

# 基于 Pipe 通信的数据同步系统的设计与实现<sup>\*</sup>

王冰,王红胜,李雄伟,向凯全

(军械工程学院 计算机工程系,石家庄 050003)

**摘要:** JXTA 是由 SUN 公司推出的一组开放的通用 P2P 协议,在 JXTA 架构中,Peer 之间主要基于 Pipe 通信机制进行通信.本文首先对 JXTA 软件架构进行了分析,然后对该架构中管道通信的概念、管道类型和管道通信过程进行了深入的介绍,最后设计并编程实现了一个数据同步系统,对实现代码进行了分析并点明了该系统的优缺点.

**关键词:** JXTA;管道;同步

**中图分类号:** TP393.03

**文献标识码:** A

**文章编号:** 1006-0707(2009)07-0092-03

P2P 网络是一种与传统的 Client/Server 或多层服务器网络完全不同的网络体系结构,P2P 网络中的 Peer 是彼此直接通信的,这种通信无需依赖集中式服务器或资源就可完成.基于 P2P 技术的软件也逐渐涵盖了日常生活的多个方面,如文件共享、分布式计算、即时通信、以及分布式存储等<sup>[1]</sup>.

JXTA 是为了构建 P2P 网络而制订的一组协议,JXTA 标准协议规范介绍为:JXTA 由 6 个协议组成,这些协议专为特定的、分布式的、对等的网络计算而设计的.使用这些协议,Peer 可以互相合作来建立自我组织、自我管理的对等组,而不必关心它们在网络中所处的位置(网络边缘或者防火墙后面),并且也不需要集中的管理机构.

## 1 JXTA 软件架构分析

基于 JXTA 的软件架构可以分为 3 层,层次结构示意图如图 1 所示.

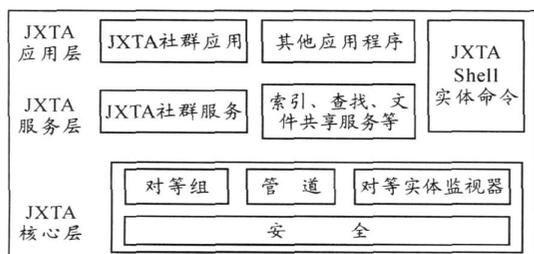


图 1 JXTA 软件架构示意图

核心层(JXTA Core)封装了 P2P 网络的一些基础的要素,

包括为 P2P 应用程序提供关键机制的模块.这些关键机制包括发现、传输、节点加入、创建组以及相关的安全原语.

服务层(JXTA Services)包括了查找、共享、索引、代码缓存和内容缓存等机制.

应用层(JXTA Application)包括了一些应用的实现,如 P2P 即时消息、文件和资源共享分布式验证系统等<sup>[2]</sup>.

## 2 基于 Pipe 的编程

### 2.1 Pipe 的基本概念

在 JXTA 架构下,Pipe 是 Peer 之间主要的信息通信机制之一,管道是异步的、单向的、不可靠的消息传输机制(安全管道除外).管道对所传输的数据没有限制,它支持所有对象的传送,比如二进制数据、字符串和 java 对象等.

Pipe 的单向性意味着通过 Pipe 机制通信,接收方首先应当与一个“Input Pipe”绑定,在另一端,发送方应当与 Output Pipe 绑定.获得一个 Pipe Advertisement 之后,就可以选择与“物理管道末端”绑定.需要发送消息的 Peer 就应当建立 Output Pipe;需要从 Pipe 中获取信息的 Peer,应当建立 Input Pipe.创建双向的通信需要创建两个独立的 Pipe.

不可靠性是指 JXTA 不保证 Pipe 传送的消息一定可到达目的主机,消息传输过程中,消息有可能丢失.

异步性是指同一 Pipe 发送的多个消息有可能不是依次抵达目的主机.

### 2.2 Pipe 的类型

Peer 基于 Pipe 相互通信是 JXTA 的通信机制之一,Pipe 的类型包括: JxtaUnicast, JxtaUnicastSecure, JxtaPropagate<sup>[3]</sup>.特点如图 2 所示.

\* 收稿日期:2009-04-20

作者简介:王冰(1984—),男,河北保定人(满族),工学硕士,主要从事计算机信息与安全研究.

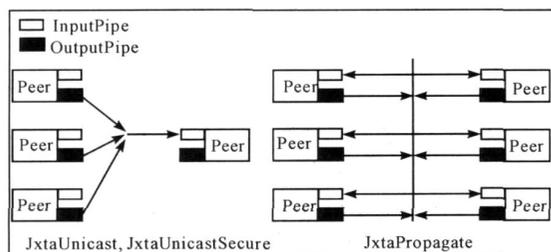


图2 不同 Pipe 类型的特点

### 2.3 Pipe 的通信过程

基于 Pipe 的服务采用注册监听器的方法对消息进行响应. 基于 JxtaUnicast Pipe 的通信过程如图 3 所示.

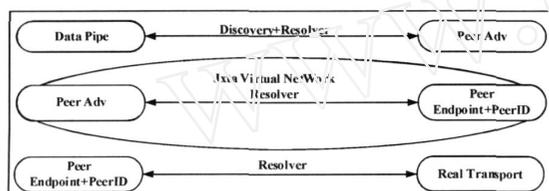


图3 基于 JxtaUnicast Pipe 的实际传输过程

1) 作为接收方的 Peer 需要具备一个管道通告文件 (Pipe Advertisement), 然后根据这个通告创建一个 Input Pipe, 等待 Messages 的到来.

2) 发送数据方的 Peer 需要使用相同的管道通告文件, 然后根据这个通告来创建 Output Pipe 以发送数据. 要创建 Output Pipe, 它要先发送一个 Pipe Binding Query Message 给所有它知道的 Peers.

3) 接收方收到这个 Pipe Binding Query Message, 看看自己缓存 Pipes 中有没有匹配的 Pipe ID. 如果有, 他就回复一个 Pipe Binding Answer Message (里面包含了自己的 Peer Advertisement) 给请求方.

4) 发送方接收到 Pipe Binding Query Message 后, 将 Peer Advertisement 从中抽取出来. 然后使用 Peer Advertisement 中的 Endpoint 信息来创建 Output pipe, 这样发送方才可以发送数据<sup>[4]</sup>.

基于 JxtaPropagate 的通信过程与 JxtaUnicast 是一致的. 由于它具有多对多消息传递的特点, 因此可以用它来开发一些特殊的应用. 本文所研究实现的数据同步系统就是基于 JxtaPropagate 管道进行实现的.

## 3 数据同步系统的设计实现

### 3.1 总体设计

数据同步系统是指 P2P 网络中任意一个 Peer 发送的消息同步显示在每一个 Peer 中, 功能类似于聊天室. JXTA 架构下的 JxtaPropagate 类管道可以实现多对多消息传输, 所以采用 JxtaPropagate 类管道建立管道连接.

系统功能由一个 XChat 类实现, 分为建立连接和收发消息两个功能模块. 在建立连接模块中, 首先创建并加入

对等组, 取得该组管道服务和发现服务, 创建 JxtaPropagate 类管道通告文件 pp. adv, 使用该通告文件建立输入管道并定时发布 Peer 与输入管道绑定的消息. 之后设置输入和输出事件监听程序. 最后使用相同的通告文件 pp. adv 建立输出管道. 创建成功则会建立一个由输出管道到输入管道的多对多连接管道.

本系统中, XChat 类继承了 JXTA 架构下的 OutputPipeListener 和 PipeMsgListener 两个类, 并在 XChat 类内部实现了输入和输出事件监听程序接口 pipeMsgEvent, outputPipeEvent, 用于处理输入和输出事件.

在 JXTA 架构中, 所有的网络资源如对等点、点组、管道和服务都是用通告来描述的. 通告是用 XML 文档描述的和语言无关的元数据结构, 对等点通过搜索相应的通告来发现资源. 本程序中所用的管道通告文件 pp. adv 的代码如下:

```
<!DOCTYPE jxta:PipeAdvertisement >
<jxta:PipeAdvertisement xmlns:jxta = http://jxta.org >
<Id >
urn:jxta:uuid - 59616261646162614A757874614D5047
25184FBC4E5D498AA0919F662E40028B04
</Id >
<Type >
JxtaPropagate
</Type >
<Name >
PipeExample1
</Name >
</jxta:PipeAdvertisement >
```

实现收发消息模块的核心是函数 newChatMessage, 该函数用于定制 Message 类的消息对象. 具体过程为: 发送消息方通过调用 newChatMessage 定制消息并通过输出管道发送消息. 对等组中的每一个 Peer 均可以监听到该消息, 输入事件监听函数对该消息进行解析处理并显示, 这样就实现了同一消息在不同 Peer 上的数据同步.

本实验实现了 3 个 Peer 之间的数据同步, 分别模拟本地 8000、8001、8002 端口, 均在本地调试运行, 使用同一个管道通告文件 pp. adv. 运行环境为 jdk1.6, JXTA 版本为 JXTA2.5.

### 3.2 具体实现和代码分析

在类的构造函数中初始化并加入对等组, 并取得该对等组的发现服务和管道通信服务. 主要代码如下:

```
public XChat1() {
    pg = new NetPeerGroupFactory().getInterface(); // 初始化
    JXTA 平台
    ps = pg.getPipeService(); // 取得点组的管道通信服务
    disc = pg.getDiscoveryService(); // 取得点组的对等点
    发现服务
    创建输入管道对象的主要代码如下:
    if (pml != null)
```

```

ip = ps.createInputPipe(inputPipeAdv, pml);
else ip = ps.createInputPipe(inputPipeAdv, this); // 创建
输入管道对象并绑定输入事件处理程序。
if(ip != null) {
disc.remotePublish(inputPipeAdv, 10 * 60 * 1000);
disc.publish(inputPipeAdv, DiscoveryService.ADV, 10 *
60 * 1000); // 定时发布 Peer 与输入管道的绑定信息
this.sleep(10 * 60 * 1000);
System.out.println( 输入管道创建成功! );}
创建输出管道对象的主要代码如下:
System.out.println( 开始创建输出管道! );
op = ps.createOutputPipe(outputPipeAdv, 1000);
if(opl != null) System.out.println( 输出管道创建成
功! );

```

消息类构造函数的主要代码如下:

```

public Message newChatMessage(String message, String out-
putPipe)
{
Message msg = new Message();
StringMessageElement me = new StringMessageElement(
source, outputPipe, null); // 添加消息源
StringMessageElement me1 = new StringMessageElement
(info, message, null); // 添加消息内容
msg.addMessageElement(null, me);
msg.addMessageElement(null, me1);
return msg;
}

```

pipeMsgEvent 函数用来实现对输入管道中来到消息的监听和处理. 主要代码如下:

```

public void pipeMsgEvent(PipeMsgEvent event) {
System.out.println( = = = = = = = = = = =
= = = = = = = = = = = = = = = = = = = = =
= );
Message msg = event.getMessage();
MessageElement msgElement = msg.getMessageElement
(null, source);
MessageElement msgElement1 = msg.getMessageElement
(null, info); // 解析 Message 类消息
System.out.println(msgElement + : + msgElement1); //

```

格式化输出消息

数据同步运行界面如图 4 所示.

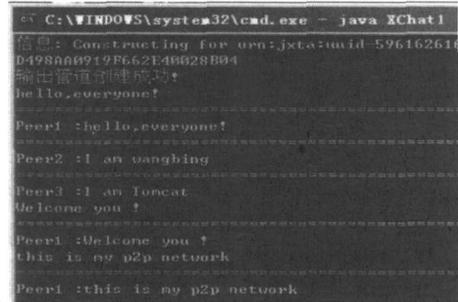


图 4 数据同步程序运行界面

### 4 结束语

JXTA 为 P2P 网络上层应用程序的开发提供了一个统一的、便利的平台和协议,使得 P2P 应用程序的开发变得高效而灵活<sup>[5]</sup>. 本文基于 JXTA 架构下的 JxtaPropagate 管道实现了一个数据同步系统,该系统具有以下优点:第一,该网络完全是 P2P 架构,可靠性和可扩展性强;第二,该系统基于 JXTA 架构下的管道通信进行开发,屏蔽了连通链路差异、防火墙等底层的细节,缩短了系统开发周期,提高了系统的稳定性. 除此之外,该系统在图形化界面、对等组安全以及功能拓展等方面需要进一步深入研究.

### 参考文献:

- [1] 杨天路,刘宏宇,张文,等. P2P 网络技术原理与系统开发实例[M]. 北京:人民邮电出版社.
- [2] 陈桂海,李振华. 对等网络:结构、应用与设计[M]. 北京:清华大学出版社.
- [3] Aniel Brookshier. JavaP2P 程序设计[M]. 北京:中国电力出版社.
- [4] 张莉,徐进. 基于 JXTA 技术的 P2P 系统性能保证体制[J]. 计算机应用, 2005, 25(2): 270 - 272.
- [5] 姜巍巍. 基于 Java 技术的 Web 服务平台的研究与应用[D]. 北京:北京理工大学, 2003.