

P2P网中基于IP组播的发现机制研究

许加兴,汪厚新,周 艳,金炜东

(西南交通大学 电气工程学院,四川 成都 610031)

摘 要:对等网络技术是近年来计算机网络技术的一个热点,而对等技术中的一个热点技术就是如何找到节点地发现机制问题。介绍了P2P技术及其发现机制和IP组播的原理,利用IP组播技术来解决P2P中对等体的发现问题。文中分析并建立了一个IP组播发现模型,在此基础上,结合SUN公司为P2P分布式计算提供的通用统一、可互操作的平台JXTA技术,实现了IP组播技术在P2P网络对等点发现机制中的应用,有效地解决了单点发送多点接收、多点发送多点接收的问题。

关键词:对等网络;IP组播;JXTA;发现机制

中图分类号:TP393.1

文献标识码:A

文章编号:1673-629X(2006)06-0032-03

Discovery Implementation by IP Multicast in Peer to Peer Network

XU Jia-xing, WANG Hou-xin, Zhou yan, JIN Wei-dong

(Electrical Engineering College, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: Peer to Peer network has become new focus of the computer network technology in recent years. Discovery Implementation is one of critical technology in Peer to Peer network. This paper introduces the P2P technology and IP multicast technology, then analyzes and proposes an IP multicast discovery model. On the basis of this, with the use of JXTA project, the application of IP multicast technology in Peer to Peer network discovery implementation is realized and the problem of information transmission from single-peer to multi-peer or multi-peer to multi-peer is effectively solved.

Key words: peer to peer; IP multicasting; JXTA; discovery

1 P2P技术和IP组播简介

1.1 P2P技术及其发现机制

P2P(Peer-to-Peer)是一种对等计算技术,其目的是将网络中不同的计算机连接起来,并充分利用和共享互联网中任何地方的闲置资源。P2P系统中对等体(Peer)既是客户端又是服务器。对等体能够在一个虚拟网络中运行,多个对等体可以组成一个对等组。这种非中心化的网络结构具有很强的优势,可有效地解决传统C/S网络结构中频繁访问服务器端单一资源造成的瓶颈问题,并且消除服务器端出现故障造成的网络瘫痪情况的发生^[1]。

P2P对等应用程序是一种大规模但又是细粒度的应用程序^[2]。每个对等点都可以进入或退出,每个对等点都关注于自己的任务。在它们短暂的活动期间,尝试完成布置给它们的任务。这些任务中的大多数都要涉及与其它对等点交互。在P2P网中,如果一个节点要和其它的节点交互,它必须先找到那个节点,这样它们才能建立连接并进行数据交换。这就使得发现在P2P网中的作用变得

越发的重要。

所谓发现,就是P2P应用程序中的对等点彼此定位以便相互之间可以交互。常用的P2P发现机制有:目录模型、Flooding模型和IP组播模型。

1.2 IP组播

组播是一种允许一个或多个发送者(组播源)发送单个数据包到多个接收者(一次且同时的)的网络技术。组播源把数据包发送到特定的组播组,而只有属于该组播组的地址(动态形成)才能收到数据包;无论有多少个目标地址,在整个网络上的任何一条链路上只传送单一的数据包,所以组播提高了数据传送效率,极大地节省了网络带宽^[3]。

IP地址方案专门为组播划出一个地址范围,在IPv4中为D类地址,范围是224.0.0.0~239.255.255.255,并将D类地址划分为局部链接组播地址、预留组播地址、管理权限组播地址^[4]。

局部链接组播地址:224.0.0.0~224.0.0.255,用于局域网,路由器不转发属于此范围的IP包;

预留组播地址:224.0.1.0~238.255.255.255,用于全球范围或网络协议;

管理权限组播地址:239.0.0.0~239.255.255.255,组织内部使用,用于限制组播范围。

收稿日期:2005-10-09

作者简介:许加兴(1977-),男,浙江诸暨人,硕士研究生,主要研究方向为信息技术;金炜东,教授,博士生导师,主要研究方向为信息技术、系统工程。

D 类地址的最后 28 bits 没有结构化,即没有网络 ID 和主机 ID 之分。响应某一个 IP 组播地址的主机构成一个主机组,主机组可跨越多个网络。主机组的成员数是动态的,主机可以通过 IGMP 协议加入或离开某个主机组,组成员关系不受地区和规模的限制;主机可以同时属于多个组;组成员关系在数据报传输过程中不是必要的,即一个主机不必是组成员就可以向一个组发送数据报^[5]。

2 IP 组播发现机制

2.1 IP 组播模型

这种模型是利用 IP 组播来发现对等体。发送站点向某一特殊 IP 地址(即 IP 组播地址)发送数据,网络中与该地址相关的接收节点都可接收到数据,无关站点将不接收这些数据,因而可有效减轻网络通信负担,避免资源浪费。每一个对等体要发布自己的资源,可以通过组播自己来宣布存在,宣布包含了主机名和一个用于正常通信的端口。对此消息感兴趣的对等体检测到这个消息后,抽取主机名和端口号,并使用该消息建立一个通信通道^[3]。IP 组播发现模型可以用图 1 来简单表示。

图中集合点(Rendezvous Peer)是一个能够处理来自其它 Peer 请求的 Peer。集合点通常拥有更多资源,并且可以存储大量的有关它周围的 Peer 的信息。

集合点可以作为发现的传递者。集合点可以转发发现请求到其它的集合点(原集合点通过与其它 Peer 的广告交互而得到了被转发集合点的信息)^[6]。每一个集合点如果本身没有被请求的信息它都会转发该信息。

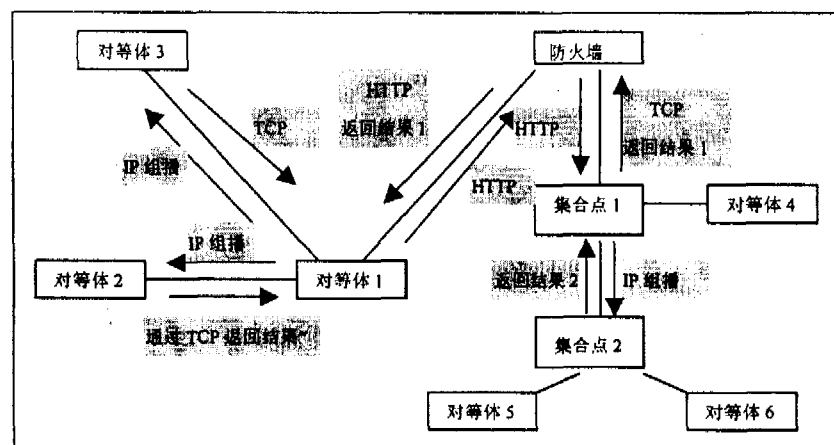


图 1 IP 组播发现模型

2.2 JXTA 工程

SUN 公司以 Java 技术为背景,开展了 JXTA 项目。其目的是为 P2P 分布式计算提供一个通用统一的、可互操作的平台来容纳任何种类的网络服务^[7],并且能够顺利地实现如 PDA、手机、传感器等数字设备无缝接入到 P2P 系统中来。

JXTA 定义了一组核心协议完成对等体之间的路由、发现、组织、监控和通信,这些协议不仅独立于程序设计语言(目前可以用 Java、C/C++、Perl 等多种语言实现),而且独立于传输协议,如可在 TCP/IP、HTTP、Bluetooth、HomePNA 等上实现。

JXTA 可用于开发如下应用和服务:

- 通过对服务器及整个 Web 的搜索,来获得有用的信息;
- 分布式地在网络上保存文件和信息,而不是只在本地硬盘上;
- 连接游戏设备,多人可在多个交互地玩同一个游戏;
- 使用任何连接设备在任何地方共同开发一个项目;
- 共享计算设备,如处理器和存储设备等,而不管系统或用户的物理位置。

JXTA 体系结构可以由图 2 来表示。

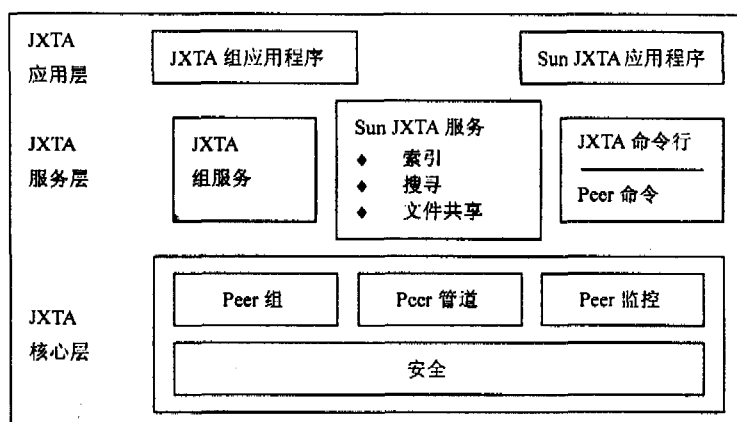


图 2 JXTA 的系统结构

JXTA 体系结构可以分成 3 层,自下往上依次是核心层、服务层、应用层^[8]。

(1)核心层(Platform layer),也叫 JXTA core,封装了构建 P2P 网络最少和最必要的组件,包括发现、传输(包括防火墙处理器)、建立结点和结点组、安全组件。

(2)服务层(Services Layer),包括网络服务,这对于 P2P 网络来说也许不是绝对必需的,但常常需要用到,例如寻找和索引、目录服务、储存系统、文件共享、分布式文件系统、资源分配、协议转换、鉴权机制。

(3)应用层(Application Layer),包括各种高层的应用,例如即时消息服务、文档资源共享、娱乐内容管理和分发、电子邮件、分布式拍卖系统和 P2P 许多其他的应用。

3 IP 组播发现机制的实现

用 JXTA 的发现服务来实现基于 IP 组播的发现,需要导入两个包:net.jxta 和 net.jxta.impl,前一个包含所有

的捆绑接口,后一个则是这些接口的实现。发现其它对等点和对等组的机制抽象到了 `net.jxta.discovery.Discovery` 类中,其实现则是 `jxta.discovery.impl.service.DiscoveryService`,每一个 JXTA 对等组都有一个 `DiscoveryService` 实例。

例子原理很简单,主要使用一个方法,此方法用于发现一个运行 `HelloService` 的对等点,然后打开一个通向它的管道,发送一个消息。

```
public void sendMessage() {
    this.discovery.getRemoteAdvertisements( null, Discovery.ADV, "name", "HelloService: *", 10 );
    try {
        Thread.sleep( 5000 ); // 等待接受请求
    } catch( InterruptedException error ) {
    }
    try {
        // 获取广告目录,对匹配的做出响应
        Enumeration pipeEnumeration = this.discovery.getLocalAdvertisements(Discovery.ADV, "name", "HelloService: *");
        if( pipeAdvertisement != null ) {
            try {
                // 建立输出管道
                OutputPipe helloPipe = this.pipes.createOutputPipe( pipeAdvertisement, Pipe.NonBlocking, -1 );
                Message helloMessage = this.pipes.createMessage();
                helloMessage.push( "sender", new ByteArrayInputStream( this.peerGroup.getPeerID().toString().getBytes() ) );
                // 通过管道传递信息
                helloPipe.send( helloMessage );
                System.out.println( "message sent!" );
            } catch( IOException error ) {
                System.err.println( "problem sending the message to the peer" );
            }
        } finally {
            helloPipe.close();
        }
    } else {
        System.err.println( "could not find a pipe to talk to" );
    } catch( IOException error ) {
        System.err.println( "problem retrieving advertisements from the local cache" );
    }
}
```

组中的一个节点要想被其计算机发现,必须向组发送管道广告。下面程序给出了管道建立方法。

```
private PipeAdvertisement createPipeAdvertisement() {
    try {
        // 创建一个管道对象
        PipeAdvertisement pipeAdvertisement =
```

```
(PipeAdvertisement) AdvertisementFactory.newAdvertisement(
    PipeAdvertisement.getAdvertisementType());
    pipeAdvertisement.setPipeID( new PipeID( this.peerGroup.getID() ));
    // 定义管道名称
    pipeAdvertisement.setName( "HelloService:" + this.peerGroup.getPeerID() );
    return pipeAdvertisement;
    } catch( InvocationTargetException error ) {
        System.err.println( "problem creating the pipe advertisement" );
    }
    return null;
}
```

管道建立完成后,就可以开始建立一个服务。这个服务用上面的方法创建一个管道后先向组发送管道广告以使自己能被发现,然后监听到来的消息,当收到消息后,会打印出这个消息。

```
public int startApp( String[] arguments ) {
    // 创建监听消息管道
    try {
        // 管道广告发布
        this.helloPipeAdvertisement = createPipeAdvertisement();
        if( this.helloPipeAdvertisement == null )
            throw new PeerGroupException();
        this.discovery.publish( this.helloPipeAdvertisement, Discovery.ADV );
        this.discovery.remotePublish( this.helloPipeAdvertisement, Discovery.ADV );
        // 创建输入管道
        this.helloPipe = this.pipes.createInputPipe( this.helloPipeAdvertisement );
    } catch( IOException error ) {
        System.err.println( "problem publishing the advertisement for the input pipe" );
    } catch( Exception error ) {
        System.err.println( "problem opening the pipe to read from" );
    }
    } finally {
        if( this.helloPipe == null ) {
            System.err.println( "do not have a pipe to read from" );
            return 1;
        }
    }
```

4 结束语

对等体的定位和发现是 P2P 网络应用中的一个重要组成部分和衡量 P2P 网络性能的重要指标。IP 组播技术有效地解决了单点发送多点接收、多点发送多点接收的问题,实现了 IP 网络中点到多点的高效数据传送,能够有效

(下转第 37 页)

一步验证支持向量机在分类识别率的优越性,将表1样本中的决策属性类别由四类改成三类后进行测试,其测试结果如表5,6所示。

表5 十交叉训练试验结果(四)

分类识别率/%	训练时间/s	测试时间/s
98.652	2.9612	0.023

表6 BP算法试验结果(五)

分类识别率/%	训练时间/s	测试时间/s	神经元个数
92.407	3.565	0.03	4
92.352	3.875	0.04	5
92.407	4.4415	0.04	6

2.2 结果分析

从上面的测试结果可以得到如下结论:

(1) 采用支持向量机进行股票预测的准确率在决策属性类别为四类的时候明显高于BP算法和RBF算法,在决策属性的类别为三类的时候虽然BP算法取得了较好的分类识别率,但支持向量机仍然比BP算法的分类识别率高出了将近7个百分点。因此,相比传统的利用神经网络来预测股票的几种方法来说,支持向量机在预测精度上明显提高。

(2) 从表2和表5的对比中发现,随着决策属性类别的增加,利用支持向量机所获得分类识别率也有所降低,但仍然优于同等情况下的BP算法和RBF算法的测试结果,虽然在预测每股收益具体数值的性能有待提高,但对股票是否盈利的预测取得了很好的效果。因此支持向量机进行股票预测有着良好的可行性和应用性。

(3) 股票市场受到政策、经济以及人为因素的干扰和影响,从而大大增加预测的难度,要想非常准确地进行股票预测并进入实用阶段,仍然有很长的路要走。但是通过测试可以发现利用对上市公司本身经营情况的分析可以排除掉部分人为因素的干扰,达到较好的预测效果。

(上接第34页)

地节约网络带宽、降低网络负载。使用组播也存在一些缺点,如一些防火墙和路由器会阻塞组播的消息,同时还存在其它IP组播的障碍,比如个人防火墙、子网路由器。不过JXTA还支持其它多种发现机制,在实际应用中,可以通过多种发现的机制的结合应用来达到更好的目的^[9]。

参考文献:

- [1] Milojicic D S, Kalogeraki V, Lukose R, et al. Peer to Peer Computing[R]. Palo Alto: HP Laboratories, 2002. 2-6.
- [2] Bawa M, Cooper B F, Crespo A. Peer-to-Peer Research at Stanford[R]. USA: Stanford University, 2003.
- [3] 吴胜浩, 钟亦平, 张世永. 用IP组播发现同位体发现机制[J]. 计算机工程, 2004, 30(3): 119-121.

3 结束语

利用支持向量机并采用对上市公司股票走势有着重要影响的相关数据来达到对股票进行预测的目的,从测试上来看预测精度明显高于传统的神经网络的预测方法,提高了分类的准确率。所采用的支持向量机克服了BP算法固有的缺陷,如学习过程收敛速度慢、网络性能差、可能存在局部极小值等。虽然股票市场非常复杂,但是如果能够对反应上市公司经营状况的数据进行合理分析,从而提取影响该上市公司股票的关键数据,再使用支持向量机对股票走势进行预测,仍然可以得到令人满意的结果。

参考文献:

- [1] Baba N, Kozaki M. An Intelligent Forecasting System of Stock Price Using Neural Networks[A]. In Proceedings of IJCNN [C]. Los Alamitos: IEEE PRESS, 1992. 652-657.
- [2] Raymond S, Lee T. iJADE Stock Advisor: An Intelligent Agent Based Stock Prediction System Using Hybrid RBF Recurrent Network[J]. IEEE Transactions on Systems, Man, and Cybernetics - part A: Systems and Humans, 2004, 34(3): 421-428.
- [3] 张铃, 张钊. 人工神经网络理论及应用[M]. 杭州: 浙江科技出版社, 1995.
- [4] 张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, 26(1): 32-41.
- [5] Burgers B C. A Tutorial on Support Vector Machines for Pattern Recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 121-167.
- [6] 吴微, 陈维强, 刘波. 用BP神经网络预测股票市场涨跌[J]. 大连理工大学学报, 2001(1): 11-15.
- [7] 张建, 陈勇, 何永保. 人工神经网络之股票预测[J]. 计算机工程, 1997(2): 52-55.
- [8] 财政部注册会计师考试委员会办公室. 财务成本管理[M]. 北京: 经济科学出版社, 2003.

- [4] 刘波. IP组播通信机制及其实现[J]. 计算机工程, 2003, 29(6): 131-133.
- [5] Johnson V, Johnson M. How IP Multicast Works[EB/OL]. <http://ipmulticast.com/community/whitepapers/howipmc-works.pdf>, 2003-11-26.
- [6] Moore D, Hebler J. 对等网[M]. 苏忠, 战晓雷等译. 北京: 清华大学出版社, 2003.
- [7] Krikorian R. HelloJXTA! [EB/OL] <http://www.onjava.com/pub/a/onjava/2001/04/25/jxta.html? page=1>, 2001.
- [8] Oaks S, Traversat B, Li Gong. JXTA技术手册[M]. 技桥译. 北京: 清华大学出版社, 2004.
- [9] 许斌. JXTA—Java P2P网络编程技术[M]. 北京: 清华大学出版社, 2003.