

观察news.163.com的头信息

请求:

Accept-Encoding:gzip,deflate,sdch

响应:

Content-Encoding:gzip

Content-Length:36093

再把页面另存下来,观察,约10W字节,实际传输的36093字节

原因-----就在于gzip压缩上.

原理:

浏览器---请求----> 声明可以接受 gzip压缩 或 deflate压缩 或compress 或 sdch
压缩

从http协议的角度看--请求头 声明 accept-encoding: gzip deflate sdch (是指
压缩算法,其中sdch是google倡导的一种压缩方式,目前支持的服务器尚不多)

服务器-->回应---把内容用gzip方式压缩---->发给浏览器

浏览<-----解码gzip-----接收gzip压缩内容-----

推算一下节省的带宽:

假设 news.163.com PV 2亿

$2 \times 10^8 * 9 \times 10^4$ 字节 ==

$2 \times 10^8 * 9 * 10^4 * 10^{-9} = 12 \text{K} * \text{G} = 18\text{T}$

节省的带宽是非常惊人的

gzip配置的常用参数

gzip on|off; #是否开启gzip

gzip_buffers 32 4K| 16 8K #缓冲(压缩在内存中缓冲几块? 每块多大?)

gzip_comp_level [1-9] #推荐6 压缩级别(级别越高,压的越小,越浪费CPU计算资源)

gzip_disable #正则匹配UA 什么样的Uri不进行gzip

gzip_min_length 200 # 开始压缩的最小长度(再小就不要压缩了,意义不在)

gzip_http_version 1.0|1.1 # 开始压缩的http协议版本(可以不设置,目前几乎全是1.1
协议)

gzip_proxied # 设置请求者代理服务器,该如何缓存内容

```
gzip_types text/plain application/xml # 对哪些类型的文件用压缩 如txt,xml,html  
,css  
gzip_vary on|off # 是否传输gzip压缩标志
```

注意:

图片/mp3这样的二进制文件,不必压缩

因为压缩率比较小, 比如100->80字节,而且压缩也是耗费CPU资源的.

比较小的文件不必压缩