

基于 RIA 架构的企业级即时通讯系统^①

何云, 姚能坚, 李辉, 赵榕生

(广州军区空军 指挥自动化工作站, 广州 510071)

摘要: 企业级即时通讯(Enterprise Instant Messaging 简称 EIM)技术是近来人们普遍研究的热点问题, 通过对企业级即时通讯的 C/S 架构、B/S 架构、基于富因特网(Rich Internet Application 简称 RIA)架构等实现技术的比较分析, 认为基于 RIA 架构的企业级即时通讯具有较大的优势。给出 Flex 技术开发 RIA 架构企业级即时通讯系统的开发原理, 描述多用户交互关系过程以及消息交互、音视频交互、用户列表管理、全局状态变量管理等核心功能, 实现基于 RIA 的企业级即时通讯系统。项目实践表明基于 RIA 的企业级即时通讯系统具有一定的实用价值和商业前景。

关键词: 即时通讯; RIA 架构; Flex 技术; 消息交互; 视频交互

Enterprise Instant Messaging System Based on RIA Architecture

HE Yun, YAO Neng-Jian, LI Hui, ZHAO Rong-Sheng

(Command Automation Office, Guangzhou Military Region, Air Force, Guangzhou 510071, China)

Abstract: Enterprise instant messaging(EIM) technology is a widespread research hotspot recently. Through the analysis of EIM of C/S architecture, B/S architecture, RIA architecture application technology, this paper thinks EIM based on RIA architecture has many advantages. It gives the principle of Flex technology development EIM system based on RIA, and describes multi-user interaction relationship and core functions of message interactive, audio and video interaction, user information management, global state variables management. It realizes EIM system based on RIA. Project practice shows that EIM system based on RIA has certain practical value and commercial prospects.

Key words: instant messaging; RIA architecture; flex technology; message interactive; video interaction

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]。自 1996 年面世以来, 特别是近几年的迅速发展, 即时通讯已经发展成集交流、资讯、娱乐、搜索、电子商务、办公协作和企业客户服务等为一体的综合化信息平台。即时通讯按其应用场合可以分为个人即时通讯(PIM)和企业级即时通讯(EIM)。EIM 是一种面向企业终端使用者的网络交互工具服务, 使用者可进行两人或多人之间的实时沟通。交流内容包括文字、界面、音视频及文件互发等^[3,4]。开发

功能完善的 EIM 具有一定的实用价值和商业前景。

1 几种实现技术比较

目前, 中国市场上的企业级即时通讯工具主要包括: 腾讯通 RTX、华夏易联 e-Link、GKE、Ucstar、金谷 Gu、百度 HI、TQ 等。普遍采用 C/S 架构, 基于 TCP/UDP 协议, 通过服务器协作, 利用防火墙穿透(代理)或基于 UDP 的 NAT 穿透技术, 保持客户端之间的持续长连接, 实现客户端之间信息的实时交互。但该类模式下的软件系统要求下载安装专用的客户端程序, 导致系统部署成本高昂, 面临系统维护困难、用户本地资源保护和网络安全问题^[5]。因此, 这种模式的 EIM 技术在网络环境下的广泛使用还存在局限性。

① 收稿时间:2011-02-18;收到修改稿时间:2011-03-20

基于 HTTP 协议的 Web-based EIM 采用 B/S 模式，客户端通过访问特定的网页而实现即时通讯。这种即时通讯技术以网页为载体，避免下载安装客户端程序，系统功能在服务器端统一维护，既减少了系统部署费用，也降低了维护难度。因此，Web-based EIM 具有更大的优势^[6]。然而，Web-based EIM 在技术实现上存在难以逾越的困难：首先，HTTP 的非连接、无状态特性将导致通讯状态管理非常困难；其次，HTTP 访问的单向性只允许客户端主动去联系服务器，而服务端却无法主动联系特定的客户端。新的 XMPP 协议技术^[7-9]、Ajax 定时刷新技术^[10,11]、Comet 服务器推送技术^[12,13]可以帮助实现 Web-based EIM 的消息交互功能，但企业级即时通讯系统要求的音视频交互与大数据传输等核心功能无法实现。

采用富因特网架构(Rich Internet Application 简称 RIA)设计企业级即时通讯系统是一个合适的选择。RIA 是一种具有高度互动性和丰富用户体验的网络应用程序技术，它兼具桌面应用程序和 Web 应用程序的特点。RIA 通讯特点则包括实时互动的声音和视频。客户机在 RIA 中的作用不仅是展示页面，它可以在幕后与用户请求异步地进行计算、传送和检索数据、显示集成的用户界面和综合使用声音和视频，这一切都可以在不依靠客户机连接的服务器或后端的情况下进行。基于富因特网技术的应用系统采用基于 Smart Client 技术，它整合了 Fat Client/Thin Client 两者的技术优点，充分利用客户端本地资源，使数据处理过程对于服务器端的依赖性大大减少，对系统平台的依赖性更小，用户分布更加灵活、处理逻辑更加集中，在结构上为进一步提高系统的可重用性、安全性、可扩展性和维护性创造了良好的条件^[14,15]。

当今流行的主要 RIA 架构开发技术有 Adobe 公司的 Flex，微软公司的 SilverLight 和开源的 Laszlo 技术。基本原理是：在客户端安装 RIA 插件（SilverLight 或 Flash 插件），服务端提供 RIA 服务（Flash Media Server 等）和 Web 服务（IIS 等）。以各自定义的扩展标记语言来编写业务逻辑，在服务端运行编译后的托管代码，根据用户发出的请求，将服务端生成的特定格式数据发送给浏览器的插件解释，把解释后的内容显示到浏览器中，返回给用户。

2 企业级即时通讯系统实现

企业级即时通讯系统的核心功能应包括：消息交互、音视频交互、用户列表管理等。基于 Flex 技术的应用平台包括：数据库服务、媒体服务（Flash Media Server）、IIS 服务，开发平台包括：Flex 开发（Adobe Flash Builder）、WebService 开发（Visual Studio）。企业级即时通讯系统的各个功能模块和多用户交互关系如图 1。以用户 A 的通讯交互为例将图 1 交互流程描述如下：

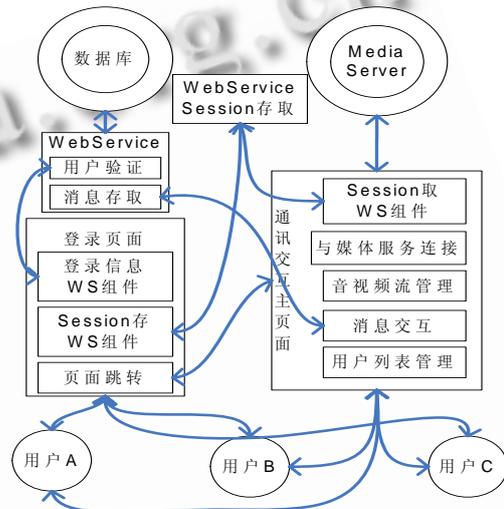


图 1 即时通讯功能模块与交互关系

用户在登录页面填写用户名、密码等登录信息，登录信息通过 Flex 的 WebService 组件（WS 组件）将参数传递给用户验证服务，用户验证服务将参数与数据库交互并返回用户验证信息给登录页面；验证通过后的用户信息通过 WS 组件传递给 Session 存取服务保存并跳转到通讯交互主页面。通讯交互主页面通过 WS 组件向 Session 存取服务索取用户信息，通过用户 ID 号与媒体服务建立连接，管理用户的音视频流交互、消息交互、用户列表等。

2.1 用户列表管理

多用户即时通讯系统首先应该解决的是用户列表管理问题，在 Flex 架构中可以利用共享对象(Shared Object)技术来实现。用户在通讯交互主页面与媒体服务建立连接后，在媒体服务端初始化一个用户列表共享对象，将用户的 ID 等信息记录在媒体服务端的共享对象中，通过对象远程调用方法(SharedObject.GetRemote)将服务端的共享对象映射到通讯交互主页面。

面。这样只要有用户加入或退出，媒体服务端的共享对象与主页面的映射对象都会同步更新，从而实现用户列表更新和管理的功能。

2.2 全局状态变量管理

用户登录信息的存取在 B/S 应用中常被用于用户验证、页面安全与管理等，在 B/S 架构中运用 Session 或 Cookie 对象技术容易实现。由于 RIA 架构的安全机制对本地资源有保护限制，且 Flex 没有提供类似 Session 或 Cookie 对象技术，虽然 Flex 有共享对象技术，但共享对象的存在必须以与媒体服务建立连接为前提，所以维持多页面间的全局状态变量非常困难^[14,15]。我们研究了两种方法可以实现多页面全局状态变量存取：

方法一：Flex 的外部接口对象（ExternalInterface）可以与 JavaScript 脚本交互。在 Flex 编译后的页面中嵌入 Cookie 存取脚本，通过外部接口对象调用 JavaScript 脚本能实现全局状态变量的存取服务。但该方法破坏了 RIA 架构保护本地资源的安全性，而且外部接口对象存在调用参数返回不稳定的缺陷。

方法二：通过 WebService 组件可以与基于 IIS 的 WebService 交互。我们把全局状态变量的存取统一设计成 IIS 服务的 Session 存取过程。Session 的维持在 IIS 服务端，该方法没有破坏 RIA 架构的安全机制，以 XML 格式返回参数，参数统一经过加解密处理，这种 Session 存取调用方法稳定且保证数据安全。

客户端交互页面请求 Session 变量的关键代码如下：

```
<mx:WebService id="WSSession" wsdl="http://www.remote.com/servicedata/SessionService.asmx?WSDL"
fault="Alert.show(event.fault.faultString, 'Error')">
  <mx:operation name="GetSessionData" resultFormat="object" />
  private function WSHandle():void//执行<mx: WebService
>组件请求函数
  {
    WSSession. GetSessionData.send();//请求 WebMethod
    WSSession.addListener(ResultEvent.RESULT_onGedata); //监听}
  private function _onGedata(event:ResultEvent):void//监听
  处理函数
  {
```

```
this.strUserId=event.result. userid;
this.strUsername=event.result. username;
}
```

IIS 服务端取 Session 变量的关键代码如下：

```
public class SessionService : System. Web. Services.
WebService
{
  public SessionService () {}//默认构造函数
  public class _User//自定义类，用以存储数据
  {
    public string userid;
    public string username;
  }
  [WebMethod]
  [System.Xml.Serialization.XmlInclude(typeof(_User))]/
  /“User”类可写入 XML
  Public _User GetSessionData()/获得 Session 数据
  { _User ur = new _User();//定义“_User”类实例
    ur.userid = Session["userid"].ToString();
    ur.username = Session["username"].ToString();
    return ur;//返回 XML 参数}
}
```

2.3 消息交互

消息交互是即时通讯系统的核心功能。在技术比较部分介绍有 XMPP 协议技术、Ajax 定时刷新技术、Comet 服务器推送技术等实现方法。在 RIA 架构中有两种实现方法：

方法一：通过 Flex 的共享对象（SharedObject）技术来实现。具体与用户列表管理的实现方法一致，但该方法是多用户共享一个对象，所以只适用于多用户公开讨论场景，无法满足点对点或点对多点信息交互应用，也无法保证消息的私密性。

方法二：通过 RIA 架构的媒体服务消息中转技术实现。在媒体服务端建立一个 Action Script 应用，一般命名为：main.asc 或 main.far（编译后）。该应用的功能包括：与媒体服务端的连接管理、服务端共享变量管理、客户端页面与服务端的远程函数相互调用和事件管理等。多用户消息交互主要利用客户端页面与服务端的远程函数调用和事件触发机制来实现，服务端应用须包含消息中转函数，客户端交互页面包含消息发送与消息接收函数。用户 A 向用户 B 发送消息的中转过程如图 2。媒体服务端（包含在 Main.asc 中）

的消息中转函数关键代码如下:

```
Client.prototype.SendMessage = function(userID,reUsrId,
sendMsg)
{ for(var i=0;i<application.clients.length;i++)
{ if(application.clients[i].userid == userID)
{ application.clients[i].call("AcceptServerMessage",null
,reUsrId,sendMsg); }
}
return "消息转发成功!";}
```

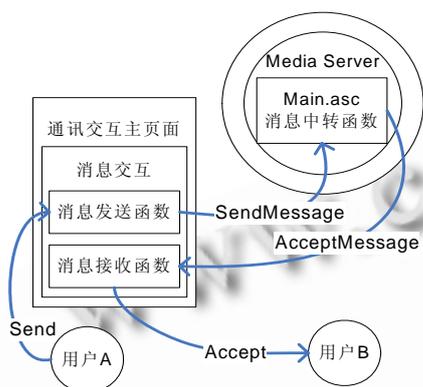


图2 消息的中转过程

2.4 音视频流管理

媒体服务 (Flash Media Server) 提供强大的流媒体发布与分发功能,使得RIA架构下实现多用户音视频交互相对简单。方法如下:

在客户端页面初始化摄像头和麦克风对象以及本地和远端两个视频 (Video) 对象,在与服务端连接上初始化一个音视频流 (NetStream)。采集本地摄像头和麦克风的音视频流,音视频流通过与服务端连接发布到媒体服务端,媒体服务端将音视频流分发到指定客户端远端视频对象组件,这样可以实现多用户音视频交互。

值得指出的是由于多用户音视频交互是采用流媒体发布与分发机制,使得媒体服务端会维持多个媒体流。媒体流数量 (定为 S) 与用户数 (定为 n) 的关系是: $S=n^2-n$ 。显然这种音视频交互方法会加重媒体服务端的负荷和网络流量。

3 项目应用

我们设计的基于RIA架构的企业级即时通讯系统在某《职称远程答辩系统》项目中得到应用。《职称远程答辩系统》要求基于浏览器模式、易于安装部署、

便于维护。用户体验效果要模拟答辩考生和考官面对面提问与回答的场景。主要流程为:答辩考生现场选题,考官宣读题目并将考题内容以文字形式下发,答辩考生现场回答并将答案整理后上传给考官确认。该项目与基于RIA架构的企业级即时通讯消息交互、音视频交互等功能需求和技术特点吻合,即时通讯系统作为核心模块嵌入到《职称远程答辩系统》中。

4 结语

本文对实现企业级即时通讯的C/S架构、B/S架构、RIA架构等技术进行比较分析,认为基于RIA架构的企业级即时通讯具有较大的技术优势,给出Flex技术开发RIA架构企业级即时通讯系统的开发原理,描述消息交互、音视频交互、用户列表管理、全局状态变量管理等核心功能以及多用户交互关系过程。该实现方法在某《职称远程答辩系统》项目中得到成功运用,项目实践表明我们设计的企业级即时通讯系统具有一定的实用价值和商业前景。

参考文献

- 1 张文茂,章森,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 Turek M. Instant Messaging: Time for It to Pay Attention. Business Communications Review, 2004,34(1): 50-53.
- 4 张永强,乔世权,孙亮.企业级即时通讯系统的设计与实现.陕西科技大学学报,2008,22(1):110-113.
- 5 陈立浩.基于B/S和C/S的即时通信系统.计算机工程,2009,35(15):270-274.
- 6 崔杰,徐浩,丁月华,李巍.跨协议的企业分布式即时通信系统的设计与实现.计算机应用,2009,29(10):2655-2658.
- 7 RFC3920-3921.Extensible Messaging and Presence Protocol (XMPP):Core/Introduction. 2004.
- 8 张云川.标准化的即时通信协议-SIMPLE和XMPP的对比研究.武汉科技大学学报(自然科学版),2005,28(4):375-377.
- 9 周文琼,王乐球,周桐,周春光.基于XMPP的企业即时通信系统研究与应用.吉林大学学报(信息科学版),2010,28(1):106-111.

(下转第242页)

从表 2 可以看出, 占用运行时间最长的是函数 `Internal filter`, 占整个程序运行时间的 59.49%, 函数 `Internal filter` 就是我们要找的核心函数。

6 结语

文章采用模拟器 `SimpleScalar` 对多媒体应用程序 `EPIC` 进行测试, 发现 `EPIC` 编码程序在执行过程中, 绝大部分的时间在运行函数 `Internal filter`, 再利用前面提出的程序特征分析工具集 `PCA` 对 `Internal filter` 进行测试分析, 从结果可以得出, `Internal filter` 是在做内部滤波(卷积), 从 `Internal filter` 的源代码中可以看到, 其核心循环是模板与原始图像九个部分分别求卷积, 其中存在并行性。并行性可以通过循环展开进行优化, 为其设计专门的体系结构自动调度循环执行, 最终实现循环内部操作、表达式、循环体的多层次并行, 加快程序执行速度, 全面加速计算机对多媒体应用程序的计算与处理的速度, 这些是我们下一步将要进行的工作。

参考文献

- 1 Austin TM. A User's and Hacker's Guide to the SimpleScalar Architectural Research Tool Set, 1997.
- 2 Bishop B, Kelliher TP, Irwin MJ. A Detailed Analysis of MediaBench. Proc. of the IEEE Workshop on Signal Processing Systems, 1999.
- 3 Bhargava R, John L, Evans B, Radhakrishnan R. Evaluating MMX technology using DSP and multimedia applications. Proc. of IEEE/ACM Sym. on Microarchit-Ecture, Dec. 1998: 37-46.
- 4 Ausrin T, Larson E, Ernst D. SimpleScalar: An infrastructure for computer system modeling. IEEE Computer, 2002, 35(2): 59-67.
- 5 彭绍亮, 窦勇, 李姗姗. 面向多媒体应用的程序特征分析与研究. 计算机工程与科学, 2004, 26(3).
- 6 陈剑龙, 傅忠传, 崔刚. SimpleScalar 模拟器内核分析及应用. 哈尔滨工业大学学报, 2004, 36(5).
- 7 Brooks D, Tiwari V, Martonosi M. Wattch: A framework for architectural-level Power analysis and optimizations. Proc. of 27th Ann Int'l Symp Computer Architecture. Los Alamitos: IEEE CS Press, 2000. 83-94.
- 10 周旋, 王丽芳, 蒋泽军. 基于 Ajax 的即时消息系统的设计与实现. 科学技术与工程, 2009, 9(2): 446-450.
- 11 Chatterjee S, Abhichandani T, Li HQ. Instant Messaging and Presence Technologies for College Campuses. IEEE Network, 2005, 19(3): 4-13.
- 12 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.
- 13 Alexandre D, Calvary Karin C. COMET(s), A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces. 15th International Workshop on Interactive Systems: Design, Specification, and Verification (DSV-IS 2008), Canada: Kingston, 2008.
- 14 王非. WebServices 应用研究与 RIA 系统中的实现. 计算机应用与软件, 2010, 27(3): 168-171.
- 15 曲海成, 李洋. 一种基于 Web 服务的 RIA 系统集成应用. 计算机系统应用, 2010, 19(9): 15-18.

(上接第 14 页)