

如果某服务器的 Vary 首部看起来像下面这样, 大量不同的 User-Agent 和 Cookie 值将会产生非常多的变体:

```
Vary: User-Agent, Cookie
```

缓存必须为每个变体保存其相应的文档版本。当缓存执行查找时, 首先会对内容协商首部集进行内容匹配, 然后比较请求的变体与缓存的变体。如果无法匹配, 缓存就从原始服务器获取文档。

17.5 转码

403

我们已经讨论了一个机制, 该机制可以让客户端和服务端从某个 URL 的一系列文档中挑选出最适合客户端的文档。实现这些机制的前提是, 存在一些满足客户端需求的文档——不管是完全满足还是在一定程度上满足。

然而, 如果服务器没有能满足客户端需求的文档会怎么样呢? 服务器可以给出一个错误响应。但理论上, 服务器可以把现存的文档转换成某种客户端可用的文档。这种选项称为转码。

表 17-4 列出了一些假设的转码。

表17-4 假设的转码

转换之前	转换之后
HTML 文档	WML 文档
高分辨率图像	低分辨率图像
彩色图像	黑白图像
有多个框架的复杂页面	没有很多框架或图像的简单文本页面
有 Java 小应用程序的 HTML 页面	没有 Java 小应用程序的 HTML 页面
有广告的面	去除广告的面

有 3 种类别的转码: 格式转换、信息综合以及内容注入。

17.5.1 格式转换

格式转换是指将数据从一种格式转换成另一种格式, 使之可以被客户端查看。通过 HTML 到 WML 的转换, 无线设备就可以访问通常供桌面客户端查看的文档了。通过慢速连接访问 Web 页面的客户端并不需要接收高分辨率图像, 如果通过格式转换降低图像分辨率和颜色来减小图像文件大小的话, 这类客户端就能更容易地查看图像比较丰富的页面了。