

iOS视频边下边播--缓存播放数据流



夜千寻墨 (/u/527ecf8c8753) ✓ 已关注

2016.02.15 15:36* 字数 2389 阅读 41906 评论 272 喜欢 692 赞赏 11

(/u/527ecf8c8753)

google搜索“iOS视频变下边播”，有好几篇博客写到了实现方法，其实只有一篇，其他都是copy的，不过他们都是使用的本地代理服务器的方式，原理很简单，但是缺点也很明显，需要自己写一个本地代理服务器或者使用第三方库httpSever。如果使用httpSever作为本地代理服务器，如果只缓存一个视频是没有问题的，如果缓存多个视频互相切换，本地代理服务器提供的数据很不稳定，crash概率非常大。

这里我采用ios7以后系统自带的方法实现视频边下边播，这里的边下边播不是单独开一个子线程去下载，而是把视频播放的数据给保存到本地。简而言之，就是使用一遍的流量，既播放了视频，也保存了视频。

用到的框架：<AVFoundation/AVFoundation.h>

用到的播放器：AVplayer

先说一下avplayer自身的播放原理，当我们给播放器设置好url等一些参数后，播放器就会向url所在的服务器发送请求(请求参数有两个值，一个是offset偏移量，另一个是length长度，其实就相当于NSRange一样)，服务器就根据range参数给播放器返回数据。这就是大致的原理，当然实际的过程还是略微比较复杂。

下面进入主题

产品需求：

- 1.支持正常播放器的一切功能，包括暂停、播放和拖拽
- 2.如果视频加载完成且完整，将视频文件保存到本地cache，下一次播放本地cache中的视频，不再请求网络数据
- 3.如果视频没有加载完（半路关闭或者拖拽）就不用保存到本地cache

实现方案：

- 1.需要在视频播放器和服务器之间添加一层类似代理的机制，视频播放器不再直接访问服务器，而是访问代理对象，代理对象去访问服务器获得数据，之后返回给视频播放器，同时代理对象根据一定的策略缓存数据。
- 2.AVURLAsset中的resourceLoader可以实现这个机制，resourceLoader的delegate就是上述的代理对象。
- 3.视频播放器在开始播放之前首先检测是本地cache中是否有此视频，如果没有才通过代理获得数据，如果有，则直接播放本地cache中的视频即可。

视频播放器需要实现的功能

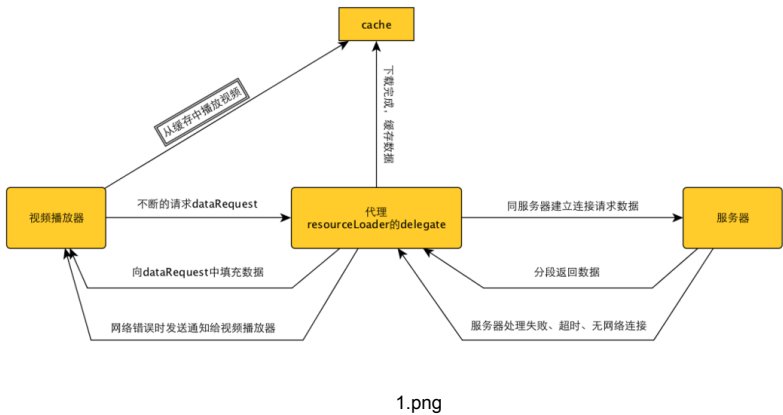


- 1.有开始暂停按钮
 - 2.显示播放进度及总时长
 - 3.可以通过拖拽从任意位置开始播放视频
 - 4.视频加载中的过程和加载失败需要有相应的提示
-

代理对象需要实现的功能

- 1.接收视频播放器的请求，并根据请求的range向服务器请求本地没有获得的数据
 - 2.缓存向服务器请求回的数据到本地
 - 3.如果向服务器的请求出现错误，需要通知给视频播放器，以便视频播放器对用户进行提示
-

具体流程图



视频播放器处理流程

- 1.当开始播放视频时，通过视频url判断本地cache中是否已经缓存当前视频，如果有，则直接播放本地cache中视频
 - 2.如果本地cache中没有视频，则视频播放器向代理请求数据
 - 3.加载视频时展示正在加载的提示（菊花转）
 - 4.如果可以正常播放视频，则去掉加载提示，播放视频，如果加载失败，去掉加载提示并显示失败提示
 - 5.在播放过程中如果由于网络过慢或拖拽原因导致没有播放数据时，要展示加载提示，跳转到第4步
-

代理对象处理流程

- 1.当视频播放器向代理请求dataRequest时，判断代理是否已经向服务器发起了请求，如果没有，则发起下载整个视频文件的请求

- 2.如果代理已经和服务器建立链接，则判断当前的dataRequest请求的offset是否大于当前已经缓存的文件的offset，如果大于则取消当前与服务器的请求，并从offset开始到文件尾向服务器发起请求（此时应该是由播放器向后拖拽，并且超过了已缓存的数据时才会出现）
- 3.如果当前的dataRequest请求的offset小于已经缓存的文件的offset，同时大于代理向服务器请求的range的offset，说明有一部分已经缓存的数据可以传给播放器，则将这部分数据返回给播放器（此时应该是由播放器向前拖拽，请求的数据已经缓存过才会出现）
- 4.如果当前的dataRequest请求的offset小于代理向服务器请求的range的offset，则取消当前与服务器的请求，并从offset开始到文件尾向服务器发起请求（此时应该是由播放器向前拖拽，并且超过了已缓存的数据时才会出现）
- 5.只要代理重新向服务器发起请求，就会导致缓存的数据不连续，则加载结束后不用将缓存的数据放入本地cache
- 6.如果代理和服务器的链接超时，重试一次，如果还是错误则通知播放器网络错误
- 7.如果服务器返回其他错误，则代理通知播放器网络错误

resourceLoader的难点处理

```

- (BOOL)resourceLoader:(AVAssetResourceLoader *)resourceLoader shouldWaitForLoadingOfResources:(AVAsset *)asset {
    {
        [self.pendingRequests addObject:loadingRequest];
        [self dealWithLoadingRequest:loadingRequest];

        return YES;
    }
}

```

播放器发出的数据请求从这里开始，我们保存从这里发出的所有请求存放数组，自己来处理这些请求，当一个请求完成后，对请求发出finishLoading消息，并从数组中移除。正常状态下，当播放器发出下一个请求的时候，会把上一个请求给finish。

下面这个方法发出的请求说明播放器自己关闭了这个请求，我们不需要再对这个请求进行处理，系统每次结束一个旧的请求，便必然会发出一个或多个新的请求，除了播放器已经获得整个视频完整的数据，这时候就不会再发起请求。

```

- (void)resourceLoader:(AVAssetResourceLoader *)resourceLoader didCancelLoadingRequest:(AVAsset *)asset {
    {
        [self.pendingRequests removeObject:loadingRequest];
    }
}

```

下面这个方法是对播放器发出的请求进行填充数据



```

- (BOOL)respondWithDataForRequest:(AVAssetResourceLoadingDataRequest *)dataRequest
{
    long long startOffset = dataRequest.requestedOffset;

    if (dataRequest.currentOffset != 0) {
        startOffset = dataRequest.currentOffset;
    }

    if ((self.task.offset + self.task.downLoadingOffset) < startOffset)
    {
        //NSLog(@"NO DATA FOR REQUEST");
        return NO;
    }

    if (startOffset < self.task.offset) {
        return NO;
    }

    NSData *filedata = [NSData dataWithContentsOfURL:[NSURL URLWithString:_videoPath]];

    // This is the total data we have from startOffset to whatever has been downloaded
    NSInteger unreadBytes = self.task.downLoadingOffset - ((NSInteger)startOffset) - 1;

    // Respond with whatever is available if we can't satisfy the request fully yet
    NSInteger numberOfBytesToRespondWith = MIN((NSInteger)dataRequest.requestedLength, unreadBytes);

    [dataRequest responseData:[filedata subdataWithRange:NSMakeRange((NSInteger)startOffset, numberOfBytesToRespondWith)]];

    long long endOffset = startOffset + dataRequest.requestedLength;
    BOOL didRespondFully = (self.task.offset + self.task.downLoadingOffset) >= endOffset;

    return didRespondFully;
}

```

这是对存放所有的请求的数组进行处理

```

- (void)processPendingRequests
{
    NSMutableArray *requestsCompleted = [NSMutableArray array]; //请求完成的数组
    //每次下载一块数据都是一次请求，把这些请求放到数组，遍历数组
    for (AVAssetResourceLoadingRequest *loadingRequest in self.pendingRequests)
    {
        [self fillInContentInformation:loadingRequest.contentInformationRequest]; //填充内容信息

        BOOL didRespondCompletely = [self responseDataForRequest:loadingRequest];

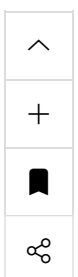
        if (didRespondCompletely) {
            [requestsCompleted addObject:loadingRequest]; //如果完整，把此次请求放进 请求完成的数组
            [loadingRequest finishLoading];
        }
    }

    [self.pendingRequests removeObjectsWithIdentifiers:requestsCompleted]; //在所有请求的数组中移除已经完成的请求
}

```

resourceLoader的难点基本上就是上面这点了，说到播放器，下面便顺便讲下AVPlayer的难点。

难点：对播放器状态的捕获



举个简单的例子，视频总长度60分，现在缓冲的数据才10分钟，然后拖动到20分钟的位置进行播放，在网速较慢的时候，视频从当前位置开始播放，必然会出现一段时间的卡顿，为了有一个更好的用户体验，在卡顿的时候，我们需要加一个菊花转的状态，现在问题就来了。

在拖动到未缓冲区域内，是否需要加菊花转，如果加，要显示多久再消失，而且如果在网速很慢的时候，播放器如果等了太久，哪怕最后有数据了，播放器也已经“死”了，它自己无法恢复播放，这个时候需要我们人为的去恢复播放，如果恢复播放不成功，那么过一段时间需要再次恢复播放，是否恢复播放成功，这里也需要捕获其状态。所以，如果要有一个好的用户体验，我们需要时时知道播放器的状态。

有两个状态需要捕获，一个是正在缓冲，一个是正在播放，监听播放的“`playbackBufferEmpty`”属性就可以捕获正在缓冲状态，播放器的时间监听器则可以捕获正在播放状态，我的demo中一共有4个状态：

```
typedef NS_ENUM(NSInteger, TBPlayerState) {
    TBPlayerStateBuffering = 1,
    TBPlayerStatePlaying   = 2,
    TBPlayerStateStopped   = 3,
    TBPlayerStatePause     = 4
};
```

这样可以对播放器更好的把握和处理了。

然后说一说在缓冲时候的处理，以及缓冲后多久去播放，处理方法：

进入缓冲状态后，缓冲2秒后去手动播放，如果播放不成功（缓冲的数据太少，还不足以播放），那就再缓冲2秒再次播放，如此循环，看详细代码：

```
- (void)bufferingSomeSecond
{
    // playbackBufferEmpty会反复进入，因此在bufferingOneSecond延时播放执行完之前再调用bufferingSomeSecond
    static BOOL isBuffering = NO;
    if (isBuffering) {
        return;
    }
    isBuffering = YES;

    // 需要先暂停一小会之后再播放，否则网络状况不好的时候时间在走，声音播放不出来
    [self.player pause];
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(2 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{

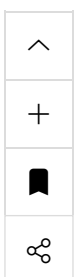
        // 如果此时用户已经暂停了，则不再需要开启播放了
        if (self.isPauseByUser) {
            isBuffering = NO;
            return;
        }

        [self.player play];
        // 如果执行了play还是没有播放则说明还没有缓存好，则再次缓存一段时间
        isBuffering = NO;
        if (!self.currentPlayerItem.isPlaybackLikelyToKeepUp) {
            [self bufferingSomeSecond];
        }
    });
}
```

这个demo花了我很长的时间，实现这个demo我也遇到了很多坑最后才完成的，现在我奉献出来，也许对你会有所帮助。如果你觉得不错，还请为我Star一个，也算是对我的支持和鼓励。

参考文章 (<http://vombat.tumblr.com/post/86294492874/caching-audio-streamed-using-avplayer>)

demo下载地址 (<https://github.com/suifengqjn/TBPlayer>)





夜千寻墨 (/u/527ecf8c8753) ♂

写了 32134 字，被 1226 人关注，获得了 1308 个喜欢

(/u/527ecf8c8753)

✓ 已关注

iOS Developer，记录对技术思考和学习的点滴！

多谢打赏！

赞赏支持




喜欢 692



更多分享


(http://cwb.assets.jianshu.io/notes/images/3012594



写下你的评论...

272条评论 只看作者

按喜欢排序 按时间正序 按时间倒序



WS_0909 (/u/c693e77d617c)

144楼 · 2017.05.31 10:02

(/u/c693e77d617c)


楼主 Demo 台挺好的 但是现在还不支持划屏快进快退吧 能不能把这个工能也优化上呢 期待期待期待@_@

赞 回复

洛洛爱吃肉 (/u/c546ef1696d8): 滑屏快进快退啥意思

2017.06.30 09:58 回复

添加新评论



MrLittleWhite (/u/c31a50d1e8fc)

145楼 · 2017.06.09 14:26

(/u/c31a50d1e8fc)

可以参考一翻,使用avplayer发现,有时候缓存比较多了,调用play依然没办法播放,必须等缓存很多很多了,才会播放

赞 回复



子墨兮 (/u/e3ba403fa75b)


146楼 · 2017.07.04 11:09

(/u/e3ba403fa75b)

http://group.file.dachentech.com.cn/o_1bjrg7i8p40211ad4si1b94c4o4s

(http://group.file.dachentech.com.cn/o_1bjrg7i8p40211ad4si1b94c4o4s) 播放不了，浏览器播放正常

赞 回复



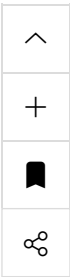
小帅er (/u/413c90573214)

147楼 · 2017.07.04 14:35

(/u/413c90573214)

下了demo但是没有缓存到本地 每次进去还是重新加载

赞 回复





Rumbles (/u/732caf2741b1)
148楼 · 2017.07.13 14:56

(/u/732caf2741b1)
我很奇怪，用大神写的这个播放视频可以边下边播，播放音乐就必须全部下载完才可以播

👍 赞 💬 回复



YoChar (/u/093c8f822f69)
149楼 · 2017.07.14 15:50

(/u/093c8f822f69)
楼主，你这个框架有记忆播放的功能吗。就是比如播放列表有几个视频，每个视频播放了一段换另一个视频播放。待会回来播放刚才播放过的视频，就从该视频之前退出的地方继续播放。

👍 赞 💬 回复



雪豹1119 (/u/b9f9752e5c93)
150楼 · 2017.07.28 10:10

(/u/b9f9752e5c93)
楼主，有一个问题，当我循环播放视频时，有可能导致重新下载数据，请问怎么解决呢？用github上的demo也是同样的问题。

👍 赞 💬 回复

田伟_a4d9 (/u/4a9a6d37177a): 你在第一次播放后，文件全部缓存下来了？
2017.07.29 16:53 💬 回复

雪豹1119 (/u/b9f9752e5c93): @田伟_a4d9 (/u/4a9a6d37177a) 是的，文件都已经下载到了本地，都可以导出来看到，但是有时AVPlayer还是会重新请求网络。按理说，avplayer第一次全部播放完成，会缓存起来，当循环播放的时候不会再次请求网络，会用缓存的数据。
2017.07.31 15:31 💬 回复

✍️ 添加新评论



田伟_a4d9 (/u/4a9a6d37177a)
151楼 · 2017.07.29 16:47

(/u/4a9a6d37177a)
lz你好，用你的思路，很好的解决了我的问题，感谢。
但我还1个疑问和1个问题，
疑问：shouldWaitForLoadingOfRequestedResource中，即在正常播放，没有任何快进已退的情况下，播放器每次请求的offset顺序感觉完全是乱来的，感觉应该按顺序来请求offset才对呢，这个情况lz知道么？
问题：我在播放几百m以下的小文件时没有问题，在播放如上G的文件，播放时会出现固定在某个时间点开始，画面不动，声音和时间都继续走的情况，播放多个大文件都有些情况，lz可曾遇到过这样的问题？

👍 赞 💬 回复



正直走 (/u/5022b74c23cd)
152楼 · 2017.08.16 17:29

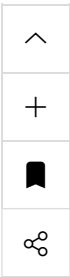
(/u/5022b74c23cd)
你好在看完运行完您代码后，发现TBPlayer在dismiss后重新点击播放按钮，则会出现莫名的问题即有时候无法弹出界面有时播放出现问题，可能是 [self.player replaceCurrentItemWithPlayerItem:self.currentPlayerItem]处问题，望作者指点，急等谢谢。

👍 赞 💬 回复



劍靈 (/u/3409608c2b57)
153楼 · 2017.09.14 15:48

(/u/3409608c2b57)



博主大神，请问下播放大视频的时候崩掉了，崩在这行代码了
[dataRequest responseData:[filedata
subdataWithRange:NSMakeRange((NSInteger)startOffset- self.task.offset,
(NSInteger)numberOfBytesToRespondWith)]];
请问这是为什么呢？

👍 赞 💬 回复



Sky521 (/u/a1d994e5ca49)
154楼 · 2017.09.26 17:32

(/u/a1d994e5ca49)
博主，AVFoundation怎么控制缓存进度？当加载url播放，它是直接一次性的缓存全部的
视频数据到内存中，可以控制这个进度在当前播放时间的1min内吗？我看爱奇艺就是这样
做的，感谢

👍 赞 💬 回复



HAORAN_Z (/u/9d0bfdb01ac3)
155楼 · 2017.10.19 09:47

(/u/9d0bfdb01ac3)
你好，麻烦请问一下，你在TBPlayer中自定义了多个通知名称，我只看到发送自定义的
通知，没有看到接收自定义的通知，你可以解释一下吗？

👍 赞 💬 回复



射手豆豆 (/u/962595458b35)
156楼 · 2017.10.24 20:55

(/u/962595458b35)
请教大侠，在4G环境下，暂停视频播放的同时也想暂停视频的缓冲，从而节省流量，有
什么方法吗？

👍 赞 💬 回复

被以下专题收入，发现更多相似内容

+ 收入我的专题



首页投稿 (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)



iOS进阶市集 (/c/49bfa36740cc?utm_source=desktop&utm_medium=notes-included-collection)



iOS学习 (/c/1332c736fe39?utm_source=desktop&utm_medium=notes-included-collection)



寒哥管理的技术专题 (/c/5be41e88940c?utm_source=desktop&utm_medium=notes-included-collection)



就是你 (/c/77c11d34b443?utm_source=desktop&utm_medium=notes-included-collection)

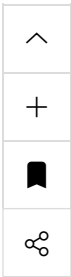


IT (/c/602efb9441ff?utm_source=desktop&utm_medium=notes-included-collection)



iOS开发 (/c/da553370c834?utm_source=desktop&utm_medium=notes-included-collection)

展开更多



2015-10-22--退出键盘的几种方式,获得手势位置 (/p/6dd007b90f08?utm_c...

1.最常用的textView取消第一响应者: [textViewresignFirstResponder];2.ios7以后系统短信方式, 退出键盘, 可拉可压,只要是scrollView的子类都可以实现scrollView.keyboardDismissMode=UISc...

夜千寻墨 (/u/527ecf8c8753?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

通过dsYM定位线上版crash位置 (/p/0885d8b3d2a7?u... (/p/0885d8b3d2a7?
utm_campaign=maleskine&utm_content=note&utrn

先看友盟统计的错误详情再看看dsYM的界面怎么使用一目了然, 不过值得注意的是: 每次提交appStore的时候, 需要把你的app的dsYM文件保存起来。ds...

夜千寻墨 (/u/527ecf8c8753?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

无论你经历什么样的面试, 你都可以这样去表达 (/p/bb... (/p/bb4608cc7a24?
utm_campaign=maleskine&utm_content=note&utrn

01 你觉得面试难在什么地方? 发挥不出真实水平? 还是感觉面对那种"你的优点 缺点"等世界级的问题, 总是难以组织语言? 其实更可怕的是, 即使你在一场...

道长是名思维贩子 (/u/92eb338437ee?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

我28岁, 仍相信一见钟情 (/p/a7b4a07e87d8?utm_ca... (/p/a7b4a07e87d8?
utm_campaign=maleskine&utm_content=note&utrn

文|菜七 别人说, 女人过了27岁就不要再幻想什么一见钟情了, 那是犯花痴。我可能脑袋有瑕疵, 今年28了, 居然还相信一见钟情。去年八月初, 我打算和...

菜七 (/u/0d3d417cbd02?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

请允许我, 短暂地跟你告别 (/p/0e43aae467f9?utm_c... (/p/0e43aae467f9?
utm_campaign=maleskine&utm_content=note&utrn

01 最近的状态不太好, 消极, 悲观, 自我否定, 常常盯着一处, 思绪漫游, 脑子空白。我尝试把注意力聚焦到一个点上, 专心致志做一件事情, 却发现, ...

寒烟衰草 (/u/6f05d462e90c?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

