

# J2EE 平台上邮件服务器的实现

郑 澜,吉国力

(厦门大学自动化系,福建 厦门 361005)

**摘要:** 在简单介绍了 J2EE 平台和 JavaMail API 的基础上,讨论了在 J2EE 平台上创建一个电子邮件系统的基本方法.该系统利用 JavaMail API 技术、采用 Java 编程,具有较好的可移植性(可以在 Windows, Linux, Unix 等多种服务器系统平台上运行)、可扩充性(可以在原有的基础上任意扩充其功能)和安全性(具有密码加密功能).

**关键词:** 电子邮件; J2EE; JavaMail; SMTP; POP3

**中图分类号:** TP 311.52

**文献标识码:** A

**文章编号:** 0438-0479(2005) Sup-0255-03

Java 技术具有强大的网络功能和全新的开发设计理念.而 J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构. J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版. J2EE 不仅巩固了标准版中的许多优点,同时还提供了对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 以及 XML 技术的全面支持.

E-mail 的收发是 Internet 中重要的组成部分, WebMail 是基于 WWW 方式进行 E-mail 收发的一种电子邮件系统<sup>[1]</sup>, 该系统操作简单、通用性强、使用方便, 深受用户欢迎. 用户无须安装任何 E-mail 客户端软件, 只需使用浏览器打开放在服务器上的特定网页 (JavaApplet) 即可进行 E-mail 收发及相关操作.

## 1 J2EE 平台及 JavaMail 的简介

### 1.1 J2EE 平台的特点

J2EE 以 Java2 平台的标准版为核心, 使用多层的分布式应用模型, 应用逻辑按功能划分为组件, 各个应用组件根据他们所在的层分布在不同的机器上. 在传统模式中, 客户端担当了过多的角色而显得臃肿, 在这种模式中, 第一次部署的时候比较容易, 但难于升级或改进, 可伸展性也不理想, 而且经常基于某种专用的协议<sup>4/4</sup>通常是某种数据库协议. 它使得重用业务逻辑和界面逻辑非常困难. 现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层. 一个多层化应用能够为不同的每种服务提供一个独立的层. J2EE 典型的四层结构包括: 运行在客户端机器上

的客户层组件、运行在 J2EE 服务器上的 Web 层组件、运行在 J2EE 服务器上的业务逻辑层组件以及运行在 EIS 服务器上的企业信息系统 (Enterprise information system) 层软件<sup>[2]</sup> (如图 1 所示).



图 1 J2EE 的基本架构

Fig. 1 The base frame of J2EE

### 1.2 JavaMail 简介<sup>[3]</sup>

尽管 World Wide Web 非常流行, 但使用得最多的 Internet 服务还是电子邮件. 可以肯定的是, 能够发送消息给同事、朋友和爱人的功能尤其是对于“在网络上的应用”来说, 是最基本的. 但是, 包装、发送和接收消息活动所涉及的细节问题仍然很复杂, 并且其中包含了对不同协议的混合使用. 从典型的 Email 客户来看, 这个过程很简单, 但要在程序中提供这些相同的功能却需要掌握大量的 API, 并且要持久地使应用程序与这些 API 所输出的特定协议紧密的耦合.

但是在 JavaMail 规范中, 提供了一些抽象类的集合, 这些抽象类为通常的邮件系统定义了通用的类和类的接口. 通过为邮件的会话、消息、传输和存储定义接口, Sun 对当前正在使用的许多协议给 Java 程序员提供了容易使用的、可扩展的面向对象的视图.

J2EE1.3 版本以上集成的 JavaMail API 是 Java 对电子邮件处理的延伸, 它提供了和通讯协定无关的 Java 解决方案, 可以处理某些最基本的协议和目前在

收稿日期: 2005-03-10

作者简介: 郑澜(1981-), 女, 硕士研究生.

Internet 上的规范,其中包括简单邮件传输协议(SMTP)、Internet 报文访问协议(IMAP)、多功能的 Internet 邮件扩充服务(MIME)和 POP3 等。

在邮件服务中,一般来说有 2 种不同类型的协议需要实现.传输(Transport)是 Sun 用于服务的原语,该服务能够把消息发送到他的目的地.最常使用的传输类型是普遍采用的 SMTP 传输.而另一个类型是存储(Store),它也是服务,可以连接到这个服务来取得已经传输到用户邮箱中的消息.虽然现在 IMAP 存储越来越流行,但大多数的用户还是熟悉 POP3 存储。

因为 POP3 和 IMAP 实现了相同的接口,所以任何一个能够使用 JavaMail 的应用程序都可以按照相同的方式来访问 POP3 存储和 IMAP 存储,也可以访问提供者中存在的其他存储(如 NNTP, P7, mbox 等).像 SMTP 和 X.400 那样的具体的消息传输也可以互换使用。

## 2 J2EE 平台上电子邮件系统的实现

### 2.1 多层邮件服务器的总体设计

我们在 J2EE 平台上构建邮件服务器.其基本流程为,用户在客户层上通过 Web 页面提交请求,请求被提交给 Web 层的 JSP 页面,JSP 根据用户的请求通过相应的 Java Bean 对象调用业务层上的 EJB,通过 JDBC 访问底层的数据库层。

根据 J2EE 的四层结构可得该系统的具体结构设计如下:

1) 客户层:客户层的类型有基于浏览器的(如 IE 等)、用 Java applet 实现的 Java 客户端以及 J2EE 客户端。

2) Web 层:由 JSP 页面或 Servlet 以及可选的一些 JavaBean 对象,通过一定的接口访问下面的业务层.上面客户层中的基于浏览器的用户必须通过这一层访问下面的业务层.而对于后两种基于 Java 的客户端和基于 J2EE 的客户端,可以通过 TCP/IP 方式访问业务层提供的端口进行交互。

3) 业务层:这里是邮件服务器的核心结构.其中包括 EJB 容器,JDBC 数据库连接,访问接口,SMTP 和 POP3 服务器端口处理接口(25 和 110).其中 SMTP 和 POP3 服务器端口处理接口用来满足客户端的请求。

4) 数据层:业务层通过 JDBC 访问这一层存放的数据,以实现查询、修改等功能.这一层即服务器的数据库层,存放服务器处理的数据,其主要作用是通过不同类型数据库的特点组织数据和优化查询以期达到最

优的效果。

### 2.2 使用 JavaMail 收发邮件

#### 2.2.1 利用 SMTP 协议发送 E-mail

SMTP<sup>[4]</sup>(Simple Mail Transfer Protocol)是 TCP/IP 协议的一部分,用于在 Internet 上对电子邮件进行路由传送,是 Internet 上应用最广泛的上层协议之一.其基本模型<sup>[5]</sup>如图 2 所示。

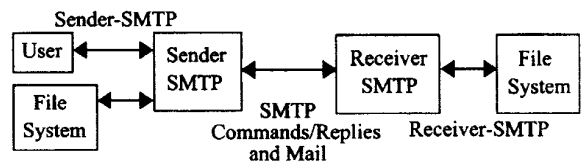


图 2 SMTP 模型

Fig.2 Model of SMTP

使用 JavaMail 实现邮件发送的流程如图 3 所示。

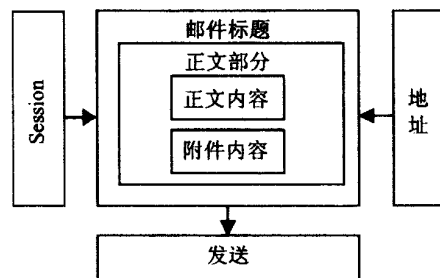


图 3 利用 JavaMail 发送邮件

Fig.3 Use JavaMail to send E-mail

其核心代码如下<sup>[6]</sup>:

```
MimeMessage msg = new MimeMessage(session);
msg.setFrom(new InternetAddress(addr,name));
if(!to.equals("")) //收件人
msg.setRecipients(Message.RecipientType.To,
InternetAddress.parse(to,false));
if(!cc.equals("")) //抄送人
msg.setRecipients(Message.RecipientType.CC,
InternetAddress.parse(cc,false));
if(!bcc.equals("")) //密送人
msg.setRecipients(Message.RecipientType.BCC,
InternetAddress.parse(bcc,false));
msg.setSubject(subject);
MimeBodyPart mbp1 = new MimeBodyPart(); //标题
mbp1.setText(msgText);
Multipart mp = new MimeMultipart(); //邮件正文
mp.addBodyPart(mbp1);
msg.setContent(mp); //处理附件
msg.setSentDate(new Date()); //发送时间
Transport.send(msg); //发送
```

### 2.2.2 利用 POP3 协议发送 E-mail

POP3 (Post Office Protocol 3)<sup>[4]</sup> 是 TCP/IP 网络经常使用的邮政协议标准的最新版本. POP3 协议允许客户通过暂时的 TCP/IP 连接, 从 POP 主机上取得电子邮件. POP 协议最大的优点是不需要与网络保持不间断的连接, 就可收取电子邮件.

在默认情况下, POP3 的端口号是 110<sup>[5]</sup>, 其工作流程分为下面几步:

- 1) 客户机发出请求, 请求和服务器连接;
- 2) POP3 服务器应答, 连接建立;
- 3) 客户机和服务器交互命令/ 应答和数据;
- 4) 结束连接.

利用 JavaMail 其实现流程如图 4 所示.

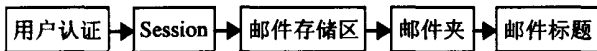


图 4 利用 JavaMail 接收邮件

Fig. 4 Use JavaMail to accept E-mail

相应的实现代码如下:

```

// 连接邮件服务器, 获得所有邮件列表
public Message[] getMail(String host, String name,
    String password)
{
    Message[] msg = null;
    Properties prop = new Properties();
    try {
        prop.put("mail.pop3.host", host);
        Session session = Session.getDefaultInstance(prop);
        Store store = session.getStore("pop3");
        store.connect(host, name, password); // 连接服务器
        Folder inbox = store.getDefaultFolder();
        getFolder("INBOX");
        inbox.open(Folder.READ_ONLY);
        Message[] msg = inbox.getMessages();
        FetchProfile profile = new FetchProfile();
        profile.add(FetchProfile.Item.ENVELOPE);
        inbox.fetch(msg, profile);
    }
    catch (Exception ex)

```

```

{
    System.err.println(ex + "收取邮件出错!");
}
return(msg);
}

```

## 3 结束语

本系统在 J2EE 框架上实现了邮件服务器的一些基本功能. 用户可以登陆服务器使用收发邮件, 修改密码等基本功能. 该系统继承了 Java 语言的良好可移植性, 其内部通过使用数据库连接的配置文件而极大的增强了系统的可移植性.

本系统在用户使用角度还可继续扩展, 加强邮件服务器的抵御病毒、过滤垃圾邮件等功能. 在性能方面可以考虑如何合理分配各部分的任务、提高它们的性能来尽量协调消除瓶颈, 以期达到整体性能的最优. 例如 (当大量用户同时访问数据库时, 系统可采用局部性原理来提高性能).

在邮件服务器实现的基础上可以根据特定用户特点开发一些相关的增值服务, 如在某些情况下使用短信形式把邮件内容发送给用户 (如使用优信通等硬件设备).

## 参考文献:

- [1] 伊斌, 江崇礼, 董明. 基于 Java 的 Web 邮件系统[J]. 计算机工程, 2004, 14: 183 - 184.
- [2] Subrahmanyam Allamaraju, Karl Avedal, Richard Browett. J2EE 服务器端高级编程[M]. 闻道工作室, 译. 北京: 机械工业出版社, 2001.
- [3] 王勇, 陈楠. 基于 Java 语言的电子邮件系统的实现[J]. 现代电子技术, 2004, 9: 26 - 30.
- [4] Williams A I. Java 2 网络协议内幕[M]. 何雄, 等译. 北京: 中国水利水电出版社, 2002.
- [5] RFC 2821 和 RPC1734 [EB/OL]. <http://www.rfc-editor.org>, 2002-08-12/2003-04-10.
- [6] SUN Microsystems Inc. Java 2 SDK, standard edition documentation Version 1.3.1 [EB/OL]. <http://java.sun.com/j2se/>, 2004.

## Implementation of an E-mail Server Based on J2EE Platform

ZHENG Lan, JI Guo-li

(Department of Automation, Xiamen University, Xiamen 361005, China)

**Abstract:** Base on a brief introduction of J2EE and JavaMail API, the development of a Email system based on J2EE was discussed. The system utilized JavaMail API and code in java. It can easily be run on many kinds of platforms (just like Windows, Linux, Unix and so on), extend its function easily and be secure (it can decode some information freely).

**Key words:** Email; J2EE; JavaMail; SMTP; POP3