

一种基于多移动 Agent 的对等计算动态协作模型

徐小龙¹⁾ 王汝传^{1),2)}

¹⁾(南京邮电大学计算机学院 南京 210003)

²⁾(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 对等计算技术为 Internet 上海量节点之间计算和信息资源的共享提供了基础平台,对等节点的协作可以有效利用分布于各对等节点上的资源以解决单节点难以完成的复杂任务,然而对等计算环境的动态性、不可预测性等特征也为顺利的协作带来了困难.提出一种基于多移动 Agent 的对等计算动态协作模型,利用多 Agent 技术构建 P2P 平台的协作层,将对等节点之间的协作转换为 Agent 之间的协作,再通过移动 Agent 来动态部署针对具体任务的功能组件.还给出了该动态协作模型在密码分析中的应用示范和对比实验数据结果,对该模型的实用价值进行验证.

关键词 对等计算;协作联盟;多移动 Agent;密码分析

中图法分类号 TP311

A Dynamic Mobile Multi-Agent-Based Collaboration Model of Peer-to-Peer (P2P) Computing

XU Xiao-Long¹⁾ WANG Ru-Chuan^{1),2)}

¹⁾(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003)

²⁾(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract The Peer-to-Peer Computing (P2P) technology is able to provide the foundational platform for computing and information sharing among peers of Internet. The cooperation among peers makes it easy to utilize resources resided on each peer to accomplish complex tasks which are too difficult for any single peer to implement. However, the dynamic and unpredictable P2P environment makes it hard for peers to collaborate successfully. This paper proposes a flexible and dynamic collaboration model of P2P system based on the mobile multi-agent technology. The collaboration layer of P2P is constructed with multi-agents, which converts the collaboration relation from peers to agents. And the functional components for a certain task are deployed by mobile agents dynamically. In the last part of this paper, the cryptanalytic application of the collaboration model and the performance test results are given, which prove the practical value of this model.

Keywords peer-to-peer computing; collaboration model; mobile multi-agent; cryptanalysis

1 引 言

目前,网格计算技术(Grid Computing)、对等计

算技术(Peer-to-Peer Computing, P2P)和普适计算技术(Pervasive Computing)已经成为分布式计算技术领域的三大研究热点. Foster 等对其中的 P2P 计算技术作出如下定义^[1]: P2P 计算技术为加入 In-

ternet 的各种资源的使用主体和提供主体提供了非中心化的、自组织的、所有的或大部分联系是对称的分布式环境,在广域的范围实现了对数据信息、存储空间、计算能力、功能组件、通信资源的充分利用。P2P 环境可以包容海量的对等节点(Peer)。目前 P2P 技术主要应用于基于 Internet 的文件共享系统,其中著名的如 Napster、Gnutella、Freenet 等;还有应用于科学计算领域的对等计算系统,如加利福尼亚大学的 SETI@Home 项目,利用 P2P 的计算能力来分析无线电望远镜拾取的信号试图寻找外星智能生物。

P2P 技术的价值还在于为开放的、动态变化的网络环境里各主体间的协同工作提供了基础运行平台,以完成单个节点难以完成的复杂任务。然而,单纯的 P2P 环境中只包含了提供驻留于各对等节点上的各类资源(计算资源、信息资源等)、负责资源发现和定位等的功能的管理系统以及基于 Internet 之上的通信协议等;P2P 环境中资源的数量、质量和种类存在着较大的差异,任务表现出多样性;P2P 环境中的对等节点随机加入和退出网络,使得环境具有不可预测性,任务的承担者难以事先确定。这些特征都为在 P2P 环境中进行复杂任务的协同工作造成了困难。

本文的贡献在于:提出一种基于多移动 Agent (Mobile Multi-Agent) 的 P2P 动态协作模型。该模型主要思想是对等节点及其资源、节点间协作和对资源的使用进行分层剥离:底层的对等节点只负责提供和管理各类资源以及基本的通信机制;各对等节点通过多 Agent (Multi-Agent) 形成基于 P2P 平台的协作层,负责对等节点间的协作;对等节点通过移动 Agent (Mobile Agent) 来动态部署完成特定任务所需要的功能组件,被部署特定组件后的对等节点就成为针对某个任务的一个工作节点。这种分层剥离、动态部署的方法使得在 P2P 环境中进行协作的灵活性和鲁棒性得到提高,同时使针对特定任务的复杂协作系统的动态建立和正常运行成为可能。本文重点阐述基于多 Agent 的 P2P 协作模型的形成、任务以及如何解决保障任务执行的可靠性等问题、如何通过移动 Agent 来动态部署 P2P 中的各对等节点上的特定的功能组件,并给出在密码分析中的应用示范。

2 多 Agent 技术和移动 Agent 技术

Agent) 的定义,Agent 是存在于某一环境中的实体,具有自主性 (Autonomy)、主动性 (Activity)、反应性 (Reactivity)、社会性 (Sociality)、智能性 (Intelligence) 等拟人特征,能根据具有的知识信念以及周围发生的事件进行感知、推理、规划、通信,并反作用于环境^{①[2]}。

多 Agent 系统^[3] 是多个 Agent 所组成的系统,强调了 Agent 社会性特征。多 Agent 之间彼此在逻辑上相互独立,通过共享知识、任务和中间结果,协同在工作中形成问题的解决方案。因此,Agent 之间的交互过程不是简单地交换数据,而是参与某种社会行为,具体表现在以下 3 个方面:

(1) 协调 (Coordination)。具有不同目标的多个 Agent 对其目标、资源等进行合理安排,以规划各自行为,最大限度地实现各自目标。

(2) 协作 (Collaboration)。多个 Agent 通过协调各自行为,合作完成共同目标。

(3) 协商 (Negotiation)。多个 Agent 借助通信,交换各自目标,直到多 Agent 的目标一致或不能达成协作。

简言之,协调是对环境的适应,协作是非对抗的 Agent 之间保持行为协调的一个特例。

移动 Agent^[4] 是一类特殊的软件 Agent,它除了具有软件 Agent 的基本特性外,还具有移动性,并可以代表用户完成指定的任务。由于移动 Agent 可以在异构的软、硬件网络环境中自由移动,因此这种新的计算模式能有效地降低分布式计算中的网络负载、提高通信效率、动态适应变化了的网络环境,并具有很好的安全性和容错能力。移动 Agent 不同于远程过程调用 (Remote Procedure Call, RPC)、一般的进程迁移或是 Java 语言中的 Applet,因为 Agent 的迁移方式、时间和地点是可以根据自身需要进行选择的,并可在客户机和服务器之间进行双向移动。移动 Agent 适用于在地理上或逻辑上分布、自主或异构的结点间提供应用服务或中间件服务,这使移动 Agent 适宜应用于 P2P 的环境中^[5]。

多移动 Agent 系统则包含了可以进行协同工作的多 Agent,也包含了可在各个节点迁移的移动 Agent,如图 1 所示。其中,协调、协作和协商存在于同一节点的 Agent 之间,也存在于不同节点的 Agent 之间,而移动 Agent 一般是由本地静态 Agent

孵化(spawn)生成,并可携带数据、程序和状态信息在节点间迁移。

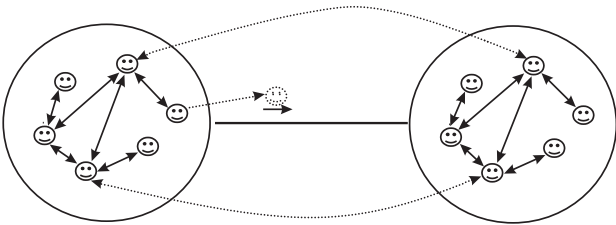


图 1 多移动 Agent 系统

3 基于多移动 Agent 的 P2P 协作模型

3.1 P2P 的层次结构

本文的主要思想就是在 P2P 体系中引入多移动 Agent 技术,从而将 P2P 的层次结构进行重新划分.共划分为三层:P2P 基础服务设施层(Foundation Service Layer)、协作层(Collaboration Layer)和应用层(Application Layer),如图 2 所示。

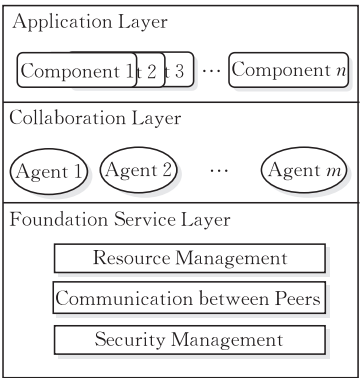


图 2 P2P 平台分层结构

P2P 基础服务设施层中的资源管理模块(Resource Management)主要包括如下部分:资源的管理模式;资源的描述;资源的注册;资源的集成;资源的可靠性管理;资源的管理策略;资源的查找;资源的发现等等.对等通信模块(Communication between Peers)实现了对等节点发现和注册、对等组织的建立和对等节点之间通信.安全管理模块(Security Management)主要负责了对等节点之间的身份认证和访问控制等安全行为。

P2P 协作层事实上是由多 Agent 技术构建的一个面向任务协作的 Agent 社会,这个社会是由对等节点内部的 Agent 和对等节点之间的 Agent 共同组成.这种设计表示对等节点在需要进行协同工作的时候,所有与协调、协作和协商的工作全部由多 Agent 构建的协作层完成.这就将协作层从 P2P 计

算体系中剥离出来,使得对等节点之间的协作转化为 Agent 之间的协作,这就使得复杂的协作活动可以更灵活、更可靠、更顺利地完成。

P2P 应用层是由各功能组件(component)构成的.这些组件并非预先安装在对等节点上,而是根据协作层多 Agent 协商的结果,由移动 Agent 按需动态携带、迁移并部署到目的节点上.功能组件可由任务的发起者来制作和发行.这种机制就使得资源和对资源的具体使用进行了剥离,使得对等节点本身可以更单纯,更容易在不可预测的环境里面进行暂时的合作,并联合完成任务。

3.2 协作模型的建立

本文采用基于时序逻辑和添加了与 Agent 行为相关的模态算子的逻辑框架^[2,6]建立基于多 Agent 的协作层模型,其中包含了:

信念算子 *Bel*. $Bel_i\varphi$ 表示 Agent *i* 具有相信能够实现目标 φ ;

能力算子 *Can*. $Can_i\varphi$ 表示 Agent *i* 具有实现目标 φ 的能力;

动作算子 *Act*. $Act_i\alpha$ 表示 Agent *i* 可以采取动作 α ;

达成算子 *Ach*. $Ach_i\alpha\varphi$ 表示 Agent *i* 可以采取动作 α 实现目标 φ ;

承诺算子 *Com*. $Com_i\varphi$ 表示 Agent *i* 承诺将实现目标 φ 。

如上所述,将多 Agent 应用于 P2P 计算环境后,在 P2P 的计算平台上形成了一个 Agent 社会.对于一项工作需要多个对等节点及其之上的 Agent 协作完成的原因是任务的发起者自身的能力不足,这里的能力主要依据为本地的资源情况(包括资源种类、效率等),即任务发起者的本身的资源不足以完成该任务,或者代价太大,或是由于无法在规定的期限内完成该任务,从而产生的协作的愿望.如果用 *G* 来表示该任务的实现目标,用 *T* 表示时间限制因素,用 *R* 表示资源限制因素,则协作愿望可表示为 $Bel_i\rightarrow (Can_i(G\wedge T\wedge R))\wedge Bel_i\Diamond (T\wedge R\text{ Until }G)$,其中 \rightarrow 表示“非”, \Diamond 是表示“有时或最终”,Until 是表示“直到...”。

基于协作愿望,任务的发起者需要承担遴选工作协作者的任务,即在 Agent 社会里面,选取一个能够完成该任务的 Agent 子集来联合完成该任务.受于各自能力的限制,在该子集里面的 Agent 都只能完成其中的一部分工作.这需要将任务分割为可独立完成的子任务,即将整体目标 *G* 进行逐步细化

为多个相对独立的子目标,这实质是形成一个“*And* 目标集”:

$$GoalGroup = \{G_1, G_2, \dots, G_n\}.$$

本文基于认为划分的目的就是实现目标集合中所有的目标,从而达到实现整体目标 G 的目的,因此 $GoalGroup$ 是“*And* 目标集”:

$$\forall 1 \leq i \leq n (And\ GoalGroup(G_i, G)) \leftrightarrow (Bel_a(\bigwedge_{i=1}^n (Ach_{ax_i} G_i) \rightarrow (Ach_{ay} G))).$$

在“*And* 目标集” $GoalGroup$ 中各目标的执行次序受到限制,一般情形下主要包括以下几种情况:

- (1) 一个目标的实现结果是另一个目标进行的前提,即直接先后关系,在目标间形成实现因果链;
- (2) 两个目标的执行过程需要并发进行,在目标间有紧密合作关系;
- (3) 不具有上述两种的直接相关的关系,但可能存在间接相关关系.

由目标之间的直接先后关系和并发关系可容易得出 $GoalGroup$ 中各目标所组成的全局执行次序有向关系图,如图 3 所示.

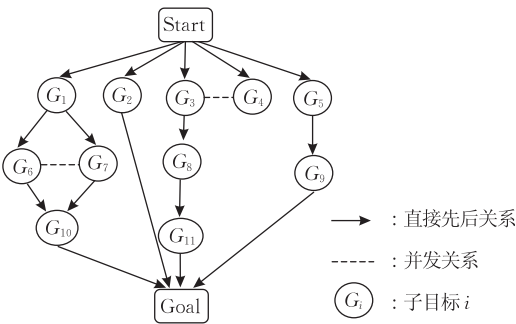


图 3 全局执行次序有向关系图

建立有向关系图后,需要检测其中是否存在死循环情况,如果存在死循环情况,则需要修改 $GoalGroup$ 集合中的目标元素及其全局执行次序有向关系图.采用以下算法予以检测.

列出直接先后关系矩阵:

$$R_{GoalGroup} = \begin{bmatrix} r_{G_1 G_1} & \cdots & r_{G_1 G_n} \\ \vdots & \ddots & \vdots \\ r_{G_n G_1} & \cdots & r_{G_n G_n} \end{bmatrix},$$

其中, $r_{G_i G_j} = \begin{cases} 1, & \text{当 } G_i \text{ 和 } G_j \text{ 有直接先后关系时} \\ 0, & \text{当 } G_i \text{ 和 } G_j \text{ 没有直接先后关系时} \end{cases}$.

$R_{GoalGroup}$ 是布尔矩阵,对 $R_{GoalGroup}$ 进行正闭包运算:

$$R_{GoalGroup}^+ = R_{GoalGroup}^1 + R_{GoalGroup}^2 + \cdots + R_{GoalGroup}^n,$$

这里的运算都是布尔逻辑运算.如果在所得出

的 $R_{GoalGroup}^+$ 中存在 $r_{G_i G_i} = 1$ 的情况,则表明存在该有向关系图存在死循环问题,需要予以消除.如果确实需要循环,可在循环发起点上设置计数器,使得计数器的值不断消减,从而跳出循环;如果是无意识的循环,则需要修改目标集合中的目标及目标间的关系.

但是,鉴于 P2P 计算环境的不可预测的动态性特点,需要根据各子目标的重要性、复杂性、难易性程度,在选择任务执行者时,增加一定的冗余度,即选取多个主体来同时来执行同一子任务,或是采用待定备份的方式,以降低因为某一个目标的未实现而导致整体目标无法达成的概率. P2P 网络环境中海量数目的节点状况易于实现这一要求. 针对每个目标 G_i 建立一个子目标集 G_{iGroup} :

$$G_{iGroup} = \{G_{i1}, G_{i2}, G_{i3}, \dots, G_{im}\},$$

显然这是一个“*Or* 目标集”,因为只要实现集合 G_{iGroup} 中的任意一个目标,就表明实现了 G_i , 即有

$$\forall 1 \leq j \leq m (And\ G_{iGroup}(G_{ij}, G_i)) \leftrightarrow (Bel_a(\bigvee_{j=1}^m (Ach_{ax_{ij}} G_{ij}) \rightarrow (Ach_{ay} G_i))).$$

在子目标集 G_{iGroup} 中的各个执行主体可以并行的执行,或是当最佳执行者失败的情况下在轮次执行,因此执行的次序方式只跟该子目标 G_i 的重要程度和时间限制相关.

利用合同网(Contract-net)^[7]的思想,任务发起者将每个子目标 G_i (或是集合 G_{iGroup} 中的目标元素)及其时间限制因素 T_i 和资源限制因素 R_i 等信息制作成标书发布于 P2P 网络中,由各个对等节点进行竞争投标,任务发起者从中选择最适合实现对应子目标的主体,实施合同,将该子目标所对应的任务派发给拥有相应资源的对等节点.而基于上述的冗余思想,或者将任务也同时发给同样拥有相应资源的对等节点并指示同时执行,或者发给待定通知,在最佳对等节点失败的情况下,再予以执行.

总之,在协作层的 Agent 与 P2P 基础服务设施层的资源标识形成了对应关系,即不同对等节点上的不同资源分别由相应的 Agent 来代表,并作为 Agent 的能力参量;而本地对等节点的对资源的使用策略即为 Agent 的意愿参量.

这实质是将目标集和协作层 Agent 集合形成了映射关系,即将目标(代表任务)投射到 Agent 集合上,目标执行关系转移为对等节点上 Agent 之间的一种协作关系.协作的依据是根据本地的资源和策略. Agent 之间必须建立信任关系,才能使后续的协作成为可能.

这种信任包括两个过程,首先是任务发起者和其协作伙伴之间必须建立相互信任的关系,其中首先是任务发起者需要信任声称能够接受并完成任务的协作伙伴.这里存在两种情况,我们分别给出定义.

定义 1. 强信任关系($sInitiatorBelief$),即任务发起者需要信任甚至只是承担任务为实现“Or 目标集” G_{iGroup} 中的任意一个目标的所有协作伙伴,才能发起该工作,如下式:

$$sInitiatorBelief = Bel_{initiator} \sum_{i=1}^n (Ach_{a_i} x_i G_i),$$

$$\bigwedge_{j=1}^m (Ach_{a_{ij}} x_i G_{ij}) \rightarrow (Ach_{a_i} x_i G_i),$$

其中 $initiator$ 代表发起者, a_i 为承诺实现目标 G_i 的协作主体 i 或是主体集合,而 a_{ij} 为该集合中的主体 j , x_i 为实现该目标需要采取的动作.

定义 2. 弱信任关系($wInitiatorBelief$),即任务发起者只需信任“And 目标集” $GoalGroup$ 中各目标都能够被完成,而如果是“Or 目标集” G_{iGroup} 则其中任一目标可被实现,就可发起该工作,如下式:

$$wInitiatorBelief = Bel_{initiator} \sum_{i=1}^n (Ach_{a_i} x_i G_i),$$

$$\bigvee_{j=1}^m (Ach_{a_{ij}} x_i G_{ij}) \rightarrow (Ach_{a_i} x_i G_i).$$

因为 G_{iGroup} 是为了提高任务完成成功概率而设计的具有冗余性质的集合,集合中任一目标被实现,就表明实现了目标 G_i ,因此本文中发起者与协作伙伴的信任关系采用弱信任关系.这里协作伙伴同时也作为任务承担者,而任务发起者虽然一般也作为某一个子任务的承担者,并同处于某一对等节点,但这里我们为了使模型结构清晰、灵活,将其身份按照角色还是予以剥离,即分别用 Agent 代表.同时,有直接关联的(先后关系或并发关系)的协作伙伴之间也必须相互信任,这种信任可由任务发起者对各协作伙伴的信任关系而间接建立,即因为发起者信任而信任,这种设想是基于任务发起者理所当然希望任务顺利完成的;如果需要建立起全局的强信任关系,则参与协作的 Agent 两两之间都必须建立信任关系,本文采用前一种方式.

3.3 协作组件的发布

一般来说,不同的任务常常需要不同的程序组件来具体执行,常常利用相同的资源如 CPU、内存等计算资源,或是数据、文件等信息资源,而这些资源是对等节点本身所拥有的.而预先在对等节点上安装所有的功能组件是不可行的.本文基于资源固有而组件则采取按需发布的思想来完成某个具体的

任务.方法是在对等节点之间建立了协作关系后,由任务发起节点或通过移动 Agent 来动态部署针对本次任务的各个功能组件.具体步骤如下:

1. 任务发起者根据各协作 Agent 所声称实现的目标,准备相应的可执行相应任务的功能组件;
2. 根据协作 Agent 所在的对等节点的地址及其标识,通过移动 Agent 携带功能组件移至目标节点;
3. 目标节点对移进的 Agent 进行身份验证等安全性核查,若不通过则拒绝接收并抛弃该 Agent,若通过转至步 4;
4. 对该 Agent 所携带的组件进行检查,查看其是否是自己所声称实现的目标所对应的功能组件,其输入、输出和功能是否符合事先的设定,若不符合则转步 5,若符合则转步 6;
5. 抛弃该组件并通知任务发起者重发,转步 2;
6. 通知任务发起者已接收,安装该组件,根据协作模型及事先的设定运行该组件.

4 实验系统

4.1 应用示范

为了验证和应用本文提出的基于多移动 Agent 的对等计算动态协作模型,我们选择需要高计算能力的密码分析(cryptanalysis)作为应用示范.因为密码分析问题易于将一个需要高计算能力的整体目标(在整个密钥空间中搜索密钥和恢复明文)和工作分解为若干个相对简单的子目标(在若干可能密钥集合中搜索密钥和恢复明文)和若干明晰的步骤来完成,具有典型性和代表性.

密码分析技术是检验密码技术强固性的重要手段,而对某个加密或是签名算法仅仅通过理论的分析是不够令人信服的,尤其是一些密码算法不能提供明确的数学理论基础,还有一些密码算法存在需要避开的弱密钥.这就需要我们采用强计算能力对其进行实际的抗攻击强度分析和测试,基于 Internet 的 P2P 计算环境中的海量节点可提供强大的计算能力,因此我们可以利用部署了基于多移动 Agent 协作系统的 P2P 计算环境作为我们的密码分析测试平台.我们的应用示范选择“已知明文攻击(know-plaintext attack)”分析方法,即根据已知的部分的消息密文和相应明文,通过强力密钥搜索方式来分析获得相应的密钥,再根据密钥来分析其它的密文,这里选择对称加密算法 DES(Data Encryption Standard,数据加密标准).基于多移动 Agent 协作系统的 P2P 计算环境下的密码分析系统按照如下的步骤工作:

1. 首先根据本文上一节的方法在 P2P 计算环境下建立针对本次密码分析任务的对等节点间协作关系；
2. 然后由本次密码分析的发起者节点将可能的密码空间分割成多个可能密钥集合；
3. 发起者节点通过移动 Agent 将消息密文和明文、可能密钥集合和密钥分析程序发送给参与密钥分析的各对等节点；
4. 发起者节点还将通过移动 Agent 将待分析的密文和密文分析程序发送给参与密文分析的各对等节点；
5. 各协作节点对收到的移动 Agent、数据和程序进行检验,通过后便根据协作模型的规范进行相应的任务处理；
6. 进行密钥分析的对等节点,根据已有的密文和明文,基于得到的可能密钥集合反复试探,试图获得真正的密钥；
7. 某个节点试探获得密钥后,立刻发送给任务发起者进行验证,验证通过后将该密钥发送给一直处于待定状态的密文分析节点,同时发消息让其它密钥分析节点中止本次任务；
8. 密文分析节点根据密钥将待解密的密文恢复成明文,再将结果发送给发起者节点.

我们采用 JAVA 开发语言并基于 IK++ 的 Grasshopper 平台来实现本系统. 其中由于 JAVA 语言是目前是唯一的一种能够跨越所有平台运行的语言,具有良好的适应于异构系统的可移植性,同时采用安全管理器细化安全策略,用户能够依据预定义的限制来控制程序行为,这使其适用于开放的、动态的 P2P 计算环境,同时也为通过移动 Agent 部署程序的实用性带来保障.

在 Grasshopper 平台中我们实现了两种 Agent,即移动 Agent 和静态 Agent,并具体实现了上述的应用示范. 移动 Agent 具有在 Peer 之间频繁移动的能力;移动 Agent 和静态 Agent 都利用具体的位置来交互协作. 静态和移动 Agent 实际的运行环境称之为 Agency,Agency 由 Core Agency 和一个或多个 Place 组成;Core Agency 具体的功能有通信 (Communication)、管理 (Management)、持久性 (Persistence)、注册 (Registration)、安全 (Security) 和传输 (Transport) 等;Place 是 Agent 的直接停留和运行环境,在 Agency 内部提供一个功能性的逻辑分组,每一个 Place 可以提供特定的服务.

4.2 性能验证

我们在校园网内部署我们的实验环境,构建的 P2P 计算环境中共有 4 个主机节点,节点主机配置均为 CPU: Intel Pentium 4 2.5GHz, Memory: DDR 256MB, OS: Microsoft Windows XP/2003 Server. 由于实验环境中有限的 PC 数量,因此为了能够对系统模型进行有效的验证,我们将密钥空间进行压缩,并进行了多轮对比实验(测试 A:单台主机,测试

B:应用基于多移动 Agent 的对等计算动态协作模型的多台主机),以测试系统模型的性能,如表 1 所示.

表 1 对比实验结果

计算次数	单台主机花费的 总时间/s	多台主机花费的 总时间/s
1	125	279
2	346	434
3	670	598
4	3578	1321
5	2582	1094
6	7335	2343

从实验结果可以看出,在计算量比较小的情况下所花费的总时间比单台计算机完成相同任务时不具备显著优势,甚至花费了更多的时间. 原因在于在多个节点之间建立协作关系和进行通信,以及在通过移动 Agent 部署程序时花费了额外时间;在大计算量的情况下(单台计算机花费以小时计的计算时间),本文提出的系统模型性能优势比较明显.

5 结束语

本文提出的基于多移动 Agent 的对等计算动态协作模型所基于的主要思想是将资源、对资源的使用和各对等节点间的协作进行分离. 引入多移动 Agent 技术使得这种思想的实现明确易行. 这种方法对其它的分布式系统也有一定的参考应用价值.

进一步需要解决的问题是在进行目标及任务划分时,如何能够达到合理分割,从而使得在进行协作的过程中,对等节点之间的通信量或是协作的复杂度降低,从而提高整体的工作性能. 同时,在开放的、动态的 Internet 环境中,即使采取认证等安全技术,对等节点之间的欺骗仍是难以解决的问题之一,而这也为协作的进行带来了困难. 另外,建立和维护一个由不同组织和个人组成的 P2P 系统必须向参与者提供一定的激励,而有效的激励方式是维持一个 P2P 系统持久正常运行的关键因素之一,因此这也成为需要积极探讨、解决的问题.

参 考 文 献

[1] Ripeanu M, Foster I, Iamnitchi A. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. IEEE Internet Computing, 2002, 6(1): 50-57

[2] Shi Zhong-Zhi. Intelligent Agent and Its Applications. Beijing: Science Press, 2000(in Chinese)

(史忠植. 智能主体及其应用. 北京: 科学出版社, 2000)

[3] Wooldridge M. An Introduction to Multi-Agent Systems. Chichester, England: John Wiley & Sons, 2002

[4] Gray R, Cybenko G. D' Agents: Applications and performance of a mobile-agent system. Software-Practice and Experience, 2002, 32(6): 543-573

[5] Zhang Yun-Yong, Liu Jin-De. Mobile Agent Technology. Beijing: Tsinghua University Press, 2000(in Chinese)
(张云勇, 刘锦德. 移动 Agent 技术. 北京: 清华大学出版社, 2003)

[6] Sheng Qiu-Jian, Zhao Zhi-Kun, Liu Shao-Hui, Shi Zhong-Zhong. A teamwork protocol for multi-agent. Journal of Software, 2004, 15(5): 689-696(in Chinese)
(盛秋戡, 赵志崑, 刘少辉, 史忠植. 多主体团队交互协议. 软件学报, 2004, 15(5): 689-696)

[7] Smith R. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 1980, 29(12): 1104-1113



XU Xiao-Long, born in 1977, Ph.D. candidate, lecturer. His research interests include computer software, distributed computing and agent etc.

WANG Ru-Chuan, born in 1943, professor, Ph. D. supervisor. His research interests include computer software, computer network and Grid, information security, wireless sensor network, mobile agent.

Background

This paper is one of the research results of the projects funded by National Natural Science Foundation of China and the National High Technology Research and Development Program (863 Program) and other foundations. These projects are directed towards providing a secure mobile multi-agent platform and its typical applications in distributed computing systems, e. g. P2P (Peer-to-peer Computing) systems. Now the research team has realized a prototype system of mobile multi-agent with JAVA and its applications in the P2P, network management, routing systems, etc.

In this paper, the authors mainly talked about how to apply the mobile multi-agent technology to P2P systems to improve the collaboration ability of peers. The P2P technology is able to provide the foundational platform for computing and information sharing among peers of Internet. Now researchers usually focus their research on the architecture, resource organization and allocation and routing mechanisms of P2P, while the studies of P2P collaboration mechanism are still not enough.

The cooperation among peers makes it easy to utilize re-

sources resided on each peer to accomplish complex tasks which are too difficult for a single peer to implement. However, the dynamic and unpredictable P2P environment makes it hard for peers to collaborate successfully. To solve this problem, they propose a flexible and dynamic collaboration model of P2P system based on the mobile multi-agent technology. The collaboration layer of P2P is constructed with multi-agents, which converts the collaboration relation from peers to agents. And the functional components for a certain task are deployed with mobile agents dynamically.

In this paper, the authors describe the definition and features of mobile multi-agent technology firstly. Then a flexible and dynamic collaboration model of P2P system based on mobile multi-agent technology is introduced, including three layers of P2P architecture, collaboration model based on multi-agent and the steps of components deployment based on mobile agents, etc. In the last part of this paper, the cryptanalytic application of the collaboration model and the performance test results are given, which prove the practical value of this model.