

Android 与 J2ME 平台间即时通信的研究与实现^①

Study and Implementation of Instant Messaging between Android and J2ME Platforms

姚昱旻^{1,2} 刘卫国¹

(1. 中南大学 信息科学与工程学院 湖南 长沙 410083;

2. 湖南广播电视大学 现代教育技术中心 湖南 长沙 410004)

摘要: 随着新一代智能手机平台 Android 的发布和我国 3G 网络投入运营,不同智能手机间的即时通信应用越来越重要。本文分析了实现 Android 与 J2ME 平台间即时通信的关键技术,并在模拟环境上实现了不同平台终端设备间的即时通信功能。

关键词: Android J2ME 即时通信 XMPP Jabber

1 引言

随着 2008 年 4 月我国 3G 网络投入商业试运营,更快的网络传输速度必定会催生新的移动应用,不同智能手机平台间的即时通信将越来越重要。本文通过对比研究,分析了 Android 和 J2ME 间即时通信的关键技术,并在模拟环境下实现了不同平台终端设备间的即时通信功能。

2 Android 与 J2ME 平台框架结构

2.1 Android 平台

Android 是 2007 年 11 月由开放手机联盟 (Open Handset Alliance) 推出的新一代智能手机平台,它是一个真正意义上的开放性移动设备综合平台,它包括操作系统、用户界面、中间件和应用程序,它既是一个智能手机操作系统,同时也是一个智能手机开发平台,其开源免费的特性加上业界巨头的技术支持和强力推动势必在不久的将来占有智能手机市场的可观份额。

Android 平台自底向上由以下四个层次组成:Linux 内核层、Android 运行时库和其他库层、应用框架层、应用程序层如图 1 所示:

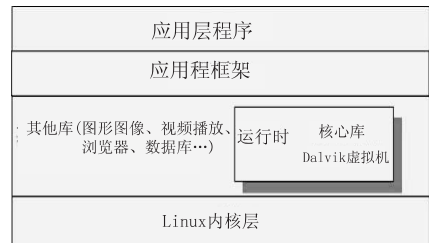


图 1 Android 平台的结构

(1) Linux 内核层: Android 底层是一个基于 Linux version 2.6.23 内核来开发的独立操作系统,主要是添加了一个名为 Goldfish 的虚拟 CPU 以及 Android 运行所需的特定驱动代码。该层用来提供系统的底层服务,包括安全机制、内存管理、进程管理、网络堆栈及一系列的驱动模块。作为一个虚拟的中间层,该层位于硬件与其它的软件层之间。

(2) Android 运行时库和其他库层: Android 运行时库包含一组核心库(提供了 Java 语言核心库内的大部分功能)和 Dalvik 虚拟机。

(3) 应用框架层: 在应用框架层 Android 开发人员可以跟那些核心应用一样拥有访问框架 APIs 的全部权限。应用的系结构化设计简化了各组件之间的重用,任何应用都可以分发自己的组件,任何应用也可以

^① 基金项目:国家自然科学基金(60676016);湖南省高校科研项目(06D064)

使用这些分发的组件(应用的使用方法需遵循框架的安全性约束)。

(4) 应用程序层: Android 本身会附带一些核心的应用程序, 包括 e-mail 客户端、短信程序、日历、地图、浏览器、通讯录等等, 目前所有的应用程序都是由 Java 语言开发的。

2.2 J2ME 平台

J2ME (Java2 platform micro edition) 是 Java 语言专门针对嵌入式电子产品开发的版本, 应用在移动电话、PDA、电视机机顶盒以及其它资源受限的设备上, J2ME 是目前移动设备上使用最广泛的开发平台^[1]。

可以发现 J2ME 的应用是建立在 JVM (或 KVM) 之上, 而本地操作系统是位于 J2ME 平台之外。这样的设计提供了跨操作系统的特性, 目前智能手机操作系统主要有: Symbian、Windows Mobile、Palm 和 Linux 四类操作系统, 它们都拥有各自操作系统上的 J2ME 平台, 因此实现了 Android 与 J2ME 间的即时通信, 就可以实现新旧不同智能手机系统间的通信。

3 即时通信系统设计

3.1 即时通信协议选择

不同平台特别是新旧平台间即时通信的实现基础是统一的即时通信协议, 这样能够保证系统的兼容性和可扩展性。目前通用的 IM (Instant Messaging) 协议主要有基于 SIP 的针对即时通信和空间平衡扩展的进程开始协议 (SIP for Instant Messaging and Presence Leveraging Extensions, SMPLE) 和基于 XML 的可扩展消息处理现场协议 (eXtensible Messaging and Presence Protocol, XMPP)。

这两大体系都源于 IETF 成立的 IMPP (即时消息和出席协议) 工作组, 都符合 2000 年 2 月发布的 RFC 2778 和 RFC 2779 两个草案的要求。IMPP 最初的目标是构建一套标准化的 IM 协议, 但后来实际上制定的是标准化的协议所应遵循的一组规范, 并由此衍生出两个不同的分支。

考虑到 XMPP 目前被广泛应用在开源的急促即时消息处理系统 (Jabber IM) 中, 发展迅速, 拥有大量的免费用户群和服务器群, 同时在相当的一段时间内仍会出现两种标准系统共存的情况, 所以这两种系统之间的互联就成了一个新的问题, 而 XMPP 支持网关扩

展^[2], 通过应用网关可以实现两类系统间的互联, 基于上述两方面考虑我们选用基于 XMPP 的开源即时通信系统 Jabber 进行后续研究与实现。

3.2 系统框架及工作流程

在整个通信过程中涉及到三个角色: 客户端、服务器、网络 (包括互联网及移动数据网)。客户端是 Android 和 J2ME 平台的应用程序^[3]。服务器端包括 Dispatch 服务器、若干个 Jabber 服务器、数据库。

客户端通过 HTTP 与服务器端进行数据交换, 交换过程中涉及到两个 XML 流: 一个流把数据包从客户端提交到服务器上; 另一个流把数据从服务器提交到客户端。当用户想进行通信时, 只要启动手机上的客户端应用程序进行登录, 发送一个包括用户登录号和密码的 HTTP 包给服务器端 Dispatch。Dispatch 根据用户的登录号和密码到相应的数据库中查询、对比, 进行身份验证, 若验证成功, 给客户端程序返回一个 HTTP 应答包, 在返回的 HTTP 应答包中包含了分配给该用户的一个 Jabber 服务器的 IP 地址和通信的端口号, 随后客户端就向指定 Jabber 服务器发起 HTTP 连接请求, 获得 HTTP 连接标识, 这样就可以利用这个标识进行一系列的通信活动, 在整个系统中的 Dispatch 起到路由和调度的作用, 它能根据用户的登录号段来分配 Jabber 服务器。若服务器端没有通过客户端的身份认证, 则会返回给客户端一个错误提示, 同时断开与该用户的连接。

3.3 系统实现难点及关键代码

3.3.1 移动终端寻址

系统采用客户端/服务器模式来构架, 但是不同于传统的单一服务器结构, 为了维护庞大的用户群体, 系统采用地域形式把整个消息空间分成不同的自治区域, 每个区域由一个单独的服务器进行管理, 同时设置应用网关与其他 IM 系统进行互联互通。为配合这种分区自治的结构, 采用 Jabber 地址 (也叫做 JID), 包含了域名标识符 (Domain), 结点标识符 (Node) 和资源标识符 (Resource) 三部分。路由处理也是基于 Jabber 地址的逻辑地址寻址方式, 其查询格式为: 节点@域名/请求源。其中域名可以在 DNS 中查找 (类似于电子邮件地址), 节点表示某个即时通信客户端上的某个应用或某项服务, 请求源为一类连接标识, 它能让某单一用户多次重复连接。如图 2 所示为 J2ME 平台客户端相关代码的依赖关系图。

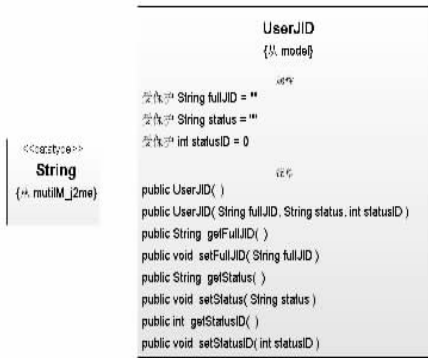


图 2 地址描述相关代码的依赖关系图

3.3.2 移动终端间通信

目前 XML 已经广泛应用于数据交换领域,因为它是纯文本的,所以具有优秀的跨平台特性。参考 Jabber 协议,设计系统实体间会话由两个平行的 XML 流组成,一个从客户端到服务器端,另一个从服务器到客户端。当一个移动客户端连接上一个服务器时,这个客户端将发起一个客户端到服务器的 XML 流,同时作为响应,服务器也将发起一个服务器到客户端的 XML 流。

在 XML 上下文数据流中,包括三个顶层 XML 元素:

<presence/ > 包含两个实体互相发送消息的内容;

<message/ > 提供关于一个实体的可用性信息。实体表现出的在线状态不是“可用”就是“不可用”。“可用”状态表示发送者可以立即收到消息。“不可用”状态表示发送者不能在当前时间收到任何数据;

<iq/ > (info/query) 意思是信息/查询。此元素主要是用来在两个实体间构建一个根本的会话,并且允许实体间来回传送 XML 格式的请求和响应。

由于该即时通信系统使用 XML 数据传输,因此 XML 的解析在移动端开发尤为重要的。J2ME 上可以调用开源的 KXML 库或使用 JSR 172 的 API 来解析 XML。Android 直接支持 DOM 解析器和 SAX 解析器两种方法,其中 SAX 方法内存占用小,更适合在手机程序开发^[4],下面是在 Android 平台客户端上构建一个 SAX 解析器的核心代码。

```

/* Android 上使用 SAXParser 解析 XML 数据

```

```

流. */
/* 分两步从 SAXParserFactory 类的实例化一个
SAXParser 对象. */
SAXParserFactory spf =
SAXParserFactory.newInstance();
SAXParser sp = spf.newSAXParser();

/* 从 SAXParser 中创建 XMLReader */
XMLReader xr = sp.getXMLReader();
/* 创建一个在 XMLReader 中使用的
ContentHandler */
ExampleHandler myExampleHandler = new Exam-
pleHandler();
xr.setContentHandler( myExampleHandler );

/* 从 URL 中解析 XML 数据. */
xr.parse( new
InputSource( url.openStream() ) );

/* ExampleHandler 用来提供解析完成的 XML 数
据. */
ParsedExampleDataSet
parsedExampleDataSet =
myExampleHandler.getParsedData();

/* 将结果显示在移动端的图形用户界面上. */
tv.setText( parsedExampleDataSet.toString() );

```

4 结语

基于 Jabber 协议在 Android 和 J2ME 各自开发平台的模拟器上实现了新旧智能手机平台间的即时通信。目前还没有使用 Android 平台的手机问世,因此还没有条件进行真实终端平台和网络环境下的测试,从模拟器的测试数据来看 Android 平台性能完全可以满足 XML 数据解析对于移动处理器和内存的要求。该系统存在的问题在于直接采用 XML 数据流传输同样的信息会比相同数据的二进制流占用更大的传输带宽。这是因为 XML 是为了清晰性和互操作性而不是

(下接第 127 页)

(上接第 120 页)

简洁性设计的一种文本标记格式,所以这种冗余与即时通信程序中常常使用的相当长的元素和属性名结合起来,XML 数据流中标记成分的大小可能远远超出传输的实际数据成分的大小,因此为了更好的在真实的移动环境中运行,需要在以后的研究中考虑 XML 数据压缩^[5]和传输性能改善的问题。

参考文献

- 1 郭春雷,李祥. 基于 J2ME 平台的手机远程监控软件的分析与实现. 计算机系统应用,2005,14(11): 86-88.
- 2 Chen MF, Lin YB, Herman C. - h. Rao, Wu Q. A mobile service platform using proxy technology. Wireless Communications and Mobile Computing. 2006, 6: 17-34.
- 3 潘永高,钟亦平,张世永. 基于网关的 J2ME Jabber 系统研究. 计算机工程,2005, 31(19): 108-110.
- 4 王冲. J2ME 开发平台上 XML 解析器的选择. 计算机工程与应用,2003, 39(31): 133-135.
- 5 王腾蛟,高军,杨冬青. 面向 XPath 执行的 XML 数据流压缩方法. 软件学报,2005, 16(5): 869-877.