

[Blog](#) [Email](#) [GitHub](#)

05 Feb 2015

使用HTTP流播放MP4

MP4 可以用 HTTP 协议来播放么？如果可以，需要我们服务器做哪些处理。咱们今天就来讨论下这个问题。

播放器

可以通过很多播放器来演示播放 HTTP 流的 MP4。比如：

- Mac 上 QuickTime Player 。
- iOS APP 中播放可以使用 AVPlayer 。
- 最简单的播放器是在浏览器中，通过 HTML5 的 video 标签来播放。比如

```
<video><source src="xx" type="video/mp4"></video>
```

当然还有很多其他平台都支持 HTTP 协议播放 MP4，欢迎大家补充。

MP4

MP4 的介绍以及协议细节，可以通过 [维基百科](#) 和 [RFC](#) 了解到，本文不细讲，主要聊聊 MP4 内部的数据结构。MP4 内部的数据结构大概如此：

```
.
|—— ftyp
|—— moov
|   |—— mvhd
|   |—— trak
|   |—— trak
|   |—— trak
|—— free
|—— free
|—— mdat
```

其中 `moov` 记录这 MP4 视频的元信息，特别是 `trak` 记录着视频播放数据的时间和空间信息。而 `mdat` 则是保存着视频音频信息。而视频的拖动，快进，都是需要根据 `moov` 和拖动至的时间，来计算要 `seek` 到的文件位置。但是 MP4 文件，不一定 `moov` 就在文件开头，也有可能是在文件末尾。

通过 HTTP 流播放

上面提到, `moov` 和 `mdat` , 如果播放本地文件, 很好办:

1. 先读取 `moov` , 头部没有, 就从尾部读取。
2. 根据 `moov` 和拖动时间, 来计算要目标文件位置后进行 `seek` 操作即可。

但是通过 HTTP 实现这两部就艰难很多:

1. 读取 `moov` 信息。做法有很多种, 我也是通过 Chrome 和 Safari 抓包得来的:
 - 直接发起 HTTP MP4 请求, 读取响应 body 的开头, 如果发现 `moov` 在开头, 就接着往下读 `mdat` 。如果发现开头没有, 先读到 `mdat` (moov 一般都比较小), 立马 RESET 这个连接, 节省流量, 通过 Range 头读取文件末尾数据, 因为前面一个 HTTP 请求已经获取到了 Content-Length , 知道了 MP4 文件的整个大小, 通过 Range 头读取部分文件尾部数据也是可以读取到的。
 - 我看 Safari 有另外一个做法, 先通过 `Range: bytes=0,1` 发起请求, 目的不在获取文件内容, 而是通过 Content-Range 获取文件大小。然后进行上面的步骤。
2. 根据 `moov` 和拖动时间, 来计算要 `seek` 的目标文件位置, 但是有可能文件仍然在下载, 目标文件位置还没有下载呢。那可以通过 Range 来重新发起 HTTP 请求, 获取指定的文件片段。

所以要通过 HTTP 流播放 MP4, 我们需要做到哪些工作:

1. 必须将 `moov` 放在 `mdat` 前面。可以通过 `ffmpeg` 指定参数 `-movflags faststart` 进行移动。
2. HTTP 请求返回的 `Content-Type` 必须是 `video/mp4` 。
3. 服务器必须支持 HTTP 1.1 的 `Range` 。
4. 确保服务器没有对 MP4 文件进行 gzip 压缩, 不然怎么读取 `moov` 啊。

最后

Apple 早已推出 [HTTP Live Streaming\(HLS\)](#) , 更适合移动网络下的播放, iOS 3.0以上 和 Android 3.0 以上都支持, 我们下次再讲。

References && Resources:

- [HTML5 VIDEO bytes on iOS](#)
- [HTML5 - How to stream large .mp4 files?](#)