第 32 卷 第 2 期 西南师范大学学报(自然科学版) 2007 年 4 月
Vol. 32 No. 2 Journal of Southwest China Normal University (Natural Science) Apr. 2007

文章编号：1000－5471(2007)02－0111－04

一种基于移动 agent 的 P2P 网络任务负载均衡策略①

吴思远[1]， 吴代贤[2]

1. 重庆邮电大学 计算机学院，重庆 400065；2. 西南大学 计算机与信息科学学院，重庆 400715

摘要：提出基于移动 agent 的任务负载均衡策略(MA-LBP). 该策略使用移动 agent 技术替代传统的 VS 技术，并在异构 P2P 系统中考虑了任务迁移时的通信耗费，使得计算结果更加符合现实情况. 同时，移动 Agent 的游走特性消除了原有的计算瓶颈，实现较好的任务均衡效果.

关 键 词：分布式哈希表；点对点系统；任务负载；移动代理

中图分类号：G434 **文献标识码**：A

大部分使用 DHT 分布式哈希表的 P2P 系统都无法实现一个较好的任务均衡分布，因为大都只考虑用哈希函数来为对象产生一个 ID[1-2]，而对任务均衡问题仅做了解释. 在这些系统中，它们大都认为 P2P 系统在资源上是统一同构的，而忽略了 P2P 网络的异构特性. 虽然针对异构分布式系统的任务调度策略[3-4]已有很多，但都无法适应 P2P 网络的下述特性：①系统中处理器节点在每个时间周期内是动态变动的；②系统的整体处理能力是随处理节点的动态加入和离开而变化的. 因此 P2P 系统的任务均衡策略要特别考虑. 文献[1]首次提出利用虚拟服务(VS)来解决任务均衡问题，但效果不尽如人意. 文献[5]假设每个节点可根据自身能力调整虚拟服务，若发生过载，则移走一些 VS. 但这样会引发任务逆行的问题. 文献[6]为基于 DHT 的 P2P 系统提出了三个简单的任务均衡策略：一对一，一对多，和多对多. 基本思想是虚拟服务从超载节点移动到轻载节点. 但这个策略也没有考虑到任务迁移的网络耗费问题. 文献[7]提出了与[6]类似的利用 VS 迁移来实现任务均衡的策略，并且引入了就近原则，使得任务迁移可以控制在局部范围内，而不用在高延迟的外网中进行任务传送，但是 VS 的轻负载节点选择成为整个任务均衡重分配过程中的瓶颈.

针对上述已有研究工作，提出了基于移动 agent 技术的任务分配策略，延用传统 P2P 网络中的 DHT 技术来产生对象 ID，使用可重构的 k 度树来保证 P2P 网络的容错性和完全分布性，替代了 VS 的移动 agent (MA)使得 VS 计算瓶颈被分配到各个超载节点的 MA 中. MA 的游走特性使得 MA 不用像 VS 一样必须掌握全局网络信息来计算就近节点，MA 只需要掌握其登陆节点的临近节点信息就可以找出就近节点，并且 MA 携带游走记录使得它不会陷入循环.

1 P2P 网络环境构建

用 8 元组来构建 P2P 网络环境 $G=<T,N,M,W,E,F,C,L>$，其中：$T=\{t|t=DHT(t_i),i=1,2,\cdots|T|\}$，$|T|$为集合 T 中元素的个数，t 是任务 t_i 由 DHT 计算出来的 ID，其值为小的非负整数；$E=\{e_{ij}|DHT$ 节点 N_i 和 N_j 之间有链接，且 $e_{ij}=e_{ji}\}$；$N=\{N_i|i\in[1,|N|]\}$是 DHT 节点集合且动态可变的；$M=\{m_j^i|m_j^i.region\subseteq N_i.region\}$，$m_j^i$ 是驻留在节点 N_i 上的移动 agent，由 k 度树的叶子节点封装形成；$W=\{w(m_j^i)|$移动 agent m_j^i 的任务负载，随 k 值变化$\}$；$F=\{f_i|i\in[1,|P|]\}$，f_i 是 DHT 节点 N_i 的负载情况标记，$f_i=0$，表示 N_i 是负载适中的节点，$f_i=-1$，表示 N_i 是轻负载节点，$f_i=1$，表示 N_i 是超载节点；C

① 收稿日期：2006－09－12

作者简介：吴思远(1974－)，男，湖北汉阳人，硕士研究生，主要研究方向为计算机网络，分布式数据库，数据库技术等.

$=\{C_i \mid i\in[1,|N|]\}$，表示网络中各节点的能力，不同的计算能力代表了节点的计算异构性，反映了系统异构性；$L=\{L_i \mid i\in[1,|N|]\}$，表示网络中各节点当前的任务负载.

基于 VS 的 P2P 网络中的 DHT 节点采用两层结构(图 1)，节点本身没有很强的计算能力，所有服务都通过 VS 实现. 基于 MA 的 DHT 节点沿用两层结构(图 2)，但节点中没有很强的计算支撑，每个移动 agent 都具有简单计算能力，即将过载 VS-Based DHT 节点的计算耗费分摊在各个 MA-Based DHT 节点的 MA 上，因此 MA-Based P2P 网络的资源分配不再存在如 VS-Based P2P 网络中的资源计算瓶颈问题.

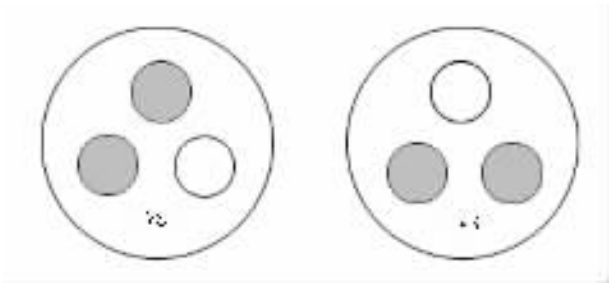

图 1 VS-Based DHT 节点

Fig. 1 VS-Based DHT Nodes

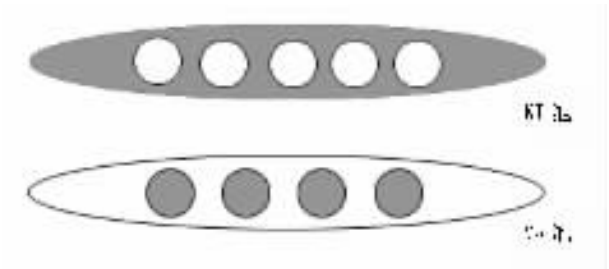

图 2 MA-Based DHT 节点

Fig. 2 MA-Based DHT Nodes

2 基于移动 agent 的任务负载均衡策略(MA-LBP)

2.1 构建分布式 KT 树

这个步骤与 VS-Based P2P 网络中构建度树类似[7]，不同之处在于 KT 树的各节点不存入 VS 中，而是直接存入其对应负责区域的 DHT 节点中，叶子节点存入 MA 池，非叶子节点存入 KT 池.

例如给定 $N=\{N_1,N_2,N_3\}$，$T=\{i/i\in[1,12]\}$，$k=2$，即生成 2 度树，且有 N_1. region$=[1,4]$，N_2. region$=[5,8]$，N_3. region$=[9,12]$. 其生成的 KT 树如图 3 所示，KT 树的各节点的存储情况如图 4 所示，图中黑圈表示叶子结点.

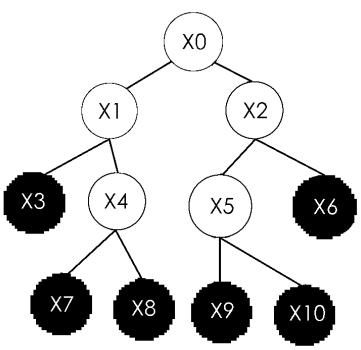

图 3 生成的 KT 树

Fig. 3 Generating a KT Tree

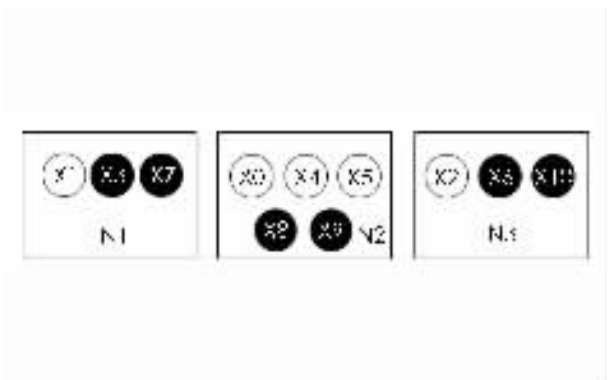

图 4 KT 树的节点存储情况

Fig. 4 The Storing of KT Nodes

2.2 划分轻载节点、超载节点和适量任务节点

由于 P2P 网络中随时有 DHT 节点离开导致其整体处理能力发生改变，所以在每次进行资源分配前都要重新计算当前网络的处理能力和负载，并由 KT 树的叶子节点即 m_j^i 将其报告给父节点并最终报告给根节点，对 N_i 来说有任务量 $L_i=\sum m_j^i$，$T_i=(L/C+\varepsilon)C_i$，ε 表示任务迁移与获得收益的交换.

$$F_i=\begin{cases}1, & L_i>T_i\\ 0, & T_i\geqslant L_{ii}>L_{i,\min}\\ -1, & L_i>T_i\end{cases}$$

当 $F_i=1$ 时，N_i 为超载节点；$F_i=0$ 时，N_i 为合适节点；$F_i=-1$ 时，N_i 为轻载节点.

2.3 任务重分配及传递过程

文献[7]采用模糊的 m 维坐标来标记物理位置相近的 DHT 节点，并通过映射函数[8]将其与 DHT 的标志空间对应起来，此后再决定将超载节点上的任务传递到物理位置相近的轻载节点上. 这种方法使得 VS 必须具备较强的计算能力，在获得全局网络信息的基础上选择相近轻节点，这是整个 P2P 网络任务均衡过程中的瓶颈步骤. 使用 MA 技术替代 VS，则每个 DHT 节点只需掌握局部信息(邻居节点信息)就可以实现任务均衡负载. 每个移动 agent(m_j^i)由 2 部分构成，第一部分为缓存，用来存放从登陆节点获得的局部网络信息，另一部分为智能体，用于判断离开还是登陆. 智能体算法 MA_Reconfig(N_i)描述如下：

MA_Reconfig(N_i)

{/ * Suppose Ni is a heavy node

l 是使得 $L_i - \sum_{j=1}^{l} w(m_j^i) \leqslant T_i$ 满足的最小整数；

for each m_j^i, do

{$Trace(m_j^i)=\{N_i\}$

/ * 移动 agent 的移动路径记录

log: $Nei(N_i)=\{N_t^{F_t} \mid e_{it}$ 存在$\}-Trace(m_j^i)$；

/ * 找出 N_t 的邻居节点信息

If $Nei(N_i)$中存在 $F_i=-1$ 的邻居节点　then

{选择 e_{it} 最小的 N_t^{-1} 作为 m_j^i 的登陆节点；

更新 $Trace(m_j^i)$，$N_t^{-1} \rightarrow Trace(m_j^i)$；

If $w(m_j^i)+L_t > T_i$　　then

{$L_i=N_i^{-1}$; $go\ log$; }}

else {选择 e_{it} 最小的 $N_t^{F_t}$ 作为 m_j^i 的登陆节点；

$L_i=N_t^{F_t}$; $go\ log$;}

m_j^i 迁移结束}}

3 实例及模拟试验

如图 5 所示 P2P 网络中，DHT 节点集合为$\{A^1, B^{-1}, C^{-1}, D^0, E^1, F^{-1}\}$，其 e_{ij} 值如图所示. A 为过载节点，建立邻居信息$\{B^{-1}, C^{-1}, D^0\}$. α 为移动 agent α；α 选择邻居中通信耗费最小的轻载节点 B^{-1}，更新 α 的 trace 集合为$\{A,B\}$，α 登陆 B^{-1}. 假设有 $w(\alpha)+L_B > T_B$，则 α 需继续迁移，此时 B^{-1}有邻居节点信息$\{A^1, D^0, E^1\}$，因为 A 已在 trace(α)中，因此 $Nei(B)=\{D^0, E^1\}$，都不是轻载节点，则选择通信耗费最少的 E^1 做为登陆节点，此时 trace(α)$=\{A,B,E\}$. $Nei(E)=\{B^{-1}, D^0, F^{-1}\}-trace(\alpha)=\{D^0, F^{-1}\}$. 选择 F^{-1}登陆，假设此时满足条件，α 结束迁移. 移动 agent α 完成重分配过程.

虽然 MA 的迁移不能保证总是移动到离 A 节点最近的可行轻负载节点上，但由于最近的轻负载节点是可行轻负载节点的概率在是 $1/(|P|-1)$，在最好情况即只有一个轻负载邻居节点的情况下为 1. P2P 网中节点间两两互连的情况总是少数，而全部节点都为轻负载节点的情况更少，因此第一次就选中可行轻负载节点的概率相对是较高的(图 6). 另一种找到最近可行轻负载节点的办法是 MA 回退，但 MA 记录的信息量会相应增多，且迁移的开销也会增大.

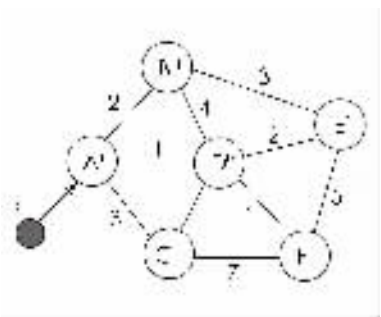

图 5　样例 P2P 网

Fig. 5　P2P Network Example

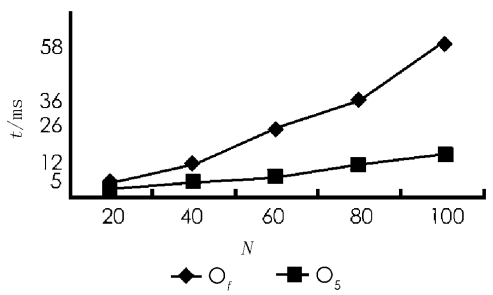

图 6　MA_LBP 的时间耗费图

Fig. 6　Time Cost of MA-LBP

图 6 是模拟试验中移动 agent 随网络中节点数的不同最终完成任务分配的时间耗费. 曲线 O_f 代表的是网络为全连接的情况，而 O_5 则代表网络为非全连接，且平均邻居节点为 5 的情况.

4 结论及将来工作

本文提出的基于移动 agent 的任务负载均衡策略(MA-LBP)是使用移动 agent 技术替代传统的 VS 技术，既考虑了系统的异构特性，也考虑了任务迁移时的通信耗费，并且 MA 的游走特性也消除了原有的计

算瓶颈，实现了较好的任务均衡效果. 但是这里没有考虑移动 agent 的安全性问题，在以后的工作中，我们会在使用尽可能少的开销下对移动 agent 进行安全性保证.

参考文献：

[1] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, . Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications[A]. Proc. ACM SIGCOMM[C]. U. C. San Diego: ACM. 2001: 149－160.

[2] A. Rowstron, P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems[A]. Proc. 18th IFIP/ACM Int'l Conf Distributed System Platforms (Middleware)[C]. Canada: Springer, 2001: 329－350.

[3] Leonidas Georgiadis, Christos Nikolaou. A fair workload allocation policy for heterogeneous systems[J]. Journal of Parallel and Distributed Computing, 2004, Vol. 64(1): 507－519.

[4] Yang Juan, Baiyun, Yuhui Qiu. DJSM-A Generally Dynamic Job scaling Mathematic Model for Parallel Applications[A]. Proceedings of The 2005 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'05)[C]. Las Vegas: NV. 2005: 1099－1105.

[5] F. Dabek, M. F. Kaashoek. Wide-Area Cooperative Storage with CFS[A]. Proc. 18th ACM Symp. Operating ystems Principles (SOSP)[C]. Banff : ACM. 2001: 202－215.

[6] A. Rao, K. Lakshminarayanan. Load Balancing in Structured P2P Systems[A]. Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS)[C]. CA : ACM. 2003: 68－79.

[7] Y. Zhu, Y. M. Hu. Efficient, Proximity-Aware Load Balancing For DHT-Based P2P Systems[J]. IEEE Tran. Parallel and Distributed Systems. 2005, 16(4): 349－361.

[8] T. Asano, D. Ranjan, T. Roos, E. Welzl. Widmaier. Space Filling Curves and Their Use in Geometric Data Structure [J]. Theoretical Computer Science. 1997, 181: 3－15.

A Mobile Agent Based on Load Balancing Policy Used in P2P Networks

WU Si-yuan, Wu Dai-xian

1. College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
2. Department of Computer Science and Information Technology, Southwest University, Chongqing 400715, China

Abstract: Most DHT based on P2P systems simply resorts to the hash function to generate the object Ids while dealing with the load balance problem. Some load balancing policies have been proposed to solve the problem in homogeneous environment. The authors propose a mobile agent based on load balancing policy which not only considered the heterogeneity property of the P2P systems but also considered the communication cost generated during the loads transferring. MA-LBP efficiently avoid the computing bottleneck caused by the traditional technique Virtual Servers and achieve the better load balancing.

Key words: distributed hash table; peer to peer systems; load balance; mobile agent

责任编辑 张 枸