

小心。分组确实会被复制，如果来自之前连接的复制分组插入了具有相同连接值的新 TCP 流，会破坏 TCP 数据。

2MSL 的连接关闭延迟通常不是什么问题，但在性能基准环境下就可能会成为一个问题。进行性能基准测试时，通常只有一台或几台用来产生流量的计算机连接到某系统中去，这样就限制了连接到服务器的客户端 IP 地址数。而且，服务器通常会在 HTTP 的默认 TCP 端口 80 上进行监听。用 TIME_WAIT 防止端口号重用，这些情况也限制了可用的连接值组合。

在只有一个客户端和一台 Web 服务器的异常情况下，构建一条 TCP 连接的 4 个值：

```
<source-IP-address, source-port, destination-IP-address, destination-port>
```

其中的 3 个都是固定的——只有源端口号可以随意改变：

```
<client-IP, source-port, server-IP, 80>
```

客户端每次连接到服务器上去时，都会获得一个新的源端口，以实现连接的唯一性。但由于可用源端口的数量有限（比如，60 000 个），而且在 2MSL 秒（比如，120 秒）内连接是无法重用的，连接率就被限制在了 $60\,000/120=500$ 次/秒。如果再不

85

断进行优化，并且服务器的连接率不高于 500 次/秒，就可确保不会遇到 TIME_WAIT 端口耗尽问题。要修正这个问题，可以增加客户端负载生成机器的数量，或者确保客户端和服务端在循环使用几个虚拟 IP 地址以增加更多的连接组合。

即使没有遇到端口耗尽问题，也要特别小心有大量连接处于打开状态的情况，或为处于等待状态的连接分配了大量控制块的情况。在有大量打开连接或控制块的情况下，有些操作系统的速度会严重减缓。

4.3 HTTP连接的处理

本章的前两节对 TCP 连接及其性能含义进行了精要的介绍。要想学习更多与 TCP 联网有关的知识，请参见本章末尾的资源列表。

现在我们切回到 HTTP 上来。本章其余部分将解释操作和优化连接的 HTTP 技术。我们从 HTTP 的 Connection 首部开始介绍，这是 HTTP 连接管理中一个很容易被误解，但又很重要的部分。然后会介绍一些 HTTP 连接优化技术。

4.3.1 常被误解的Connection首部

HTTP 允许在客户端和最终的源端服务器之间存在一串 HTTP 中间实体（代理、高速缓存等）。可以从客户端开始，逐跳地将 HTTP 报文经过这些中间设备，转发到源