基于WebSocket 的实时技术

肖在昌 杨文晖 刘 兵
(成都理工大学信息科学与技术学院，四川 成都 610059)

[摘 要] 很多 Web 应用如股票系统、网页游戏、在线订票系统，都需要将服务器端的变化实时发送到客户端，无需客户端不断发送请求和刷新来实现数据的实时性。本文对目前实时技术进行分析，结合 HTML5 新特性 WebSocket 与基于事件驱动的 NodeJS 构建了一种新的 Web 实时技术。

[关键词] 实时技术；WebSocket；事件驱动；NodeJS

1. 引 言

随着计算机技术的飞速发展，人们对信息实时性需求越来越多，如 :在线订票系统、即时通信系统、股票交易系统等等。在这些需求当中，客户端数据的实时性非常重要。当服务器数据发生变化的时候，需要主动地向客户端实时地发送消息，将最新的数据或事件通知给用户。在传统的 C/S 系统中，这种需求没有任何问题，因为客户端和服务器之间通常存在着持久的连接，这个连接可以双向传递各种数据。而基于 HTTP 协议的 Web 应用却不行。在 Web 世界中，服务器永远是被动地发送数据，前提是客户端必须先发送请求。浏览器其实并不知道服务器的信息什么时候会有改变。于是，针对这种问题的解决方案变得越来越重要。

2. 目前 Web 实时性技术

通常，浏览器访问Web 页面时，一般会向页面所在的 Web 服务器发送一个 HTTP 请求。Web 服务器识别请求后作出响应。当内容从服务器返回到页面上时，可能没有了时效性。随着实时信息对用户越来越重要，不能通过不断手动刷新 Web 页面来获取实时信息。目前，Web 应用基本上通过轮询和其它服务器推送实现信息实时性的，最常用是轮询和长轮询技术。

(1) 使用轮询，浏览器定期发送 HTTP 请求，并随即接受响应，这是最早的一种 Web 实时推送方案。客户端以一定的时间间隔向服务端发出请求，以频繁请求的方式来保持客户端和服务器端的同步。这种同步方案的最大问题是，当客户端以固定频率向服务器发起请求的时候，服务器端的数据可能并没有更新，这样会带来很多无谓的网络传输，所以这是一种非常低效的实时方案。

(2) 长轮询就是浏览器发送一请求到服务器，在一段时间内服务器都处于打开状态。服务器在打开状态时如果收到一个请求，就会发送响应到客户端。长轮询会造出非常多的请求，不断的请求可能会造成的影响是数据顺序无法得到保证。

长连接的技术主要是用 Comet，采用的技术分浏览器来实现，IE 上用 iframe，别的浏览器用 xhr 来实现。长连接即时性非常好，管理起来也相对方便，但由于对服务器的资源会有一个占用，所以相对开销会比较大。

综上所述，目前实现实时数据的方式都会涉及 HTTP 请求和响应报头，这其中有很多不必要的数据会造成传输延迟。

3. 解决方案用到的技术

3.1 WebSocket 介绍

WebSocket 是 HTML5 重要功能，WebSocket 通信协议实现的是基于浏览器的原生 socket，使客户端浏览器具备像 C/S 架构下桌面系统的实时通讯能力。基本就是通过浏览器的支持在 Web 上实现了与服务器端的 socket 通信。WebSocket 没有试图在 HTTP 之上模拟 server 推送，而是直接在 TCP 之上定义了帧协议，因此 WebSocket 能够支持浏览器与服务器之间的双向通信，解决了 Web 实时的问题，对传统的轮询方式和 WebSocket 调用方式作了一个详细的测试和比较，将一个简单的 Web 应用分别用轮询方式和 WebSocket 方式来实现，在这里引用一下他们的测试结果图(图 1)：

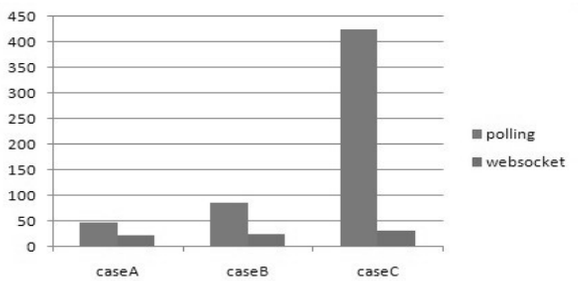

图 1 轮询和 WebSocket 实现方式的网络负载对比图

通过这张图可以清楚看出，在流量和负载增大的情况下，WebSocket 方案相比传统的 Ajax 轮询方案有极大的性能优势。这也是为什么我们认为 WebSocket 是未来实时 Web 服务器推送的首选方案的原因。WebSocket 客户端与服务端建立连接的过程类似于 Http 的建立连接过程，不同的

作者简介 :肖在昌，男，山东济南人，硕士，研究方向 :数字媒体技术，计算机网络，模式识别技术。

是，建立连接之后，接下来客户端和服务端的任何交互都需要再有这个动作。WebSocket 客户端与服务端建立连接的过程如下：

客户端与服务端建立握手，发送如下信息：

GET /echo HTTP/1.1
Upgrade：WebSocket
Connection：Upgrade
Host：127.0.0.1：8088
Origin：http：//127.0.0.1
服务端会发回如下信息：
HTTP/1.1 101 Web Socket Protocol Handshake
Upgrade：WebSocket
Connection：Upgrade
WebSocket-Origin：http：//127.0.0.1
WebSocket-Location：ws：//127.0.0.1：8088/echo

3．2 NodeJS 介绍

在 Web 应用中，客户端与服务器之间的交互响应通常是通过 Ajax 实现的，每次客户端发送一个请求时，都要在网络里走一个来回，服务器都要做出响应并为这个连接生成一个新线程。随着用户增多，做出响应的连接数目就会越来越多，这样会出现内存溢出、逻辑交错带来的冲突、网络瘫痪、系统崩溃等等一系列的问题。用户的每次请求可能需要不同服务器响应，因此，所有的资源要在全部服务器之间共享。这样，Web 应用架构中的瓶颈是：服务器能处理的并发连接的最大数量。

运行 NodeJS 的服务器改变了连接方式，每次连接客户端用发送一个事件到 Node 引擎的进程中取代了为连接生成新线程。NodeJS 不允许使用锁，当然也不会发生死锁现象，它不会直接阻塞 I/O 调用，这样的服务器能支持数万级别的并发连接。对于网络密集型应用，对实时性要求极高，而 Node．js 在网络 I/O 上优势完全可以满足。使用 NodeJS 开发实时 Web 应用服务器的优势有以下几点：

(1) I/O 与高可伸缩性的优势，IO 密集型的应用采用 NodeJS 是最合适的，可达到最好的可伸缩性。

(2) Node．js 采用单线程，使它在处理复杂逻辑的时候无需考虑线程同步、锁、死锁等一系列问题，减少了很多逻辑错误。由多进程 Node．js 组成的服务器群是非常理想的应用架构。

(3) 可扩展性好，Node．js 的模块比较小，所有的库、工具都可以用 npm module 的形式扩展进来，任何第三方都可以根据自己需要开发自定义 module。

(4) 语言优势，使用 JavaScript 开发可以实现快速迭代，客户端使用 HTML5，可实现代码公用，并且与 HTML5 的 WebSocket 完美结合，以达到最大的实时性。

4．构建 Websocket 与 NodeJS 的实时 Web 通信技术

对于客户端而言，WebSocket API 是基于事件的，而对于服务端的处理，可以简单地理解为实现 websocket 协议的 socket server 开发。NodeJS 本身就是一个高效的服务端语言，可以直接使用 javascript 直接来处理来自客户端的请求，这样如果服务端这边需要大量的业务逻辑开发，则可以直接使用 node 开发。通过 Node 和 websocket 的结合可以开发出很多实时性要求很高的 web 应用，如游戏、直播、股票、监控、IM 等等。

NodeJS 如何实现 WebSocket 的支持，其实原理也是很简单，就是用 NodeJS 实现了 WebSocket draft-76 的协议，同时它对外提供了 API，可以方便其它应用程序简化编程。

它继承了 NodeJS 的 http．Server 的事件和方法，这样它简化了服务端的编程，同时可以处理 Http 的请求。为了实现连接之间的通信和消息的广播，它实现了一个 manager 类，给每一个连接创建一个 id，然后在内存中维护一个连接链表，并提供了上线和下线的自动管理。

它还提供对以下几个事件的接口：

listening 当服务器准备好接受客户端请求时
request 当一个 Http 请求发生时触发
stream
close
clientError
error

WebSocket 将服务器端推送的问题变得简单。相比 long polling 和 iframe 隐藏域来说更加容易使用。NodeJS 的事件驱动机制能更好地支持这种服务器端推送。只要在得到连接时注册监听函数监听在某个消息上。当服务器需要统一推送数据时发送消息，则会调用 callback 函数向客户端推送数据。程序的设计会变得简单。

5．浏览器对 WebSocket 的支持

目前一直没有统一的 HTML5 标准，各主流浏览器对 HTML5 个新特性支持也不一样。下面是主流浏览器对 HTML5 WebSocket 的支持情况见表 1：

表 1 各大浏览器对 WebSocket 的支持情况

浏览器	支持情况
Chrome	Supported in version 4+
Firefox	Supported in version 4+
Internet Explorer	Supported in version 9+
Opera	Supported in version 10+
Safari	Supported in version 5+

6．结束语

Web 实时技术有着广泛的应用需求，但是 WebSocket 作为一个正在演变中的 Web 规范，我们也要看到目前用 Websocket 构建应用程序的一些风险。WebSocket 规范目前还处于草案阶段，各浏览器对 HTML5 的支持参差不齐。而 NodeJS 不是适合所有的 Web 应用中，它非常适合处理分布式基于事件的有很高流量的 I/O，不适合大数据处理（例如视频等大型数据）。NodeJS 处在初级阶段，但发展很迅速，相信

很快会在互联网开发中普及。

参考文献：

[1] 张丽．服务器推送技术在 Web 中的应用研究[J]．新乡学院报，2010，27(04)：1- 3.

[2] 姜毅，王兆青，曹丽．基于 HTTP 的实时信息传输方法[J]．计算机工程与设计，2008，29(10)：1- 4.

[3] 景慎艳．基于 Pushlet 的服务器推送技术研究与应用[J]．现代计算机，2009，31(07)：2- 3.

[4] 刘犇，王猛．基于服务器推送技术的 Web 数据实时更新[J]．电脑开发与应用，2011，24(06)：1- 2.

[5] 高鹰，南方新闻网．浅谈基于 HTTP 的长连接原理及应用[J]．信息与电脑，2010，19(04)：1- 2.

Real-Time Technology of Web Based on WebSocket

Xiao Zaichang　Yang Wenhui　Liu Bing
(Chengdu Univercity of Technology，Chengdu 610059，Sichuan)

【Abstract】Many Web applications，such as stock systems，web games and online booking system，should send the changes of the server to the client in real time，so the client don't need to send requests and refresh to achieve real-time data. This acticle focuses on the analysis of server push technology，and builds the new real-time technology based on event-driven NodeJS and WebSocket，which is the new feature of HTML5.

【Keywords】real-time；WebSocket；NodeJS；event-driven

(上接第 39 页)

层面上理解 HTML5 的含义及设计理念，使其充分切入到移动互联网应用当中。

参考文献：

[1] 杨栋梁．移动互联网发展趋势的研究[J]．电脑知识与技术，2012，05：1039- 1042.

[2] 李慧云，何震苇，李丽，陆钢．HTML5 技术与应用模式研究[J]．电信科学，2012，05：24- 29.

[3] 顾旻霞．构建更加开放的移动互联网[J]．信息通信技术，2011，04：54- 57.

[4] 宋明艳．移动互联网应用及其发展分析[J]．电脑与电信，2012，10：35- 37.

Application of HTML5 in the Mobile Internet

Ren Jinbo
(Nanjing Forestry University Library，Nanjing 210037，Jiangsu)

【Abstract】With the rapid development of mobile communication technology innovation and the Internet industry，mobile data services，of which the mobile Internet as a representative，present a rapid growth momentum. And the emergence of HTML5 has brought new changes for mobile Internet. This paper first describes the status of the development and business model of the mobile Internet；and then describes the development and technical characteristics of HTML5；then describes the application of HTML5 in the mobile internet，and gives some examples.

【Keywords】HTML5；mobile internet；web application