



傍晚雨

勇于正视任何问题

HTTP/2.0 简单总结（转载）

HTTP/2.0 简单总结（转载于  
https://linjunzhu.github.io/blog/2016/03/10/http2-zongjie/）  
如何使用上 HTTP/2.0

- 1. 需要浏览器的支持，目前最新版的 Chrome、Opera、FireFox、IE11、edge 都已经支持了
- 2. 需要 WEB 服务器的支持，比如 Nginx , H2O

如果浏览器或服务器有一方不支持，那么会自动变成 Http/1.1

HTTP/1.1 时代

HTTP/1.1 对 HTTP/1.0 做了许多优化，也是当今使用得最多的 HTTP 协议：

- 1. 持久化连接以支持连接重用
- 2. 分块传输编码以支持流式响应
- 3. 请求管道以支持并行请求处理
- 4. 字节服务以支持基于范围的资源请求
- 5. 改进的更好的缓存机制

持久连接

在 HTTP/1.0 时代，每一个请求都会重新建立一个 TCP 连接，一旦响应返回，就关闭连接。而建立一个连接，则需要进行三次握手(https的话则是9次握手)，这极大的浪费了性能

因此 HTTP/1.1 新增了「keep-alive」功能，当浏览器建立一个 TCP 连接时，多个请求都会使用这条连接。（现如今大多数浏览器默认都是开启的）

PipeLining 管道

持久连接解决了连接复用问题，但还是存在着一个问题：在一个 TCP 连接中，同一时间只能够发送一个请求，并且需要等响应完成才能够发送第二个请求。

因此 HTTP/1.1 制订了 PipeLining 管道，通过这个管道，浏览器的多个请求可以同时发到服务器，但是服务器的响应只能够一个接着一个的返回（但各大浏览器有些不支持/默认关闭,因此这功能可以说是鸡肋）

因为每一条连接同时只能够返回一个响应，因此浏览器为了改善这种情况，会同时开启4~8个 TCP 连接进行发送请求。

小结

在 HTTP/1.1 时代主要增加了：

- 1. keep-alive 选项，建立连接后，在一定时间内不会断开，其他请求都可以使用这条连接。

2019年2月						
<	日	一	二	三	四	五
	27	28	29	30	31	1
	3	4	5	6	7	8
	10	11	12	13	14	15
	17	18	19	20	21	22
	24	25	26	27	28	1
	3	4	5	6	7	8

我的标签

Sql Server(2)  
SSRS(1)  
sum() over()(1)  
tool(1)  
WindowBatch(1)  
反射泛型类(1)  
前端(1)  
网络原理(1)  
系统设计(1)  
Javascript(1)  
更多

随笔分类

Batch(1)  
C#(1)  
CSS(1)  
EntityFramework(2)  
Html(3)  
Javascript(3)  
jQuery(3)  
Net(1)  
Office(1)  
Oracle(1)  
Reporting Service(1)  
SharePoint2010  
Silverlight(3)  
SqlServer(6)  
编程方法论  
编程思想(1)  
计算机网络(1)

随笔档案

2018年7月 (7)  
2018年2月 (1)  
2017年12月 (1)  
2017年6月 (2)  
2017年5月 (2)  
2016年12月 (1)  
2016年11月 (1)

2. pipelining 管道，通过这个管道，浏览器的多个请求可以同时发到服务器，但是服务器的响应只能一个接着一个的返回（但各大浏览器有些不支持/默认关闭,因此这功能可以说是鸡肋）

## HTTP/1.1 时代的优化

### 1、连接和拼接

连接或拼接JS和CSS文件，雪碧图，以减少HTTP请求，同时浏览器可缓存这些静态资源，为下次访问节约时间。但是这样带来的副作用是，维护成本高，其中某一个小改动都会使得整个拼接后的文件发生改变，重新缓存。

当然并不是说无止境的拼接，建议大小为： 30~50 KB

### 2、域名分区

由于浏览器的限制，同一个域下最多只能建立6个连接。我们通常使用子域名来减少所有资源在只有一个连接时的产生的排队延迟。这个显然不适用在HTTP2中，因为不同的域需要建立不同的连接。

### 3、资源内嵌

对于不常用的，较小资源内嵌在文档中，比如base64的图片，以减少HTTP请求，但是这样的资源不能在浏览器中缓存，也不可能被其他页面共享，同时还有可能编码之后的资源变等更大了。在HTTP2中，这样的资源就可以使用SERVER PUSH来推送。

建议：

1. 只考虑嵌入1~2 KB 以下的资源，因为小于这个标准的资源经常会导致比它自身更高的HTTP 开销
2. 如果文件很小，而且只有个别页面使用，可以考虑嵌入。理想情况下，最好是只用一次的资源
3. 如果文件很小，但需要在多个页面中重用，应该考虑集中打包
4. 如果小文件经常需要更新，就不要嵌入
5. 通过减少 HTTP cookie的大小将协议开销最小化

## SPDY 时代

由于现代网页的不断丰富， HTTP/1.1 协议的性能也逐渐吃不消，因此2012年google如一声惊雷提出了SPDY的方案，实际上，HTTP/2.0 也是以 SPDY 作为原型进行开发的。

### SPDY基础功能

#### 多路复用（multiplexing）

多路复用通过多个请求stream共享一个tcp连接的方式，解决了http1.x holb（head of line blocking）的问题，降低了延迟同时提高了带宽的利用率。

#### 请求优先级（request prioritization）

多路复用带来一个新的问题是，在连接共享的基础之上有可能导致关键请求被阻塞。SPDY允许给每个request设置优先级，这样重要的请求就会优先得到响应。比如浏览器加载首页，首页的html内容应该优先展示，之后才是各种静态资源文件，脚本文件等加载，这样可以保证用户能第一时间看到网页内容。

#### header压缩

前面提到过几次http1.x的header很多时候都是重复多余的。选择合适的压缩算法可以减小包的大小和数量。SPDY对header的压缩率可以达到80%以上，低带宽环境下效果很大。

SPDY 现已经被大多数浏览器以及 WEB 服务器所支持，但为了推进 HTTP/2.0, Google 已经宣布在 2016年对其停止开发。

## HTTP/2.0 时代

2015年5月, HTTP/2.0 在万众瞩目下以RFC 7540正式发表。（热烈鼓掌~啦啦啦啦~~）

## 二进制分帧

在应用层与传输层之间增加一个二进制分帧层，以此达到“在不改动HTTP的语义，HTTP 方法、状态码、URI及首部字段的情况下，突破HTTP1.1的性能限制，改进传输性能，实现低延迟和高吞吐量。”

在二进制分帧层上，HTTP2.0会将所有传输的信息分割为更小的消息和帧,并对它们采用二进制格式的编码，其中HTTP1.x的首部信息会被封装到Headers帧，而我们的request body则封装到Data帧里面。

## 压缩头部

2014年12月 (1)

2013年6月 (1)

2013年3月 (1)

2012年12月 (1)

2012年11月 (2)

2012年10月 (3)

2012年9月 (2)

2012年8月 (1)

### 积分与排名

积分 - 19767

排名 - 26612

### 阅读排行榜

1. JQuery window、document、body(19655)
2. oracle sum(x) over( partition by y ORDER BY z ) 分析(8682)
3. JQuery Offset实验与应用(8619)
4. jquery attr style 选择器在不同浏览器中的表现(2910)
5. HTTP/2.0 简单总结（转载） (978)

### 评论排行榜

1. JQuery Offset实验与应用(11)
2. oracle sum(x) over( partition by y ORDER BY z ) 分析(3)
3. SSRS 使用总结(2)
4. 用代码从文件中导入数据到SQL Server(1)

### 推荐排行榜

1. JQuery Offset实验与应用(6)
2. oracle sum(x) over( partition by y ORDER BY z ) 分析(4)
3. Javascript unescape 的 Sql Server 实现 (2)
4. jquery attr style 选择器在不同浏览器中的表现(1)
5. JQuery window、document、 body(1)

HTTP/2.0规定了在客户端和服务端会使用并且维护「首部表」来跟踪和存储之前发送的键值对，对于相同的头部，不必再通过请求发送，只需发送一次

事实上,如果请求中不包含首部（例如对同一资源的轮询请求），那么首部开销就是零字节。此时所有首部都自动使用之前请求发送的首部。

如果首部发生了变化了，那么只需要发送变化了数据在Headers帧里面，新增或修改的首部帧会被追加到“首部表”。首部表在 HTTP2.0的连接存续期内始终存在,由客户端和服务端共同渐进地更新。

## 多路复用

HTTP/2.0 时代拥有了「多路复用」功能，意思是: 在一条连接上，我可以同时发起无数个请求，并且响应可以同时返回。（这个难点终于被解决了）

客户端和服务端可以把HTTP消息分解为互不依赖的帧，然后乱序发送，最后再在另一端把它们重新组合起来。注意，同一链接上有多个不同方向的数据流在传输。客户端可以一边乱序发送stream，也可以一边接收者服务器的响应，而服务器那端同理。

也就是说，HTTP2.0通信都在一个连接上完成，这个连接可以承载任意数量的双向数据流。就好比，我请求一个页面

<http://www.qq.com%E3%80%82%E9%A1%B5%E9%9D%A2%E4%B8%8A%E6%89%80%E6%9C%89%E7%9A%84%E8%B5%84%E6%BA%90%E8%AF%B7%E6%B1%82%E9%83%BD%E6%98%AF%E5%AE%A2%E6%88%B7%E7%AB%AF%E4%B8%8E%E6%9C%8D%E5%8A%A1%E5%99%A8%E4%B8%8A%E7%9A%84%E4%B8%80%E6%9D%A1TCP%E4%B8%8A%E8%AF%B7%E6%B1%82%E5%92%8C%E5%93%8D%E5%BA%94%E7%9A%84%EF%BC%81>

注意，对一个域名，只需要开启一条 TCP 连接，请求都在这条 TCP 连接上干活。

因此在 HTTP/2.0 时代，之前的合并 JS、CSS 文件技巧，反而不适用了。

## 请求优先级

既然所有资源都是并行发送，那么就需要「优先级」的概念了，这样就可以对重要的文件进行先传输，加速页面的渲染。

## 服务器推送

在 HTTP2.0中，服务器推送是指在客户端请求之前发送数据的机制。如果一个请求是由你的主页发起的，服务器很可能响应主页内容、logo以及样式表，因为它知道客户端会用到这些东西。这相当于在一个 HTML 文档内集合了所有的资源，不过与之相比，服务器推送有一个很大的优势：可以缓存！

## 强制 SSL

虽然 HTTP/2.0 协议并没声明一定要用 SSL，但是 Google Chrome 等浏览器强制要求使用 HTTP/2.0 必须要用上 SSL，也就是说必须要：<https://>

<http://> 将继续使用 http/1.0

## 对优化的影响：

1. 因为“所有的HTTP2.0的请求都在一个TCP链接上”，“资源合并减少请求”，比如CSS Sprites,多个JS文件、CSS文件合并等手段没有效果，或者说没有必要。
2. 因为“多路复用”，采用“cdn1.cn,cdn2.cn,cdn3.cn”，打开多个TCP会话，突破浏览器对同一域名的链接数的限制”的手段是没有必要的。因为因为资源都是并行交错发送，且没有限制，不需要额外的多域名并行下载。
3. 因为“服务器推送”，内嵌资源的优化手段也变得没有意义了。而且使用服务器推送的资源的方式更加高效，因为客户端还可以缓存起来，甚至可以由不同的页面共享（依旧遵循同源策略）

分类： 计算机网络

标签： 网络原理

好文置顶

关注我

收藏该文



傍晚雨  
关注 - 86  
粉丝 - 8

0

0

[+加关注](#)

« 上一篇: [oracle sum\(x\) over\(,partition by y ORDER BY z \) 分析](#)

» 下一篇: [SQL Server 高级SQL](#)

posted @ 2016-11-25 10:08 傍晚雨 阅读(977) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！

【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

相关博文：

- [【转载】.NET 2.0下简单的FTP访问程序](#)
- [\[转载\]RSS 2.0规范](#)
- [HTTP详解\(转载\)](#)
- [HTTP参考 - 转载](#)
- [\(转载\)HTTP解析](#)

最新新闻：

- 网页端Gmail更新：引入更灵活的右键菜单设置
  - Mozilla署名的公开信要求Facebook采取“实际行动”来打击虚假信息
  - 豆瓣App被打低分背后：“打一星”成为表明立场的手段
  - 科学家发现近2000种此前未知的人类肠道细菌
  - 微软和谷歌警告投资者：糟糕的AI会损害他们的品牌
- » [更多新闻...](#)