异构系统动态负载平衡的扩散算法

金之雁[1)] 王鼎兴[2)]

[1)](中国气象科学研究院 北京 100081)

[2)](清华大学计算机科学与技术系 北京 100084)

摘 要 动态负载平衡是大规模并行计算中的一个十分重要的研究领域.它的主要方法是将计算负载通过并行计算机节点间的互连网络从负载高的节点移至负载低的节点.以前的学者针对同构系统提出了扩散算法等,对于异构系统研究得很少.该文研究了在异构系统中的扩散算法,在理论上证明了该方法的守恒性与收敛性,提出了一种构造异构系统的扩散矩阵的方法,并在不同规模的二维格栅网结构上进行试验,初步试验表明,该方法能够有效地对异构系统进行负载平衡,对于规模较小的系统收敛速度较快,而对于较大的系统,收敛速度慢一些.

关键词 异构分布并行系统;动态负载平衡;扩散算法

中图法分类号 TP301

Diffusion Algorithm of Dynamic Load Balancing for Heterogeneous System

JIN Zhi-Yan[1)] WANG Ding-Xing[2)]

[1)](*Chinese Academy of Meterological Sciences*, *Beijing* 100081)

[2)](*Department of Computer Science and Technology*, *Tsinghua University*, *Beijing* 100084)

Abstract One of the key issues in distributed parallel systems is the dynamic load balancing. The idea is to migrate the load from the busy nodes to idle nodes along the links between them. The algorithm to redistribute the workload over various network have been studied intensively in recent years, but most of the work assumes the system is homogeneous. This paper presents a diffusion algorithm for heterogeneous distributed parallel system. The conservation and convergence of the algorithm is proved. A method to construct the diffusion matrix is also given. A test on 2-D mesh network is carried out and primary result shows that is can balance the load on the heterogeneous system. The convergence rate is higher for small system and relative slow for larger system.

Keywords heterogeneous distributed parallel system; dynamic load balancing; diffusion algorithm

1 引 言

许多工厂、企业和教育科研单位都有自己的工作站和微机网络系统.这些微机和工作站在很多时间都处在空闲状态.这些空闲的计算机资源的总和往往超过一台超级计算机.利用这些空闲的计算资源开展高性能计算是很有意义的.高效使用这些计算资源必须解决异构分布并行处理的负载平衡问题.

负载平衡是并行处理的一个重要研究领域[1~3].如果计算负载可以在运行之前确定,并可事先将负载按需要划分,这种问题属于静态负载平衡问题;如

收稿日期:2001-11-27;修改稿收到日期:2003-07-03.本课题得到国家自然科学基金(40245023,60273007,60121160743)和国家科技攻关计划项目(2001DA607B)资助.**金之雁**,男,1962 年生,高级工程师,主要研究领域为数值天气预报、并行计算.E-mail:jinzy@cma.gov.cn.**王鼎兴**,男,1937 年生,教授,博士生导师,主要研究领域为并行计算机结构、高性能计算编译技术等.

果计算负载只能在运行时进行实时测量并确定负载划分,这种问题属于动态负载平衡问题.本文主要讨论动态负载平衡问题.

分布并行处理系统是由多个节点按照一定的结构相互连接构成的,在本文中我们假定每个节点只能与其周围直接连接的节点通信,计算负载也只能通过这些连接线移动.每个节点只能与其周围直接连接的节点平衡负载,通过多次循环迭代实现系统的负载平衡[4].采用这种方式的有维交换法(dimension exchange method)[5]和梯度法(gradient based method)[6]等.Cybenko 提出了同构系统的扩散方法(diffusion method)[6],该方法归结为对一个实对称扩散矩阵的循环迭代过程,该文证明了迭代过程的守恒性与收敛性.本文提出了适用性更广的异构扩散方法,该方法所得到的是非对称扩散矩阵.本文证明了该算法的收敛性和守恒性,并对不同规模的二维平面格栅网并行系统进行了试验.

2 异构系统负载平衡的数学模型

设并行处理系统是由某种拓扑结构的网络相连的处理速度不同的节点构成,可用图 $G=(V,E)$ 表示,其中 V 是节点,E 是边,代表节点之间的通信连接.$P=|V|$ 是节点数量.将节点从 1 至 P 顺序编号,使每个节点有唯一的序号,节点 i 可用 p_i 表示.将边从 1 至 $|E|$ 编号,使每个边也有唯一的序号,边 i 可用 e_i 表示,也可用 e_{ij} 表示连接节点 i,j 的边,$|E|$ 是边的数量.节点 i 的处理速度是 s_i,计算负载是 w_i,计算时间是 $l_i=\frac{w_i}{s_i}$,整个系统的最短计算时间是

$$\bar{l}=\frac{\sum_{i=1}^{P}w_i}{\sum_{i=1}^{P}s_i}=\frac{\sum_{i=1}^{P}l_is_i}{\sum_{i=1}^{P}s_i} \tag{1}$$

负载平衡问题可以归结为通过调整计算时间 l_i,使该节点的计算时间为 $\bar{l}$,系统达到负载平衡,节点计算时间变化向量是

$$\boldsymbol{b}=(\bar{l}-l_1,\bar{l}-l_2,\bar{l}-l_3,\cdots,\bar{l}-l_P)^{\mathrm{T}} \tag{2}$$

其中 $(\ )^{\mathrm{T}}$ 表示 $(\ )$ 的转置矩阵.设沿边 e_i 计算负载的转移量是 δ_i,设计算负载转移向量是 $\boldsymbol{x}=(\delta_1,\delta_2,\delta_3,\cdots,\delta_{|E|})^{\mathrm{T}}$,沿边 e_{ij} 负载转移量也可写为 δ_{ij},对于节点 i,它的计算负载总变化量是 $\sum_{j\leftrightarrow i}\delta_{ij}$,其中 $j\leftrightarrow i$ 表示节点 j 与节点 i 之间有边直接相连,称为节点 j 与节点 i 相邻.如果整个系统达到负载平衡,对于节点 i 有

$$\frac{\sum_{j\leftrightarrow i}\delta_{ij}}{s_i}=\bar{l}-l_i,\ i=1,2,\cdots,P$$

写成矩阵的形式为

$$\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \tag{3}$$

其中 $\boldsymbol{S}$ 为对角矩阵,$\boldsymbol{S}=\mathrm{diag}(s_1,s_2,s_3,\cdots,s_P)$,$\boldsymbol{A}$ 为图 G 的邻接矩阵,其维数为 $|V|\times|E|$,其元素 a_{ij} 为

$$a_{ij}=\begin{cases}1, & \text{边 } j \text{ 与节点 } i \text{ 相连且是起点}\\ -1, & \text{边 } j \text{ 与节点 } i \text{ 相连且是终点}\\ 0, & \text{其它}\end{cases}$$

因此,负载平衡算法归结为求出向量 $\boldsymbol{x}$,使各节点计算时间相同.

对于给定的系统,已知速度对角矩阵 $\boldsymbol{S}$ 和邻接矩阵 $\boldsymbol{A}$,向量 $\boldsymbol{b}$ 可以通过测量处理节点的处理时间 l_i 并通过式(1),(2)得出,因此,动态负载平衡问题归结于求解代数方程式(3).它有 $|E|$ 个未知变量,有 $P=|V|$ 个方程,一般 $|E|>P$,所以式(3)的解不唯一,存在多种方法求解该方程,各种方法的结果也可以各不相同.

3 异构扩散算法模型

如图 1 所示,用一组相互连接的水桶来建立异构扩散算法的模型.设有 P 个圆柱形容器,其水平截面积各不相同,在容器底部有一些管道将容器连接起来,水可通过管道从一个容器中流到另一个容器,称为连通器.所有容器的底面都处在同一个水平面上.如果水平面有高有低,水就会在容器间流动,直至所有容器的水平面高度相同为止.假定该系统是连通的.在此模型中,水的体积与计算量相对应,水桶的水位与节点的计算时间相对应,容器的横截面积与节点的处理速度相对应,水桶之间的连接管

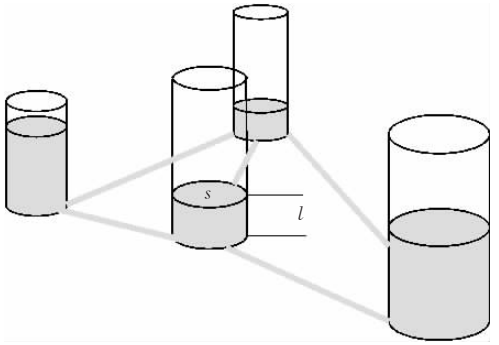


图 1 连通器模型

与节点之间的通信线路对应. 如果计算时间(水位)不相同,一些计算负载(水)会从通过处理节点间的通信连接(管道)从一个节点转移到另一个节点. 如果计算时间相等,整个系统达到负载均衡,计算负载不再移动. 显然在此模型中,管道的粗细对实现整个系统的最终平衡没有影响,但是它对各管道的具体流量是有关的.

设节点 i,j 相邻,一个时间步内通过 e_{ij} 到达 i 的流量正比于它们的水位差. 设在第 t 次迭代时,各节点的运算时间是 $l_i^{(t)}$,扩散算法的公式是:$l_i^{(t+1)}=l_i^{(t)}+\frac{1}{s_i}\sum_{j\leftrightarrow i}\tau_{ij}(l_j^{(t)}-l_i^{(t)})$,其中 τ_{ij} 是非负的常量. 此式表示每一步,节点 i,j 之间交换的计算负载为 $\tau_{ij}(l_j^t-l_i^t)$,引起的计算时间的变化为 $\frac{1}{s_i}\tau_{ij}(l_j^t-l_i^t)$. 如果此量为正,表示计算量由 j 转移到 i,如果为负,表示转移方向相反. 假定通信线路是双向的,因此 $\tau_{ji}=\tau_{ij}$. 扩散公式可改写为 $l_i^{t+1}=\left(1-\frac{1}{s_i}\sum_{j\leftrightarrow i}\tau_{ij}\right)l_i^t+\frac{1}{s_i}\sum_{j\leftrightarrow i}\tau_{ij}l_j^t$,如果与节点 i 相邻的节点在初始时刻的计算时间为零,在一个迭代步内,节点 i 上的计算时间不能为负,所以对于常数 τ_{ij} 的两个约束条件是

$$\begin{cases}\tau_{ij}>0\\ \left(1-\frac{1}{s_i}\sum_j\tau_{ij}\right)\geq 0\end{cases}\tag{4}$$

令 $\boldsymbol{l}^{(t)}=(l_1^{(t)},l_2^{(t)},\cdots,l_P^{(t)})^{\mathrm{T}}$,扩散公式写成矩阵形式为

$$\boldsymbol{l}^{(t+1)}=\boldsymbol{M}\boldsymbol{l}^{(t)}\tag{5}$$

其中,$\boldsymbol{M}=(m_{ij})$ 为 $|V|$ 阶方阵,其元素

$$m_{ij}=\begin{cases}\frac{\tau_{ij}}{s_i}, & \text{如果 } i\leftrightarrow j\\ 0, & \text{其它}\\ 1-\frac{1}{s_i}\sum_k\tau_{ik}, & \text{如果 } i=j\end{cases}\tag{6}$$

显然,对于其中的每一行元素都有 $\sum_j m_{ij}=1$,容易验证 $\boldsymbol{M}=\boldsymbol{I}-\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^{\mathrm{T}}$,其中 $\boldsymbol{I}$ 为单位矩阵,$\boldsymbol{W}=\mathrm{diag}(\tau_1,\tau_2,\tau_3,\cdots,\tau_{|E|})$,$\boldsymbol{A}$ 为图 G 的邻接矩阵,$\boldsymbol{S}=\mathrm{diag}(s_1,s_2,s_3,\cdots,s_P)$,令 $\boldsymbol{y}^{(t+1)}=\boldsymbol{W}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{l}^{(t)}$,则方程(3)的解 $\boldsymbol{x}=\sum_{t=1}^{\infty}\boldsymbol{y}^{(t)}$,显然,扩散矩阵 $\boldsymbol{M}$ 不是对称矩阵. 以下证明该算法的收敛性.

4　扩散方法的收敛性

在异构系统的负载平衡过程中,总计算时间(各节点计算时间之和)不守恒,但是总的计算量守恒.

定理 1.　在式(5)中的迭代循环中,总计算量 $\boldsymbol{W}^{(t)}=\sum_{i\in V}s_i l_i^{(t)}$ 不变. 即 $\boldsymbol{W}^{(t)}=\boldsymbol{W}^{(0)}$,$t=1,2,\cdots$.

证明.　设 $\boldsymbol{s}=(s_1,s_2,\cdots,s_P)^{\mathrm{T}}$,注意到 $\tau_{ji}=\tau_{ij}$,可以得到 $\boldsymbol{W}^{(1)}=\boldsymbol{s}\cdot\boldsymbol{l}^{(1)}=\boldsymbol{s}\cdot\boldsymbol{M}\boldsymbol{l}^{(0)}=\boldsymbol{s}\cdot\boldsymbol{l}^{(0)}=\boldsymbol{W}^{(0)}$,同理 $\boldsymbol{W}^{(2)}=\boldsymbol{W}^{(1)}$,所以 $\boldsymbol{W}^{(t)}=\boldsymbol{W}^{(0)}$.　证毕.

所以循环迭代(式(5))的总计算量保持不变,只是对计算负载进行了重新分配. 根据线性代数理论,式(5)的收敛性取决于扩散矩阵 $\boldsymbol{M}$ 的特征值,需要考察扩散矩阵 $\boldsymbol{M}$ 的特征值情况. 实对称矩阵的所有特征值都是实数,但是,扩散矩阵 $\boldsymbol{M}$ 为非对称矩阵. 可以证明,$\boldsymbol{M}$ 的所有特征值仍为实数.

定理 2.　由式(6)定义的方阵 $\boldsymbol{M}$ 的特征值为实数.

证明.　设 $\boldsymbol{M}'=(m'_{ij})$ 为实对称矩阵,其中,

$$m'_{ij}=\begin{cases}\tau_{ij}, & i\leftrightarrow j,\\ s_i-\sum_{k\leftrightarrow i}\tau_{ik}, & i=j,\\ 0, & \text{其它.}\end{cases}$$

所以 $\boldsymbol{M}=\boldsymbol{S}^{-1}\boldsymbol{M}'$,设 $\boldsymbol{M}$ 的特征值为 λ,特征向量为 $\boldsymbol{X}$,$\overline{(\ )}$ 为 $(\ )$ 的共轭矩阵,$(\ )^{\mathrm{T}}$ 为 $(\ )$ 的转置矩阵. 有 $\overline{(\boldsymbol{M}\boldsymbol{X})^{\mathrm{T}}}=\overline{(\lambda\boldsymbol{X})^{\mathrm{T}}}$,$\overline{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}}=\bar{\lambda}\overline{\boldsymbol{X}^{\mathrm{T}}}$,$\overline{\boldsymbol{X}^{\mathrm{T}}}\ \overline{\boldsymbol{M}'\boldsymbol{S}^{-1}}=\bar{\lambda}\overline{\boldsymbol{X}^{\mathrm{T}}}$,$\overline{\boldsymbol{X}^{\mathrm{T}}}\boldsymbol{M}'\boldsymbol{S}^{-1}\boldsymbol{S}\boldsymbol{X}=\bar{\lambda}\overline{\boldsymbol{X}^{\mathrm{T}}}\boldsymbol{S}\boldsymbol{X}$,$\lambda\overline{\boldsymbol{X}^{\mathrm{T}}}\boldsymbol{S}\boldsymbol{X}=\bar{\lambda}\overline{\boldsymbol{X}^{\mathrm{T}}}\boldsymbol{S}\boldsymbol{X}$,因为 $\overline{\boldsymbol{X}^{\mathrm{T}}}\boldsymbol{S}\boldsymbol{X}\neq 0$,所以 $\lambda=\bar{\lambda}$.　证毕.

迭代过程(5)的收敛性取决于矩阵 $\boldsymbol{M}$ 特征值的绝对值情况,如果存在绝对值大于 1 的特征值,迭代过程(5)不收敛. 我们下面证明 $\boldsymbol{M}$ 没有绝对值大于 1 的特征值.

定理 3.　如式(6)给定矩阵 $\boldsymbol{M}$,它的所有特征值的模小于等于 1.

证明. 设 $\boldsymbol{M}$ 的特征值是 λ,相应的特征向量是 $\boldsymbol{X}$,因为 $\boldsymbol{M}\boldsymbol{X}=\lambda\boldsymbol{X}$,对于其中的第 i 行有 $\sum_j m_{ij}x_j=\lambda x_i$,记特征向量是 $\boldsymbol{X}$ 中的最大分量为 $x_k=\max_{1\leq j\leq|V|}|x_j|$,注意到 $\boldsymbol{M}$ 的所有元素均大于等于零和 $\sum_j m_{ij}=1$,所以 $|\lambda|=\left|\frac{\lambda x_k}{x_k}\right|=\left|\frac{\sum_{j=1}^{|V|}m_{kj}x_j}{x_k}\right|=\left|\sum_{j=1}^{|V|}m_{kj}\frac{x_j}{x_k}\right|\leq\sum_{j=1}^{|V|}|m_{kj}|\left|\frac{x_j}{x_k}\right|\leq\left|\sum_{j=1}^{|V|}m_{kj}\right|=1$.　证毕.

所以,$\boldsymbol{M}$ 的特征值在闭区间[1,−1]内. 从证明过程可以看到,如果 $\lambda=1$,则只有在 $\frac{x_j}{x_k}=1$,当 $m_{k,j}\neq 0$ 时,才有等式成立,即 $x_j=x_k$,$j\leftrightarrow k$,根据连同性假

定可得特征向量的所有分量均相等,即 $x_j=c, j=1, P$,所以 $\boldsymbol{M}$ 的最大特征值是 1,相应的特征向量是常数向量 $\boldsymbol{X}=(c,c,\cdots,c)^{\mathrm{T}}$. 并且只有常数特征向量,$c$ 为常数. 所以,特征值 $\lambda=1$ 的代数重数是 1,它是负载平衡的特征向量.

定理 4. 如式(6)给定图 G 相应的矩阵 $\boldsymbol{M}$,且满足约束条件(4),-1 不是 $\boldsymbol{M}$ 的特征值的充分必要条件是:

(i)$\boldsymbol{M}$ 的主对角元素不全为 0;

(ii)G 不是二部图.

证明.

假定 $\boldsymbol{M}$ 有特征值 -1,相应的特征向量为 $\boldsymbol{X}$,由于 $\boldsymbol{M}$ 的所有元素均为非负元素,且由假定 $\boldsymbol{MX}=-\boldsymbol{X}$,所以特征向量 $\boldsymbol{X}$ 中一定存在大于和小于零的元素,不妨令 $\boldsymbol{X}$ 的元素中绝对值最大的元素为 -1,并通过调整 $\boldsymbol{X}$ 元素的顺序和 $\boldsymbol{M}$ 行列的顺序,将 $\boldsymbol{X}$ 中的元素按大小逆序排序,设 $\boldsymbol{X}$ 中有 k 个小于零的元素,m 个零元素,n 个大于零的元素,所以 $\boldsymbol{X}$ 可以写成如下形式:

$$\boldsymbol{X}=(-1,\cdots,-x_k,0,\cdots,0,x_{k+m+1},\cdots,x_P)^{\mathrm{T}}=\begin{pmatrix}-\boldsymbol{X}_1\\ \boldsymbol{0}\\ \boldsymbol{X}_3\end{pmatrix},$$

$$\boldsymbol{MX}=\begin{pmatrix}
m_{1,1} & \cdots & m_{1,k} & m_{1,k+1} & \cdots & m_{1,k+m} & m_{1,k+m+1} & \cdots & m_{1,P}\\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots\\
m_{k,1} & \cdots & m_{k,k} & m_{k,k+1} & \cdots & m_{k,k+m} & m_{k,k+m+1} & \cdots & m_{k,P}\\
m_{k+1,1} & \cdots & m_{k+1,k} & m_{k+1,k+1} & \cdots & m_{k+1,k+m} & m_{k+1,k+m+1} & \cdots & m_{k+1,P}\\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots\\
m_{k+m,1} & \cdots & m_{k+m,k} & m_{k+m,k+1} & \cdots & m_{k+m,k+m} & m_{k+m,k+m+1} & \cdots & m_{k+m,P}\\
m_{k+m+1,1} & \cdots & m_{k+m+1,k} & m_{k+m+1,k+1} & \cdots & m_{k+m+1,k+m} & m_{k+m+1,k+m+1} & \cdots & m_{k+m+1,P}\\
\vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots\\
m_{P,1} & \cdots & m_{P,k} & m_{P,k+1} & \cdots & m_{P,k+m} & m_{P,k+m+1} & \cdots & m_{P,P}
\end{pmatrix}
\begin{pmatrix}-1\\ \vdots\\ -x_k\\ 0\\ \vdots\\ 0\\ x_{k+m+1}\\ \vdots\\ x_P\end{pmatrix}=
\begin{pmatrix}1\\ \vdots\\ x_k\\ 0\\ \vdots\\ 0\\ -x_{k+m+1}\\ \vdots\\ -x_P\end{pmatrix}$$

写成简化的形式为

$$\begin{pmatrix}\boldsymbol{M}_{1,1} & \boldsymbol{M}_{1,2} & \boldsymbol{M}_{1,3}\\ \boldsymbol{M}_{2,1} & \boldsymbol{M}_{2,2} & \boldsymbol{M}_{2,3}\\ \boldsymbol{M}_{3,1} & \boldsymbol{M}_{3,2} & \boldsymbol{M}_{3,3}\end{pmatrix}\begin{pmatrix}-\boldsymbol{X}_1\\ \boldsymbol{0}\\ \boldsymbol{X}_3\end{pmatrix}=\begin{pmatrix}\boldsymbol{X}_1\\ \boldsymbol{0}\\ -\boldsymbol{X}_3\end{pmatrix}.$$

其中,$\boldsymbol{M}_{i,j}$ 为矩阵 $\boldsymbol{M}$ 分块后相应的子阵. 由第一行:

$$-(m_{1,1}+m_{1,2}x_2+\cdots+m_{1,k}x_k)+$$
$$(m_{1,k+m+1}x_{k+m+1}+\cdots+m_{1,P}x_P)=1,$$

等式左边两项的绝对值均小于等于 1,因此,等式成立必定有第一项为零,第二项为 1. 由第一项为零,有

$$m_{1,1}+m_{1,2}x_2+\cdots+m_{1,k}x_k=0.$$

根据假定,$1\geq x_n>0, n=1,2,\cdots,k$,所以

$$m_{1,1}=m_{1,2}=\cdots=m_{1,k}=0.$$

由第二项为 1,有

$$(m_{1,k+m+1}x_{k+m+1}+\cdots+m_{1,P}x_P)=1.$$

由于 $0<x_{k+m+1},\cdots,x_P\leq 1$,所以 $(m_{1,k+m+1}+\cdots+m_{1,P})\geq 1$.

注意到

$$(m_{1,k+1}+\cdots+m_{1,k+m})+(m_{1,k+m+1}+\cdots+m_{1,P})=1,$$
$$m_{1,k+1}=\cdots=m_{1,k+m}=0,$$

$x_i=1$,当 $m_{1,i}\neq 0, i\in(k+m+1,P)$.

所以节点 1 仅与 $k+m+1,\cdots,P$ 中的处理节点相连,由网络连通假定,它至少与其中之一相连,设为处理节点 $i, i\in(k+m+1,P)$,所以按照与第一行相似的分析过程,对第 i 行进行分析,可以得到

$$m_{i,k+1}=\cdots=m_{i,P}=0,\ x_j=1,$$

当 $m_{i,j}\neq 0,\ j\in(1,2,\cdots,k), j\leftrightarrow i$.

由于网络是连通的,所以通过节点 1 可以到达任何一个节点,循环上述过程,可以到达矩阵的所有行,所以矩阵 $\boldsymbol{M}$ 的分块子阵:

$$\boldsymbol{M}_{1,1}=\boldsymbol{0},\boldsymbol{M}_{1,2}=\boldsymbol{0},\boldsymbol{M}_{3,2}=\boldsymbol{0},\boldsymbol{M}_{3,3}=\boldsymbol{0},$$
$$x_i=1,\ i=1,\cdots,k,k+m+1,\cdots,P.$$

其中 $\boldsymbol{0}$ 是相应阶数的矩阵,所有元素为零. 根据 $\boldsymbol{M}$ 构造方法,$\boldsymbol{M}$ 中的零元素是对称的,所以

$$\boldsymbol{M}_{2,1}=\boldsymbol{0},\boldsymbol{M}_{2,3}=\boldsymbol{0},$$

这时,可以看到,节点 $i, i\in(k+1,k+m)$不与该范围以外的处理节点相连,与连通假设矛盾,所以 $m=0$,即在特征向量中不存在零元素.

所以特征向量可以重新写为

$$\boldsymbol{X}=\begin{pmatrix}\boldsymbol{X}_1\\ -\boldsymbol{X}_3\end{pmatrix},$$

其中 $\boldsymbol{X}_1=\boldsymbol{1},\boldsymbol{X}_3=\boldsymbol{1}$.

矩阵 $\boldsymbol{M}$ 具有如下形式

$$\boldsymbol{M}=\begin{pmatrix}\boldsymbol{0} & \boldsymbol{M}_{1,3}\\ \boldsymbol{M}_{3,1} & \boldsymbol{0}\end{pmatrix} \tag{7}$$

其中 $\boldsymbol{0}$ 是相应阶数的矩阵,其元素均为 0.

以上证明了如果矩阵 $\boldsymbol{M}$ 有特征值 -1,则经过重新对行列的调整,$\boldsymbol{M}$ 一定可写为式(7)的形式.下面证明如果 $\boldsymbol{M}$ 可写为式(7)的形式,则它存在特征值 -1.

构造向量 $\boldsymbol{X}=\begin{pmatrix}-\boldsymbol{X}_1\\ \boldsymbol{X}_1\end{pmatrix}$,其中 $\boldsymbol{X}_1=\boldsymbol{1}$,则

$$\boldsymbol{MX}=\begin{pmatrix}\boldsymbol{0} & \boldsymbol{M}_{1,3}\\ \boldsymbol{M}_{3,1} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}-\boldsymbol{X}_1\\ \boldsymbol{X}_1\end{pmatrix}=-\begin{pmatrix}-\boldsymbol{X}_1\\ \boldsymbol{X}_1\end{pmatrix}.$$

因此 $\boldsymbol{M}$ 确有 -1 的特征值.

因此 $\boldsymbol{M}$ 没有 -1 特征值的充要条件是 $\boldsymbol{M}$ 没有式(7)的形式,即

(i)$\boldsymbol{M}$ 的主对角元素不全为 0.

(ii)G 不是二部图.　　证毕.

定理 5.　给定图 $G=(V,E)$ 相应的矩阵 $\boldsymbol{M}$,满足约束条件(4),且 $\boldsymbol{M}$ 的主对角元素不全为 0,G 不是二部图,则迭代过程(5)收敛于均匀解.

证明.

对于 $\lambda=1$,特征向量 $\boldsymbol{X}=\boldsymbol{1}$,所以对于 $\lambda\neq1$ 的特征向量 $\boldsymbol{X}$ 均与 $\boldsymbol{X}=\boldsymbol{1}$ 正交.构造矩阵 $\boldsymbol{B}=\boldsymbol{M}-\boldsymbol{A}$,其中 $\boldsymbol{A}$ 为常数矩阵,其元素为 $\frac{1}{P}$,容易验证,对于 $\boldsymbol{M}$ 的 $\lambda\neq1$ 的特征值和特征向量,也是矩阵 $\boldsymbol{B}$ 的特征值和特征向量,$\boldsymbol{X}=\boldsymbol{1}$ 是 $\boldsymbol{B}$ 关于 $\lambda=0$ 的特征向量.所以,矩阵 $\boldsymbol{B}$ 的谱半径小于 1.对于任意初始向量 $\boldsymbol{u}^{(0)}$ 都有循环 $\boldsymbol{u}^{(t+1)}=\boldsymbol{B}\boldsymbol{u}^{(t)}$ 收敛于 0.注意到 $\boldsymbol{A}\boldsymbol{u}^{(t)}=\bar{\boldsymbol{u}}^{(t)}$,$\boldsymbol{B}\bar{\boldsymbol{u}}^{(t)}=\boldsymbol{0}$,若令 $\boldsymbol{u}^{(0)}=\boldsymbol{l}^{(0)}$,$\boldsymbol{l}^{(t+1)}=\boldsymbol{M}\boldsymbol{l}^{(t)}=\boldsymbol{B}\boldsymbol{l}^{(t)}+\boldsymbol{A}\boldsymbol{l}^{(t)}=\boldsymbol{B}\boldsymbol{u}^{(t)}+\bar{\boldsymbol{l}}^{(t)}$.根据定理 1 有 $\bar{\boldsymbol{W}}=\boldsymbol{l}^{(t+1)}\cdot\boldsymbol{s}=\boldsymbol{u}^{(t+1)}\cdot\boldsymbol{s}+\bar{\boldsymbol{l}}^{(t)}\cdot\boldsymbol{s}$,因为 $\lim\limits_{t\to\infty}\boldsymbol{u}^{(t+1)}\cdot\boldsymbol{s}=0$,有

$$\lim_{t\to\infty}\bar{\boldsymbol{l}}^{(t)}\cdot\boldsymbol{s}=\lim_{t\to\infty}\bar{\boldsymbol{l}}^{(t)}\boldsymbol{s}=\bar{\boldsymbol{l}}\boldsymbol{s}=\bar{\boldsymbol{W}},$$

因此 $\lim\limits_{t\to\infty}\bar{\boldsymbol{l}}^{(t)}=\bar{\boldsymbol{l}}$,$\lim\limits_{t\to\infty}\boldsymbol{l}^{(t+1)}=\bar{\boldsymbol{l}}$.　　证毕.

对 $\boldsymbol{l}^{(t)}$ 由 $t=1,2,\cdots,\infty$ 求和可得 $\boldsymbol{l}^{(\infty)}-\boldsymbol{l}^{(0)}=-\boldsymbol{S}^{-1}\boldsymbol{A}\sum\limits_{t=1}^{\infty}\boldsymbol{W}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{l}^{(t)}$,由于 $\boldsymbol{l}^{(\infty)}=\bar{\boldsymbol{l}}$,若令 $\boldsymbol{x}=\sum\limits_{t=1}^{\infty}\boldsymbol{W}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{l}^{(t)}$,有 $\boldsymbol{b}=\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{x}$.故 $\boldsymbol{x}$ 是方程(7)的解.事实上,不可能循环至无穷,若在 $k+1$ 结束循环,我们得到 $\boldsymbol{b}^{(k+1)}=\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{x}^{(k+1)}$,其中 $\boldsymbol{b}^{(k+1)}=\boldsymbol{l}^{(k+1)}-\boldsymbol{l}^{(0)}$,

$$\boldsymbol{x}^{(k+1)}=\sum_{t=1}^{k}\boldsymbol{W}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{l}^{(t)} \tag{8}$$

是方程(3)的近似解.

4　扩散方法的并行计算步骤

我们可以直接用式(2)和式(8)来计算节点的当前计算时间和沿每一条边负载的移动量,得到计算算法如下.

步骤 a:初始化

每个处理节点找出它的所有邻居 j

计算 $m_{ij}=\frac{\tau_{ij}}{s_i}$ 和 $m_{ii}=1-\sum\limits_{j}\frac{\tau_{ij}}{s_i}$.

步骤 b:循环迭代

对属于本节点邻居的 j 循环,循环体开始

向节点 j 非阻塞发送本节点的计算时间 $l_i^{(k)}$,

从节点 j 阻塞接受该节点的计算时间 $l_j^{(k)}$,

$w=w+m_{ij}l_j^k$.

循环体结束.

计算节点的计算时间 $d_i^{(k+1)}=d_i^k+l_i^k$.

计算 $l_i^{(k+1)}=m_{ii}l_i^k+w$.

计算负载平衡函数(见本文式(9)),如果大于给定阈值,转到步骤 b,否则转到步骤 c.

步骤 c:计算负载转移量.

对属于本节点邻居的 j 循环,循环体开始

向节点 j 非阻塞发送本节点的 d_i^k.

从节点 j 阻塞接受该节点的 $d_j^{(k)}$.

循环体结束.

对属于本节点邻居的 j 循环,循环体开始

计算 $w=\tau_{ij}(d_i-d_j)$.

如果 $w<0$,节点 j 向 i 移动计算负载 w,否则,节点 i 向 j 移动计算负载 w.

循环体结束.

该算法同时求出了每个节点的计算时间和对每个边的负载移动量.

5　扩散矩阵的构造

从理论上讲,由式(6)定义的扩散矩阵 $\boldsymbol{M}$,只要满足式(4),从任何初始值开始,迭代循环式(5)都可以收敛于均匀解.本文提出用一种方法构造扩散矩阵.其元素为

$$m_{ij}=\begin{cases}\tau'_{ij}, & i\leftrightarrow j\\ 1-\sum\limits_{k\leftrightarrow i}\tau'_{ik}, & i=j\\ 0, & \text{其它}\end{cases}.$$

其中

$$\tau'_{ij}=\min(s(i),s(j))\times\min\left(\frac{1}{d(i)+1},\frac{1}{d(j)+1}\right).$$

显然,这种构造扩散矩阵的方法能够满足式(4)的要求.

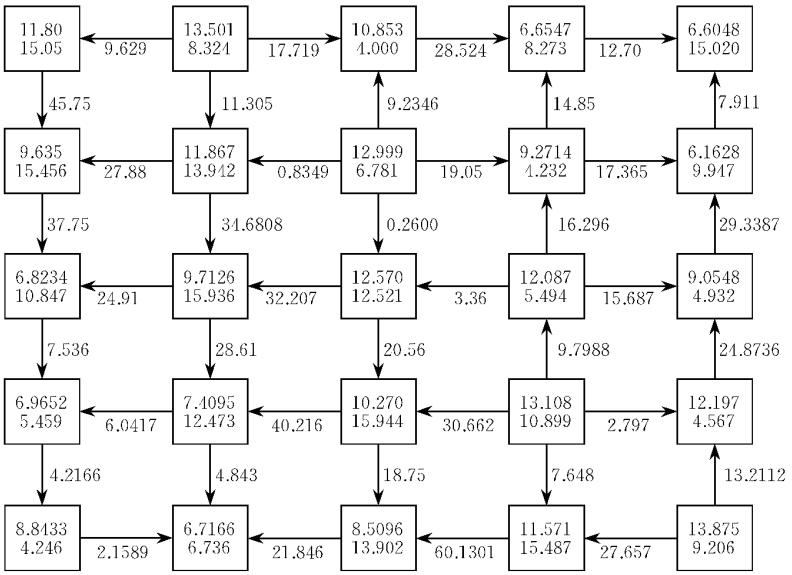
6 数值试验与性能分析

定义负载平衡函数

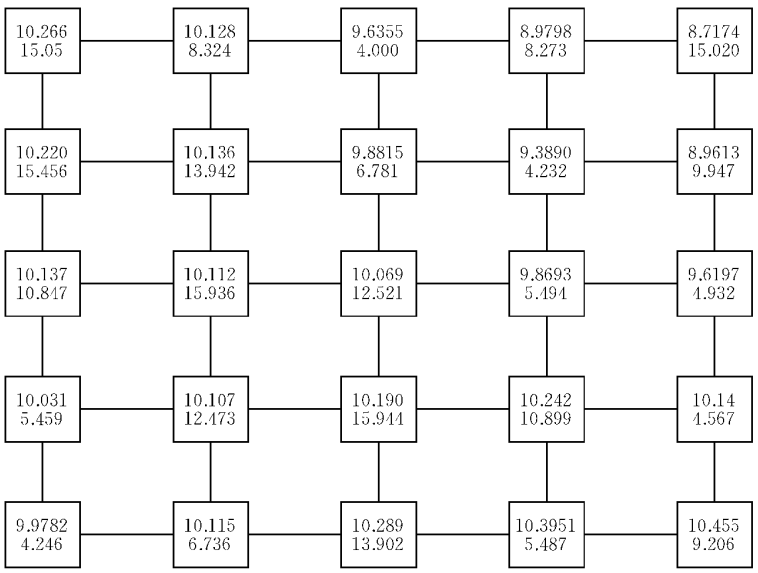
$$I_m = \frac{\max_{i=1}^{P}(l_i) \times \sum_{i=1}^{P} s_i - \sum_{i=1}^{P} s_i l_i}{\sum_{i=1}^{P} s_i l_i} = \frac{1}{E} - 1 \quad (9)$$

由于二维格栅网(mesh)是大规模数据并行应用中最常见的结构,实验在这种结构的网络上进行. 节点处理速度设为 4～16 之间的随机数,最快节点速度约为最慢节点的 4 倍. 设定 $l_i^{(0)}$ 的初值为 5～15 之间的随机数,最长的计算时间约为最短计算时间的 3 倍.

我们考察了一个 5×5 的二维网的收敛情况,在初始状态下,它的负载平衡函数约为 50%,经过 14 步迭代以后,下降到 5%以下. 图 2 是 14 步迭代过程前后计算时间和负载移动示意图. 其中每个方框为一个处理节点,方框内的第一行数字为当前该节点的运算时间,第二行数字为该节点的处理速度,方框之间的连线代表处理节点间的通信线路,箭头代表计算负载的移动方向,箭头边的数字是沿该方向移动的计算负载总量. 图 2(a)是负载平衡以前的情况,图 2(b)是负载平衡以后的情况.



(a) 负载平衡以前的运行时间和计算负载移动方向及数值



(b) 实现负载平衡以后的计算负载情况

图 2 负载平衡试验实例(每个方框为一个节点,方框内的第 1 行数字为当前该节点的运算时间,第 2 行数字为该节点的处理速度,方框之间的连线代表节点间的通信线路,箭头代表计算负载的移动方向,箭头边的数字是沿该方向移动的计算负载总量)

另一组试验是在系统规模分别为 16～256 个节点的二维格栅网络系统进行的,在试验中采用了与第一组试验相同的处理机速度和 $l_i^{(0)}$ 初值的设定方法,根据式(9),可计算出每一个循环迭代的负载平衡函数 I_m,图 3 是 I_m 随循环变量 t 变化的情况.

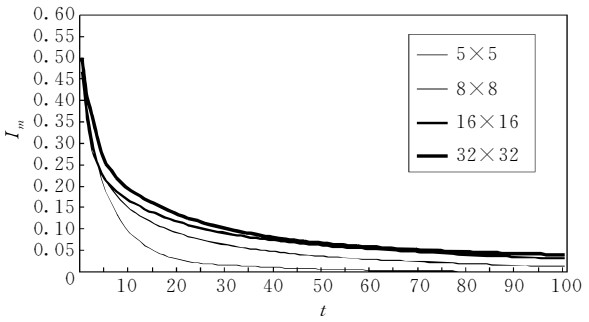


图 3　几种不同网络的 I_m 随循环变量 t 变化的试验结果

可以看出,负载平衡函数随循环次数的增加而减少,在开始阶段下降很快,但是当下降到 10% 以后,下降速度明显减慢,所以如果节点数量在 1024 个以下,要求负载不平衡不超过 5%～10%,该方法的效率比较高,如果要求达到更高,迭代次数就比较多了.

在本文中,我们假定计算负载是无限可分的,但是在实际应用中这是不可能的,计算负载有最小的可划分单位,负载不可能达到绝对平衡.从这个意义上讲,只能将负载平衡维持在适当的限度之内.所以,当负载平衡函数下降到某个规定的阈值时即可终止循环过程.

7　结论和今后的工作方向

本文提出了在异构系统中的负载平衡的数学模型,根据连通器模型建立了异构扩散方法的算法,并在理论上证明了该算法能够在异构环境中从任意计算负载分布下,通过迭代过程,将各节点的负载通过节点间的通信网络在节点之间移动,从而实现各节点间的负载平衡.该算法的特点是,除最后确定终止迭代外,每个节点只需要与其相邻的节点通信.本文同时给出了如何从根据一个具体的异构系统构造扩散矩阵的方法.并在此基础上对在工程计算中最常用的二维格栅网结构的并行系统进行了数值计算试验.实验证明,虽然该算法的收敛速度随二维格栅网的结点数量的增多而降低,但是,在 1024 个节点以内的二维格栅网组成的并行系统上收敛速度比较高,可以满足大多数系统的需要.

参　考　文　献

1　Casavant T L, Kuhl J G. A taxonomy of scheduling in general-purpose distributed computing. Systems IEEE Transactions on Software Engineering, 1988,14(2):141～154

2　Willebeek-LeMair M H, Reeves A P. Strategies for dynamic load balancing on highly parallel computer. IEEE Transactions on Parallel and Distributed System, 1993, 4(9):979～993

3　Xu C Z, Lau F C M. Iterative dynamic load balancing in multicomputers. Journal of Operational Research Soc. 1994, 45(7): 786～796

4　Jaja J, Ryu K W. Load balancing on the hypercube and related networks. In: Proceedings of International Conference Parallel Processing, Dupage County, Ill. USA, 1990, 1:203～210

5　Lin C H, Keller R M. The gradient model load distribution methods. IEEE Transactions on Software Engineering, 1987,13(1):32～38

6　Cybenko G. Dynamic load balancing for distributed memory multiprocessores. Journal of Parallel and Distributed Computing, 1989,7(2):279～301

JIN Zhi-Yan, born on 1962, senior research fellow at Chinese Academy of meterological Sciences. His research interests include numerical weather prediction, parallel processing.

WANG Ding-Xing, born in 1937, professor, Ph. D. supervisor. His research interests include parallel architecture, compiler techniques for high performance computing, etc.