文章编号:1002-0640(2014)增刊-0181-03

基于WebSocket的Web实时通信系统

叶忠文,黄 鹏,施金金
(北方信息控制集团有限公司,南京 211153)

摘 要:传统军事信息系统一般采用C/S架构,但是其系统部署成本高,软件升级维护困难。而采用B/S架构则不会存在此问题,但是B/S架构通信量大、网络带宽占用高,不利于在战场环境中进行服务器推送。针对此问题,首先对目前存在的WEB服务器推送技术进行了分析和总结;接着实现了基于WebSocket的服务器推送技术,并通过实验证明了该方法网络带宽占用小,效率高;最后详细介绍了如何实现一个WEB实时通信系统。

关键词:WebSocket,实时通信,B/S
中图分类号:TP393 文献标识码:A

Web Real-time Communication System Based on the Websocket

YE Zhong-wen,HUANG Peng,SHI Jin-jin
(*North Information Control Group Co. LTD*,*Nanjing* 211153,*China*)

Abstract:The military field generally uses the C/S architecture for communication,but it has the high cost of system deployment and the difficulties of upgrade. However,the B/S architecture does not have the problems. In order to solve the problem of high-bandwidth in B/S architecture,firstly,the existing WEB-server-push technologies are analyzed and summarized,and then the paper realizes the WEB-server-push technology based on WebSocket. The experimental results prove that this method is high efficiency and small bandwidth. Finally,a real-time communication system which is based on Web is introduced.

Key words:WebSocket,Real time communication,B/S

引 言

传统军事软件一般采用C/S架构,但是这种架构下,要求服务端和客户端需安装相配套的软件,才可以进行实时通信,这就直接导致系统部署时间长,软件的升级维护也比较困难。而B/S架构客户端只需要部署安装浏览器软件,软件的升级维护只需要在服务端,使得软件的升级维护变得简单便捷,因此,采用B/S模式开发协同平台就显得是大势所趋了,但在军事领域中带宽和传输速率有限,如何能够在较小的网络负载条件下,将消息推送出去就成了问题;并且由于传统桌面软件不可能突然转变为WEB软件,而浏览器端不能够与传统桌面软件进行通信,因此,如何同原有软件进行衔接也成了亟待解决的问题。

1 研究背景

传统上,服务器推送的实时通信方案,比较常用的有短轮询、长轮询以及基于IFrame流的方式,然而这些方案都存在着浪费资源、占用带宽等问题[1]。下面详细介绍这几种服务器推送方式:

⑴ 短轮询。这是最早的一种实现服务器向浏览器端推送数据的方式,其核心思想就是浏览器定期的向服务器发送一个http请求,如果服务器有数据需要向浏览器推送,那么就将数据放在应答包的数据域中。该方法有两个显著的缺点:

①无法确定合适的时间间隔来向服务器发送http请求,如果时间间隔短了服务器压力过大,如果时间间隔长了,时延较长,不能够满足实时向浏览器推送这一要求;

②由于服务器端大部分时间都不会推送数据给客户端,因此,大部分应答包中的数据域为空,浪费了大量的带宽和其他网络资源。

⑵ 长轮询。长轮询是将针对短轮询的无法确定合适的轮询时间等缺点提出的,由浏览器端向服务器发出请求,如果服务器有数据需要向浏览器推送,那么就将数据放在应答包中,如果暂时没有数据需要推送,那么就暂时不进行应答,等到有数据需要推送时再应答[2]。长轮询解决了短轮询无法

收稿日期:2014-05-18 修回日期:2014-06-21
作者简介:叶忠文(1987-),男,安徽芜湖人,硕士研究生。研究方向:WEB应用技术。

确定轮询时间的问题。但是依旧存在着大量占用服务器资源,大量占用带宽和其他网络资源的问题。

⑶ IFrame 流的方式。iframe 流方式是在页面中插入一个隐藏的 iframe,利用其 src 属性在服务器和客户端之间创建一条长链接,服务器向 iframe 传输数据(通常是 HTML,内有负责插入信息的 javascript),来实时更新页面。但使用该方法也会有一个弊端:该方法会阻止 onload 事件的加载。当然可以通过 Google 的一个称为"htmlfile"的 ActiveX 控件解决,但需要安装 ActiveX 控件,在服务器推送上不具有通用性。

2　WebSocket 技术

WebSocket 是 HTML5 开始提供的一种浏览器与服务器间进行全双工通信的网络技术,2011 年被 IETF 定为标准 RFC 6455,WebSocketAPI 被 W3C 定为标准。

实际上 WebSocket 协议可以看成是另一种形式的 Tcp 协议,整个流程与 Tcp 协议极为相似,首先浏览器(支持 Websocket 的浏览器)发起一个请求,然后等待服务端的响应服务器返回握手响应,告诉浏览器请将后续的数据按照 websocket 制定的数据格式传过来浏览器和服务器的 socket 连接不中断,此时这个连接是双工的了。

在 WebSocket 协议中,数据的传输是通过一系列的帧传输实现的,具体的 WebSocket 传输的数据帧格式如图 1 所示。

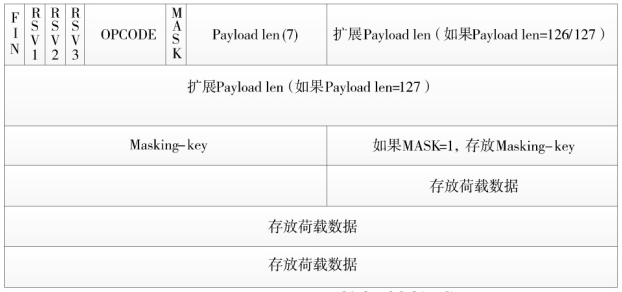

图 1　WebSocket 数据帧格式

其中参数具体意义如表 1 所示。

表 1　数据帧参数表

位名称	所占位数	作用
FIN	1	表明这是消息最后的消息片段
RSV1,RSV2,RSV3	1	自定义协议位
Opcode	4	定义有效负载数据
MASK	1	是否有掩码位
Payload Length	7 或者 7+16 或者 7+64	如果这个值以字节表示是 0~125 这个范围,那这个值就表示传输数据的长度;如果这个值是 126,则随后的两个字节表示的是一个 16 进制无符号数,用来表示传输数据的长度;如果这个值是 127,则随后的是 8 个字节表示的一个 64 位无符号数,这个数用来表示传输数据的长度
Masking-key	0 或 4	掩码位

3　WebSocket 效率分析

本节将对 WebScoket 用于服务器推送的效率进行分析,并且将其与其他推送方法通过实验进行比较。由于 Iframe 流的方式实现上存在着明显的缺陷,而这些缺陷限制了其在服务器推送上的应用,而短轮询的方法,网络带宽占用特别大,在实际应用领域由于已经被长轮询所取代,因此,本文不将这两者作为比较目标。下文比较了 2 种不同的服务器推送技术,一种是基于 Ajax 的长轮询,一种是基于 WebSocket,对比这两种服务器推送技术的效率,从而验证 WebSocket 在实时通信方面的优势。

在该实验中,HTTP 请求头部约为 580 个字节,服务器推送的数据平均长度大约为 20 个字节。建立好连接后,WebSocket 数据帧可以在客户端和服务器之间以全双工模式传输,最小帧只有 2 个字节。当分别有 1 000 个,10 000 个,100 000 个客户端需要进行推送时,网络的吞吐量如图 2 所示。

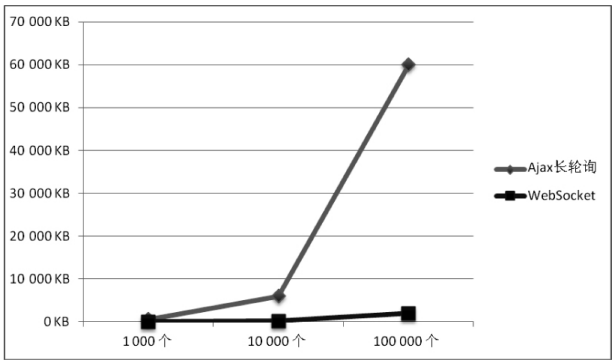

图 2　网络吞吐量对比图

从图 2 可以看出在网络吞吐量方面,基于 Ajax 的长轮询方式的服务器推送技术系统消耗明显高于使用 WebSocket 的方式。这是因为每次进行 Ajax 轮询时便会产生一个完整的 HTTP 报文,每次都会发送 600 个字节,而使用 WebSocket 方式进行推送时,只有在连接建立的阶段需要发送 580 个字节,在建立连接后每次只需要发送数据报文,即只需要发送 20 个字节即可。由于在军用信息系统中,跨车进行数据访问的情况并不多,每次 20 字节完全能够在军用窄带条件下使用,因此,军用信息系统可以采用 WebScoket 技术来进行实时通信。

4　实时通信系统实现

本节将分两个部分详细介绍如何使用 WebSocket 技术建立起一套实时通信程序。第 1 部分详细介绍浏览器端之间如何进行实时通信,第 2 部分主要实现 C/S 架构的程序与浏览器端实时通信。

4.1　浏览器端之间的实时通信

⑴ 服务器端总体架构

服务端实现了一个 MyWebSocketServlet,MyWebSocketServlet 继承自 WebSocketServlet,该程序负责响应浏览器发出的建立 WebSocket 的请求,当 MyWebSocketServlet 收到用户请求建立 WebSocket 的请求后,将该消息发送至 MessageCenter,MessageCenter 负责保存各个 WebSocket 连接,对消息进行分发。

⑵ 登录过程

当用户登录成功之后,前端页面向后台 MyWebSocketServlet 请求一个 WebSocket 连接,MyWebSocketServlet 收到请求后,将该消息发送至 MessageCenter,MessageCenter 收到请求连接的消息后,为该连接分配资源,并将该连接存入连接

池中,最后将该连接返回给浏览器端。

⑶ 消息发送过程

当用户 A 需要发送消息给用户 B 时,首先前端页面将消息封装成如表 2 所示的 JSON 结构,并通过已经建立起来的 WebSocket 发送给 MessageCenter,MessageCenter 收到之后,首先解析数据域中的该 JSON 结构,查找连接池如果发现收消息方已经建立过连接,则取得该连接,并通过该连接将消息推送给用户 B。具体消息发送顺序图如图 3 所示。

表 2 JSON 消息结构表

From	消息发送者
To	消息接收者
Data	数据域
Emergency	紧急消息

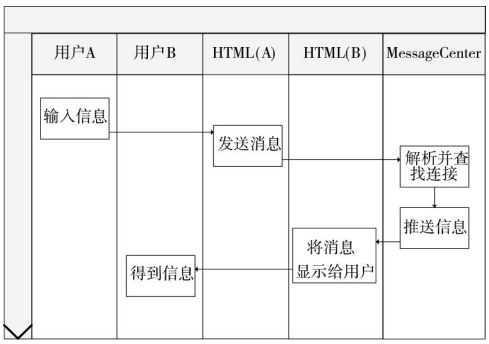

图 3 消息发送顺序图

这样就实现了一个基于 WebSocket 的实时通信程序,服务器能够通过 WebSocket 向浏览器进行消息的推送,通过该程序用户能够通过浏览器进行实时通信。

4.2 浏览器端与桌面程序之间的实时通信

桌面程序将浏览器端订阅的信息发送给消息中间件,消息中间件根据策略将消息分发给消息代理 (本系统使用 JAVA 编写代理),消息代理得到消息后通过 HTTP 请求,将消息交给 WEB 服务器处理,处理后通过 WebSocket 发给相应的浏览器端;最后浏览器从 WebSocket 中获取消息,这样就实现 C/S 架构程序与浏览器的通信。具体的流程图如图 4 所示。

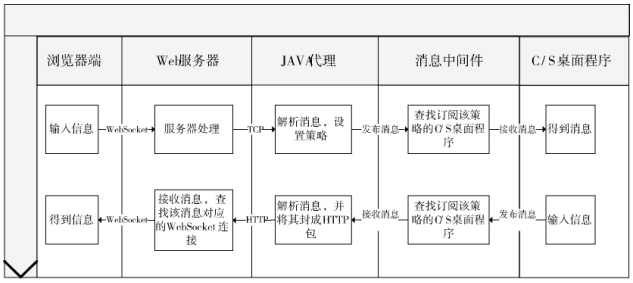

图 4 异构消息发送流程图

5 结 论

近年来,随着互联网技术和 Web 技术的发展,将实时应用引入到 Web 应用领域逐渐成为一种趋势。在这种趋势之下,一些实时 Web 应用技术被开发出来并得到了应用,也使得实时 Web 应用技术成为了当前 Web 技术的一个重要研究方向。本文首先介绍了传统的服务器推送技术,并对各种服务器推送技术的优缺点做了分析,之后着重介绍了 WebSocket 技术,接着通过实验比较 WebSocket、轮询的网络吞吐量,得出 WebSocket 在服务器推送的实时通信领域具有明显的优势;最后详细介绍了基于 WebSocket 的实时通信系统,该系统不仅能够完成浏览器之间信息的实时交互,并且通过消息中间件能够与原有的 C/S 桌面程序进行实时通信。

参考文献:

[1] 孙清国,朱 玮,刘华军,等.Web 应用中的服务器推送技术研究综述 [J]. 计算机系统应用,2008,17 (11):116-120.
[2] 蔡骥然,曹海传.B/S 架构下基于 OPC 与 Comet 技术的实时监控系统 [J].计算机应用,2012,32 (S2):214-216.

(上接第 180 页)

后的密文中相同字符的明文加密出的密文是不一样的,没有规律可循。第二列采用的是单表体制的加密算法,从表中可以看出字符“中国梦”、“春之声”重复出现的时候,加密出的密文是一致的。在密文中出现的字母是随机产生的,不是加密算法计算出来的,是属于混淆密文的一种手段。单表体制的加密,可以通过重复出现的密文进行推测判断并加以破译。相比之下,双表体制加密的密文的破译难度就明显增加。RSA 加密算法的公钥私钥是在考虑了时间开销的前提下选择的,在采用双表体制的情况下即使密文解密的时候被各种破译手段破译出私钥以后,字符也是经过在加工处理后加密的,不知道加密算法机制,没有加密算法中的 X,是无法在很短的时间内正确解析出明文的。

3 结 论

Eclipse 为创建可扩展的集成开发环境提供了一个开放源码平台,Eclipse 插件开发机制解决了不同开发商所开发的软件之间的无缝集成问题[2]。插件式的加密工具方便了二次开发人员的使用。文件加密工具采用的双表体制的 RSA 加密算法,能够一定程度上解决大素数的产生技术的限制和幂乘运算的大量的时间开销。双表体制加密算法能够以折中的方式考虑和解决安全性问题,能够避免被概论攻击法破译。本文的算法提高了加密算法的速度,但网页解密的效率和安全性还有待做进一步的测试。

参考文献:

[1] 林培通. 文件加密系统设计与实现 [J]. 电脑知识与技术,2011,14 (7):3299-3301.
[2] 谷 钰,杨艳斌,王泽生. Eclipse 插件体系结构的研究 [J]. 电脑知识与技术,2009,31 (5):8706-8708,8711.
[3] 宋秀丽. 现代密码学原理与应用 [M]. 北京:机械工业出版社,2012.
[4] 田军舰,寇应展,陈财森. RSA 公钥密码计时攻击研究与仿真 [J].计算机技术与发展,2010,20 (8),150-158.
[5] 姜 楠,金英善,崔晓锋,等. 基于 RSA 文件加密系统的设计 [J].辽宁民族学院报,2013,15 (5):535-538.