如今在这个特殊的时代下：flash将死未死，微软和IE的历史问题，html5标准未定，苹果和谷歌的闭源和开源之争，移动互联网的大势所趋，浏览器各自为战...这些都导致web开发者在设计视频解决方案的时候相当困惑。本文围绕这个主题，来探讨一下相关的技术，原理和工具。

### **编码与格式的误区**

很多人将编码和格式误认为是同一个东西，往往以视频文件的后缀来唯一确定视频文件的支持程度。而事实上，用一句话来概括就是：视频的文件后缀（假设没有恶意修改后缀）实际上代表一种封装格式，而视频或者音频的编码算法与封装格式本身无直接的关系：同样的封装格式（即同样的后缀）可以封装不同编码算法的视频和音频。而视频播放设备或软件是否支持视频的播放，不仅仅要看封装格式，还要看编码算法。认清这一点是理解和排查问题的基础。

封装格式规定了视频的所有内容，包括图像，声音，字幕，系统控制等，其中以图像和声音最为关键。

### **从MPEG说起**

MPEG是一个定义视频规格的国际组织，他们曾经推出的MPEG-1和MPEG-2实际上分别就是大家熟知的VCD和DVD，不过这都是远古的东西了。我们来看看跟本文主题有关的MPEG-4规范。

[MPEG-4](http://zh.wikipedia.org/wiki/MPEG-4" \t "http://www.cnblogs.com/P_Chou/p/_blank)规范规定了文件后缀名为.mp4，目前包括三种图像编码和压缩算法：[Xvid](http://zh.wikipedia.org/wiki/XviD" \t "http://www.cnblogs.com/P_Chou/p/_blank)\DivX\AVC(H.264)，其中Xvid和DivX也可以统称为MPEG-4 Part 2或者MPEG-4 Visual，而更为知名的H.264和AVC是相同的概念。音频方面则是AAC。以下关于兼容的内容，来源于[维基百科](http://en.wikipedia.org/wiki/Html_video_tag" \t "http://www.cnblogs.com/P_Chou/p/_blank)和格式工厂以及笔者的测试：

* Android浏览器：支持DivX和AVC，Xvid应该不支持
* iPhone和iPad（iOS）：支持DivX和AVC，Xvid不支持
* Chrome：支持AVC，不支持DivX和Xvid。谷歌曾在2011年初[宣布](http://www.guao.hk/posts/chrome-removes-h-264-codec.html" \t "http://www.cnblogs.com/P_Chou/p/_blank)由于许可问题，将移除Chrome浏览器对AVC(H.264)的支持。但是直到目前的版本，AVC还在被支持。另外，实际测试下来，如果是DivX和AAC封装在mp4中的话，chrome可以播放，但是只有声音(AAC)。
* Firefox和Opera：还是由于许可的问题，Firefox和Opera逐渐动摇了对AVC的支持，笔者在最新的Firefox中测试AVC仍然可以播放（[维基百科](http://en.wikipedia.org/wiki/Html_video_tag" \t "http://www.cnblogs.com/P_Chou/p/_blank)的解释是可能与系统本身具有解码器有关）；至于DivX和Xvid，笔者在Firefox下的测试结果是不支持。从维基百科的兼容列表看，Opera对AVC支持的不好。
* IE：笔者的IE11能够支持AVC，不支持DivX和Xvid

### **WebM的倡导**

由于AVC(H.264)的授权问题，以Chrome、Firefox、Opera为首的开源阵营开始动摇对AVC的支持，尽管目前这些浏览器仍然能够支持AVC，但是它们也倾向于一个叫做[WebM](http://en.wikipedia.org/wiki/WebM_Project" \t "http://www.cnblogs.com/P_Chou/p/_blank)的开源多媒体项目，该项目包括一个叫VP8的新的开源视频编解码方案。目前VP8已经发展到了VP9。作为封装格式的WebM具有.webm的后缀和video/webm的MIME类型。在音频方面，可以使用[Vorbis](http://en.wikipedia.org/wiki/Vorbis)/[Opus](http://en.wikipedia.org/wiki/Opus_(audio_format))。从兼容性看，Chrome、Firefox、Opera对VP8的兼容性相当理想，但是Safari和IE几乎无法支持。

### **开源的Ogg**

[Ogg](http://en.wikipedia.org/wiki/Ogg" \t "http://www.cnblogs.com/P_Chou/p/_blank)几乎与WebM相同，开源，被广泛的在开源平台支持。其视频编码方案称为[Theora](http://en.wikipedia.org/wiki/Theora" \t "http://www.cnblogs.com/P_Chou/p/_blank)(有VP3发展而来，由[Xiph.org基金会开发](http://xiph.org/)，可被用于任何封装格式)，音频为[Vorbis](http://en.wikipedia.org/wiki/Vorbis)。后缀通常为.ogv或.ogg，MIME类型为video/ogg。在兼容性上，Chrome、Firefox、Opera能够支持（但是Opera在移动平台上无法支持），但是Safari和IE几乎无法支持。

### **Html5方案**

以上的讨论实际上的大前提是：视频基于Html5的<video>方案。现在我们来总结一下兼容性：

| **Codecs/container** | **IE** | **Firefox** | **Safari** | **Chrome** | **Opera** | **iPhone** | **Android** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Theora+Vorbis+Ogg** | · | 3.5+ |  | 5.0+ | 10.5+ | · | · |
| **H.264+AAC+MP4** | 9.0+ | · | 3.0+ | 5.0+‡ | · | 3.0+ | 2.0+ |
| **WebM** | 9.0+\* | 4.0+ |  | 6.0+ | 10.6+ | · | 2.3+ |
| \* IE9 “[只有当用户安装了VP8的编解码器时](http://windowsteamblog.com/windows/b/bloggingwindows/archive/2010/05/19/another-follow-up-on-html5-video-in-ie9.aspx)”才能支持VP8。 | | | | | | | |
| ‡ Google Chrome 2011年[宣布](http://blog.chromium.org/2011/01/more-about-chrome-html-video-codec.html) [放弃H.264](http://blog.chromium.org/2011/01/html-video-codec-support-in-chrome.html), 但是“还没兑现”。 | | | | | | | |

可以看出现在主流的仍然是MP4(AVC)，但是为了解决“开源阵营”对AVC的摇摆不定，可以选择利用video的多源方案，在AVC的基础上额外提供对webm或ogg的支持：

<video poster="movie.jpg" controls>

<source src="movie.webm" type='video/webm; codecs="vp8.0, vorbis"'>

<source src="movie.ogg" type='video/ogg; codecs="theora, vorbis"'>

<source src="movie.mp4" type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'>

<p>This is fallback content</p>

</video>

浏览器会根据自己的偏好来选择具体加载那种格式的流媒体文件，当然服务端必须对同一个视频提供多种格式的支持，具体可以这么做：

1. 提供一个WebM的视频版本（VP8+Vorbis）
2. 提供一个MP4的视频版本（H.264+AAC(low complexity)）
3. 提供Ogg版本（Theora+Vorbis）

服务端推荐使用nginx，尽量注意MIME类型的配置正确

### **旧版本的IE和flash**

在html5流行之前，通用的视频播放解决方案是flash和flv（flash从9开始支持h.264的mp4）。但是随着ios设备的流行，flash已经不是万能药了，越来越多的视频网站提供多元的解决方案，而且偏向于html5：也就是说，通过检测agent是否支持html5来决定使用video还是flash。在面对IE8以下的浏览器时，flash几乎是唯一的选择(silverlight的接受度普遍不高)。

当然针对flash和flv的方案，也有多种实现方法，笔者能够想到的有如下两种：

* 服务端根据agent的类型，输出不同的html，如果支持html5就输出video+mp4(avc)和webm（或者ogg），否则输出flash相关的标签或脚本
* 使用[html5shiv](https://github.com/aFarkas/html5shiv" \t "http://www.cnblogs.com/P_Chou/p/_blank)和[html5-video](http://diveintohtml5.info/j/html5-video.js" \t "http://www.cnblogs.com/P_Chou/p/_blank)是IE也能够支持video标签，并且使用Flash播放器来代替原生的video播放，[参考](http://webdesign.about.com/od/video/ss/html5-video_9.htm" \t "http://www.cnblogs.com/P_Chou/p/_blank)
* 将object内嵌在video中：

<video id="movie" width="320" height="240" preload controls>

<source src="pr6.webm" type="video/webm; codecs=vp8,vorbis" />

<source src="pr6.ogv" type="video/ogg; codecs=theora,vorbis" />

<source src="pr6.mp4" />

<object width="320" height="240" type="application/x-shockwave-flash"

data="flowplayer-3.2.1.swf">

<param name="movie" value="flowplayer-3.2.1.swf" />

<param name="allowfullscreen" value="true" />

<param name="flashvars" value="config={'clip': {'url': 'http://wearehugh.com/dih5/pr6.mp4', 'autoPlay':false, 'autoBuffering':true}}" />

<p>Download video as <a href="pr6.mp4">MP4</a>, <a href="pr6.webm">WebM</a>, or <a href="pr6.ogv">Ogg</a>.</p>

</object>

</video>