

基于 JXTA 的 P2P 文件共享系统的实现研究 *

陈 锋¹ 罗逢吉² 文俊浩²

(重庆工学院计算机学院 400050)¹ (重庆大学软件学院 重庆 400030)²

摘 要 P2P 对等网络是对互联网传统计算模式的一次技术上的革命。在 P2P 技术的研究领域里, JXTA 作为一种新的解决方案正日益受到业界的关注, 它为对等应用实现了底层的基础服务, 使开发者可直接在其上建立应用。针对目前在 P2P 应用研究领域尚无通用一致性平台的现状, 本文分析了 JXTA 技术的体系结构及其核心协议, 通过构建一个实例深入研究了利用 JXTA 技术实现 P2P 文件共享系统的特点。

关键词 P2P, JXTA, 文件共享

The Realization Research of Applying JXTA in P2P File Share System

CHEN Feng¹ LUO Feng-Ji² WEN Jun-Hao²

(College of Computer Science and Technology, Chongqing Institute of Technology, Chongqing 400050)¹

(College of Software Engineering, Chongqing University, Chongqing 400030)²

Abstract The P2P network is a technique revolution of traditional Internet computing mode. JXTA, which acts as a new solution is gradually becoming the focus in the domain of P2P technique. JXTA implements the foundational services for the applications, and allow the developers to build the applications upon it. Aimed at the none of the universal platform in the domain of the P2P application research at present, the author analyzes the architecture and the core protocols of JXTA in the paper, and by building a application instance, the author deeply studies the characters of using JXTA technique to implement P2P file sharing system.

Keywords P2P, JXTA, File sharing

1 引言

P2P(Peer to Peer), 即对等互联网络技术, 是目前计算机技术中的研究热点之一。P2P 技术是一种用于不同的用户间, 不经过中继设备直接交换数据或设备的技术。它允许网络用户直接使用对方的数据, 各用户可以直接连接到其他用户的计算机, 并进行文件的交换, 这在极大程度上打破了客户端/服务器的计算模式^[1]。目前, 围绕着 P2P 技术的优势, 已经开发出了许多应用, 如实时聊天、文件共享、深度搜索、分布式计算等。其中, 文件共享是最为广泛也最具代表性的应用, 具有很高的工程应用价值和广阔的应用研究前景。

JXTA 技术是美国 Sun 公司针对原有 P2P 系统的缺陷而设计的一种解决现代 P2P 分布计算的编程和计算平台^[5]。该技术致力于创建一个通用的平台, 提供 P2P 程序所需的底层服务, 以简单有效的方式构建特定的分布式应用, 使得开发者不需过多考虑如何解决分布计算的底层细节, 而专注于如何实现与完善可扩展性强、具有高可用性的高层应用。

2 JXTA 核心技术分析

Sun 公司为 JXTA 建立了一个三层的体系结构^[2], 如图 1 所示。

该结构分为三层。最底层是平台层即核心层, 这一层包括了 P2P 网络最基本的要素, 如对等体(Peer)和对等体组(Peer Group)的建立、资源发现、通信管理及其他一些底层服

务; 中间层是服务层, 这一层提供了诸如索引、查找、文件共享、协议转化、授权服务、PKI(Public Key Infrastructure)服务之类的高级服务; 最上层是应用层, 即通过使用 JXTA 技术而构建出的应用程序。

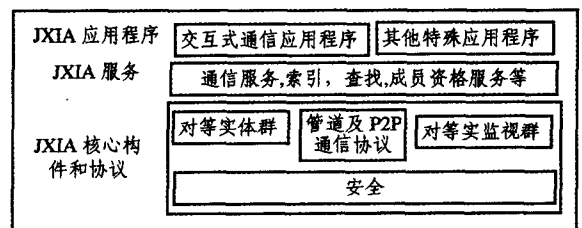


图 1 JXTA 的体系结构

JXTA 技术的核心在于一套独立于编程语言、系统平台和网络平台的协议集。这套协议集描述和实现了 P2P 应用最基本的需求, 其中的每一个协议都可以很容易地实现和集成到 P2P 的服务和应用中。这样, 不同的 P2P 程序之间可以方便地互相通信, 协同工作, 向对方提供服务^[3]。这套协议集包括了以下六个协议:

(1) 对等体发现协议(Peer Discovery Protocol, PDP)。该协议是包括世界对等组在内的所有对等体组的默认发现协议, 它可以被替换。对等体通过 PDP 发现已经发布的 JXTA 资源。

(2) 对等体解析协议(Peer Resolver Protocol, PRP)。在

* 重庆市自然科学基金项目(CSTC, 2006BB2234)、重庆大学研究生创新基金。陈 锋 博士研究生, 副教授, 研究方向为软件工程、图像处理; 文俊浩 副教授, 博士研究生, 主要研究方向: 软件工程、分布式计算、数据挖掘; 罗逢吉 研究方向: 软件工程、分布式计算。

JXTA 网络中,对等体向其它对等体发送请求,以寻求所需服务,PRP 用于标准化请求的格式。

(3)对等体信息协议(Peer Information Protocol, PIP)。PIP 使得一个对等体能够了解其他对等体的状态。例如,可以发送一个 Ping 消息来判断一个对等体是否已经激活。如果对等体的属性有名字和值,可以通过 PIP 得到它的属性信息。

(4)对等体成员资格协议(Peer Membership Protocol, PMP)。对等体可通过 PMP 加入或离开对等体组。PMP 包括了四个相对独立的部分:申请、加入、更新、取消,每一部分都定义了相应的 JXTA 消息格式。权力认证和安全管理可以提供相应等级的保护。

(5)管道绑定协议(Pipe Binding Protocol, PBP)。在 JXTA 环境中,管道是一种异步的、单向的、用于收发消息的虚拟信息通路。通过 PBP,可在运行时绑定到管道的一端来访问服务,还可以创建新的服务或解除管道绑定。

(6)端点路由协议(Endpoint Routing Protocol, ERP)。该协议可以保证一个对等体向另一个路由对等体询问到达目的地的可用路径。一般而言,两个通信的对等体并不直接相连,例如两个使用不同网络协议的对等体或被防火墙和网络地址转换分隔的对等体。对等体路由用有效的路由信息来响应请求,通常路由信息是这条路径所经过的网关列表。

3 利用 JXTA 构建 P2P 文件共享的实例

通过设计和构建一个实例,来分析如何利用 JXTA 技术来实现 P2P 文件共享系统。

系统分为三个功能模块,分别是:点对点实时聊天、文件共享和文件搜索。这三个功能也代表了当前 P2P 文件共享系统的最常见需求。以下将分别对每个模块的设计实现方案进行分析讨论。

3.1 点对点实时聊天

利用 JXTA 实现点对点实时聊天,核心在于管道技术。管道是 JXTA 中对等节点之间传输数据的通道。在 P2P 网络中,连接不一定是直接的,可能存在路由节点对消息的转发。而管道对开发人员隐藏了底层网络的所有内幕,它为开发人员在网络底层的 IP 和路由之上建立了一个逻辑上的连接机制,使得开发人员不必关心底层网络传输的细节,而将注意力集中于节点与节点之间的逻辑连接上^[4],如图 2 所示。

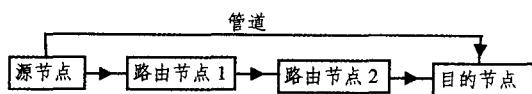


图 2 管道机制

JXTA 的管道分为两种:输入管道和输出管道。管道有一个重要的特性,即它是单向的,输入管道只能用于接收信息,而输出管道只能用于发送信息。这个特性是由 P2P 网络的本质所决定的。如果仅有的路径是通过 HTTP,那么完整的双向能力很难实现。

在创建管道时,一个输入管道所使用的管道通告应该和一个输出管道所使用的管道通告相一致。这样,当创建了一个输出管道后,JXTA 会自动解析创建了响应输入管道的节点 IP。

对点对点实时聊天的设计可采用如下方案:在启动应用程序平台后,本地节点将创建一个输入管道,并将管道通告在网络中进行发布。当远程节点试图向本地节点发送消息时,

远程节点应搜索本地节点的输入管道通告,并根据该通告创建输出管道。类似地,当本地节点试图发送消息给远程节点时,本地节点应搜索远程节点的输入管道通告,并根据该通告创建输出管道。在双方对对方的管道进行了绑定之后,就可以进行消息交换。该方案的设计流程图如图 3 所示。

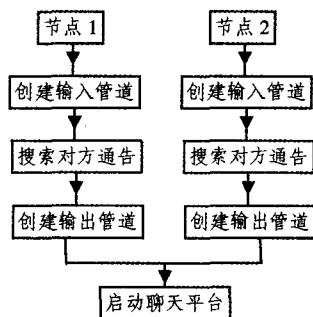


图 3 点对点聊天的设计方案

本地节点创建输入管道的过程如下:

```

PipeManage pm = new PipeManage(myShareGroup);
String pipeName = myPeer.GetID();
String description = "Input Pipe of Local Peer";
InputPipe myInputPipe = pm.createInputPipe(pipeName, description, PipeService.UnicastType, true);
  
```

输入管道一经创建后,JXTA 将生成一个对应的管道通告,并发布到 JXTA 网络中。远程节点根据搜索到的输入管道通告创建输出管道与之绑定的函数如下所示:

```

Public OutputPipe createOutPipe(Pipe.Advertisement adv, int timeout)
{
    PipeService ps = StartMyShare.getMyShareGroup().getPipeService();
    OutputPipe outPipe = null;
    try
    {
        OutPutPipe outPipe = ps.createOutputPipe(adv, timeout)
    }
    Catch(Exception e)
    {
        System.out.println("Create OutputPipe Failed!");
        e.printStackTrace();
    }
    return outPipe;
}
  
```

3.2 文件共享

文件共享模块是本系统的核心模块。在 P2P 网络中,任何一个节点都扮演着双重角色,在一些时刻它是服务器,在一些时刻它是客户端,甚至在某些时刻,节点既是服务器也是客户端^[5]。对于文件共享来讲,发起文件下载请求的节点担当起客户端的角色,而发送文件数据的节点则作为服务器。要在节点之间实现文件共享,需要解决两个主要问题:如何获取对等节点的共享文件信息以及如何在对等节点之间传输文件。下面,将讨论在两点之间共享文件的模型,这样一个模式很容易扩展到多点的网络环境中。在以下的讨论中,把请求获取共享文件信息或请求下载文件的节点称为客户端节点,把提供共享文件信息或发送文件数据的节点成为服务器端节点。

当客户端节点连接到服务器端节点后,首先应发出请求来获取服务器端节点的共享文件信息。这一点,笔者给出的解决方案如下:每一个节点都指定一个文件夹作为共享文件夹,共享文件夹中的文件与数据都为该节点的共享文件,可供远程节点访问和下载。当节点收到客户端远程节点查看共享文件信息的请求后,将本地共享文件的信息组织成 XML 的结构发送给客户端节点,客户端节点收到响应信息后,解析该 XML,获得共享文件信息,并显示在界面上。

可在程序中定义一个文件信息类,其中的字段代表了文件的各个属性,如名称、大小等。当服务器端节点收到获取共享文件信息的请求时,它遍历共享文件夹中的文件。对于每一个文件,获取它的各个属性,将其包装到一个文件信息对象中。在遍历完成之后,将所有文件信息对象中的信息存储在一个字符串中,以便发送给客户端节点。该字符串实际上是一个具有 XML 层次结构的字符串,其中具有文件的名称,大小等标签对,它可被解析为 XML 文档。

当客户端节点收到该结构化的字符串后,需要将其解析,以得到服务器端节点共享文件的各个信息。这需要用到 Java 的 DocumentBuilder 类,该类的 parse 函数可将字符串输入流解析为结构化文档,如下所示:

```
StringBufferInputStream=new StringBufferInputStream(s);
DocumentBuilder builder=new DocumentBuilder();
Document doc = builder.parse(is);
```

在客户端节点获取到了服务器端节点的共享文件信息后,另一个需要考虑的问题是对等节点间共享文件的传输问题。对此,作者采取的解决方案如下:在启动应用程序平台后,本地节点将创建一个服务器端套接字,以监听来自远程节点(客户端节点)的连接。当客户端节点从服务器端节点的共享文件列表中选择一个文件进行下载时,通过套接字向服务器端节点发送文件名称,服务器端节点收到文件名后,从本地共享文件夹中查找该文件,并向客户端节点传送文件字节流,如图 4 所示。

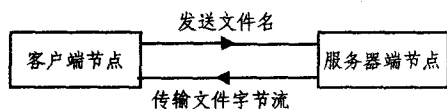


图 4 传输文件数据

服务器端节点在创建套接字时,需要用到 JXTA 套接字。JXTA 套接字是建立在管道基础上的,因此在建立 JXTA 套接字时,应首先建立一个输入管道,然后在管道基础上建立服务器端套接字。

客户端节点在与服务器端节点建立连接时,首先应在网络中搜索服务器端节点创建套接字时所使用的管道通告。获取该通告后,客户端节点在该通告基础上创建套接字。由于服务器端节点与客户端节点在创建 JXTA 套接字使用的是相同的管道通告,JXTA 平台会自动将服务器端节点与客户端节点进行绑定,如图 5 所示。

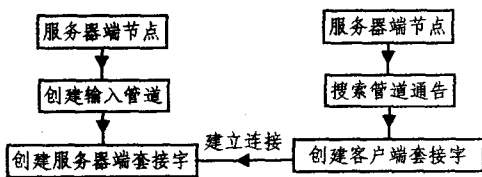


图 5 在节点之间建立 JXTA 套接字连接

在传输文件数据时,服务器端节点需要使用两个流:一个是读取本地文件的文件输入流,一个是从套接字打开的输出流,用于将缓冲区中的数据传输到远程节点。在客户端节点接收字节流时,作者所采取的策略是“一边接,一边写”,即客户端节点一块一块地从套接字输入流中读取文件信息,同时将读取的数据写入到磁盘上。这样做的优点是可以不需要将文件字节流发送完毕便可打开文件。在实际应用中,对于视频文件的传送,这一方案较为实用。整个过程如图 6 所示。

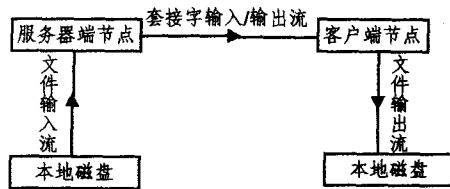


图 6 利用套接字传输文件

3.3 文件搜索

在 P2P 网络中,如何搜索和定位资源是一个重要的课题。但是,由于 P2P 网络的不稳定性和不可靠性,使得 P2P 网络的信息搜索技术更加复杂。下面将首先介绍传统的泛洪搜索算法,然后讨论如何对该算法进行改进以使其更有效地运用于本系统中。

传统的泛洪搜索算法的基本思想如下^[6],网络上的任何一个节点发起搜索时,首先根据查询内容生成一个查询消息,并将此消息发送给网络中与其相连的邻接节点。收到该查询消息的节点将搜索自身的资源,如果有与之匹配的文件,则形成一个查询应答发送给源节点;如果没有,则不发送。然后,该收到查询消息的节点,又将查询消息转发给它所已知的、除了源节点之外的其他节点。

泛洪算法在搜索深度和广度上有着不可比拟的优越性。由于每个节点都将搜索消息转发给它所知道的节点,搜索消息将迅速扩散到整个网络,这样就极大地提高了信息的挖掘度,使得遍布于网络中的所有与搜索消息相匹配的数据都可能被搜索到。

但是传统的泛洪算法也有着极大的缺陷性,即它过多地占用了网络带宽。由于每个节点只是简单地将消息转发,从而导致网络中消息的数量以指数级数增长,极大地降低了网络带宽的利用率,很容易造成网络阻塞。同时,一些恶意节点也很容易利用这一缺陷性对网络进行攻击。

针对传统的泛洪算法的缺陷性,结合考虑该算法的优点,可以采取以下几点措施,对其进行改进。

一是给搜索消息增加 TTL(Time To Live,生命期)值,它为每条搜索消息指定其在网络中的生存时间。当一个节点收到资源搜索消息时,它首先检查消息的 TTL 值,如果大于 0,则将消息进行转发,并将 TTL 减 1;如果等于 0,则不进行转发。通过定义 TTL 值,可以限定消息在网络中传播的时间。以发送一个 TTL 值为 3 的搜索消息为例,如图 7 所示,当经过三次转发到达节点 d 后,由于搜索消息的 TTL 值已减为 0,节点 d 简单地丢弃该消息。

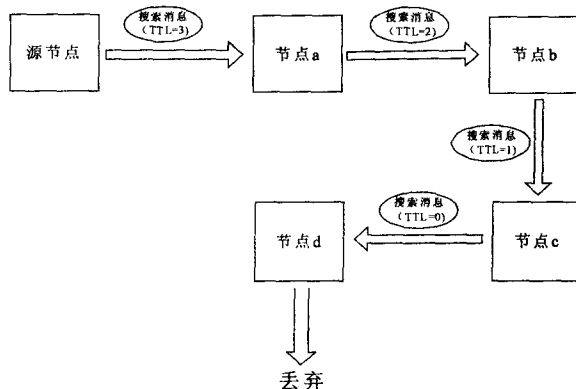


图 7 为搜索消息增加 TTL 值

- sion data. *Bioinformatics*, 2005
- 39 Knijnenburg T A, Reinders M J T, Wessels L F A. Artifacts of Markov blanket filtering based on discretized features in small sample size applications. *Pattern Recognition Letters*, 2006
 - 40 Shipp M A, et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 2002,8(1):68~74
 - 41 Goh Liang, Song Qun, Kasabov N. A Novel Feature Selection Method to Improve Classification of Gene Expression Data. In: *Proceedings of the Second Conference on Asia-Pacific Bioinformatics*, vol. 29, 2004. 161~166
 - 42 Cho J H, Lee D, Park J H, et al. New gene selection for classification of cancer subtype considering within-class variation. *FEBS Letters*, 2003, 551(1):3~7
 - 43 Xu Xian, Zhang Aidong. Boost Feature Subset Selection: A New Gene Selection Algorithm for Microarray Dataset. *Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v3992 LNCS-II, Computational Science-ICCS 2006 In: 6th International Conference, Proceedings, 2006. 670~677
 - 44 Tang Yuchun, Zhang Yan-Qing, Huang Zhen, et al. Granular SVM-RFE Gene Selection Algorithm for Reliable Prostate Cancer Classification on Microarray Expression Data. *BIBE*, 2005. 290~293
 - 45 Kluger Y, Basri R, Chang J T, et al. Spectral biclustering of microarray cancer data; co-clustering genes and conditions, *Genome Res*, 2003
 - 46 Liu Bing, Wan C, Wang Lipo. An efficient semi-supervised gene selection method via spectral biclustering, *Nanobioscience, IEEE Transations on*, 2006,5(2)
 - 47 Chu F, Xie W, Wang L P. Gene selection and cancer classification using a fuzzy neural network. In: *Proc. IEEE Annu Meeting North Amer Fuzzy Information Processing Soc*, vol 2, 2004. 555~559
 - 48 Tibshirani R, Hastie T, Narashiman B, et al. Diagnosis of multiple cancer types by shrunken centroids of gene expression. In: *Proc. Natl Acad Sci USA*, Vol 99,2002. 6567~6572
 - 49 Tibshirani R, Hastie T, Narashiman B, et al. Class prediction by nearest shrunken centroids with applications to DNA microarrays. *Statistical Science*,2003,18:104~117
 - 50 Deutsch J M. Evolutionary algorithms for finding optimal gene sets in microarray prediction. *Bioinformatics*, 2003,19:45~52
 - 51 Tan Feng, Fu Xuezheng, Zhang Yanqing, et al. Improving feature subset selection using a genetic algorithm for microarray gene expression data. *Evolutionary Computation*. In: *IEEE Congress*, 2006. 2529~2534
 - 52 Li Shutao, Liao Chen, Kwok J T. Wavelet-based feature extraction for microarray data classification. In: *International Joint Conference on Neural Networks*, Vancouver, Canada, July 2006
 - 53 Krishnapuram B, Carin L, Hartemink A. Gene expression analysis: Joint feature selection and classifier design. In: Scholkopf B, Tsuda K, Vert J-P, eds. *Kernel Methods in Computational Biology*, MIT, 2004. 299~318
 - 54 Al-Mubaid H, Ghaffari M. Identifying the most significant genes from gene expression profiles for sample classification. In: *2006IEEE International Conference on Granular Computing*, 2006. 655~658
 - 55 Paul T K, Iba H. Extraction of information genes from microarray data. In: *GECCO'05*, Washington, DC, USA, 2005

(上接第 128 页)

二是为每条消息设定唯一标识符(Unique Identifier, UID)。当搜索消息从源节点发出时,给每条消息都加上一个唯一的标识符。每个节点都需要维护一个消息的 UID 列表,当某一节点收到搜索消息后,取出消息中的 UID 字段,把它同自身所维护的 UID 列表中的值相比较。如果该 UID 已经存在于列表中,说明该节点此前已经收到过该消息,那么该节点简单地丢弃该消息;否则,节点将该消息进行转发,并将其 UID 加入到列表中。如图 8 所示,当源节点发送一条 UID 为 m3 的搜索消息给节点 a 和节点 b 时,节点 a 与节点 b 分别检查各自的 UID 列表,节点 a 的 UID 列表中没有 m3 的记录,因此节点 a 将 m3 加入到列表中,并将消息进行转发;而节点 b 的 UID 列表中已存在 m3,因此节点 b 丢弃该消息。

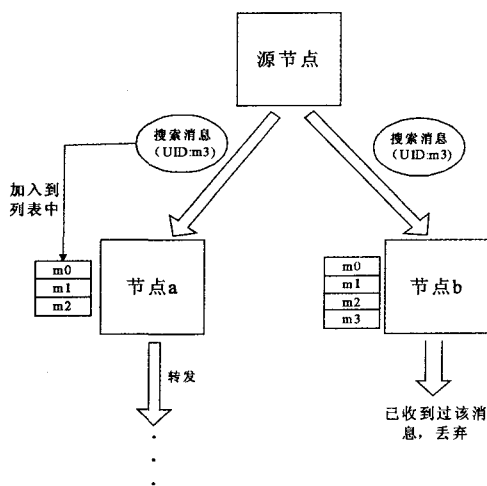


图 8 为搜索消息增加 UID

三是设定消息的重复发送次数^[7]。通过事先设定消息的

重复发送次数,如果搜索消息第一次发送后没有搜索到所期望的结果,则将消息的 TTL 再加一次发送。由于 TTL 值的增加,第二次搜索消息的搜索范围将比第一次搜索的范围更广。如果第二次搜索仍然未能找到所期望的结果,则将 TTL 值加 1 再发送。以此类推,直到发送次数达到预先设定的重复发送次数限制为止。

结束语 上述针对 JXTA 技术进行分析,提出了一种利用 JXTA 技术来构建 P2P 文件共享系统的设计方案,论述了对点对点实时聊天,文件共享和文件搜索三个功能模块的建模方案,并讨论了如何对传统的文件搜索算法进行改进。

参考文献

- 1 Granville L Z, da Rosa D M, Panisson A, et al. A methodology for P2P file-sharing traffic detection [J]. *Managing Computer Networks Using Peer-to-peer Technologies*, 2005,7:62~68
- 2 Traversat B, Abdelaziz M, Doolin D, et al. Project JXTA-C: enabling a Web of things [C]. In: *Proceedings of the 36th Annual Hawaii International Conference*, 2003
- 3 Kim K, Park D. Mobile NodeID based P2P algorithm for the heterogeneous network [C]. In: *Proceedings of the Second Embedded Software and Systems International Conference*, 2005
- 4 Hsing Mei, Chang S. PP-COSE: a P2P community search scheme [C]. In: *Proceedings of the Fourth Computer and Information Technology International Conference*, 2004,9:416~423
- 5 Maibaum N, Munclt T. JXTA: a technology facilitating mobile peer-to-peer networks [J]. *Mobility and Wireless Access Workshop*, 2002(10):7~13
- 6 庄雷,潘春建,郭永强,等. Gnutella 网络的连接管理[J]. *软件学报*, 2005, 16(1): 158~164
- 7 黄维维,黄铭钧,陈建利,等. 一种基于自配置策略的新型 Peer to Peer 平台系统[J]. *软件学报*,2003,14(2): 237~246