

Web Cache学习总结

Web Cache在客户端与服务器中间，对服务器响应的页面、文件、图片等进行缓存，再遇到客户端发来的同样的情况，根据情况返回缓存的数据，不需要再从服务器拉取数据。

Web Cache的作用

Web Cache的作用主要是：

- 减少页面响应延迟
- 减少网络开销，节省带宽

Web Cache的类型

1. 浏览器缓存

浏览器划分一部分磁盘空间存储用户所见的展现信息。在用户点击后退按钮或者点击之前看过的页面的时候最有用。

3. 代理缓存

代理为成百上千的用户提供服务。大型公司和ISPs经常在防火墙上安装，或作为独立的设备（通常也称为中间件）。

由于代理缓存既不属于客户端也不属于服务端，独立于网络外，请求需要以某种方式路由到它。一种方法是手动设置浏览器的代理；另一方式是使用Interception proxies存在于网络中，将请求重定向到代理上，这样客户端不需要配置代理，甚至不知道代理的存在。代理缓存是一个共享的缓存。在减少延时和网络开销方面能起到很好的作用。

5. 网关缓存

反向代理缓存、替代缓存。网关缓存也是中间件。不是由网络管理员为了减少带宽而创建。而是由web管理者创建，使站点更易扩展、更可靠、提供性能更好的服务。

CDN (content delivery network) 也属于网关缓存

Web Cache控制缓存的几个Http header

- expires (属于freshness)
需要设置一个绝对的GMT时间作为缓存过期的时间
- cache-control (属于freshness)
 - max-age 生存的秒数，由expires的绝对时间改为相对时间，更灵活
 - s-maxage 与max-age类似，只对代理缓存起作用
 - public 标记响应可以被缓存，如果链接是有权限控制的，默认为私有的
 - private 允许缓存给单独的用户，比如浏览器缓存。共享缓存则不被支持
 - no-cache 在提供缓存版本前，强制缓存服务向后台发送确认信息
 - no-store 任何情况下都不存储副本
 - must-revalidate 告诉缓存必须遵守freshness信息
 - proxy-revalidate 类似must-revalidate，不同在于只适用于代理缓存
- Last-Modified (属于validators)

可以使用这个字段向服务器询问，自从这个时间以后是否有所改变，请求头包含If-Modified-

Since字段。

- ETag (属于validators)

HTTP1.1引入了一个新validator叫做ETag。Tags是服务器生成的唯一标识，每一次展现改变时随之改变。由于服务器控制ETag如何生成，缓存服务器可以通过带有If-None-Match请求来确认展现是否还是原来的。

几乎所有的缓存使用Last-Modified时间作为validators。ETag验证也正在变得流行。

使用Web Cache的几点建议

- 使用一致的URLs。
- 使用一个通用的图片库
- 使缓存存储不经常改变的图片 and 页面。使用Cache-Control: max-age, 赋予一个较大的值。
- 使缓存能够识别定期更新的页面。设置合适的max-age或过期时间。
- 如果一个资源（尤其是下供下载的资源）改变了，改变它的名字。这样可以使它在很远的未来过期，仍然保证正确的版本在线。链接到它的额页面则需要一个短的过期时间。
- 不要改变不必要的文件。如果这样做，所有的东西将有一个错误的Last-Modified时间。
- 只在必要的时候使用cookies。cookies难于缓存，大多数情况下并不需要，如果必须使用cookie，限制它使用在动态页面上。
- 使用REDBot检查页面。