

基于 Agent 与分布式缓存的 P2P 资源发现策略

陈杰, 彭德巍

(武汉理工大学计算机科学与技术学院, 武汉 430063)

摘要: 提出一种基于 multi-agent 与分布式缓存技术的资源发现模型, 静态 agent 可维护本地资源信息, 移动 agents 可通过在网络中的迁移来完成资源的发现, 分布式缓存可以提供一种高效的搜索方式。在 P2P 网络中, 使用该模型可快速找到资源, 减少网络阻塞, 改善扩展性与健壮性。

关键词: 移动代理; 资源发现; 蚂蚁算法; 信息素; 启发式

Strategy of P2P Resource Discovery Based on Agent and Distributed Caches

CHEN Jie, PENG De-wei

(Academy of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063)

【Abstract】 This paper presents a novel resource discovery strategy based on the multi-agent and distributed caches technology. Agents maintain local resource information and mobile agents migrate on network to discover shared resource. Distributed caches can provide an effective searching method. System model and decision algorithm is described. Resource can be quickly acquired and network traffic can be reduced greatly, scalability and robustness can be improved based on this model in P2P network.

【Key words】 Mobile Agent(MA); resource discovery; ant algorithm; pheromone; heuristic

1 概述

随着网络技术的飞速发展和网络规模的不断扩大, 计算机的性能得到了极大提高。但目前的互联网仍然是以C/S模式为主, Web技术的发展使得许多Web服务器成为信息的主要来源, 整个Internet 系统依附于这些少量的服务器节点, 导致网络中大量的边缘节点上的资源无法被充分利用, 形成大量的信息孤岛。对等网络(Peer to Peer, P2P)指分布式系统中的各个节点是逻辑对等的, 这种计算技术的目的就是希望能够充分利用互联网中所蕴含的潜在资源^[1]。

但由于P2P网络中高度的动态性、异构性以及分布式的资源, 导致发现资源变得非常困难。尤其随着节点的增加, 对资源模型的有效性和扩展性提出了更高的要求^[2]。目前已有一些模型, 比如: (1)以Napster为代表的集中式资源发现模型; (2)以Gnutella为代表的纯分布式资源发现模型; (3)以Chord^[3], CAN为代表的基于DHT的资源发现模型等。这些模型由于对共享资源缺乏启发式的机制, 且先前搜索结果对以后的搜索没有起到借鉴的作用, 因此它们更加关注资源的管理。同时, 它们在源对等体与目标对等体间可能产生巨大的网络阻塞, 增加网络的负担。

2 基于 multi-agent 与分布式缓存的资源发现模型

2.1 multi-agent 技术

Agent是一个具有智能性、自治性等特征的软件实体, 它能与所处环境进行交互并完成用户交给的任务。移动代理(Mobile Agent, MA)是一个能在异构网络中自主地从一台主机迁移到另一台, 并可与其他代理或资源交互的程序, 它可以自主地决定去哪儿、做什么、存活多久^[4]。适合在P2P网络中进行资源发现的特征如下:

(1)自治与协作: 在 P2P 系统中, 每个节点是一个独立与自治的实体, 节点的动态加入、离开以及异质的网络结构, 使得网络的规模与拓扑结构不断变化, 因此, 根据网络的状态改变与配置自身是很有必要的。agent 的这个特征可以满足此需求。此外, multi-agents 形成了一个 agent 组织, 可以通过彼此协作来高效地解决问题。

(2)MA 的迁移: MA 可在网络的节点范围内进行迁移, 并使用节点提供的资源与服务来完成用户给定的任务, 这样 MA 通过本地通信来进行高效的工作, 即将计算移往数据而不是把数据移往计算, 因此, 它可以减少网上原始数据的流量, 大大减轻网络负载, 消除网络延迟, 有效节约带宽。

(3)异步性: 网络中的信息量巨大, 在其中进行分布式的信息检索往往需要大量的时间, 同时由于 P2P 网络中存在大量的移动设备, 它们通常依赖于昂贵而脆弱的网络连接, 如果要求移动设备与网络之间必须保持连续的连接, 既不经济, 在技术上也不可行。为解决这一问题, 可将任务嵌入到 MA 中, MA 被派遣到网络上后, 移动设备就不必与网络保持连接, MA 可以独立创建它的进程, 异步、自主地完成资源发现任务。之后移动设备连接上网络, 收回 MA, 取得资源发现的结果。如图 1 所示。

(4)健壮性与容错性: 由于 MA 具有对错误和事件动态作出响应的能力, 因此减少了健壮和容错的分布式系统的难度。当某个 P2P 对等点上的信息发现错误时, 可以给正在运行的

基金项目: 武汉理工大学博士基金资助项目(471-38300846)

作者简介: 陈杰(1981-), 男, 硕士研究生, 主研方向: 计算机网络, 分布式处理; 彭德巍, 副教授、博士

收稿日期: 2007-11-30 **E-mail:** summer747@126.com

MA 发出警告，它们可以在很短的时间内移动到网络的对应点上，并且继续执行。

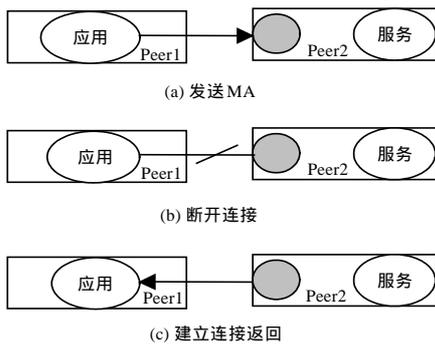


图1 MA 的异步性

2.2 分布式缓存

本文假设一个多 agent 系统，有 N 个节点或 agents，每个 agent 上提供一定数量的资源，例如本系统中的 IA。假定有 R 个不同的资源类型，每个 agent 可以在本地存储部分所知的中间 agent。假设，每个 agent 保存一个大小为 K 的私有缓存表，比如 agent A 存储着 K 个不同资源的缓存信息，分别对应 K 个提供资源的 Agent，在 A 缓存中的 agent 称为 A 的邻居。

最后，节点缓存构成了一个有向图 $G(V, E)$ ，称为缓存网络(cache network)。图中的每个节点相当于 1 个 agent 的缓存，节点 A 到每个邻居都有一条边。由于 1 个节点可提供 2 个或多个资源，因此该 agent 可能会出现在不同的 agent 缓存表中。从而，每个节点最多有 K 个邻居，可能存在 1 个节点的出度小于 K 。实例如图 2 所示。

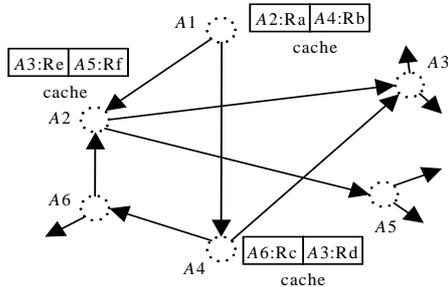


图2 部分缓存网络($K=2$)

2.3 资源发现模型

本文设计了 2 类 agent：(1)静态 agent，驻留在每个对等体上，记录环境、资源与缓存信息；(2)移动 agent，通过迁移到别处并与静态 agent 交互完成资源发现和获得。同时，设计了 2 类移动 agent，一类负责发现拥有资源的对等体，另一类负责搜索对应对等体上的资源。图 3 描述了本系统模型和 agent 间的交互过程。每个 agent 的功能描述如下：

(1)Control Agent(CA)：每个对等体上都有此静态 agent，它负责创建其他 Agents，接受用户的请求并把搜索结果返回给用户。

(2)Explore Agent(EA)：EA 被 CA 创建。它具有迁移的能力，负责发现拥有请求资源的对等体，并向 CA 返回对等体的地址。

(3)Search Agent(SA)：SA 也是由 CA 创建并且被 CA 派遣。从 CA 上得到迁移路径后，迁移到对等体上去搜索资源，并把结果返回 CA。

(4)Information Agent(IA)：IA 是静态的，它记录并提供本地资源信息，并保存用户每次的搜索结果和一个缓存表，这些结果信息可以被 EA 和 SA 利用。

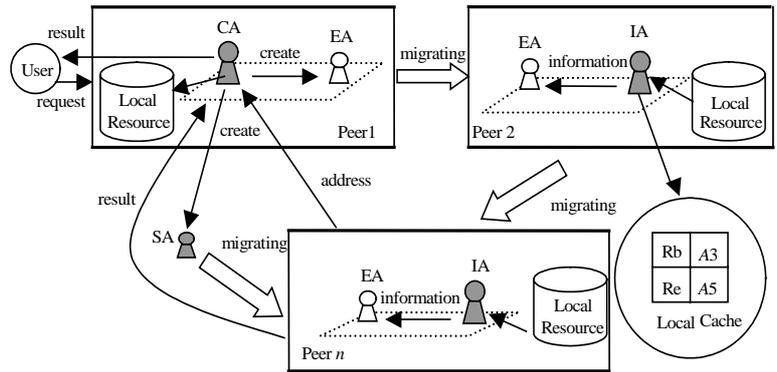


图3 资源发现与 Agent 交互模型

3 算法与策略

研究结果表明如果发现一个节点提供较多的共享资源，那么下次它很可能再被选择。根据这个特点，通过 agent 的自治与响应来充分利用先前的经验非常重要。另外，探测新节点适应 P2P 的动态网络也很重要。蚂蚁系统中的信息素与本地启发式信息反映了上述的特征。

3.1 蚂蚁算法

蚂蚁算法可以用来求解如 TSP(Traveling Salesman Problem)、分配问题、JSP(Job Shop Scheduling)问题等复杂的最优化难题。研究表明，蚁群总能找到一条从食物到巢穴之间的最短路径。这是因为蚂蚁在寻找路径时会在路径上释放一种特殊的信息素，形成信息素轨迹。蚂蚁能够感知信息素并选择信息浓度较高的路径，而其他的路径上激素浓度会随着时间的流逝而挥发。这种趋势使得蚁群找到最短路径成为可能^[5]。

3.2 基于蚂蚁算法的资源发现策略

蚂蚁算法是从以信息素为表现形式的历史经验信息和本地启发式信息综合考虑的，不同的参数值控制这 2 个方面的影响程度。另外，这个最佳方案是由一群蚂蚁得出的，信息素为蚂蚁的异步通信提供了一个特殊通道。

基于 multi-agent 技术，在 P2P 系统中使用通过信息素的学习机制以及通过本地启发式信息的探测机制去解决资源发现问题是可行的。具有某种记忆存储和学习能力的 agent 很适合做智能蚂蚁，资源发现过程就是自主学习与知识积累的过程。通过学习，EA 能够完成节点的发现，SA 利用他的结果去发现节点上的资源。

另外，P2P 网络中的资源发现和蚂蚁系统中食物的定位有很大的不同。例如，在蚂蚁系统中，食物是固定的，但在 P2P 系统中，有很多分布式的资源，网络拓扑与状态都在动态改变。因此，除了向其他节点学习外，Agent 也应该能够探测新加入的拥有共享资源的节点。针对 P2P 网络的动态性，本文采用了探测机制。

3.2.1 参数定义

通过以上的分析，为 EA 定义以下参数：

(1) d_{ij} ：对等体 i 和 j 间的网络延迟时间；

(2) $l_i(j)$ ：记录存储在 i 上关于 j 的负载信息，它是从等待队列里计算出来的平均等待时间；

(3) ReItem：资源内容，用字符串来描述资源；

(4) τ_{ij}^R, τ_{ij}^L : 前者是资源信息素的值, 后者是负载信息素的值。

IA 负责存储以下表的信息: (1)ResTable(RT): 保存自己的资源信息。表结构为 (ResName, ResType, τ_{ij}^R)。

(2)VisitedAgentTable(VAT): 记录访问本节点的 SA 和被它们访问的资源。表结构为 (AntOrigin, ResName, ForwNode)。

(3)ForwTable(FT): 保存曾经作为它下一个邻节点的信息。表结构为 (NerNodeAddress, Rank)。Rank, 代表了邻节点的重要程度。

(4)LoadTable(LT): 保存负载信息, 表结构为 (d_{ij} , $l_i(j)$, τ_{ij}^L)。

(5)CacheTable(CT): 保存缓存信息, 表结构为 (ResName, NodeAddress)。

3.2.2 迁移准则

迁移准则使用信息素与本地启发式信息特定比值来计算概率, 直接影响下一个节点的迁移选择。由于网络状态变化很快, 因此算法必须高效。本文采用式(1)^[6]计算:

$$P_{ij} = \begin{cases} 1 & j = \arg \max A_i(j) \quad l \in allowed_k \\ 0 & otherwise \end{cases} \quad q \leq q_0$$

$$A_i(j) = \frac{(\tau_{ij}^L)^\alpha \cdot (\tau_{ij}^R)^\beta \cdot (\frac{1}{d_{ij} + l_i(j)})^\gamma}{\sum_{n \in allowed_k} [(\tau_{in}^L)^\alpha \cdot (\tau_{in}^R)^\beta \cdot (\frac{1}{d_{in} + l_i(n)})^\gamma]} \quad q > q_0$$

$q_0 \in (0,1), q \in [0,1]$

3.2.3 资源发现过程

(1)当收到用户的搜索请求之后, CA 与 IA 进行交互, 在本地节点查询资源。如果资源可用, CA 把结果返回给用户。

(2)如果本地的 RT 表中无法找到可用资源, 则查询本地 CT 缓存表看是否有匹配的资源, 如果有, 转到下一步。否则 CA 创建一个 EA 并加上一定的约束, EA 根据式(1)计算迁移路径, 此路径也作为 SA 的路径。

(3)CA 创建一个 SA, 并给定约束及迁移路径, 然后 SA 迁移进网络。

(4)SA 根据路径与节点上的 IA 交互并更新相应节点 VAT 表。

(5)根据本地存在的资源, 返回给 SA 一个元组 (localAvailability, referredNode):

1)IF localAvailability=False, 表示本地资源不可用。IA 决定是采用基于概率的探测机制还是采用顺序机制。如果采

用探测机制, IA 将 refferedNode 值设置为 FT 表中 Rank 值最大的节点, SA 接着访问此节点, 同时更新相应的 VAT 表。如果采用顺序机制 SA 将继续访问路径中的下一个节点。同时表明此节点已经访问过, 因此, FT 表中的 Rank 值增 1。

2)IF localAvailability=True, 表明请求资源可用。如果满足约束, SA 返回到原始节点并把资源信息传送给 CA, 然后 CA 创建一个下载 agent 去下载发现节点上的资源。如果约束条件不满足, SA 根据路径继续迁移。

当一个对等体离开 P2P 系统, 在以上表中本节点的相应信息应该删除。同时, 一个新的节点加入, 它的信息应该被它的邻节点添加到以上表中, 且在 FT 中 Rank 的值设置为最大值与最小值之间的一个随机数。这样保存了新加入的节点有机会变成 referredNode。

4 结束语

本文使用 agent 与缓存技术来解决 P2P 网络中的资源发现问题, 通过 multi-agent 间的协作与交互完成资源发现。信息素构建了一种特殊的通信通道使 agent 可以学习先前的经验, 而且, 具有探测新节点的能力, 可以反映 P2P 网络中的动态特征。今后还有一些问题需要解决: (1)采用精确的方法去评估主机的负载状况并监视网络状态; (2)寻找一种更高效的缓存表的更新策略; (3)解决 P2P 与 Agent 的安全; (4)更进一步地研究与验证在其他网络中的应用, 如无线网络。

参考文献

- [1] Milojicic S. Peer-to-Peer Computing[R]. Palo Alto, USA: HP Laboratories, Technical Report: HPL-2002-57, 2002.
- [2] Hari B, Frans M. Looking up Data in P2P System[J]. Communication of the ACM, 2003, 46(2): 43-48.
- [3] Stoica I, Morris R. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications[C]//Proc. of ACM SIGCOMM. New York, USA: ACM Press, 2001.
- [4] Lange D B, Oshima M. Seven Good Reasons for Mobile Agents[J]. Communications of the ACM, 1999, 42(3): 88-89.
- [5] Dorigo V, Maniezzo A. The Ant System: Optimization by a Colony of Cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1996, 26(1): 29-41.
- [6] Dorigo M, DiCaro G. Ant Algorithms for Discrete Optimization[J]. Artificial Life, 1999, 5(3): 137-172.

(上接第 158 页)

路, 在实践中具有良好的应用性与可行性。通过以上分析可以得出, 就安全性而言, 本方案能够很好地抵御成员欺骗, 但如何防止管理者欺骗没有给出研究, 这是作者今后研究的方向。

参考文献

- [1] Shamir A. How to Share a Secret[J]. Communications of the ACM, 1979, 22(11): 612-613.
- [2] Backly G. R. Safeguarding Cryptographic Keys[C]//Proc. of the National Computer Conference of AFIPS. New Jersey, USA: AFIPS Press, 1979: 313-317.
- [3] Deng J L. Control Problems of Gray System[J]. Systems and Control Letters, 1982, 1(5): 285-294.
- [4] Deng J L. Introduction to Gray System Theory[J]. The Journal of

Grey System, 1989, 1(1): 1-24.

- [5] Shi K Q, Chen T S. A Grey General Lock and Central Public Cryptosystem(I)[J]. The Journal of Grey System, 2000, 12(4): 331-340.
- [6] Shi K Q, Chen T S. A Grey General Lock and Central Public Cryptosystem(II)[J]. The Journal of Grey System, 2001, 13(1): 57-64.
- [7] Shi K Q, Chen T S. On the Grey Encryption Problems of Information Security(I)[J]. The Journal of Grey System, 2000, 12(4): 215-224.
- [8] Shi K Q, Chen T S. On the Grey Encryption Problems of Information Security(II)[J]. The Journal of Grey System, 2000, 12(3): 255-262.