

基于 php 和服务端推技术的 Web 即时聊天系统^①

王振兴, 黄 静

(天津大学 管理与经济学部, 天津 300072)

摘 要: 基于 http 协议应用于 Web 端, 实现一个浏览器无关的、便于移植的、高性能的 Web 即时聊天系统. 系统使用服务器推技术中的 ajax 长轮询模型构建 http 通讯模型, 利用开源 LAMP 架构搭建服务器端程序, 并使用 XML 文件系统存储即时聊天内容, 前端使用 javascript 的 jquery 框架实现与浏览器无关的 ajax 前端程序. 提供包括文本表情模式的聊天, 便于整合到社交类型的网站中. 项目实践表明, 基于 php 和服务端推技术的 Web 即时聊天系统稳定性高, 具有一定的使用价值.

关键词: Web; 即时聊天; 服务器推; php; ajax 长轮询

Web Instant Messaging System Based on Server Push Technology and php

WANG Zhen-Xing, HUANG Jing

(Department of Management and Economics, Tianjin University, Tianjin 300072, China)

Abstract: Based on the http protocol used in the Web side, it realizes a not browser based, easy to transplant, high-performance real-time web chat system. System uses the ajax long polling model of the server push technique to build the http communication, using the open source LAMP architecture to build server-side programs, and use the xml file system to store real-time chat content, the front using the javascript jquery framework to achieve and browse device-independent ajax front-end program. Text chat and expressions, and ease of integration into the sns type of site. Project practice shows that the web php-based server push model real time chat system stability, it has a certain value.

Key words: web; instant messaging system; server push technology; php; ajax long polling

随着 web2.0 的发展, 即时聊天也开始 Web 化. 即时通讯(Instant Messaging, 简称 IM) 是指使用在线识别用户和实时交换信息技术, 依靠互联网平台和移动通讯平台, 以多种信息格式(文字、图片、声音、视频等)沟通为目的, 通过多平台、多终端的通讯技术来实现的同平台、跨平台的低成本高效率的综合通讯方式^[1,2]. Web 即时聊天是即时通讯的一种实现, 主要信息格式是文字. 现在很多 Web2.0 模式下的网站都有即时聊天功能, 可以和在线的好友聊天. 例如人人网、新浪微博、腾讯微博和 facebook 等. Web 即时聊天相对于传统 C/S 结构下的即时聊天, 有一定的便捷性, 部署方便, 使用浏览器作为载体可以很好的和所属网站结合, 成为网站中用户和用户之间、用户和网站之间良好的沟通模式.

Web 即时聊天是实现在 http 协议下的通讯模式, 使

用浏览器作为客户端, 就需要兼容各种浏览器. 为了实现实时性, 服务器端需要一个高效稳定的架构模式. 针对 http 协议是无连接状态, 本文实现的系统使用 comet 服务器推中的 ajax 长轮询模式, 实现一个 http 长连接, 这样就能保证聊天的即时性. 在客户端使用 jquery 框架编写 javascript 使系统前端的界面兼容各个浏览器. 后端使用 php 实现服务器推的机制, 使用现在十分流行的 LAMP 架构, 满足整体较高的效率要求. 在 Web2.0 网络开发平台方面, LAMP(Linux +Apache+ Mysql +php /perl /python) 受到 IT 界越来越多的关注^[3]. LAMP 架构由于稳定性、高效性在 Web 开发上有相当多的应用, 对比.NET 和 JSP 有一定的优势, 所以本文系统使用 php 来进行开发. 本系统在数据存储为, 为了提高访问效率, 使用 XML 文件系统来存储即时聊天中的信息, 这样能

^① 基金项目:国家科技支撑计划(2011BAH15B04)

收稿时间:2012-04-17;收到修改稿时间:2012-05-18

减少对数据库过于频繁的访问提高数据库提取的效率。综合分析,系统使用 comet 服务器推模型中的 ajax 长轮询模式,使用 jquery 框架编写 javascript 来兼容各种浏览器,后端使用 LAMP 架构,在数据存储方面加入 XML 文件存储系统提高效率。

1 系统框架概述

系统整体是 B/S 结构。总体分成三部分:前端 Web 页面、业务逻辑和数据存储。系统总体框图如图 1 所示。

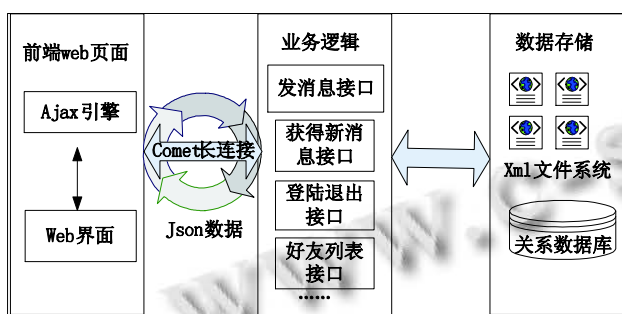


图 1 系统总体框图

前端 Web 界面: 使用 html、css 和 javascript 构建前端 Web 页面。使用框架 JQuery 编写 ajax 引擎,通过 ajax 和后端通讯。通讯的数据使用 json 格式,相对 xml 大大减少通讯量。前端 Web 页面也可以嵌入其他的网站中,灵活性大。系统使用浏览器作为客户端。

业务逻辑: 通过 php 实现系统的各个接口,实现服务器推的服务器端程序。使用 php 编写 xml 文件系统,将聊天的实时数据存入 xml 文件,用户的信息通过数据库读取和管理。

数据存储: 数据存储分为两个结构,一个是 xml 文件系统,存储聊天的数据;另一个是关系型数据库,存储用户的信息等数据。用户的信息和其他相关数据完全可以使用其他系统的数据,只要和对应读取和验证接口连接,整体就可以很方便的嵌入其他网站系统。

2 系统通讯模式选择和功能分析

2.1 comet 服务器推通讯模式选择

Web 聊天需要实时性,但是 http 协议是无连接的状态,不能像 c/s 系统那样进行持续的通讯。现在的技术通过浏览器拉和服务器推实现 Web 即时通讯,对比现有技术的稳定性和效率,服务器推有明显的优势。

服务器推送(Server Push)即 Comet 技术,是由

2006 年由 Alex Russell(Dojo Toolkit 的项目领导人)在他的文章中提出来的概念。它是一种基于 HTTP 长连接、无须在浏览器端安装插件而实现 Web 服务器主动向浏览器客户端发送消息的技术。其工作模式是浏览器和服务器建立连接之后,服务器发送一批数据,浏览器显示这些数据,同时保证与服务器的连接。当服务器需要再次发送一批数据时,浏览器仍然可以显示数据并保持连接。服务器推送技术一个优点是显而易见的高效性,它消除了浏览器等待与轮询的时间消耗,减少了建立或关闭新的 HTTP 连接的时间开销,客户端不会像使用轮询那样生成大量无用的通信量,并且服务器端在事件发生后可立即发布给客户端^[4]。

实现服务器推送主要有如下机制: 轮询方式、长轮询方式、基于 IFrame 的流(streaming)方式、基于客户端插件套接口的 Comet 技术^[5]。

轮询方式主要在前端使用 ajax 间隔一定时间不断向后端请求数据,类似刷新页面,这种模式不仅有一定的间隔延时而且会对服务器造成巨大压力。

长轮询也是前端使用 ajax 向后端请求数据,但是不是不断的请求而是请求后一直等待服务器推来数据,服务器端阻塞请求直到有新数据或者超时。客户端在获得数据或者超时就会重新发起连接,这样即保证了实时性,也大大减少了对服务器的压力。

基于 Iframe 及 htmlfile 的流(streaming)方式通过在 HTML 页面里嵌入一个隐藏帧,然后将这个隐藏帧的 SRC 属性设为对一个长连接的请求,服务器端就能源源不断地往客户端输入数据。类似长轮询。但是没有利用 ajax 实现的长轮询灵活。

基于客户端插件套接口的 Comet 技术,如: Flash XML Socket、Java 小程序(Applet) 等,需要客户端安装对应的插件,对平台和客户端有一定要求。

另外 HTML5 中的 Web Socket 技术,可以实现客户端浏览器的 socket 通讯。但是现在还没有真正推广。

综合以上技术模式,comet 下基于 ajax 长轮询的模式很适合 Web 即时聊天中信息的传递。

2.2 系统功能分析

整体系统需要实现良好的可移植性,便于整合到其他系统中。因此,Web 即时聊天系统需要一个比较通用的几套前端界面,以方便嵌入各种类型的网站中。功能上需要实现如下核心接口:

登陆和退出聊天系统接口,负责登陆和退出,登

陆后相当于在线；聊天接口，用户发出聊天内容，获取新聊天内容，这个是最核心功能，需要实现 comet 服务器推的 ajax 长连接；获取用户接口，获取聊天对象的信息，和在线成员信息；在数据读取上，除了读取数据库外，需要一个读取聊天 xml 文件的接口，这个接口在效率上要求较高。

3 系统核心技术实现

3.1 Web 前端 ajax 实现和数据传输格式

前端页面的各种行为主要通过 js 实现。包括进入，退出，发送和接收数据等。整体 js 的 ajax 实现都是用 jquery 框架提供的接口。

前端和服务器传送使用 json 格式的数据，例如一段在聊天中的 json 数据：

```
{ "code":0,"data":{"chatlist":[],"memberlist":[{"userid":"56","username":"\u98ce\u4e4b\u7ffc","logtime":"1330822219","headpic":"upload\album\56\hpic\56hpic4f4d63ad5016d.gif","mlastmodify":"1330822219.2965","chatlastmodify":"1330822219.2969","chatkey":"13308508674902445"}],"onlinenum":1}}
```

json 格式数据对应传统 ajax 使用的 xml 格式数据有明显优势。同样信息量数据，json 数据只有 xml 数据的 10% 的大小。这样大大减少了数据传输，并且 json 都是键值对应的数据格式。在处理上也提高了效率。

3.2 长轮询的后端实现

服务器推模型后端需要一个提供最新信息的程序，下面这段代码是后端长轮询中的核心内容。

```
if($chatroom->getOneMember($userid)!=false&&
$chatroom->checkChatKey($userid,$chatkey))
{ //判断是否是在聊天室中，通过在线和聊天
//key 值判读通讯合法性
// 先获得一次新消息
$SrsArr=$chatroom->getAll($massid,$userid,$username);
// 如果没有进入循环获得
while(count($SrsArr['chatlist'])==0&&count($SrsArr['memberlist'])==0 && $i<60){
// 休眠 1 秒
sleep(1);
if($chatroom->getOneMember($userid)== false)
{ //如果已经推出聊天 循环等待结束
```

```
break;
}
if(!$chatroom->checkChatKey($userid,$chatkey))
{
// 暂时关闭 循环等待结束
echo returnMessage(104);
exit;
}
// 获得最新消息
$SrsArr=$chatroom->getAll($massid,$userid,$username);
//计数器增加
$i++;
} // 循环末尾
//如果有新数据也在拿一次在线成员
$SrsArr['onlinenum']=
$chatroom->getMemberNum();
//数据返回
echo returnMessage(0,$SrsArr);
}else{
echo returnMessage(104);
}
```

RequestCheck 类的处理接受 post 或者 get 传来的值。returnMessage 类是处理返回对应的 json 格式数据。

这里核心的轮询次数是 60 次，程序主要是建立一个获取循环，如果每次没有获得变化，即新的留言变化和成员变化信息，并且满足对应循环次数，就会不停的循环，每次循环进程自动 sleep 一定的时间，这里是 1 秒。

```
while(count($SrsArr['chatlist'])==0&&count($SrsArr['memberlist'])==0 && $i<60){
.....
}
```

这样在 1min 内有信息最大延迟 1 秒(不算传输时间)，基本上做到了实时。在实际使用中，用户体验都是实时的反馈。前端在接到信息或者连接超时和断开都会重新启动一个新的长链接。这里整体轮询的次数和对应的服务器配置相关和网络路线有关。

3.3 xml 文件系统实现

数据存储使用关系型数据库和 xml 文件系统存储来共同实现。

即时聊天的即时内容数据存储主要使用 xml 格式的数据, 用户的登陆和个人信息使用 mysql 数据库存储. 本文实现的系统实现了一个 xml 文件系统来存储即时聊天的信息. 考虑到在线聊天会有很大的频率, 特别是群聊, 这样如果存入关系型数据库, 对数据库的读写很多, 这样就会拉低整体系统的效率. 所以综合考虑整体系统效率, 使用 xml 文件存储即时聊天内容. 存储通过日期建立文件夹. 每一个聊天有两个文件, 一个存储成员, 一个存储聊天内容. 格式如下两种.

① 聊天内容文件

```
<?xml version="1.0" ?>
<chatroom>
  <mass>215</mass>
  <chatcontent>
    <username>王振兴</username>
    <userid>56</userid>
    <content>你好</content>
    <pubtime>1330408066.3712</pubtime>
    <headpic>image/NoPictures.jpg</headpic>
  </chatcontent>
</chatroom>
```

这里面的 mass 对应相应的聊天编号, chatcontent 是聊天内容, 其中包括聊天人的姓名(username)、用户编号(userid)、内容(content)、发布时间(pubtime)和头像信息(headpic), 每条聊天的记录按照时间增量添加.

② 成员 xml

```
<?xml version="1.0" ?>
<chatroom>
  <mass>215</mass>
  <member>1</member>
  <lastmodify>1330408401.5466</lastmodify>
<chatmember>
  <username>王振兴</username>
  <userid>56</userid>
```

```
<logtime>1330408062</logtime>
<mlastmodify>1330408401.5468</mlastmodify>
<chatlastmodify>1330408524.13
</chatlastmodify>
<headpic>upload/album/56/hpic/56hpic1323655
62.jpg</headpic>
<chatkey>13304084053759047</chatkey>
</chatmember>
</chatroom>
```

这里存储一组聊天的成员信息, mass 是聊天的编号, member 是聊天人数, chatmember 是聊天人的相关信息. 存储聊天人姓名(username)、编号(userid)、登陆时间(logtime)、成员变动时间(mlastmodify)、内容变动时间(chatlastmodify)、头像信息(headpic)和聊天密钥(chatkey).

这种结构支持 1 对 1 聊天和群聊(多对多), 1 对 1 聊天的文件目录和群聊文件系统格式有些不一样, 是把两人的 id 号对应的 hash 值, 作为一个文件夹, 其中存储对应的聊天信息. 聊天信息按照日期存储. 其他格式一样.

3.4 读取聊天数据的算法

大量聊天内容读取和多人聊天读取需要一个灵活高效的数据读取算法. 本系统聊天的核心算法用到了三个存在成员 xml 文件的时间参数 lastmodify、mlastmodify 和 chatlastmodify. lastmodify 是一组群聊或者 1 对 1 聊天聊天内容变动的最后时间, 也就是说, 一旦有人发言就将存储对应聊天内容 xml 文件中的发言时间改为最后发言的时间; mlastmodify 和 chatlastmodify 是对应每个参与聊天人的成员变动的最后时间和聊天内容读取的最后时间. 如果一个人的 chatlastmodify 时间或者 mlastmodify 时间小于 lastmodify, 也就是说最近有更新, 那么就会根据个人的 chatlastmodify 时间去获取这个时间段之后的聊天内容. 同理 mlastmodify 时间不对应将会获得全部成员的信息, 例如在线情况. 这样可以增量的获得相关的聊天内容.

3.5 实际项目中的应用

本系统已经应用在一款社交类网站中, 作为其中班级群组的聊天室. 实际环境中使用 FreeBSD 系统, apache2.2, php 5.3.1, mysql 5.0, 实际运行良好. 页面样式如图 2.

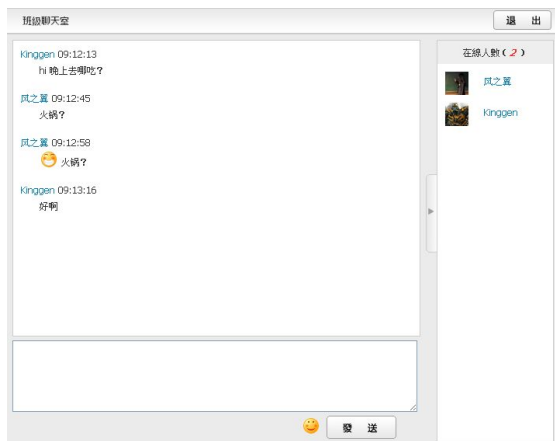


图 2 班级聊天界面

4 结语

本系统选择服务器推模型中的 ajax 长轮询模式作为 Web 即时聊天的通讯模型. 使用 jquery 解决前端 ajax 和聊天前端 js 的浏览器通用性问题, 利用开源的 LAMP 架构搭建高性能的后端服务程序, 使用关系型数据库和 xml 文件系统存储数据. 本文的系统已经全部实现并且运行在一个社交类型的网站. 实践证明整体程序运行良好. 如果实际中需要更大的性能可以在 LAMP 架构中加入缓存层进行进一步优化.

(上接第 11 页)

方式, 以线程为运行的基本单位, 实现了主控制软件和游戏程序的及时通信, 避免了访问数据过程中误读误取的现象. 主软件在数据采集之后, 实时将专注度、速度等信息传入到虚拟现实系统中, 在测得病患的专注度达到一定阈值, 判断其适合参与虚拟现实后, 开启游戏, 并将病患的速度信息实时地显示反映在游戏人物的运行快慢上, 从而让患者在康复训练中能更好地集中注意力并沉浸到虚拟现实中去, 进而让整个恢复锻炼的过程变得有趣而不乏味. 同时, 这种人机交互的方式更好地训练了患者的专注度, 调动了患者的积极性, 也更有利于病人的康复.

参考文献

- 1 唐磊. 下肢康复训练机器人机械结构及控制策略设计. 洛阳: 河南科技大学, 2009.
- 2 Krebs HI, Hogan N, Aisen ML, et al. Robot-aided Neuro-

参考文献

- 1 张文茂, 章淼, 毕军, 覃征. 互联网即时消息 (Instant Messaging, IM) 的研究现状与展望. 小型微型计算机系统, 2007, 28(7): 1162-1168.
- 2 庞怡, 许洪光, 姜媛. 即时通讯工具现状及发展趋势分析. 科技情报开发与经济, 2006, 16(6): 169-170.
- 3 三扬科技. 大道 PHP:LAMP+Zend+开源框架整合开发与与实践. 北京: 电子工业出版社, 2009. 3-11.
- 4 薛真真. 基于服务器推送和事件流处理技术的实时 Web 系统研究 [硕士学位论文]. 杭州: 浙江大学, 2008.
- 5 孙清国, 朱玮, 刘华军, 张鹏. web 应用中的服务器推送技术研究综述. 计算机系统应用, 2008, 17(11): 116-120.
- 6 IBM developerWorks 中国. Comet: 基于 HTTP 长连接的“服务器推”技术. [2007-08-31] <http://www.ibm.com/developerworks/cn/web/wa-lo-comet/>
- 7 PHP Manual. <http://www.php.net/manual/en/>. 2010-03-20.
- 8 Kevin M. Pro PHP: Patterns, Frameworks, Testing and More. United States of America: Apress, 2008, 50-65.
- 9 王秋云. XML 数据存储方法的研究. 重庆科技学院学报 (自然科学版), 2007, (4): 89-91.
- 10 李昕. 脚本语言 PHP 的 XML 应用支持. 计算机与数字工程, 2007, (6) 137-139.

rehabilitation. IEEE Trans. on Rehabilitation Engineering, 1998, 6(1): 75-87.

- 3 Loureiro R, Amirabdollahian F, Topping M, et al. Upper Limb Robot Mediated Stroke Therapy—GENTLE/s Approach. Reading: Autonomous Robots, 2003, 15(1): 35-51.
- 4 王月皎. 网络化一对多康复机器人测控系统及软件设计. 南京: 东南大学, 2011.
- 5 孔祥战. 仿生外骨骼式下肢康复机器人研究. 天津: 河北工业大学, 2010.
- 6 程方. 减重步行康复训练机器人研究进展. 中国康复医学杂志, 2008.
- 7 董亦鸣. 下肢康复医疗外骨骼训练控制系统研究与初步实现. 杭州: 浙江大学机械与能源工程学院, 2008.
- 8 文忠, 钱晋武, 沈林勇, 章亚男. 基于阻抗控制的步行康复训练机器人的轨迹自适应. 机器人, 2011, 33(2), 18-22.