

面向异构网络的动态负载均衡算法及其收敛性分析

张浩军 朱艳琴 纪其进*

(苏州大学计算机科学与技术学院 苏州 215006)

(江苏省计算机信息处理技术重点实验室 苏州 215006)

摘要: 资源共享是分布式计算的核心问题, 负载均衡是实现资源共享的基本手段。现有的负载均衡算法多局限于同构网络, 伴随计算终端多元化的发展趋势, 面向异构网络的负载均衡需求也不断增加。该文基于经典的扩散算法, 提出一种适用于异构网络环境的新的动态负载均衡算法, 并通过严格的数学方法证明了算法中各结点可收敛于平衡点。实验结果表明, 该算法的收敛性能优于 Rotaru 等人(2004)提出的 GDA 算法, 并且对于网状网络、星型网络和花环网络等经典结构都有良好的收敛性, 即便在随机生成的网络结构中, 仍然保持了较快的收敛速度。

关键词: 分布式计算; 异构网络; 动态负载均衡; 收敛性分析

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2013)09-2247-07

DOI: 10.3724/SP.J.1146.2012.01360

A Diffusion-based Dynamic Load Balancing Algorithm for Heterogeneous Networks and Its Convergence Analysis

Zhang Hao-jun Zhu Yan-qin Ji Qi-jin

(School of Computer Science & Technology, Soochow University, Suzhou 215006, China)

(Jiangsu Provincial Key Laboratory of Computer Information Processing, Suzhou 215006, China)

Abstract: Resource sharing is one of the key issues of distributed computing, and load balancing is the fundamental approach to sharing scarce resource in a distributed computing system. However, the existing load balancing methods are mostly confined to the homogeneous networks. With the diversity of the computing terminals, there is increasing requirement for the study of load balancing toward the heterogeneous networks. In this paper, a diffusion-based dynamic load balancing algorithm is proposed for the heterogeneous networks and it is proved with mathematical strictness that all nodes will converge to the expected balanced point. The numerical results show that the algorithm is better than the GDA algorithm proposed by Rotaru *et al.* (2004) and has ideal convergence property over many network topologies including Mesh, Star and Tours *etc.* And it converges rather fast even in a randomly generated network.

Key words: Distributed computing; Heterogeneous networks; Dynamic load balancing; Converge analysis

1 引言

随着计算机硬件成本的降低和网络的快速发展, 基于网络的集群和网络计算逐渐流行起来。目前, 网络上的主机差异很大, 不仅体现在处理能力上, 同时可能在操作系统和接入网络环境上存在差异。在这样的异构系统上负载均衡是提高系统资源利用率和并行计算性能的一个关键技术^[1]。总体而言, 负载均衡法可分为静态和动态两类。如果系统负载可以在运行之前确定并事先进行划分, 则属于静态负载均衡问题; 若负载只能在运行时测量并动

态确定负载划分, 则属于动态负载均衡(Dynamic Load Balancing, DLB)问题。由于分布式系统中各结点之间的通信采用报文传递的方式进行, 通信产生的不可预测延迟使得难以事先进行计算负载的估算, 因而静态负载均衡法逐渐不被使用。动态负载均衡的决策取决于系统当前的状态, 对各个结点的负载动态地调整, 更加适用于当前的大多数系统。本文研究异构网络环境中的动态负载均衡问题。

为了避免中心化方法的通信负载高并且难以扩展等问题, 近邻法^[2]被广泛应用于同构网络下的动态负载均衡问题。近邻法主要包括扩散法^[3]、维交换法^[3]和梯度法^[4]。在该策略中, 每个处理器和相邻的处理器交换负载, 实现和相邻处理器间的负载均衡, 并经过多次迭代达到全局负载均衡。相对于同构网络, 关于异构网络中 DLB 问题的研究较少。Hui 等人^[5]在结点计算能力不同但通信参数相同的条件下,

2012-10-23 收到, 2013-07-02 改回

国家自然科学基金(61070170), 东南大学计算机网络和信息集成教育部重点实验室开放研究基金和苏州市应用基础研究计划项目(SYG201238)资助课题

*通信作者: 纪其进 ji@suda.edu.cn

提出了水动力学负载均衡法,用连通器中水的流动和水的位能在均衡时最低的原理来计算负载的移动。Diekmann 等人^[6]提出了具有相同计算能力和不同通信参数的异构系统中的扩散法。Rotaru 等人^[7]和 Elsasser 等人^[8]将这些方法拓展到具有不同相对速度的处理器和不同通信参数的异构系统上,给出了扩散算法的一般化问题。并且, Rotaru 等人从理论上证明了所给的一般扩散算法的收敛速度快于水动力学。Franceschelli 等人^[9]提出一个基于 Gossip 的算法,分别解决同构^[9]和异构^[10]情况下的负载均衡问题。Zhao 等人^[11]改进了 Elsasser 提出的扩散策略,并且提高了数值结果的稳定性。Sun 等人^[12]提出使用模糊逻辑方法解决在异构无线网络中的负载均衡问题。在无线自组织网络环境中,为了应对结点的自私性,一些经济学方法,如竞争均衡模型^[13]、合作博弈模型^[14]等被用来分析和设计异构网络下的负载均衡算法。

本文基于经典的扩散算法^[3],提出一种适用于异构网络环境的新的动态负载均衡算法。相比现存的用于异构网络环境中的 DLB 算法,本文提出的算法具有更一般的形式,适用于处理器能力不一致且通信参数也不一致的异构网络。此外,本文提出的分布式算法更加简单,不需要计算通信网络上的拉普拉斯算子的特征值等参数,并且通过严格的数学推导,证明了算法的收敛性。实验结果表明,该算法优于 Rotaru 等人^[7]提出的 GDA 算法并且对于网状网络、星型网络和花环网络等经典结构都有良好的收敛性,即便在随机生成的网络结构中,仍然保持了较快的收敛速度。

2 异构网络上的动态负载均衡算法

2.1 异构网络模型

我们用无向连通图 $G=(N, E)$ 来描述一个包含 n 个结点的异构网络系统。其中 N 是 n 个异步异构处理器(后文称其为结点)的集合, E 为网络拓扑结构,用邻接矩阵表示。假设负载可无限分割且用正实数表示,负载能力统一量化为一个正实数。利用网络协议,可以限定处理器之间的消息时延在一定的范围内。模型中每个结点都是算法执行的主体,即它们是异步执行算法的。另外假设算法从开始执行到首次判定系统达到全局平衡过程中,没有任何负载被加入或者被消耗。这里的假设只是为了观察算法的收敛速度。在实际应用中,负载可以被加入也可以被消耗。在后面的实验中,我们发现只要负载量足够大,那么在考虑负载消耗的情况下,系统仍能达到负载均衡。

2.2 动态负载均衡算法

扩散模型是一种经典的 DLB 算法。在该算法的每次迭代中,每个处理器将部分负载传递给具有较低负载的相邻处理器,同时从其它具有较高负载的相邻处理器接收负载,通过交换适当数量的负载来达到更加均匀的状态。本文提出的异构网络下的 DLB 算法即基于扩散模型。以下用到的主要符号定义见表 1。

表 1 符号定义表

符号	说明	符号	说明
t	表示时刻,是一个非负整数	$h_i(t)$	$h_i(t)=l_i(t)/c_i$ 是 t 时刻处理器 i 的加权负载值
L	网络总负载量,是正实数	$h_j^i(t)$	t 时刻处理器 i 对 j 的加权负载值的估计
$A(i)$	处理器 i 的邻居的集合	$s_{ij}(t)$	t 时刻处理器 i 向处理器 j 发送的负载量
c_i	处理器 i 的负载能力,我们将其量化为一个正实数	$v_{ij}(t)$	t 时刻之前,已从处理器 i 送出但还未被处理器 j 收到的负载量
$l_i(t)$	t 时刻处理器 i 上的负载量	$r_{ij}(t)$	t 时刻处理器 j 从 i 收到的负载量

假设系统从某时刻($t=0$ 时)开始运行,对于每一个处理器 i ,都有一个与之对应的时刻的序列 T^i ,当 $t \in T^i$ 时,结点开始执行算法。为了使算法可以在有限时间内收敛,还需附加以下两个假设。假设 1 是对异步条件和时间的限定,而假设 2 是对转移量的限制。

假设 1 存在一个正整数 B 满足:

(1)对每个 i 和 $t \geq 0$,集合 $\{t, t+1, \dots, t+B-1\}$ 中至少有一个元素属于 T^i ;

(2)对所有的 i 和 t ,以及 $j \in A(i)$ 总有 $t-B < \Gamma_j^i(t) \leq t$;

(3)在某时刻 $t \in T^i$ 从处理器 i 送往处理器 j 的负载 $s_{ij}(t)$ 将在 $t+B$ 时刻前被处理器 j 收到。

假设 2

(1)对每个 i 和 $t \in T^i$ 总有 $a \in (0,1)$,使得存在 $j \in A(i)$ 满足 $h_j^i(t) = \min_{k \in A(i)} h_k^i(t)$ 和 $s_{ij}(t) \geq a \frac{c_i c_j}{c_i + c_j} \cdot (h_i(t) - h_j^i(t))$ 。并且如果 $h_i(t) \leq h_j^i(t)$,那么 $s_{ij}(t)=0$;

(2)对于任意的 $i, t \in T^i$ 和满足 $h_i(t) > h_j^i(t)$ 的 $j \in A(i)$ 有: $h_i(t) - \frac{\sum_{k \in A(i)} s_{ik}(t)}{c_i} \geq h_j^i(t) + \frac{s_{ij}(t)}{c_j}$ 。

算法分为两个部分:交换信息和负载转移。表 2 给出了结点执行算法 GNDA(General Neighbor

表2 面向异构网络的动态负载均衡算法 GNDA 流程

```

(1)  $t=0$ ; 获取邻居结点的负载能力与负载信息;
(2) 结点负载均衡的操作时间  $T=1$ ;
(3) while( $\|h(t) - h(t-1)\|_2 < 0.1$ ) do
(4)   if(需要更新邻居信息)更新邻居负载信息;
(5)   if( $T==t$ ) do
(6)     minNode=INF;
(7)     forin(node in 邻居结点集) do
(8)       IF(minNode>node) do
(9)         minNode=node; end if
(10)    end for
(11)    该结点向 minNode 结点  $j$  转移负载量  $\frac{c_i c_j (h_i(t) - h_j^i(t))}{c_i + c_j}$ 
(12)     $l_i(t+1) = l_i(t) - \sum_{j \in A(i)} s_{ij}(t) + \sum_{j \in A(i)} r_{ji}(t)$ ;
(13)  end if
(14)   $t++$ ;  $T=T+\text{randint}(1,1,B)$ ;
(15) end while

```

Diffusion Algorithm)的流程。每个结点在系统启动后会一直执行算法，直到所有任务完成。算法收敛的条件为 $\|h(t) - h(t-1)\|_2 < 0.1$ ，其中 $h(t)$ 表示在时刻 t 执行该算法的结点与其邻居结点的加权负载向量。由于收敛条件的判断局限于邻居结点，可能会造成局部的收敛，所以算法在到达收敛之后每隔一段时间会检测是否仍然满足收敛条件。虽然执行该检测操作需要一定的资源，但是这种消耗是值得的。GNDA 算法每一步迭代完成的任务主要是与邻居结点交换负载信息及转移负载，其时间复杂度仅与网络规模有关，可用 $o(n)$ 表示。算法迭代次数的影响因素主要有系统的初始状态、网络规模、通信参数和收敛条件等。

3 算法的收敛性证明

上一节给出了具体的异构网络下的负载均衡算法，该算法是在每个结点上异步执行，目的是要达到系统全局的负载均衡，即算法执行一段时间后每个结点都被分配了与自身负载能力相适应的负载。上述算法的总体思想是由高负载结点向低负载结点迁移任务，但是由于算法的异步性和网络时延带来的影响，算法的收敛性证明变得困难，本节将利用单调有界性证明算法的收敛性。

根据上一节的静态假设条件，即负载总量在算法执行过程中不变，有式(1)和式(2)成立：

$$h_i(t+1) = h_i(t) - \frac{\sum_{j \in A(i)} s_{ij}(t)}{c_i} + \frac{\sum_{j \in A(i)} r_{ji}(t)}{c_i} \quad (1)$$

$$v_{ij}(t) = \sum_{\Gamma=0}^{t-1} (s_{ij}(\Gamma) - r_{ij}(\Gamma)) \quad (2)$$

这里隐含假设了 $v_{ij}(0) = 0$ 。由于假设零时刻没有任何负载的转移，有 $\sum_{i=1}^n l_i(0) = L$ ，再根据式(1)和式(2)很容易得到负载守恒等式：

$$\sum_{i=1}^n \left(l_i(t) + \sum_{j \in A(i)} v_{ij}(t) \right) = L, \quad \forall t \geq 0 \quad (3)$$

另外假设有式(4)：

$$m(t) = \min_i \min_{t-B < \tau \leq t} h_i(\tau) \quad (4)$$

可以发现对每个 i, j 和 t ，有 $h_j^i(t) = h_j(\Gamma_j^i(t)) \geq m(t)$ 。

为了证明异构网络中结点上的负载最终收敛于平衡状态，首先引入3个引理，在此基础上再给出算法的收敛性证明。

引理1 对每个 $i, t, \Delta t > 0$ ，存在 $\beta \in (0,1)$ ，满足

$$h_i(t + \Delta t) \geq m(t) + \beta^{\Delta t} (h_i(t) - m(t)) \quad (5)$$

特别地， β 可以取 c_{\min} / nc_{\max} ，并证明序列 $m(t)$ 是非递减且收敛。其中 c_{\min} 表示所有处理器中负载能力的最小值， c_{\max} 表示所有处理器中负载能力的最大值。

证明 首先证明当 $\Delta t = 1$ 时，结论成立。如果 $t \notin T^i$ ，那么 $h_i(t+1) = h_i(t) = m(t) + (h_i(t) - m(t))$ 。由于 $h_i(t)$ 大于等于 $m(t)$ ，此时式(5)是针对任意 $\beta \in (0,1)$ 成立的。若 $t \in T^i$ 考虑集合 $A' = \{j \mid j \in A(i), h_i(t) \geq h_j^i(t)\}$ ，不妨假设其基数为 K 。式(1)和假设2(2)表明对所有的 $j \in A(i)$ 有

$$h_i(t+1) \geq h_i(t) - \frac{\sum_{j \in A(i)} s_{ij}(t)}{c_i} \geq m(t) + \frac{s_{ij}(t)}{c_{\max}}$$

累加所有 $j \in A'$ 得到

$$Kh_i(t+1) \geq Km(t) + \frac{1}{c_{\max}} \sum_{j \in A'} s_{ij}(t) \quad (6)$$

同时，从等式(1)和事实条件当 $j \notin A'$ 时 $s_{ij}(t) = 0$ 我们得到

$$\sum_{j \in A'} s_{ij}(t) = \sum_{j \in A(i)} s_{ij}(t) \geq c_i (h_i(t) - h_i(t+1)) \quad (7)$$

综合式(6)和式(7)并整理后得到： $h_i(t+1) \geq m(t) + \frac{c_i}{c_{\max}K + c_i} (h_i(t) - m(t)) \geq m(t) + \frac{c_{\min}}{nc_{\max}} (h_i(t) - m(t))$ 。所以 $\Delta t = 1$ 时，式(5)成立，那么 $h_i(t+1) \geq m(t)$ ，进而有 $m(t+1) \geq m(t)$ 。又有 $h_i(t)$ 被 L/c_{\min} 所限制，所以 $m(t)$ 也被 L/c_{\min} 限制，且因 $m(t)$ 单调递增，所以 $m(t)$ 必收敛。通过归纳法，假设式(5)对某个 Δt 成立，由于 $m(t)$ 的单调递增，所以有

$$h_i(t + \Delta t + 1) \geq m(t) + \beta^{\Delta t + 1}(h_i(t) - m(t))$$

其中最后一步是根据归纳假设。证毕

定义1 令处理机 i 处在某时刻 t_0 , 对任意 $j \in A(i)$ 和任意时刻 $t \geq t_0$, 称事件 $E_j(t)$ 发生, 如果以下两个条件成立:

(1)

$$h_j^i(t) < m(t_0) + \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \beta^{t-t_0} (h_i(t_0) - m(t_0)) \quad (8)$$

(2) 处理机 j 如假设2(1)是 i 的一个被转移负载的轻负载邻居, 特别地

$$s_{ij}(t) \geq \alpha \frac{c_i c_j}{c_i + c_j} (h_i(t) - h_j^i(t)) \quad (9)$$

引理 2 若 $j \in A(i)$, $t_1 \geq t_0$, $t_1 \in T^i$ 且事件 $E_j(t_1)$ 发生, 则对任意 $\Gamma \geq t_1 + 2B$ 且 $\Gamma \in T^i$, 事件 $E_j(\Gamma)$ 不发生。

证明 因为 $t_1 \geq t_0$, $t_1 \in T^i$, $j \in A(i)$, 并且事件 $E_j(t_1)$ 发生, 式(8)和式(9)在 $t=t_1$ 时成立, 由引理1得

$$h_i(t_1) \geq m(t_0) + \beta^{t_1-t_0} (h_i(t_0) - m(t_0)) \quad (10)$$

用式(10)减去式(8)(当 $t=t_1$), 并由条件 $\alpha < 1$ 得到

$$h_i(t_1) - h_j^i(t_1) \geq \frac{1}{2} \beta^{t_1-t_0} (h_i(t_0) - m(t_0))。$$

然后, 由不等式(9)(当 $t=t_1$)得

$$s_{ij}(t_1) \geq \frac{\alpha}{2} \frac{c_i c_j}{c_i + c_j} \beta^{t_1-t_0} (h_i(t_0) - m(t_0)) \quad (11)$$

处理机 j 会在某时刻 t_2 接收 $s_{ij}(t_1)$ 的负载, 其中 $t_1 \leq t_2 < t_1 + B$ (假设1(3))。如果 $t_2 \notin T^j$, 那么对所有 $k \in A(j)$ 有 $s_{jk}(t_2) = 0$, 由式(1)可得

$$\begin{aligned} h_j(t_2 + 1) &= h_j(t_2) + \frac{\sum_{k \in A(j)} r_{kj}(t_2)}{c_j} \geq h_j(t_2) \\ &+ \frac{r_{ij}(t_2)}{c_j} \geq h_j(t_2) + \frac{s_{ij}(t_1)}{c_j} \\ &\geq m(t_2) + \frac{s_{ij}(t_1)}{c_j} \end{aligned} \quad (12)$$

另一方面, 若 $t_2 \in T^j$, 则存在 $k^* \in A(j)$ 满足 $s_{j k^*}(t_2) > 0$, 由式(1)和假设2(2)得

$$\begin{aligned} h_j(t_2 + 1) &= h_j(t_2) - \frac{\sum_{k \in A(j)} s_{jk}(t_2)}{c_j} + \frac{\sum_{k \in A(j)} r_{kj}(t_2)}{c_j} \\ &\geq h_{k^*}^j(t_2) + \frac{r_{ij}(t_2)}{c_j} \geq m(t_2) + \frac{s_{ij}(t_1)}{c_j} \end{aligned} \quad (13)$$

综合两种情况, 再用式(11), 得

$$\begin{aligned} h_j(t_2 + 1) &\geq m(t_2) + \frac{s_{ij}(t_1)}{c_j} \\ &\geq m(t_0) + \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \beta^{t_1-t_0} (h_i(t_0) - m(t_0)) \\ &\geq m(t_0) + \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \beta^{t_2+1-t_0} (h_i(t_0) - m(t_0)) \end{aligned} \quad (14)$$

用引理1总结得到

$$\begin{aligned} h_j(t) &\geq m(t_0) + \beta^{t-t_0} \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \\ &\cdot (h_i(t_0) - m(t_0)), \quad \forall t \geq t_2 + 1 \end{aligned} \quad (15)$$

由于 $t_2 < t_1 + B$, 可以看出式(15)对所有 $t > t_1 + B$ 都成立。

取 t_3 满足 $t_3 \in T^i$ 且 $t_3 \geq t_1 + 2B$, 那么 $\Gamma_j^i(t_3) > t_3 - B \geq t_1 + B$, 代入式(15)得到

$$\begin{aligned} h_j^i(t_3) - m(t_0) &= h_j(\Gamma_j^i(t_3)) - m(t_0) \geq \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \\ &\cdot \beta^{t_3-t_0} (h_i(t_0) - m(t_0)) \end{aligned} \quad (16)$$

所以式(8)在 $t=t_3$ 时不成立, 故事件 $E_j(t_3)$ 不发生。

证毕

引理 3 对任意 i, t_0 , 可从 i 通过 l 条弧到达的 j 以及任意 $t \geq t_0 + 3nlB$, 有

$$h_j(t) \geq m(t_0) + (\eta \beta^{t-t_0})^l (h_i(t_0) - m(t_0)) \quad (17)$$

证明 首先证明 $l = 1$ 时结论成立。选取 i 和 t_0 , 令 t_1, \dots, t_n 为 T^i 的元素, 且 $t_{k-1} + 2B < t_k \leq t_{k-1} + 3B$, $k = 1, \dots, n$ 。根据引理2, 如果 $j \in A(i)$ 且 $k \neq 1$, 那么事件 $E_j(t_k)$ 和事件 $E_j(t_1)$ 不可能同时发生。也就是说, 对某个 $t_k (1 \leq k \leq n)$, 事件 $E_j(t_k)$ 在任意 $j \in A(i)$ 时都不会发生。根据假设2(1), 对某个 j^* 满足 $h_{j^*}^i(t_k) \leq h_j^i(t_k), \forall j \in A(i)$, 式(10)必然成立。又因事件 $E_{j^*}(t_k)$ 不会发生, 式(8)在 $j=j^*$ 时必然不成立, 于是有

$$\begin{aligned} h_j(\Gamma_j^i(t_k)) &= h_j^i(t_k) \geq h_{j^*}^i(t_k) \geq m(t_0) + \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \\ &\cdot \beta^{t_k-t_0} (h_i(t_0) - m(t_0)), \quad \forall j \in A(i) \end{aligned} \quad (18)$$

对任意 t 满足 $t \geq t_0 + 3nB$, 由 $t \geq t_k \geq \Gamma_j^i(t_k)$ 和引理1得

$$h_j(t) \geq m(t_0) + \frac{\alpha}{2} \frac{c_i}{c_i + c_j} \beta^B \beta^{t-t_0} (h_i(t_0) - m(t_0)) \quad (19)$$

式(19)是由 $t_k - \Gamma_j^i(t_k) \leq B$ 得到。假设对所有距离 i 有 l 的 j 结论成立, 设某个 k 距 i 有 $l+1$, 那么总存在一个距 i 为 l 的 j 且 $k \in A(j)$ 。对处理机 j 应用归纳假设得到

$$h_j(t_0 + 3nlB) \geq m(t_0) + (\eta \beta^{3nlB})^l (h_i(t_0) - m(t_0)) \quad (20)$$

然后用 $t_0 + 3nlB$ 替换 t_0 得到对任意的 $t \geq t_0 + 3nlB + 3nB$ 有

$$h_k(t) \geq m(t_0) + (\eta\beta^{t-t_0})^{l+1}(h_i(t_0) - m(t_0)) \quad (21)$$

至此完成了归纳并证明了引理。 证毕

在上述引理的基础上，下面给出本文的主要结果。

定理 1 根据假设 1 和假设 2，对任意的处理机

i ，有 $\lim_{t \rightarrow \infty} l_i(t) = c_i \left(L / \sum_{k=1}^n c_k \right)$ 成立。

证明 取适当的处理机 i 和时刻 t_0 ，由于每一个处理机到 i 的距离都小于 n ，由引理 3 得

$$h_j(t) \geq m(t_0) + (\eta\beta^{3n^2B+B})^n (h_i(t_0) - m(t_0)), \quad \forall j, \forall t \in [t_0 + 3n^2B, t_0 + 3n^2B + B] \quad (22)$$

因此，由式(4)知存在某个 $\delta > 0$ 使 $m(t_0 + 3n^2B + B) \geq m(t_0) + \delta(h_i(t_0) - m(t_0))$ 。上述不等式对每个 i 都成立，所以

$$m(t_0 + 3n^2B + B) \geq m(t_0) + \delta(\max_i h_i(t_0) - m(t_0)) \quad (23)$$

如果 $\max_i h_i(t) - m(t)$ 不收敛到 0，那么式(23)表明 $m(t)$ 会增大到无穷，这与 $m(t)$ 的有界性矛盾。所以 $\max_i h_i(t) - m(t)$ 一定收敛到 0，而且 $m(t)$ 也会收敛到某个常数 C (引理1)，所以有 $\lim_{t \rightarrow \infty} \max_i h_i(t) = C$ 。由于对所有的 j, t 都有 $m(t) \leq h_j(t) \leq \max_i h_i(t)$ ，由夹逼定理得，对所有 $j, h_j(t)$ 都收敛到 C 。反观假设 2(2) 会得到对所有 $(i, j) \in A$ ，有 $\lim_{t \rightarrow \infty} s_{ij}(t) = 0$ 。假设1(3)表明时刻 t 之前正在转移的负载 $v_{ij}(t)$ 包括了所有在区间 $(t - B, t - 1]$ 内被转移的负载。相应地有

$$v_{ij}(t) \leq \sum_{\Gamma=t-B+1}^{t-1} s_{ij}(\Gamma), \quad \forall i, \forall j \in A(i) \quad (24)$$

由式(24)可得对所有 $(i, j) \in A$ ，有 $\lim_{t \rightarrow \infty} v_{ij}(t) = 0$ 。利用式(3)可以得到

$$C \sum_{i=1}^n c_i = \lim_{t \rightarrow \infty} \sum_{i=1}^n c_i h_i(t) = \lim_{t \rightarrow \infty} \sum_{i=1}^n l_i(t) = L。由此得 $C = \left(L / \sum_{i=1}^n c_i \right)$ 。 证毕$$

4 实验结果

第3节通过严格的数学方法证明了算法的收敛性，本节通过一些经典和随机的网络结构验证算法的收敛性，分析其收敛速度，并与 Rotaru 等人^[7]提出的 GDA 算法做比较。实验平台采用 MATLAB，网络结构通过其邻接矩阵表示，在算法执行之前，给每个处理器随机分配任务量并指定其负载能力。如前所述，负载能力和任务量都是用一个正实数表示。

实验采用以下6种网络结构：树型(Tree)、网状网络(Mesh)、环状网络(Ring)、星型络(Star)、花环

网络(Torus，可以看作是 Mesh 的变形)和随机网络(Random)。对于随机网络，处理器之间的连接关系是随机的，但必须保证整个网络结构是一个连通图。

(1)算法在随机网络下的收敛情况 首先测试算法在随机网络情况下的收敛情况。这里假设处理器数量 $n=20$ ，处理器能力的上限 $c_{\max}=10$ ，总任务量为 $600n$ 。另外为了体现通信参数的异构性，对每条边 e_i 赋予随机权值 B_i ，代表该链路上的最大传输时延，实际传输时延在 $(0, B_i]$ 之间扰动，并且有 $\text{Max}\{B_i\} = 3$ ，以保证总的传输时延在可控范围内。每个处理器被指定一个小于 c_{\max} 的负载能力数值并随机分配到总任务量中的一部分。图1给出在算法没有执行之前的各处理器状态，很明显在算法没有执行之前，整个系统的负载是不平衡的。图2是在算法执行过程中3个处理器随着时间加权负载的变化情况。处理器的加权负载在前期的波动是比较大的，但是波动随着时间变化越来越小，最终所有处理器都能快速收敛于理论中的极限。

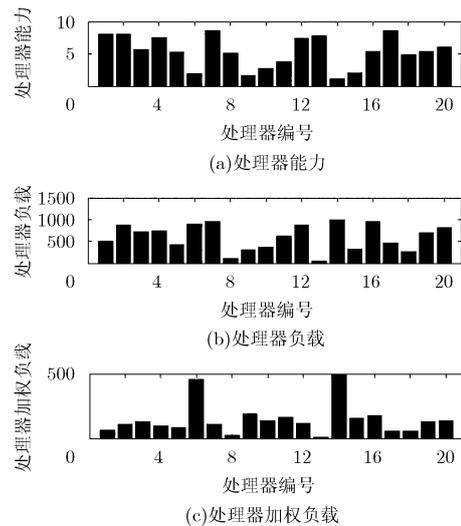


图1 随机生成的初始状态

接下来针对5种常见的网络拓扑结构和随机网络结构，考察算法与网络规模、任务量之间的关系，并基于前4种网络拓扑与 Rotaru 等人^[7]提出的 GDA 算法做比较。GDA 算法的参数设置为 $\epsilon_0 = 2 \sin^2(\pi/2n)$ ，扩散矩阵为 $M_1(\epsilon_0)$ 。

(2)收敛速度与网络规模的关系 每组测试用例的负载能力为小于其规定上限的随机生成值，网络参数设置同上，总任务量大小为 $600n$ 。假设处理器能力的上限 $c_{\max}=10$ ，总任务量随机不均匀地分配到各个处理器。记录每组收敛时间(正整数)，测试5次并去除最大和最小值取剩余3个的平均值。对于网状网络和花环网络，结点数分别取4, 9, 16, 25, 36，

其它的网络结构结点个数分别取3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36。6种网络结构随着处理器个数增加,其收敛时间的变化情况见图3。实验结果表明,星型、网状型和花环型网络的收敛时间随着处理器

个数的增加变化不大,保持着良好的收敛速度。在与 GDA 算法的比较中,从图4可以明显地看到 GNDA 算法优于 GDA 算法。

(3)收敛速度与任务量的关系 接下来验证算

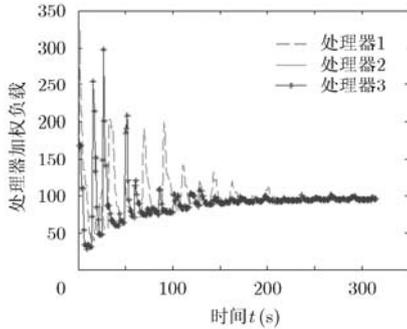


图2 3个处理器随着时间的收敛情况

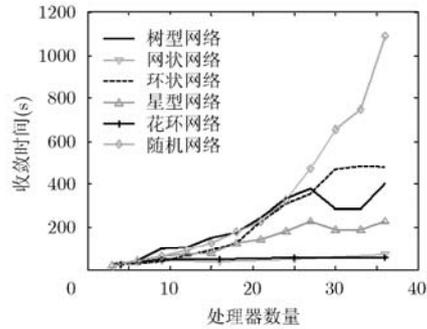
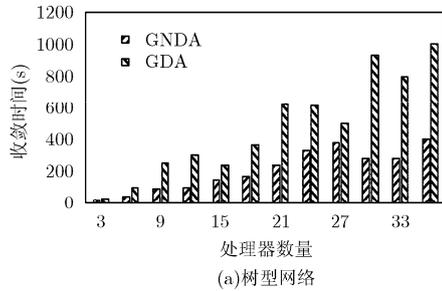
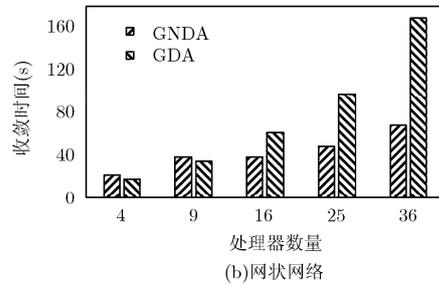


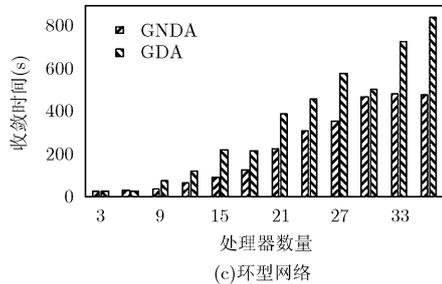
图3 网络规模与收敛时间的关系



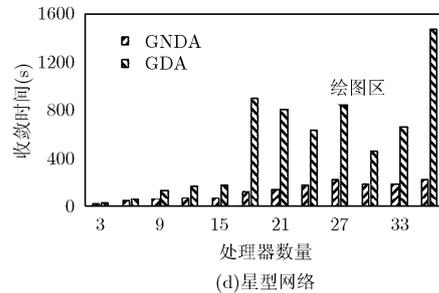
(a)树型网络



(b)网状网络



(c)环型网络



(d)星型网络

图4 GNDA 与 GDA 的收敛速度随网络规模变化情况

法的收敛时间与总任务量之间的关系。规定处理器个数为16个且其能力上限 $c_{max}=10$,网络参数设置同上。总任务量在 $600n \sim 5000n$ 之间变化,每个处理器会随机分配到总任务量中的一部分。图5表明 GNDA 算法对于随机网络和花环网络等的变化都不大,随着总负载量的增加,它们都有很好的收敛性。对于其它类型的网络拓扑,虽有一定的波动,从整体来看它们仍然具有较为平稳的性质。另外,在与 GDA 算法的比较中,从图6可以明显看到 GNDA 算法从整体上优于 GDA 算法。

5 结束语

分布式系统可以提供强大的计算能力,负载均衡是实现这种能力的关键问题。本文主要研究了一

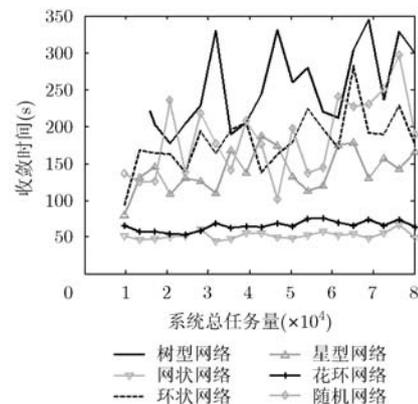


图5 任务量与收敛时间的关系

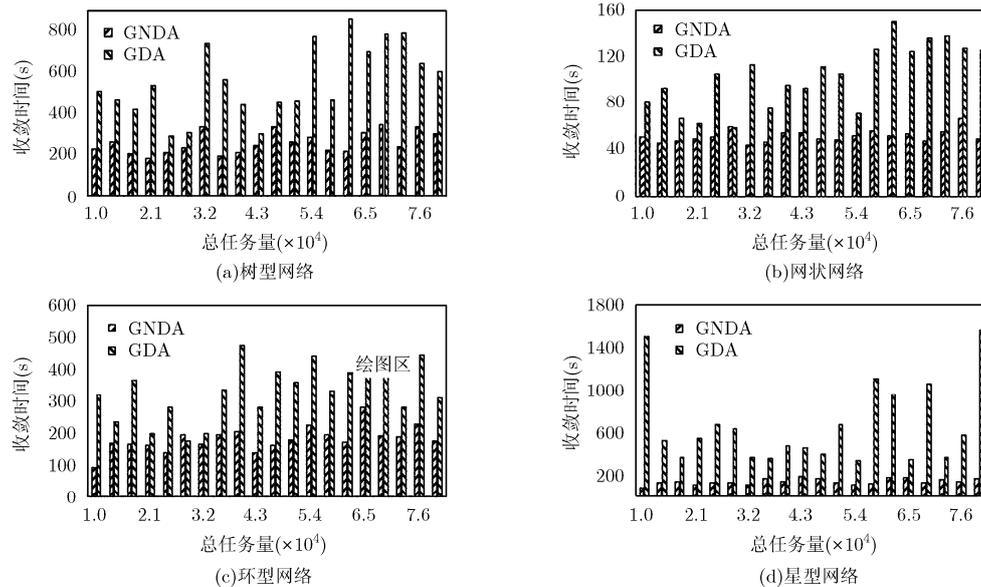


图6 GNDA 与 GDA 的收敛速度随总任务量的变化情况

种基于经典扩散方法的动态负载均衡算法，该算法可以适用于异构网络。论文通过严格的数学推导证明了算法收敛性，并通过实验在不同拓扑结构上对算法收敛性能进行了评估。另外实验结果也表明本文提出的算法收敛性能优于 Rotaru 等人提出的 GDA 算法。本文提出的面向异构网络的动态负载均衡算法与传统负载均衡算法相比，假设模型更加接近真实的网络环境，算法实现简单，在各种网络结构中具有较好的收敛性。下一步将会考虑负载任务的粒度问题，放松任务可无限细分的限制条件，从而使得模型更加接近真实的情况。

致谢 感谢赵煜同学在本文稿件完成过程中给予的帮助。

参考文献

- [1] Zheng G. Achieving high performance on extremely large parallel machines: performance prediction and load balancing [D]. [Ph.D. dissertation], University of Illinois, Urbana-Champaign, 2005.
- [2] Xu Cheng-zhong, and Francis C. Iterative dynamic load balancing in multicomputers[J]. *Journal of the Operational Research Society*, 1994, 45(7): 786-796.
- [3] Cybenko G. Dynamic load balancing for distributed memory multiprocessors[J]. *Journal of Parallel and Distributed Computing*, 1989, 7(2): 279-301.
- [4] Lin F, and Keller R. The gradient model load balancing method[J]. *IEEE Transactions on Software Engineering*, 1987, 13(1): 32-38.
- [5] Hui C, and Chanson S. Hydrodynamic load balancing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 1999, 10(11): 1118-1137.
- [6] Diekmann R, Frommer A, and Monien B. Efficient schemes for nearest neighbor load balancing[J]. *Parallel Computing*, 1999, 25 (7): 788-812.
- [7] Rotaru T and Nägeli H. Dynamic load balancing by diffusion in heterogeneous systems[J]. *Journal of Parallel and Distributed Computing*, 2004, 64(4): 481-497.
- [8] Elsässer R, Monien B, and Preis R. Diffusion schemes for load balancing on heterogeneous networks[J]. *Theory of Computing Systems*, 2002, 35(3): 305-320.
- [9] Franceschelli M, Giua A, and Seatzu C. Load balancing on networks with gossip based distributed algorithms[C]. Proceedings of 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 2007: 500-505.
- [10] Franceschelli M, Giua A, and Seatzu C. Load balancing over heterogeneous networks with gossip-based algorithms[C]. American Control Conference, St. Louis, MO, 2009: 1987-1993.
- [11] Zhao Cheng-gui and Xiao Wen-jun. Improved diffusion strategies for load balancing on general heterogeneous networks[J]. *Advanced Science Letters*, 2012, 5(2): 901-905.
- [12] Sun Jin, Wu Xuan-li, and Sha Xue-jun. Load balancing algorithm with multi-service in heterogeneous wireless networks[C]. Proceedings of 6th International ICST Conference on Communications and Networking in China, Harbin, 2011: 703-707.
- [13] Chatrapati K, Rekha J, and Babu A. Competitive equilibrium approach for load balancing a data grid[J]. *International Journal of Computer Science Issues*, 2011, 8(2): 427-437.
- [14] Penmatsa S and Chronopoulos A. Game-theoretic static load balancing for distributed systems[J]. *Journal of Parallel and Distributed Computing*, 2011, 71(4): 537-535.

张浩军：男，1989年生，研究生，研究方向为P2P网络和移动社交网络。

朱艳琴：女，1964年生，教授，博士，研究方向为计算机网络、信息安全技术、应用密码学等。

纪其进：男，1974年生，副教授，博士，研究方向为移动传感网络、社会网络与社会计算、认知网络及多agent系统。