**http连接**

**其实常谈论的http连接实际上就是tcp连接加上一些使用连接的规则**

**tcp为http提供了一条可靠的比特传输管道**

**一旦连接建立起来，在客户端和服务器的计算机之间交换的报文就永远不会丢失、受损或失序**

**过程**

1. **通常http事务发生时会经过几个步骤，下面以访问通常http事务发生时会经过几个步骤，下面以访问http://www.xxx.com:80/path/index.html为例说明：**
2. **浏览器从地址栏中解析处域名（主机名），也就是拿到www.xxx.com**
3. **浏览器根据得到的主机名查询出ip地址，比如算出ip为202.43.78.3，（中间可能经过查找host文件或去查询dns服务器）**
4. **浏览器解析出端口（http默认为80，https默认为443）**
5. **浏览器发起一条到202.43.78.3端口为80的链接**

**（重建需要经过几次确定相关参数的来回“握手”）**

1. **浏览器发起请求报文**
2. **服务器返回响应报文**
3. **浏览器关闭连接（其实浏览器和服务器都可以在不通知对方的情况关闭连接）**

http报文流给到tcp，tcp把报文分成一段一段的，然后tcp把每个tcp段交给ip，ip封装成一个ip分组，最后传输的是ip分组(通过tcp通道)

tcp段包括源端口，目的端口

ip段包括源ip，目的ip

**TCP确定一个连接**

* TCP用四个信息来唯一确定一条连接：
* 源ip地址、源端口号、目的ip地址、目的端口号
* 只要其中有一个不同，那么就不是同一条连接。在任意时刻计算机都可以有几条tcp连接在打开状态。

**对TCP性能的考虑**

* 首先相对建立tcp连接、发送http请求报文以及响应报文相比，http事务处理的时间相对短很多很多，此时延可不用讨论，除非你的服务器超负载了或正在处理复杂的运算。因此，http事务的时延往往由以下原因组成：

1. 首先客户端解析ip地址或者端口号需要时间，如果当前没有访问过相关资源，那么解析还需要查询dns服务器，此操作，造成的时延较多，可能花费数十秒。
2. 建立tcp链接会有建立时延，通常2s左右，如果当前的http事务较多，那么会很快叠加上去。
3. 传输、处理请求报文需要时间
4. 回传响应报文需要时间
5. 当然还有其他因素，比如硬件、网络负载，以及报文尺寸等

**性能聚焦区域**

建立tcp链接这个过程可能存在的时延分析，包括：经典三次“握手”、tcp慢启动拥塞控制机制等！

* 经典三次“握手”说的就是http事务在建立tcp连接是需要做的相关参数确认过程，大概如下：

1. 客户端发送携带“SYN”标记的TCP段说明发起连接请求
2. 服务端返回“SYN”和“ACK”的TCP段说明已接受
3. 最后客户端发送确认信息以确认连接

* tcp慢启动说明了，tcp连接会随着时间的增加进行自我调谐。这个主要目的防止突然的tcp连接增多导致网络瘫痪，所以它会慢慢的调整传输速度，这个机制就叫做TCP慢启动

#### HTTP连接的处理

##### 串行处理事务时延

* 此种机制描述了http事务一个一个接着发起，不能同时下载更多的资源，使得界面上用户看不到东西，体验不够好。串行连接没有很好的利用tcp/ip连接的慢启动机制！

##### 并行连接

* 浏览同时发起过个http事务，因为是并行的，所以时延也并行的，这样总时延较小，页面呈现更快，体验较好。但也不是总是这样，因为如果在网络速度很慢的时候，多个连接会去竞争本来不多的带宽，那么就谈不上加快速度了。还有就是并行连接也是需要付出代价的，比如增加系统内训消耗、服务器负载，比如有一个100客户端同时对服务发起100tcp并行连接的话，那么服务器就得负责10000个处理请求，很快的你的服务器就会爆掉。当然了，并行连接确实能带来视觉上的速度提升，因为相比于串行连接慢慢地显示数据而并行一下子能全部显示完信息，视觉上并行连接会给人速度更快的感觉！

##### 持久连接

* 如果对同ip、同端口的发起多个http事务连接，那么可以在前一个事务处理完成之后不要关闭tcp连接，以此来减小建立tcp、tcp慢启动所带来的时延。

##### 管道化连接

HTTP/1.1允许在持久连接上可选地使用请求管道。这是在keep-alive连接上的进一步性能优化。

在响应到达之前，可以将多条请求放入队列。

当第一条请求通过网络流向地球另一端的服务器时，第二条和第三条请求也可以开始发送了。在高时延网络条件下，这样做可以降低网络的环回时间，提高性能。

* 管道连接的限制
  + 如果不是持久连接就不要使用管道连接
  + 接收端必须按收到请求报文的顺序返回响应报文，因为HTTP报文中没有序列号标签。所以必须靠按序发送响应报文来达到“数据对应”
  + 发送端应该做好数据没有发送完连接就关闭的准备并开始重新发送数据。
  + HTTP客户端不应该用管道化的方式发送会产生副作用的请求（比如POST）。