

### 3.4 JMeter 的 WebSocket 实战

对于实时要求高、海量并发的应用，我们常常显得捉襟见肘，尤其在当前移动互联网蓬勃发展的趋势下，高并发与用户实时响应是我们开发 Web 应用时经常面临的需求，例如金融证券的实时信息获取、Web 导航应用中的地理位置获取、社交网络的实时消息推送、多人游戏等场景的实现。WebSocket 协议就比较擅长应用于此类场景。

WebSocket 协议是 Web 客户端和服务端之间新的通信方式，它依然架构在 HTTP 之上。使用 WebSocket 连接而不是以前的 poll 方式，Web 应用程序可以执行实时的交互，如图 3-19 所示。一个 WebSocket 协议是通过一个独立的 TCP 连接实现的、异步的、双向的和全双工的消息传递实现机制。一个全双工的系统允许同时进行双向的通信。

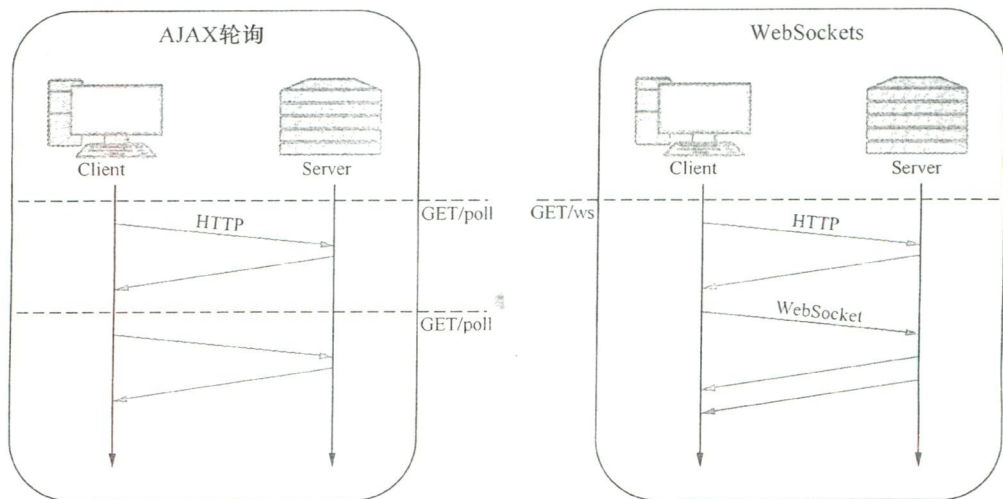


图 3-19 轮询和 WebSocket 请求交互方式

WebSocket 协议利用 HTTP 升级头信息把一个 HTTP 连接升级为一个 WebSocket 连接。HTML5 的 WebSocket 协议解决了许多导致 HTTP 不适合实时应用的问题，并且它通过避免复杂的工作方式使得应用结构很简单。虽然 WebSocket 协议与 HTTP 一样通过已建立的 TCP 连接来传输数据，但是和 HTTP 相比最大优势如下。

(1) WebSocket 协议真正实现了全双工方式。建立连接后客户端与服务器端是完全平等的，可以互相主动请求，即服务器端也能主动地推送消息和发起请求。

(2) HTTP 是传统的客户端对服务器端发起请求的模式。在 HTTP 长连接中，每次数据交换除了真正的数据部分，服务器端和客户端还要大量交换 HTTP 消息头，信息交换效率很低。WebSocket 协议通过第一个请求建立了 TCP 连接后，不需要发送 HTTP 消息头就能交换数据，这显然和原有的 HTTP 有区别，所以需要服务器端和客户端都进行升级才能实现它(主流浏览器都已支持 HTML5)。