基本都吃在cpu上，不在网络下载上。所以通过客户端性能分析工具来精细控制缓冲过程中长耗时的点。

精细调控系统资源

- **线程梳理**。视频本身逻辑放入单独线程，其他操作如图片下载，返回包解析等列入另外线程；
- **异步化**。各种ChildView按需初始化，图片加载，缓存写入时机后移；
- **重复运算做内部缓存**。比如encode, decode, md5, httpParser等。

在视频下载过程中，可以降低对CPU和IO的占用，使得系统能够调度更多的资源在解封、装、解码、渲染等与播放、下载直接相关的操作上，进而减少这部分的耗时。

优化HLS



好友动态自动播放

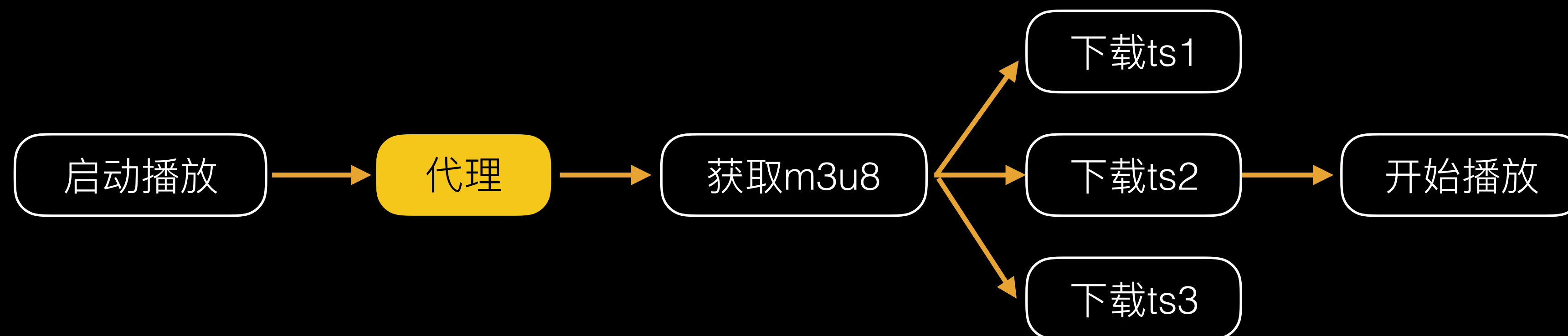
问题:

好友动态里直播的视频启用自动播放, 首次启动播放需要等待6~9秒以上。

优化思路



HLS请求链路较长，且是串行下载，需要下载3-4片TS才能启动播放。系统播放器内部逻辑无法修改，参数也是写死的，没有优化空间，系统还有各种坑。



发挥本地代理的优势使用并行加载来实现HLS加速。

优化思路

在分层架构之上，做HLS加速优化方便，思路清晰。



HLS优化效果

HLS提速明显，无缓存首次缓冲耗时由6.3秒降低为1.2秒，有缓存下由2秒降低为1秒以下。

测试数据

旧版本:

首次缓冲时间 (单位: ms, 结合版660, 机型: MI 3, 系统: Android 4.4.4)						
	1	2	3	4	5	平均
无缓存	6620	6227	5895	5938	6997	6335.4
有缓存	1973	2171	2025	1988	1860	2003.4

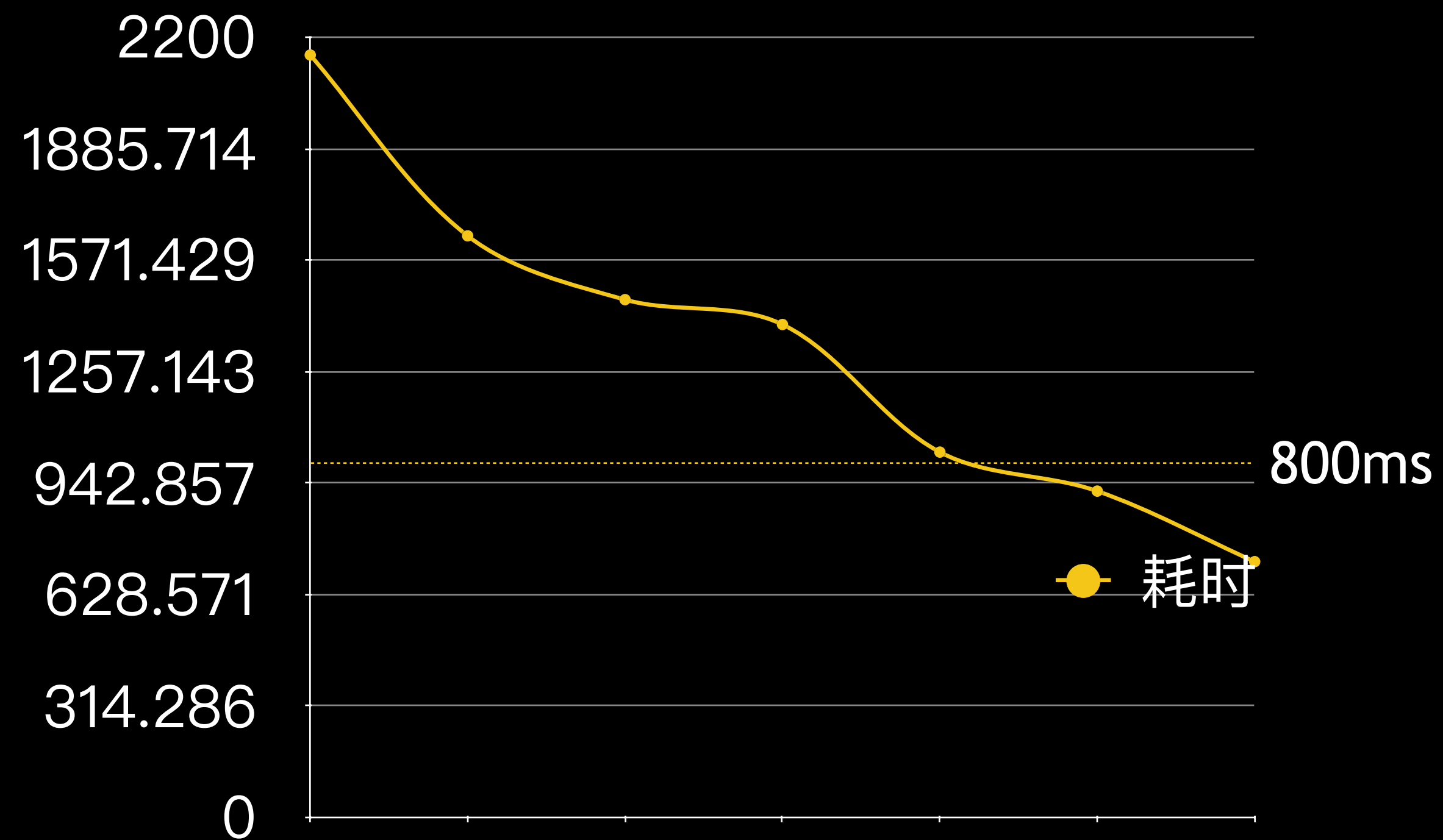
新版本:

首次缓冲时间 (单位: ms, 结合版663, 机型: MI 3, 系统: Android 4.4.4)						
	1	2	3	4	5	平均
无缓存	1288	1181	1314	1263	1120	1233.2
有缓存	972	908	1098	993	856	965.4

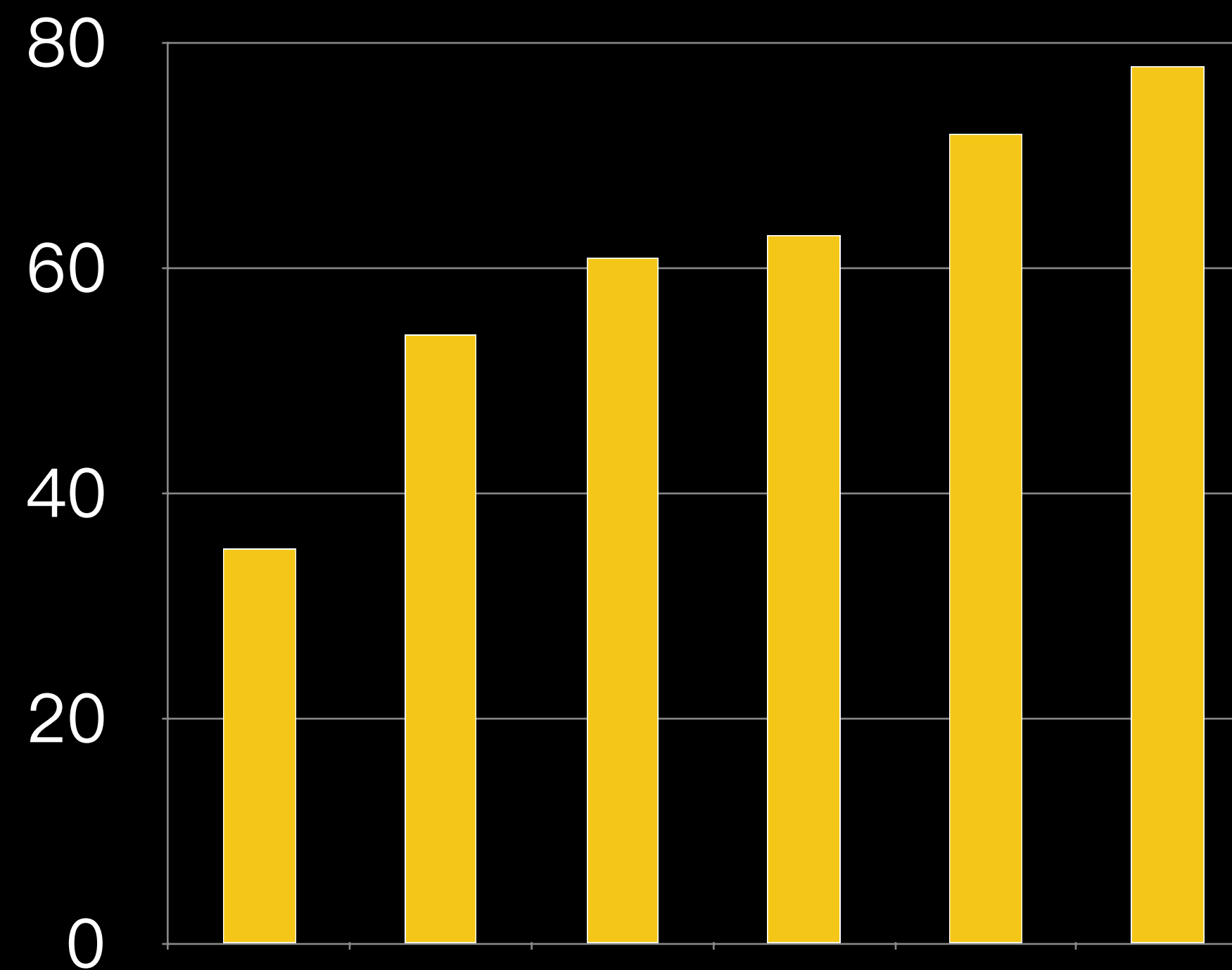
测试方法

秒开优化效果

缓冲耗时(单位: 毫秒)



0~1秒占比



体验优化-缓冲优化

你所看到的视频都不卡



缓冲优化

二次缓冲概率的定义：视频开始播放之后加载数据的次数/视频播放次数。

正常场景(10%):

播放进度

A diagram illustrating a normal playback scenario. It consists of two horizontal bars. The top bar is gray and labeled '播放进度' (Playback Progress), extending to approximately 20% of the total width. The bottom bar is yellow and labeled '缓冲数据' (Buffered Data), extending to approximately 75% of the total width.

缓冲数据

Seek场景(90%):

播放进度

A diagram illustrating a seek playback scenario. It consists of two horizontal bars. The top bar is gray and labeled '播放进度' (Playback Progress), extending to approximately 75% of the total width. The bottom bar is yellow and labeled '缓冲数据' (Buffered Data), extending to approximately 20% of the total width, with a small yellow segment at the far right end.

缓冲数据

正常场景(10%)

播放进度



缓冲数据

带宽调整:

暂停其他占用下载带宽的逻辑

低速切换IP:

下载速度 <0.5 倍码率 + 持续时长 >6 秒

网络抖动:

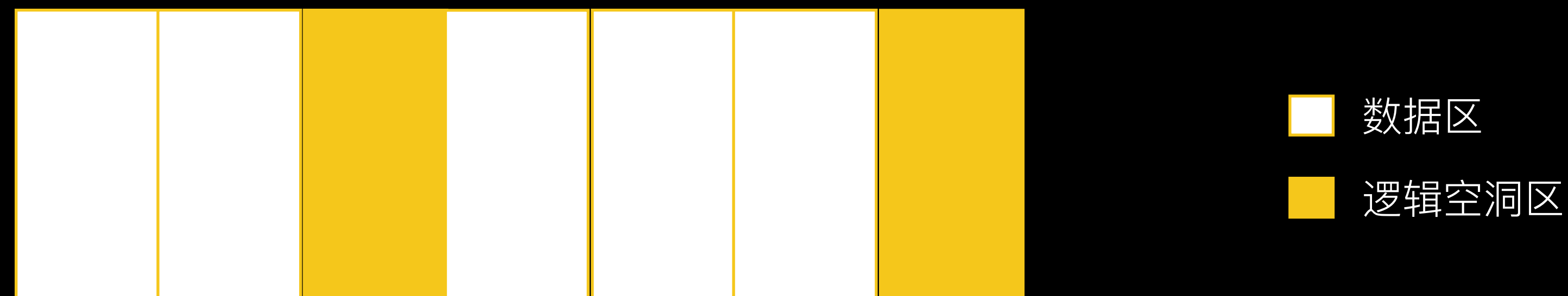
单次下载过程中发生两次以上网络切换, 突破下载网络速度限制。

Seek场景(90%)

如果使用简单的使用文件顺序缓存，会造成拖动后的区间无法进行缓存，再次恢复播放就会造成额外缓冲。



针对这种场景，我们对缓存系统进行重构，支持文件空洞。








单分片大小最大1M，约10s视频数据，例如：文件共20M，至少20个分片

缓存的设计

- 缓存文件中记录关键信息，无需数据库，回包时无需额外查询文件总大小，响应时间更快

文件名构成：唯一Key.偏移量.原始大小.文件类型（MIME）.创建时间.版本号.扩展名

Name	Date Modified	Size
 22fb86a9c8a8fad40190.0.7718840.766964656F2F6D7034.1491295470484.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.1048576.7718840.766964656F2F6D7034.1491295470492.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.2097152.7718840.766964656F2F6D7034.1491295470497.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.3145728.7718840.766964656F2F6D7034.1491295470501.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.4194304.7718840.766964656F2F6D7034.1491295470505.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.5242880.7718840.766964656F2F6D7034.1491295470510.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.6291456.7718840.766964656F2F6D7034.1491295470515.v3.dat	Today, 4:54 PM	1 MB
 22fb86a9c8a8fad40190.7340032.7718840.766964656F2F6D7034.1491295470574.v3.dat	Today, 4:54 PM	379 KB

- 以分片为单位淘汰，缓存空间利用率高，淘汰算法可以更加灵活
- 任意位置的起始数据请求都可以缓存下来
- 扩展能力强，支持任意格式的文件，可以实现数据进一步压缩加密存储
- 支持无锁多进程缓存共享

优化效果

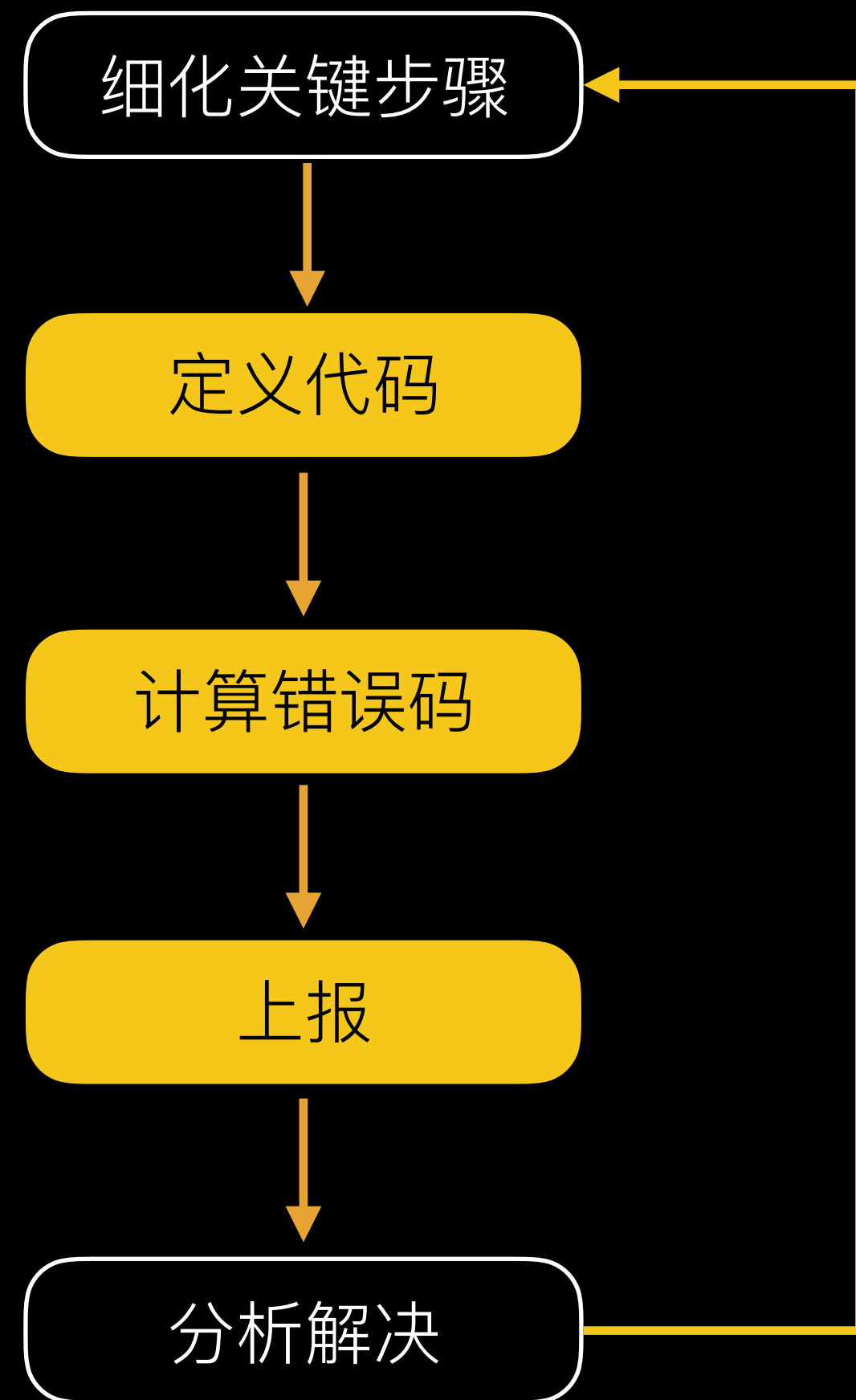
缓存命中率大幅提升，由Seek导致的缓冲大幅减少。
缓冲概率从之前的4.63%下降到0.48%。

体验优化-成功率提升

你所看到的视频都能播

成功率优化

没有捷径，case by case各个击破。



未能解决

1	错误码	播放器错误码 what	播放器错误码 extra	含义
57	-173800000	262	0	onInfo what=860引起的，播放器不支持视频编码
58	-2147485648000	-2147483648	0	非标准错误
59	-300400000	-1004	0	非标准错误
60	-203800000	-38	0	非法操作
61	-199900000	1	0	未知错误
62	-201900000	-19	0	未知错误
63	-800000000	-6000	0	启动播放60秒后没有开始播放（超时）
64	-800100000	-6001	0	setDataSource IOException
65	-800200000	-6002	0	prepareAsync IllegalState Exception
66	-800300000	-6003	0	网络提示后调用prepareAsync异常
67	-800400000	-6004	0	启动播放其它异常
68	-199900019	1	-19	系统播放器报-19，但播放实际未出错
69	-199901010	1	-1010	
70	-200000000	0	0	what和extra都是0，但抛了onError
71	下载错误引起的异常			
72	-499799999	-1000	-199799999	下载数据contentType不对引起的播放异常
73	-499399999	-1000	-199399999	下载其它HttpDataSourceException
74	-450019995	-1000	-150019995	无网络引起的播放失败（腾讯视频下载）
75	-450019976	-1000	-150019976	发出请求错误
76	-499199999	-1000	-199199999	无网络引起的播放失败（兜底方案）
77	-450019997	-1000	-150019997	DNS解析失败（腾讯视频下载）
78	-459799999	-1000	-159799999	
79	-499099999	-1000	-199099999	http头正确的情况下，返回非法的内容(360wifi下返回内容为html)
80	-459719983	-1000	-159719983	下载错误码-159719983，表示下载遇到非正常状态码 403, 404等
81	-459619983	-1000	-159619983	http 404,资源不存在
82	-449899999	-1000	-149899999	下载错误码-149899999，服务器过载或错误
83	-499599999	-1000	-199599999	下载错误码-199599999，无法连接服务器
84	-459019983	-1000	-159019983	28618
85	-450019995	-1000	-150019995	连接失败，无网络（腾讯视频下载）
86	-459619983	-1000	-159619983	http 404,资源不存在
87	-498699999	-1000	-198699999	下载错误码 -198699999 SocketTimeout 通常是网络太差引起
88	-509299999	-1100	-199299999	防盗连超时
89	-509299914	-1100	-199299914	changekey返回较慢逻辑问题
90	-509300775	-1100	-199300775	换ckey/vkey接口调用失败
91	-508599999	-1100	-198599999	因无网络导致的防盗链失败

时间	平台	版本	错误码	错误次数	错误占比	负责人	错误描述
2017-03-28	android独立版	7.0.1.288	-800000000	1	66.7%	...	40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.0.1.288	-460019983	1	33.3%	...	腾讯视频删除或者审不通过通过通知漏过，已经让腾讯视频修改
2017-03-28	android独立版	7.1.5.288	-800000000	1	29.3%	...	40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.1.1.288	-173901003	1	20%	...	ERROR_CANNOT_CONNECT
2017-03-28	android独立版	7.1.1.288	-800000000	1	20%	...	40s内连接服务器失败（包含重试）
2017-03-28	android独立版	7.1.1.288	-203800000	1	20%	...	偶现播放器非法操作
2017-03-28	android独立版	7.1.1.288	-499599999	1	20%	...	播放错误码，对应的下载错误码-199599999，无法连接腾讯视频下载服务器
2017-03-28	android独立版	7.1.1.288	-2347383648	1	20%	...	播放器报ERROR_UNKNOWN
2017-03-28	android独立版	7.1.5.288	-2347383648	1	20%	...	播放器报ERROR_UNKNOWN
2017-03-28	android独立版	7.1.5.288	-2347379543	1	12%	请更新	请更新

下载常见错误

下载错误原因	错误占比	解决方案
DNS劫持，域名解析成错误IP，导致连接失败或返回HTML	35%	多IP直出，既避免了DNS解析的过程，又可以在一个IP失败的情况下自动切换另一个IP下载保证连接成功率，后续IP直出后切HTTPS更加保险。
异常返回码403	25%	就空间的业务来说，多数是因为VKey失效，通过私密协议换VKey重连即可解决。少数是视频非法访问，如带版权的腾讯视频，只能排查出来后通过业务层规避此类视频的访问。
异常返回码404	10%	404多数为CDN上已删除的文件进入了播放逻辑，这种只需要后台进行逻辑优化即可解决
慢速网络	10%	提供多种码率规格的视频，根据网络情况智能降档。
网络不稳定	10%	重连，并无法彻底解决
异常返回码500	5%	服务器过载引起，扩容或优化负载均衡解决
内容遭劫持	5%	提取字节流对字节流进行校验，如果发现是内容遭劫持则不进入播放逻辑。各种WiFi棒会引起此问题。

播放常见错误

播放错误原因	错误占比	解决方案
视频播放器内部的错误	35%	<p>错误码：-2147483648原因：低端机器播放器的解码器在播放码率较高的视频时，解码器会直接报出2147错误，播放器兼容性问题。解决：1.某些UGC视频是由于上传时将h264压缩参数profile为high，导致编码出的视频播放器不支持，调整为baseline；2.本身高码率的PGC视频需要切换为自解码播放器软解兼容播放。</p>
播放器调度不合理导致的错误	65%	<p>错误码：-199900038原因：Android的mediaplayer内部有严格的状态控制，例如preparing过程中调用start就会抛出这个错误。 解决：在业务层对mediaplayer的调用加入状态机控制，严格杜绝非法调度</p> <p>错误码：-80020000原因：播放器prepareAsync报非法错误，无缝播放的场景，由于mediaplayer未销毁，而surface切换，小概率会偶现非法错误。 解决：重新new新的MediaPlayer和surface来解决</p>

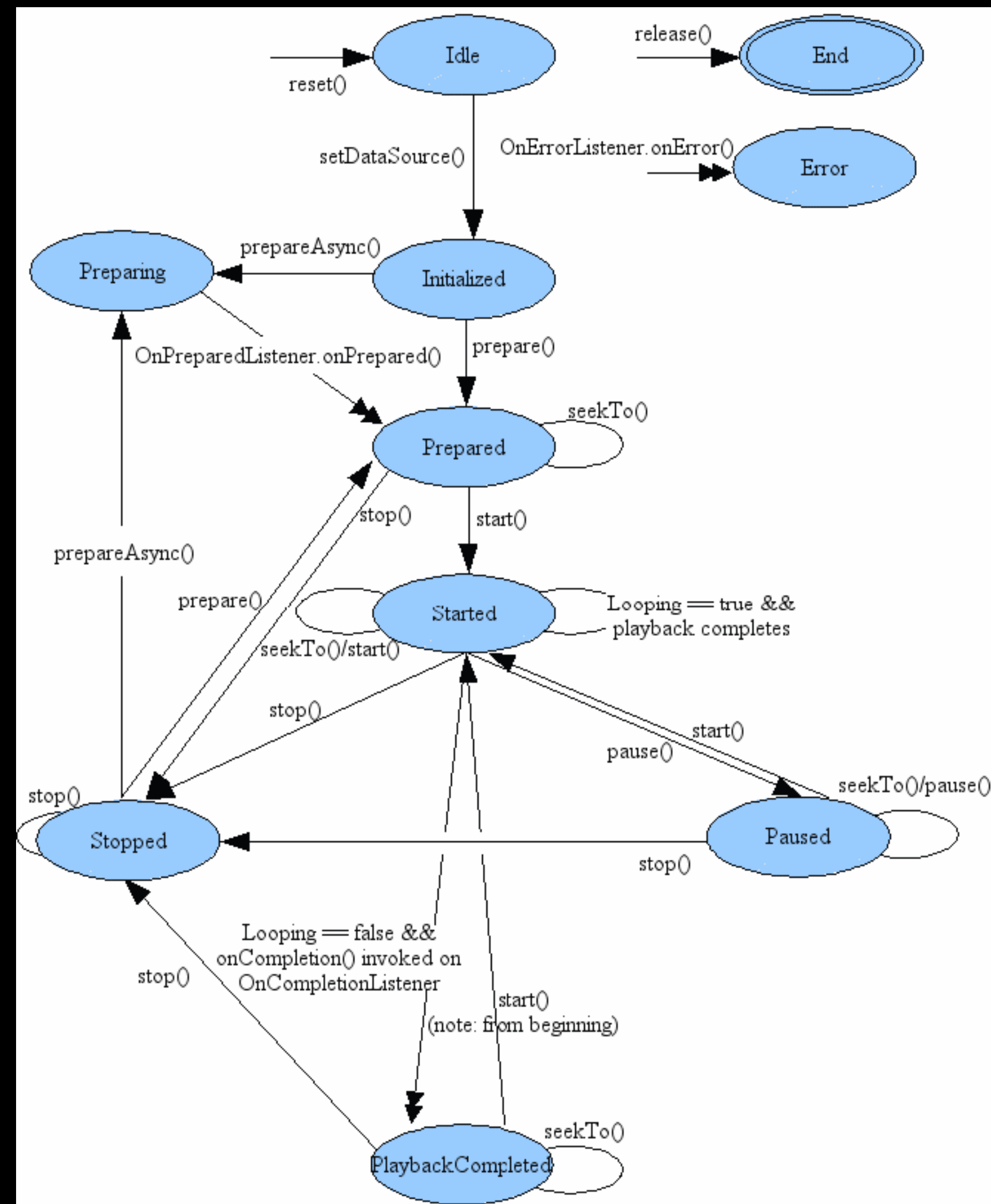
播放成功率解决

播放器调度错误产生的主要原因

播放器本身的状态较多，调度需要严格的时序。多人开发模式下，方法调用比较随意。

播放器错误解决

HOOK播放器接口与回调，实现播放器状态机，监控播放器API的调用是否合法，不合法直接告警或Crash。帮助开发快速定位问题，同时减轻测试同事的负担。封装成UI组件，使其它开发不必理解播放器



优化成果

技术指标	优化前	优化后
下载成功率	97.1%	99.9%
播放成功率	97.0%	99.9%
首次缓冲耗时	1.95s	0.7s
二次缓冲概率	4.63%	0.48%



欢迎关注、投递简历



关注QCon微信公众号，
获得更多干货！

Thanks!



INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 **Geekbang** & **InfoQ**
极客邦科技