**另一篇文章：**

**http://code.google.com/p/chromium/issues/detail?id=39206**

**简述：**

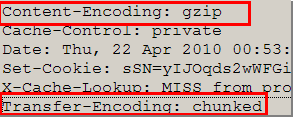
QQ空间提出一个需求，要求qzhttp支持chunked+gzip。每个chunk是一个独立的gzip压缩包，并提到雅虎是这么实现的。

**分析结论：**

1.雅虎的chunked+gzip的方式，每个chunk并不是一个独立的gzip压缩包。

2.如果采用chunked+gzip方式，并且每个chunk都是一个gzip压缩包的方式，浏览器不支持(用ie和chrome测试)。

**一. 首先对于雅虎搜索chunked+gzip的一个抓包分析。**

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static4.photo.sina.com.cn/orignal/622a9970g851e1406ebb3)

第一段chunk,前2个字节是标准的gzip包头0x1f8b

[HTTP <wbr>chunked+gzip及浏览器兼容测试](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static16.photo.sina.com.cn/orignal/622a9970g851e1412fdcf)

第二段chunk,前2个字节则不是标准的gzip包头。

[HTTP <wbr>chunked+gzip及浏览器兼容测试](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static6.photo.sina.com.cn/orignal/622a9970g851e14274195)

如果将yahoo的http回包中头字段和chunk控制字段去掉保存起来，会得到一个标准的gzip包，可以用7-zip打开。

**所以yahoo搜索的chunked+gzip模式中的每个chunk并不是一个可独立解压的gzip包。**

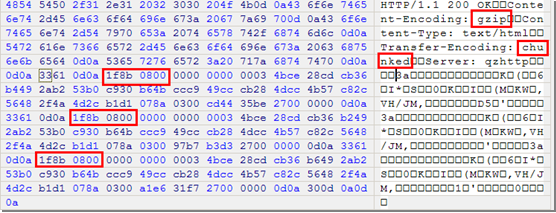
**二. 关于对浏览器对chunked+gzip的支持测试。**

为了快速测试，用php脚本实现了一个简单的web server。（见下）

**1. 每个chunk单独压缩。**

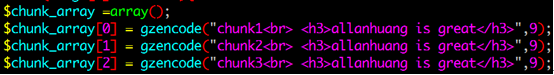
用cat参数启动server （server的php脚本见下）

./chunked\_gzip.php cat

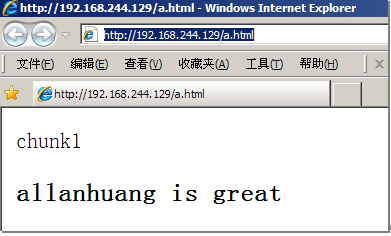
[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static1.photo.sina.com.cn/orignal/622a9970g851e1432a800)

从图中可以看到分了三个chunk，每个chunk都是一个标准的gzip压缩包。

对应php中的代码：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static13.photo.sina.com.cn/orignal/622a9970g851e1443eedc)

IE对此的反应是只显示的第一个chunk.

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static1.photo.sina.com.cn/orignal/622a9970g851e14bf9c20)

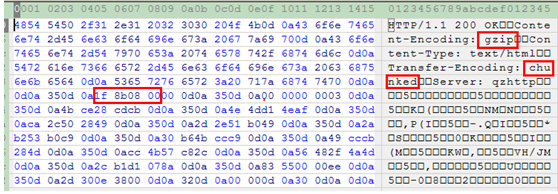
Chrome对此的反应是正确显示第一个chunk，其它的就乱来了。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static1.photo.sina.com.cn/orignal/622a9970g851e14ce67c0)

**2.** **如果将整个包体压缩成一个gzip包，然后分成多个chunk。**

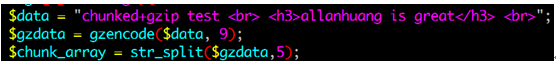
用dog参数启动server（server php脚本见附件）

./trucked\_gzip.php dog

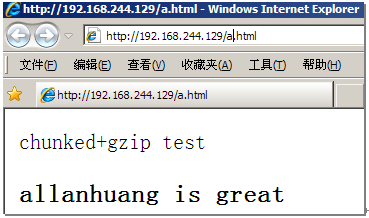
[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static3.photo.sina.com.cn/orignal/622a9970g851e14dc8e72)

从图中可以看到所有的chunk合在一起是一个压缩包，gzip标准头只出现了一次0x1f8b。

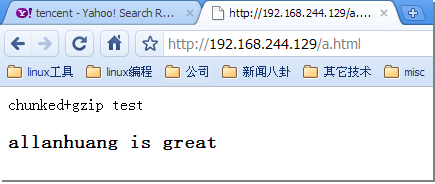
对应php server里面的代码是：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static11.photo.sina.com.cn/orignal/622a9970g851e14ea10ba)

IE对此的反应是能正确显示。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static4.photo.sina.com.cn/orignal/622a9970g851e14fc2733)

Chrome对此的反应同样的正确的。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static1.photo.sina.com.cn/orignal/622a9970g851e150ef470)

**三.** 在wikipedia上有对于chunked+gzip是不是分段压缩的明确说明

原文：HTTP servers sometimes use compression (gzip or deflate) to allow reducing time spent for transmission. Chunked transfer encoding can be used to delimit parts of the compressed object. In this case, it is worth noting that the chunks are **not** individually compressed.Instead, the complete payload is compressed and the output of the compression process is chunked using the scheme described in this article.

译：http server 有时候用压缩来减少传输时间。Chunked方式可以将压缩后的数据分成好几块。在这种情况下，值得注意的是这些chunks并**不是**分别（独立）被压缩的，相反，整个压缩处理后的输出再用前面讲到的chunked方式传输。

<http://en.wikipedia.org/wiki/Chunked_transfer_encoding>

php测试脚本代码：

[](http://photo.blog.sina.com.cn/showpic.html#blogid=622a99700100hskr&url=http://static11.photo.sina.com.cn/orignal/622a9970g851e9071042a)