P2P 计算网格路由和负载均衡算法

吴湘宁[1],胡成玉[1,2],汪 渊[3],王永骥[2]
WU Xiang-ning[1],HU Cheng-yu[1,2],WANG Yuan[3],WANG Yong-ji[2]

1.中国地质大学 计算机学院,武汉 430074
2.华中科技大学 自控系,武汉 430074
3.国防科技大学 计算机学院,长沙 410073
1.Computer Department,China University of Geosciences,Wuhan 430074,China
2.Dept. of Control Science & Engineering,Huazhong University of Science & Technology,Wuhan 430074,China
3.Computer Institute,National University of Defence Technology,Changsha 410073,China
E-mail:000sun@sohu.com

WU Xiang-ning,HU Cheng-yu,WANG Yuan,et al.Based routing and load-balancing algorithm for peer-to-peer computing grid.Computer Engineering and Applications,2007,43(32):105-107.

Abstract: Routing and load-balancing are two tasks particularly hard in Peer-to-Peer (P2P) computing grid.Traditional routing and load-balancing algorithms can not be applied to P2P networks:due to distributed and dynamic environment and the lack of central control,This paper presents a hybrid P2P routing and load-balancing algorithm which draws inspiration from ant collective intelligence,mobile agents-artificial ants deposit pheromone that used by taking routing decision and task scheduling when traveling between nodes.Simulation results show that the algorithm is effective and adapted to decentralized and self-organized P2P network.

Key words: P2P;grid computing;swarm intelligence;Ant Colony Optimization(ACO);load balancing

摘 要:路由和负载均衡是 P2P 计算网格的两个技术难题,由于 P2P 网络的分布性和动态性,以及缺乏统一的中心控制,使得传统的路由和负载均衡算法不能应用于 P2P 网络。提出了一种源自蚁群智能的混合路由和负载均衡算法,通过移动代理,即人工蚂蚁在节点间移动时所释放的信息素来作为路由和任务调度的依据。仿真结果表明该算法是有效的,且适用于具有分散和自组织特性的 P2P 网络。

关键词:P2P 网格计算;群体智能;蚁群优化算法;负载均衡
文章编号:1002-8331(2007)32-0105-03 **文献标识码**:A **中图分类号**:TP393

1 引言

对等计算网格(Peer-to-Peer Computing Grid)由许多地位均等、可提供共享计算资源的对等节点(Peer Node)所组成,是无中心节点的动态分布式网络。路由和负载均衡是 P2P 计算网格的两个重要研究内容,路由研究如何将数据包沿优质路径从源节点传送到目的节点,负载均衡则研究如何将计算负载均匀分配到各个节点上,从而提高网格的通信和计算效率。传统路由算法(如基于距离向量的路由算法 RIP、基于链接状态的路由算法 OSPF)中各个节点依靠相邻节点信息构建路由表,网络拓扑以及链路状态的变化传递较慢,且只考虑最短路径而无法避免拥塞[1]。而传统的负载均衡算法大多采用集中式调度技术,未考虑实时环境下各个节点的负载变化情况。因此,有必要研究能够适用于具有很强动态伸缩性、开放性和自组织性的 P2P 网络,而且能将路由和负载均衡有效结合的新算法。

本文介绍一种基于蚁群智能和多代理技术的计算网格分布式路由和负载均衡算法。蚁群优化算法(Ant Colony Optimization,ACO)是一种模拟蚁群合作解决复杂问题的启发式算法,蚂蚁在运动的路径上释放信息素,而且通过探测别的蚂蚁留下的信息素来决定下一步行动,整个蚁群通过这种间接通信方式完成筑巢、觅食、迁移等复杂社会活动[2]。蚁群优化算法建立在多代理系统(Multi-Agent Systems,MAS)基础上,每个自治代理相当于一只蚂蚁,可实时观察、收集环境信息,并影响和修改环境信息,所有代理协同工作,体现出一种群体智能(collective intelligence),从而完成单个代理无法完成的复杂任务。

2 系统结构与实现

2.1 系统模型

P2P 计算网格的每个对等节点可看作一个蚁巢(Ant

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60274014);湖北省自然科学基金(the Natural Science Foundation of Hubei Province of China under Grant No.12003ABA043);中国地质大学优秀青年教师基金(No.CUGQNL0617)。
作者简介:吴湘宁(1972-),男,硕士,主要研究方向:智能计算、高性能网络等;胡成玉,博士;汪渊,硕士;王永骥,教授。

Nest),蚁巢负责本地计算资源的管理、蚂蚁的创建与调度、路由、作业调度及负载均衡。蚁巢的结构如图 1 所示,它由以下模块构成:

(1)计算资源管理器(Computing Resource Manager,CRM):负责本地计算资源的管理,运行分配给当前蚁巢的作业,对本地作业队列进行维护。

(2)蚂蚁调度模块(Ant Scheduler,AS):创建、克隆和销毁蚂蚁,为到访的蚂蚁提供访问当前蚁巢的接口,可根据蚂蚁携带的信息修改蚁巢中的信息素。

(3)路由管理器(Routing Manager,RM):确定数据包的路由方向,将其通过合适的链路向邻居节点转发。

(4)作业管理器(Job Manager,JM):是本地用户提交作业的入口,可根据负载均衡算法确定最适合完成作业的目标节点,并将作业传送到目标节点,作业完成后从目标节点接收结果并返回给本地用户。另外,还可接受其他节点传过来的作业请求并将其转交给 CRM,计算完成后从 CRM 取得结果并返回给源节点。

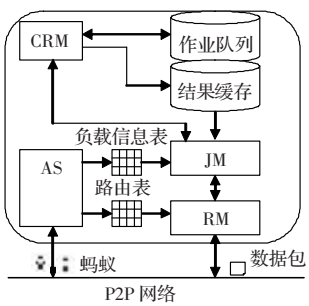

图 1 蚁巢的结构

2.2 路由机制

每个节点有一张路由表 $\boldsymbol{P}_{routing}$,它是一张信息素表,是一个 $(n-m)\times m$ 的矩阵,m 是当前节点邻居节点的数目,n 是所有节点的数目,$\boldsymbol{P}_{routing}$ 的元素 $\boldsymbol{P}_{routing}(i,j)$是信息素的强度,表示前往第 i 个节点时下一跳选择第 j 个邻居节点的概率。当前节点选择其所有邻居节点概率之和等于 1,初始化时选择各个相邻节点的概率相等。例如节点 F 有三个相邻节点 A、P、E,则 $P_{routing}(j,A)$、$P_{routing}(j,P)$和 $P_{routing}(j,E)$均被初始化为 1/3。

每个蚁巢定期派出探路蚂蚁,探路蚂蚁采用洪泛法扩散,源节点同时向其所有相邻节点发送相同的探路蚂蚁,收到探路蚂蚁的中间节点会将其克隆,并向所有相邻节点转发,但接收该蚂蚁的链路除外。源节点每次同时发出的探索蚂蚁及其所有的克隆蚂蚁均使用相同的标识号,互相称为克隆兄弟。对于具有同一标识号的蚂蚁,中间节点只对第一只到达的蚂蚁进行克隆和转发,而对后到达的该蚂蚁的克隆兄弟则就地销毁且不再克隆转发。洪泛法可保证探索蚂蚁能够到达 P2P 网络上所有节点。

每个节点收到探路蚂蚁后,会更新路由表。蚂蚁来时所通过的相邻节点所对应的信息素会增加,而其他的相邻节点的信息素会减少,增加的量和减少的总量相等,从而使通往其所有相邻节点概率之和仍然等于 1。增加和减少信息素分别使用公式(1)和式(2)[3]。公式可保证不管具有同一标识号的蚂蚁通过不同路径到达一个节点的先后次序如何,最佳路径上信息素的增长总是高于其他路径,而信息素的减少却低于其他路径。

$$P_{t+1}=\begin{cases}\dfrac{\Delta p}{1-\Delta p}P_t & \text{if } P_t\leqslant 1-\Delta p\\ \dfrac{1-\Delta p}{\Delta p}(P_t-1)+1 & \text{if } P_t>1-\Delta p\end{cases} \tag{1}$$

$$P_{t+1}=\begin{cases}\dfrac{1-\Delta p}{\Delta p}P_t & \text{if } P_t\leqslant \Delta p\\ \dfrac{\Delta p}{1-\Delta p}(P_t-1)+1 & \text{if } P_t>\Delta p\end{cases} \tag{2}$$

式中 P_t 和 P_{t+1} 分别表示更新前后的信息素浓度。Δp 是一个随路径跳数增加而减少、但随路径带宽增加而增大的参数,其值可反映一条路径的质量,即跳数越少,带宽越大,路径质量越好,则 Δp 越大。经类似归一化处理后 $0.5\leqslant\Delta p<1$,计算如公式(3)所示:

$$\Delta p\propto[\frac{1}{f(hop)}]^{\theta}[\frac{B_{bottleneck}}{M}]^{\lambda} \tag{3}$$

式中 hop 为路径跳数,$B_{bottleneck}$ 为路径瓶颈段带宽,M 为一个比值调节常数,θ、λ是跳数和带宽对 Δp 影响的权重因子。

路由机制可由图 2 所示简单网络解释,假设各个链路带宽相同,节点 B 采用洪泛法发出两只相同的探路蚂蚁,它们经不同路径到达节点 F 并修改路由表,左边蚂蚁经质量较好的路径 B–A–F 到达后,使 F 上路由表中信息素值 $P_{routing}(B,A)$有较大增长,同时使 $P_{routing}(B,E)$有较大减少,右边蚂蚁经质量较差的路径 B–C–D–E–F 到达后,使 F 上路由表中 $P_{routing}(B,E)$有少量增长,并使 $P_{routing}(B,A)$有少量减少,将两只蚂蚁的影响相加以后,$P_{routing}(B,A)$会增加,而 $P_{routing}(B,E)$会减少,因此 F 在为目的节点为 B 的数据包选择路径时,下一跳选择 A 的概率要比选择 E 的概率大。

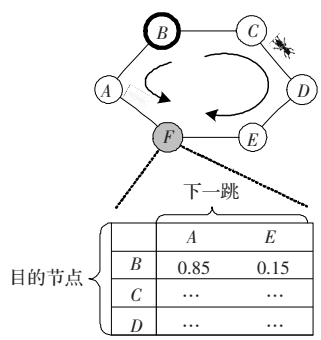

图 2 蚁群智能路由机制示例

为了防止发生停滞(stagnation)现象,即数据包总是被发往信息素浓度最强的链路而造成链路的阻塞,在路由选择策略上有意增加一定的噪音(noise),以增加探索新路径的可能性。这里设置一个常数 f_1,$0<f_1<1$,称为路由噪音因子,当前节点在选择下一跳时,有概率为 f_1 的可能是在相邻链路中随机选择,还有概率($1-f_1$)的可能是按照路由表中各个相邻链路的信息素浓度来选择,即选择概率最大的邻居节点作为下一跳[4]。

2.3 负载均衡机制

JM 做负载均衡计算时所使用的负载信息表 $\boldsymbol{\tau}_{load}$ 也是一张信息素表,其长度为 n,第 k 行元素记录第 k 个节点 N_K 的运算资源、当前负载情况,以及该节点的负载信息素浓度。负载信息素浓度表示该节点被选为运算新作业的目标节点的可能性。发自 N_K 的探路蚂蚁夹带了 N_K 最近的运算资源和负载情况信息,当该蚂蚁或其克隆兄弟中的第一只到达其他节点,会更新在这些节点上 $\boldsymbol{\tau}_{load}$ 中 N_K 对应的信息。

节点 N_K 的运算资源包括 CPU 数目和 MIPS(Million Instruction Per Second,百万条指令/秒)、存储容量 M_{Disk}、内存容量 M_{Memory}、通信能力 C。负载情况用下一作业预期启动时间 $T_{start\_next\_job}$ 来表示,即新到达作业可以开始执行的时间,等于 N_K 当前作业队列中现有作业全部处理时间的总和。为方便计算,这些参数可以按照一定权重合并为一个属性,称为节点 N_K 的固有计算能力 η_K,合并规则如公式(4)所示。式中 $MIPS(i)$ 表示第 i 个 CPU 的 $MIPS$ 值,ω_1 至 ω_5 分别为各个参数的权重系数:

$$\eta_K=\omega_1\sum_{cpu(i)}MIPS(i)+\omega_2M_{Disk}+\omega_3M_{Memory}+\omega_4C+\frac{\omega_5}{0.2+T_{start\_next\_job}} \quad (4)$$

节点 N_K 的信息素浓度体现了 N_K 处理作业的历史信息,其初始值等于 η_K,随后会随 N_K 所承担负载的情况而变化。

(1)当前节点收到本地用户提交的新作业请求后,根据任务对 MIPS、存储容量、内存容量、通信能力的要求,从负载信息表中挑选所有满足条件的节点,构成一个候选目标节点集合 $U_{allowed}$。

(2)按公式(5)分别计算 $U_{allowed}$ 中每个候选目标节点 u 的选择概率 P_u,并选择概率 P_u 最大的节点作为目标节点 N_D。式中 τ_u 是 u 的信息素浓度,η_u 是 u 的固有计算能力。α、β 是常量,分别代表历史信息素以及资源固有计算能力的重要性。

$$P_u=\frac{[\tau_u]^\alpha\cdot[\eta_u]^\beta}{\sum_{u\in U_{allowed}}^{\infty n}([\tau_u]^\alpha\cdot[\eta_u]^\beta)} \quad (5)$$

同路由策略一样,为了防止发生停滞现象,即作业总是被集中分配在信息素浓度最强的节点上而造成该节点超载,在作业分配策略上有意加上一定的噪音,以增加新分配方案的可能性。这里设置一个常数 f_2,$0<f_2<1$,称为作业分配噪音因子,当前节点在选择目标节点时,有概率为 f_2 的可能是从候选节点中随机选择一个目标节点,还有概率($1-f_2$)的可能是根据负载信息表中候选节点的运算资源信息和信息素浓度信息,计算选择概率 P_u 来决定目标节点。

(3)按公式(6)修改目标节点 N_D 所对应行上的信息素 τ_D。式中 Q 为常数,是比值调节因子,T_D 是刚分配给 N_D 的新作业的预计执行时间。

$$\tau_D=\tau_D+\Delta\tau_D \quad \Delta\tau_D=\frac{Q}{\tau_D} \quad (6)$$

(4)当前节点将新任务发送到目标节点 N_D 上,若 N_D 成功完成任务并返回结果,则将当前节点负载信息表 $\boldsymbol{\tau}_{load}$ 中 N_D 对应的信息素值提高到原来的 ρ_{reward} 倍,$\rho_{reward}=1.05$,称为奖励因子。若由于故障等原因作业超时仍未完成,则将 N_D 对应的信息素值减少为原来的 ρ_{punish} 倍,$\rho_{punish}=0.95$,称为惩罚因子。当前节点还要将此次奖励和惩罚的信息夹带在下次发出的探路蚂蚁中,以通知其他节点同步地修改 N_D 对应的信息素值。在网络上的数据包传递均采用前面介绍过的路由方法。

(5)若当前节点超时仍未收到某个节点 N_K 的探路蚂蚁,则将当前节点负载信息表 $\boldsymbol{\tau}_{load}$ 中 N_K 所对应的行屏蔽,使 N_K 暂时失去候选资格,直至重新收到 N_K 的探路蚂蚁后再行恢复。若过一段更长时间后仍未收到 N_K 的探路蚂蚁,则将 N_K 所对应的行从 $\boldsymbol{\tau}_{load}$ 中彻底删除。

(6)所有蚁巢每隔一定的时间间隔 $T_{interval}$,按公式(7)将 $\boldsymbol{\tau}_{load}$ 表中所有的信息素进行挥发。ρ 是常数,称为挥发系数,这里设为 0.005。若信息素值已小于 0.003,则不再挥发。挥发技术可防止某些优质节点的信息素值增长过高,从而给其他节点更多被选作目标节点的机会。

$$\tau_u=(1-\rho)\cdot\tau_u \quad (7)$$

3　仿真分析

结合本文给出的算法,利用网络仿真工具 OMNeT++[5]进行了仿真,生成了一个 30 个节点的 P2P 计算网格,节点间链路带宽 1 Mb/s≤B≤100 Mb/s,每个节点上有 4 个用户,1≤CPU≤2,$MIPS$=500,100 G≤M_{Disk}≤500 T,5 G≤M_{Memory}≤10 T,1 Mb/s≤C≤100 Mb/s。每个用户生成 0 到 10 个作业请求,每个作业的运算负载从 4×10^{12} 到 1×10^{14} 条指令不等,内存等要求在一定范围内随机产生。主要参数为 θ=0.4,λ=0.6,α=0.5,β=0.5,f_1=0.2,f_2=0.25。仿真结果如图 3 和图 4 所示。

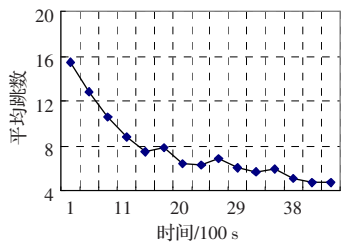

图 3　路由平均跳数随时间变化图

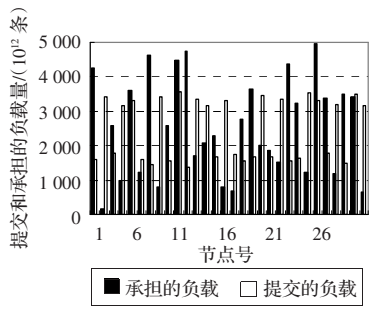

图 4　各节点提交和承担的负载量对比图

由图 3 可看出,经过一段时间以后,数据包路由所需经过的节点数明显减少,主要原因在于探路蚂蚁不断释放的信息素实现了优质路径在路由表中的正反馈,从而提高了路由效率。

在图 4 中,各个节点上提交的负载(以 10^{12} 条指令为单位)是随机变化的,150 个时钟周期后,负载基本上被合理地分配到各个节点上,其中双 CPU 节点承担的负载接近单 CPU 节点的两倍,说明优质计算资源得到了充分利用,网格总体的计算效率得到提高。

实验结果还表明,信息素浓度需要设置一定的上限以防溢出,若达到上限,可增加其挥发强度。另外,在网络拓扑结构和节点性能不是经常变动的情况下,可减少发出探路蚂蚁的频率。

4　小结

本文设计了一个基于蚁群智能的 P2P 计算网格路由及负载均衡算法,该算法模拟自然界中的蚂蚁利用信息素完成复杂社会工作的机制,通过启发方式有效解决动态的、分布的 P2P 网格环境下的路由以及计算资源的平均分配问题。仿真结果表明,算法在 P2P 计算网格上路由及负载分配上取得理想效果。不足之处在于路由方面只考虑跳数和带宽,未考虑链路的成

(下转 240 页)