静态负载平衡问题的表示与算法*)

The Description and Algorithms of Static Load Balancing in Graph Theory

秦忠国　姜弘道

(河海大学工程力学系　南京210024)

摘　要　For some numerical method such as FEM, the load balancing is a important problem influencing overall efficiency and not been handled well yet at present. In this paper the static load balancing problem which arise in most of scientific and engeering computing is described in the view of graph theory and some heuristic methods are discussed in detail.

关键词　Load balancing, Graph theroy, Spectral Method

1. 引言

对工程与科学计算中的若干数值分析问题，其并行算法已经开展了较为广泛的研究，如线性代数问题的并行求解等，但是目前在这些数值并行算法中，对于特定的并行机体系结构，怎样进行任务划分与处理机分配以求获得较高的并行效率，或者是没有考虑，或者用一些经验方法。然而，任务划分与处理机分配带来的负载平衡问题是并行计算系统的基本问题，直接影响并行计算的效率，当并行机的节点机较多、求解规模较大时更是如此。

不同的应用问题，并行算法在数据处理、数据交换、同步方面具有不同的特征，负载平衡算法有较强的依赖性。因此，并行操作系统目前一般也不具有负载平衡的能力，但在应用程序里实现这些负载平衡算法并不困难。

自七十年代以来，图论已成为数学中发展最快的分支之一。应用图论来解决运筹学、网络理论、控制理论和计算机科学中的若干问题显示了极大的优越性。图论有若干算法依赖于计算机，而图论的进一步发展又可帮助解决计算机科学中的若干问题。

本文以分布存储并行机为背景，用图论方法描述数值分析中常见的静态负载平衡问题，并讨论其相应的算法。

2. 并行机与并行应用程序的图示

在分布存储的并行机体系中，数据交换是通过网络进行的，这样可以把其看成是自带存储器的一批处理机与通信网络的结合。用平面图上的一个点集 $U=\{u_1,\cdots,u_p\}$ 表示 p 个处理机，用这些点的可能连线 $F=\{f_1,\cdots,f_k\}\subseteq U\times U$ 表示节点间的通信线路。仅考虑同构网络，即节点机的参数是相同的，假定处理机之间的消息传递能力与方向无关，因而，可以略去节点的权重，也不必考虑边的方向性，这样，分布存储的并行机可以用无向图 $H=(U,F)$ 表示(图1)。

一个并行应用程序用赋权图 $G=(V,E,\rho,\sigma)$ 来描述(图2)，其中节点 $V=\{v_1,\cdots,v_n\}$，边 $E=\{e_1,\cdots,e_m\}\subseteq V\times V$，节点权重 $\rho:V\to R$，边的权重 $\sigma:E\to R$，即用某种方式对 V、E 进行量化，节点与边的具体含义可由并行应用的具体情况而定，对于本文讨论的数值分析问题，节点可代表并行程序的各进程的数据处理(如分裂子区域的单元数量)，边代表数据通信，当用赋权图描述一个并行应用程序时，其拓扑可由图的邻接矩阵 A(或关联矩阵 A_c)表示：

$$A=(a_{ij})_{n\times n}$$

其中，$1\le i,j\le n$，$a_{ij}=1$ 表示节点 v_i,v_j 有通信，$a_{ij}=0$ 表示节点 v_i,v_j 无通信。

G 的端容量矩阵 T 为一实对称矩阵，可用来表示节点之间的通信量：

$$T=(t_{ij})_{n\times n}$$

其中，$1\le i,j\le n$，当 $i=j$ 时，令 $t_{ij}=0$，即忽略节点与自身的通信开销。

*)中国博士后基金、水利科研基金课题资助。秦忠国　博士后，主要研究领域为结构分析、并行数值计算；姜弘道　教授，博士生导师，校长，江苏省力学学会理事长，研究领域为计算力学、复杂结构分析方法等。

G的顶点容量代表节点的数据处理的计算量，用实列向量C表示，则

$$C=\begin{pmatrix}c_1\\ \cdots\\ c_n\end{pmatrix}=\begin{pmatrix}\rho(v_1)\\ \cdots\\ \rho(v_n)\end{pmatrix}$$

在对并行机与并行应用程序进行上述描述的基础上，负载平衡问题可以看成是一个图的嵌入问题[7]，对于数值分析问题，一般可认为 $n\gg p$，即图G的节点远大于图H的节点，负载平衡的任务是找到一个"好的"映射 $\pi:G\to H$，使某些判据得到极小化满足。根据并行应用的不同性质，图G可能是静态的或动态的，也就是说，程序生成的并行任务在运行期间所做的计算量是否变化。而图H对应于并行机结构，被认为是不变的。

所谓静态问题就是只需将并行应用图G向图H做一次映射，保持这个映射一直到程序运行结束，这样，静态问题的负载平衡问题退化成一个古典的映射问题，这个映射问题对应两个图的嵌入问题。对于绝大部分科学与工程计算问题，常常可以看成静态问题来研究，例如非自适应的有限元计算。

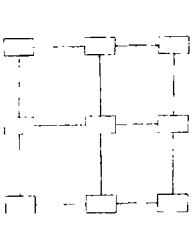

图1

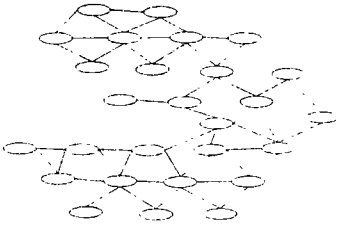

图2

3. 静态问题的负载平衡

静态问题的模型图G的拓扑结构与重量运行中是不变的。假设在程序启动前这个图是已知的，对于一般的静态问题，这个假设是成立的。例如非自适应有限元分析，计算前可根据单元数估算其运算量。

如上所述，负载平衡问题的目标是找到一个"好的"方法，将图 $G=(V,E,\rho,\sigma)$ 映射到图 $H=(U,F)$ 上，设这个映射为 $\pi:V\to U$，则 π 必须定义G中的节点与H中节点的对应关系，设这个关系由矩阵 $B=(b_{ij})_{n\times p}$ 确定，并有

$$b_{ij}=\begin{cases}1 & G\text{中}j\text{节点分配于}H\text{中的}i\text{节点}\\ 0 & G\text{中}j\text{节点不分配于}H\text{中的}i\text{节点}\end{cases}$$

另用列向量 $X=(x_1,x_2,\cdots,x_p)^T$ 表示每个处理机的运算开销，则有

$$X=BC$$

为了使并行系统达到较好的负载平衡状态，首先必须控制单个处理机承受的最大负载，在映射 π 下，这个负载将是：

$$\mathrm{load}(\pi)=\max_{u_k\in U}\sum_{v_j\in V,\pi(v_j)=u_k}\rho(v_j)=\max_{1\le i\le p}(x_i)\qquad(1)$$

设G中的某个边为 $e=\{v_1,v_2\}\in E$，经 π 映射 $\pi(v_1)=u_i\in U$，$\pi(v_2)=u_i\in U$，但在图H中，$\pi(v_1)$ 至 $\pi(v_2)$ 的链路是经 $u_2,u_3,\cdots,u_{i-1}$ 到达，并且有

$f_1=(u_1,u_2),f_2=(u_2,u_3),\cdots,f_i=(u_{i-1},u_i)$

设 w_e 为 u_1 至 u_i 的路径，则有

$w_e=f_1+f_2+\cdots+f_i$

并设 $f_i(i=1,2,\cdots)$ 的长度为1，即 $|f_i|=1$，则

$|w_e|=i$

图G中某一边在H中需要的最长通信链路

$$\mathrm{long}(\pi)=\max_{e=(v_1,v_2)\in E}|w_e|\qquad(2)$$

此外，为使并行机通信网络上的通信量达到较好的均衡状态，为此必须计算并行机处理机之间的通信线路 f_k 上最大通信流量

$$\mathrm{flow}_{f_k}=\sum_{\substack{e=(v_1,v_2)\in E\\ f_k\in w_e}}\sigma(e)$$

为达到较好的负载平衡状态，应控制通信线路上的最大通信量，所以负载平衡的第三个指标为：

$$\mathrm{flow}(\pi)=\max_{f=(u_1,u_2)\in F}\mathrm{flow}_f=\max_{f=(u_1,u_2)\in F}\sum_{\substack{e=(v_1,v_2)\in E\\ f\in w_e}}\sigma(e)\qquad(3)$$

以上(1)、(2)、(3)式给出了负载平衡问题的三个优化目标，要求在任务划分与处理机分配时得到极小满足。(1)式对应图的点集V的等分；(2)式等价于图的最短通路问题；(3)式属于网络流问题。对于一般意义上的图G和H，以上问题被认为是NP完备的，即认为不存在求解上述问题的有效方法[5]。然而对于现代并行机系统，内部的互联网络都具有很强的通信能力，有若干高效路由策略，如 Virual-Cut-Through-Routing 等。在这种情况下，处理机只要将数据发出，经由什么样的通信链路可以不予考虑，因此，图H可以看成是一个完全连接的，此时，$|W_e|\equiv 1$。这样，图G向图H的映射问题可以退化成图G的分割问题。设并行机共有p个处理机节点，则负载平衡问题转化成求图G的p个子集，且满足

$$\mathrm{cut}(\pi)=\sum_{\substack{e=(v_1,v_2)\in E\\ \pi(v_1)\neq\pi(v_2)}}\sigma(e)\qquad(4)$$

为极小，上式意义是要求在对图G进行切割时，要求被切断的边上的通信总量为最小。这是一个图的分割问题。

4. 算法

图的分割问题定义如下：给定无权图 G＝(V，E)或赋权图 G＝(V，E，ρ，σ)，及参数 p(p 为正整数)，求 V 的 p 个子集，使这 p 个子集尽可能地相等，并使被切断的边的个数或权重为极小。

以上问题被证明为 NP 完备的，即使 p＝2时也如此[6]。图的分割问题近年来得到计算机科学界普遍重视，其启发求解算法概述如下。

图的分割有总体[4,5,8]与局部[9]两类算法，总体方法将产生一个较为粗糙的分割。局部方法在整体分割的基础上作进一步的改进。总体分割可分别用几何方法和代数方法。简单的几何方法是将图的节点在某一轴投影，再沿轴向进行划分，这个轴可以是坐标轴或者是图的最小惯性轴。

总体分割应用较多的是建立在代数图理论上的谱方法[8]。设 A 是图 G＝(V，E)的邻接矩阵[1]，D 是包含 G 的节点的度的对角阵，则 L＝D－A 为 G 的 Laplace 矩阵，L 是一个正定降秩阵，其特征值 $\lambda_i \geq 0$ $(i=1,\cdots,p)$，零特征值的个数等于 G 中包含的互不连通的子图的数量，因设 G 为连通的，故只有一个零特征值，设 L 的特征值为 $\lambda_1=0\leq\lambda_2\leq\cdots\leq\lambda_p$，且 $y=(y_1,\cdots,y_p)$为对应于 λ_2的特征向量，则 y 中的每个分量指示了对应节点的某些特征，这些特征可以用来分割 G。这基于以下定理：

定理[2,3] 设 G＝(V，E)是一个连通图，y 是对应于 λ_2的特征向量，y_v 为对应 v 的分量。设有一实数 $\gamma\geq 0$，定义一个 V 的子集 $V_1(r)=\{v\in V: y_v\geq\gamma\}$，则由 V1构成的子图是连通的，相类似，对于实数 $\gamma\leq 0$，对子集 $V_2(r)=\{v\in V: y_v\leq|r|\}$，则由 $V_2(r)$构成的子图也是连通的。

由于 V1和 V2两个子集都包含与 y 中零分量对应的节点，以上定理的一个推理就是：如果对所有节点都有 $y_v\neq 0$，则两个子集 $P=\{v\in V: y_v>0\}$和 $N=\{v\in V: y_v<0\}$都是 G 的两个连通片。

因此，可以根据 y 的分量对节点进行排序，据此划分节点。这里给出这一机制的一个力学解释：把图看成一个弹性网络，则各特征向量代表的是各阶振型，其中必含有某些固有的内在特征，其分量可以看成是振动的位移，根据这个位移来分割图的节点是一个很自然的方法，而自然方法必然满足某些极值条件。

以上算法重复 n 次，可以把 G 图划分为 2^n 个子图，但事实上有可能划分为任意个，例如，当需要9个子图时，第一次按节点数分别划分5/9与4/9，再划分3/9与2/9…。

谱方法可以寻找图中较为稀疏的部位以便进行图的分割，是有效的一种图的分割法，尽管这一方法需要较多的计算量，但由于它不需要求解全部的特征值和特征向量，即使是比较复杂的问题也是有效的。

尽管总体分割可以得到比较好的结果，但仍然存在很多值得进一步改进的可能性，局部算法可以在总体分割的基础上进一步降低被切断的边的数量，其中常用的方法是寻找可以交换节点的子图，然而尝试在这些子图中交换节点，看断边数量是否会减少。

如果图 G 的规模很大，为节省运行时间，常常对原图作一些简化[2,4]，分割后再外推还原成原图。

结束语 本文从图论的角度讨论了数值计算中常见的静态的负载平衡问题，结论是：对于分布存储的并行处理机，当处理机同构时，负载平衡问题等价于一个线图的分割问题，其对应的约束条件，一是要求各处理机上的任务尽可能地相等，二是处理机之间的通信量之和为极小。因此，可用图的分割问题的谱方法进行求解。

参考文献

[1] 楼世博等，图论及其应用，人民邮电出版社，1982.7

[2] S. T. Barnard, H. D. Simon, Fast multilevel implementation of recursive spectral bisection... Concurrency: Practice and Exp. 6(2)1994

[3] M. Fiedler, Algebraic connectivity of graphs, Czech. Math. J., 23,1973

[4] M. Fiedler, A property of eigenvectors of non-negative symmetric matrices and its application to graph theory, Czech. Math. J., 25,1975

[5] B. Hendrickson, R. Leland, An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations, SIAM J. Scientific Computing 16(2) 1995

[6] G. Karypis, V. Kumar, A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, TR 95-035; CS-Dept; Univ. Minnesota

[7] B. Monien, I. H. Sudborough, Embedding One Interconnection Network in Another, Computing Suppl. 7,1990

[8] A. Pothen et al., Partitioining Sparse Matrices with Eigenvectors of Graphs, SIAM J. Math. Anal. Appl. 11/3,1990

[9] B. Ghosh et al:, Tight Analyses of Two Local Load Balancing Algorithms, 27th ACM STOC, 1995