

浏览器请求队列机制-请求为什么会阻塞

浏览器并发机制

这是一个历史问题：在浏览器刚刚流行的时候，大部分用户是通过拨号来上网，由于受当时的带宽条件的限制，无法使得用户的同时多个请求被处理。同时，当时的服务器的配置也比现在差很多，所以现在每个浏览器的连接数的大小也是有必要的。浏览器默认对同一域下的资源，只保持一定的连接数，阻塞过多的连接,以提高访问速度和解决阻塞问题。不同浏览器的默认值不一样，对于不同的HTTP协议其值也不一样,参看下表:

Browser	HTTP/1.1	HTTP/1.0
IE 6,7	2	4
IE 8	6	6
Firefox 2	2	8
Firefox 3	6	6
Safari 3,4	4	4
Chrome 1,2	6	?
Chrome 3	4	4
Chrome 4+	6	?
iPhone 2	4	?
iPhone 3	6	?
iPhone 4	4	?
Opera 9.63,10.00alpha	4	4
Opera 10.51+	8	?

1. 同时请求15个请求，在Chrome 浏览器中，看到chrome network中有15个请求，而通过charles获取发送请求中，只有6个。这个和上面数据吻合。
2. 当一个请求成功/失败后才会执行下一个请求。
3. 只要请求队列里面还有请求，在排队。那么新添加进去的请求，就必须排队。
4. 队列完毕后，再添加一条能成功发送的请求，那么能正常执行成功。因为队列已经空了，新添加的请求可以马上执行。
5. 这测试证明了浏览器请求是有队列情况，而且在ios手机上并行数目更少4,这也导致了如果有4个请求同时超时就会阻塞。后面的请求就无法发送出去。

解决办法

一. 服务对应域名-增加出口

浏览器是限制一个域名同时发送N个请求，但如果是2个域名都同时请求。那么最大发送请求数 = 域名数目 * 浏览器最大并行请求数；如chrome最大发送请求数：12=2*6。就是2个服务的站点分出去，谁阻塞就找谁服务。跨域建议使用CORS技术或者副域名相同进行跨域（a.ok.com,b.ok.com）*.ok.com 都可以进行跨域。

二. 全部请求设置timeout超时访问时间

加上timeout就会按timeout时间清除队列。保持队列定时清空，就可以发出请求。

主服务设置timeout：10000 副服务设置timeout：8000 主服务和副服务并行：主服务成功；主服务比副服务快：主服务成功；主服务比副服务慢：主服务成功；就算同时100个请求，请求都是按照timeout时间机制。

按时间来做瀑布流使整个站点维持规范，方便请求管理。