自动并行化与数据自动迁移*)

丁晓宁　朱怡安　康继昌
(西北工业大学计算机系　西安 710072)

摘　要　本文面向计算流体力学(CFD)提出了数据自动迁移的并行计算模型(ADM模型),符合该模型的并行CFD程序能够根据计算节点的计算能力及负载轻重,自动将数据迁移至计算能力强、负载轻的计算节点,从而使得并行程序能够在网络计算平台上取得较好的并行效率;本文还讨论了自动并行化系统对ADM模型的支持方法;最后给出了性能测试结果。

关键词　自动并行化,数据迁移,负载平衡,网络计算

自动并行化技术是解决并行软件开发、并行机应用困难的有效途径,其重要性已被广泛接受。西方发达国家在自动并行化技术方面的研究较多,已经开发出Polaris[1]、SUIF[2]、Tiny、KAP、SuperB、Forge90[3]等比较著名的自动并行化工具,国内主要有复旦大学的AFT[4]、华中理工大学的HZPARA[5]、西北工业大学面向CFD开发的NPUPAR[3,6]等。

目前,随着网络以及网络计算环境的发展,高性能计算逐渐转向网络平台,要求自动并行化技术也必须能够适应复杂的网络环境。现有自动并行化系统多面向同构系统,即假设并行/分布系统中各处理节点具有相同的处理能力而均匀划分数据;某些自动并行化系统即使支持异构系统,也需要事先知道异构系统中各处理节点的处理能力,然后再据此合理划分数据,生成相应的并行程序。因而,现有的自动并行化系统难以适应以异构和动态配置为特点的网络平台。

如果自动并行化系统支持数据自动迁移,即生成的并行程序能够根据计算节点的计算能力以及负载轻重,自动将数据迁移至计算能力强、负载轻的计算节点,那么不论计算平台是同构还是异构,不论各计算节点的负载轻重,并行程序都可以取得比较好的并行效率。

本文面向计算流体力学(CFD),研究了自动并行化系统支持的数据自动迁移技术。

1　CFD程序的特点

计算流体力学(Computational Fluid Dynamics,CFD)是一门利用计算机对各种类型的流体在各种速度范围内的复杂流动进行数值模拟的学科。流体的运动规律可由欧拉方程、N-S(Navier-Stokes)方程等数学方程组来描述,CFD则采用数值计算方法求解这些方程,研究流体运动特性,得出流体的空间流动规律、

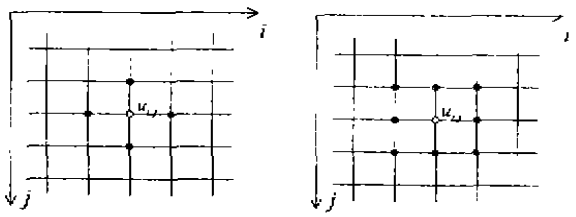
在CFD中,首先要把不规则的物理流场离散化为网格,然后把这个不规则的网格经过一定的变换映射到一个规则的网格(称为结构化网格)上,这个规则的网格就是计算区域。流场的状态由各种特征值来表述,如速度、密度、压力、温度等。CFD程序主要就是对网格点特征值进行大量迭代,当残值δ小于给定精度ε后,便得到流场在空间各点上的状态(图3)。常用的CFD的迭代算法有:点雅可比迭代、线雅可比迭代、高斯-赛德尔迭代、线高斯-赛德尔迭代、逐次超松弛迭代、线逐次超松弛迭代等。

在迭代过程中,对每个网格点进行计算时,通常只引用其周围几个网格点的值。根据对周围网格点的引用情况,可以把迭代方法分为五点格式和九点格式。以二维流场为例,用u_{ij}表示网格点(i,j)的特征值u,对于五点格式,有:

$$u_{i,j}=f(u_{i-1,j},u_{i,j-1},u_{i+1,j},u_{i,j+1},u_{i,j})$$

对于九点格式,有:

$$u_{i,j}=f(u_{i-1,j-1},u_{i,j-1},u_{i+1,j-1},u_{i-1,j},u_{i+1,j},u_{i,j+1},u_{i-1,j+1},u_{i+1,j+1},u_{i,j})$$



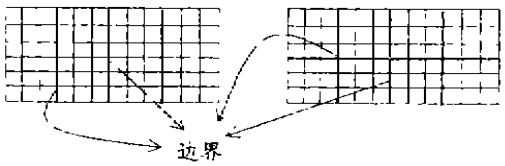
(a) 五点格式　　(b) 九点格式

图1　五点格式和九点格式

一般,并行化CFD程序需要先划分流场,常见的划分方式有一维划分、二维划分等(图2)。每个节点程

*)本研究受到国防科技重点实验室基金资助(编号:99JS94 6 1.HK0313)。

序只计算一部分网格，不同的节点程序通过交换边界网格数据共同完成整个流场网格的计算，每次迭代可分为两个阶段：计算阶段和通信阶段（图 4）。



(a)一维划分　　(b) 二维划分

图 2　一维划分与二维划分

2　数据自动迁移的 CFD 并行计算模型

对于一般的并行 CFD 程序，各节点程序负责计算的边界和网格数目都是固定的，一旦确定好数据的划分方法，各节点程序的数据量和计算量也都确定下来了。因而，对于一般的 CFD 并行计算模型而言，只有事先知道计算平台中各计算节点的相对计算能力，并且再据此划分数据，才有可能得到比较好的运行效率。

本文提出了支持数据自动迁移的 CFD 并行计算模型，简称 ADM 计算模型，其算法流程图如图 3。该模型中，划分的边界及各节点程序负责计算的网格数目都不是固定的，程序运行时根据各节点程序所在的计算节点的计算能力和负载轻重动态决定边界和网格数目，据此网格数据由计算能力弱、负载重的计算节点自动迁移到计算能力强、负载轻的节点。因而，不论计算平台是同构还是异构，不论各计算节点的负载轻重，并行程序都可以取得比较好的并行效率。

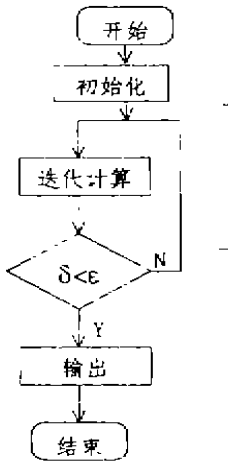


图 3　CFD 串行算法流程图

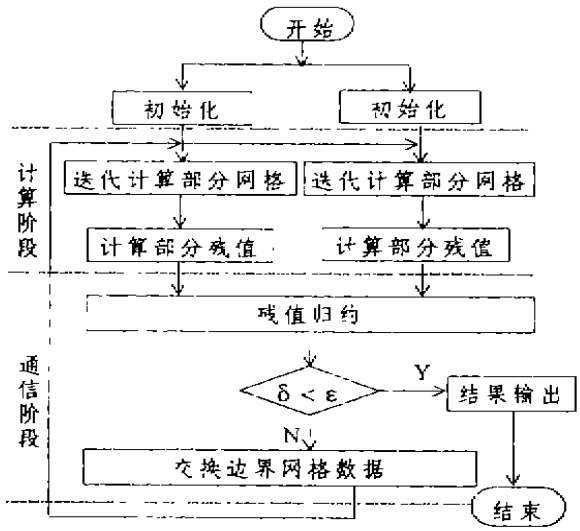


图 4　CFD 并行算法流程图

3　自动并行化系统对 ADM 计算模型的支持

与一般的自动并行化系统不同，支持 ADM 模型的自动并行化系统在程序生成部分必须考虑到边界的不确定性。例如：节点程序中循环的初值与终值必须根据当前的边界确定。对于一般的赋值语句，各节点程序也必须根据当前的边界决定是否执行。如将下面一段程序转化为两个节点程序组成的并行程序：

```
DIMENSION A(100)
DO I=1,100
   A[I]=…
END DO
A[20]= …
```

一般的自动并行化系统可将上述程序段并行化为：

```
DO I=1,50          DO I=51,100
   A[I]=…             A[I]=…
END DO             END DO
A[20]=…
节点程序 1          节点程序 2
```

而支持 ADM 模型的自动并行化系统须将上述程序段并行化为(BORDER 为动态边界)：

```
DO I=1,BORDER              DO I=BORDER,100
   A[I]=…                     A[I]=…
END DO                     END DO
IF (20.LT.BORDER) THEN     IF (20.GT.BORDER) THEN
  A[20]=…                    A[20]=…
END IF                     END IF
节点程序 1                  节点程序 2
```

此外，支持 ADM 的自动并行化系统与一般的自动并行化系统最大的区别在于需要加入代码以重新划分网格以及数据迁移的程序代码。

3.1　重新划分网格数据的判据

文[3]证明：对于 CFD 并行程序，当各节点程序负责计算的网格数目与相应计算节点的计算能力成正比时，并行程序的效率最高。这时所有计算节点在每次迭代中计算阶段所化时间相等。

因此，在ADM模型中，各节点程序（假设为N个）监测在计算阶段所花时间：$T_1, T_2, \cdots, T_n$。定义数据不平衡因子 $B=\mathrm{Max}(T_1, T_2, \cdots, T_N)/\mathrm{Min}(T_1, T_2, \cdots, T_N)$。若B超过一定的门限值，则需要重新划分网格数据，使得重新划分后的数据不平衡因子尽量接近于1。

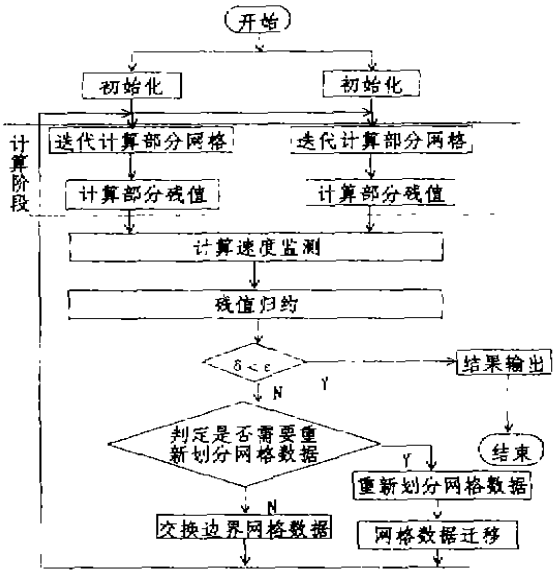


图5 ADM算法流程图

3.2 重新划分网格及数据迁移

不失一般性，假定计算问题是一个三维流场，方向分别为X、Y和Z，三个方向上流场的宽度分别为：WX、WY和WZ。现以一维划分和二维划分为例说明如何确定新的网格划分方法和需要迁移的数据。并假定并行化时，一维划分划分方向为X，二维划分划分方向为X和Y。

3.2.1 一维划分 假定并行程序包含N个节点程序 $P_1, P_2, \cdots, P_N$，分配给它们的网格数据的集合分别为 $G_1, G_2, \cdots, G_N$，在X方向上的宽度分别为 $WX_1, WX_2, \cdots, WX_N$（$\sum_{i=1}^{N} WX_i = WX$），在X方向上的左边界为 $XB_n = \sum_{i=1}^{n-1} WX_i$，右边界为 $XE_n = \sum_{i=1}^{n} WX_i$，在计算阶段所花时间分别为 $T_1, T_2, \cdots, T_N$。那么，在新的划分方法中，P_n 在X方向上的宽度应为：

$$WX'_n = \left(\frac{WX_n}{T_n} \times WX\right) / \left(\sum_{i=1}^{N} \frac{WX_i}{T_i}\right)$$

由于新划分方法中各节点程序在X方向上的宽度已经确定，因此可以确定新划分方法分配给各节点程序的网格数据集合 G'。因此：节点程序 P_n 应向节点程序 P_m 迁移的结点数据为 $G_n \cap G'_m$

3.2.2 二维划分 假定在X方向上划分数为M，Y方向上划分数为N，并行程序包含 $M \times N$ 个节点程序：$P_{11}, P_{12}, \cdots, P_{MN}$，分配给它们的网格数据的集合分别为 $G_{11}, G_{12}, \cdots, G_{MN}$。为了计算及通信的方便，规定 $P_{i1}, P_{i2}, \cdots, P_{iN}$ 在X方向上的宽度相同，为 WX_i（$\sum_{i=1}^{M} WX_i = WX$），在Y方向上的宽度可以不等，分别为 $WY_{i1}, WY_{i2}, \cdots, WY_{iN}$（$\sum_{j=1}^{N} WY_{ij} = WY$）。在计算阶段所花时间分别为 $T_{11}, T_{12}, \cdots, T_{MN}$，那么，重新划分数据后，节点程序 Pmn 在X方向上的宽度应为：

$$WX'_{mn} = \left(\sum_{i=1}^{N} \frac{WY_{mi} \times WX_{mi}}{T_{mi}} \times WX\right) / \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{WX_{ij} \times WY_{ij}}{T_{ij}}$$

在Y方向上的宽度应为：

$$WY'_{mn} = \left(\frac{WY_{mn}}{T_{mn}} \times WY\right) / \sum_{j=1}^{N} \frac{WY_{mj}}{T_{mj}}$$

由于新划分方法中各节点程序在X和Y方向上的宽度已经确定，因此可以确定新划分方法分配给各节点程序的网格数据集合 G'。因此，P_{mn} 需要向 P_{pq} 迁移的结点数据为 $G_{mn} \cap G'_{pq}$

在CFD程序中计算量基本与网格数目成正比。因此，重新划分网格后，对于一维划分各节点程序的计算时间均为 $\frac{WX}{\sum_{i=1}^{N} \frac{WX_i}{T_i}}$，对于二维划分，均为 $(WX \times WY) / \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{WX_{ij} \times WY_{ij}}{T_{ij}}$。显然，重新划分网格以及数据迁移后，从理论上数据不平衡因子B=1。

4 性能测试

本文测试了计算平台中，各计算节点计算能力差异对程序性能的影响，以及计算节点负载变化对程序性能的影响；测试的内容为：自动并行化系统在平均分配网格数据与数据自动迁移两种情况下生成的并行程序的运行时间 Ta 和 Tm；采用的测试程序采用高斯-赛德尔迭代法求解椭圆型方程，采用的计算平台为10M共享式以太网连接的多台微机，配置如下：

编号	CPU	主频(MHz)	内存(MB)
1	赛扬	500	128
2	赛扬	466	64
3、4	Pentium II	350	256
5	Pentium	166	16
6、7	Pentium	100	16

4.1 计算节点计算能力差异对程序性能的影响

下列测试中，在并行程序运行时系统中没有其他用户程序。测试一和测试二使用的微机计算性能相近，

因而 Ta 和 Tm 相差不大，而测试三中采用的微机计算性能差别较大，因而数据自动迁移大大缩短了并行程序运行的时间。

测试	使用微机	Ta(秒)	Tm(秒)
测试一	1、2	315	283
测试二	1、2、3、4	165	162
测试三	2、5、6、7	1015	421

4.2 计算节点负载变化对程序性能的影响

测试三使用微机 1、2，后台运行 CFD 计算程序，前台运行其他用户程序。测试得到：对于平均分配网格数据的并行程序，运行时间为 413 秒；对于数据自动迁移的并行程序，运行时间为 349 秒，由于采用的是在 X 方向划分，两个节点程序在 X 方向上的数据宽度随时间变化如图 5 所示。

图 6 数据宽度随时间的变化

结束语 采用 ADM 模型也会带来一定的额外开销。一般情况下由节点计算能力差异而导致的数据迁移会较大地提高并行程序的运行效率，并且额外开销较小；如果由于负载变化(特别是负载变化频繁时)导致较为频繁的数据迁移，额外开销较大，但是如果负载变化不很频繁，由于迭代过程中计算时间远远大于数据迁移所用的时间，其额外开销还是可以忽略的。

另外，ADM 模型不但适用于 CFD，对于数值天气预报等网格类计算程序也同样适用。

参 考 文 献

1 William B, et al. Parallel Programming with Polaris. IEEE Computer, 1996, 29(12): 78～82

2 Hall Mary W, et al Maximizing Multiprocessor Performance with the SUIF Compiler IEEE Computer, 1996, 29(12): 84～89

3 冯百明. 基于分区的自动并行化程序重构技术研究. [博士学位论文]. 西安：西北工业大学，1998

4 朱传琪，臧斌宇，陈彤. 程序自动并行化系统. 软件学报，1996，7(3)：180～186

5 金国华，陈福接. KD-PARPRO：一个基于知识的并行化工具-总体设计与功能描述. 软件学报，1993，4(6)：1～6

6 况正谦. 程序自动并行化的场循环相关分析与优化技术. [博士学位论文]. 西安：西北工业大学，1998

(上接第 36 页)

表 1 仿真数据

学习样本					测试样本				
No.	类	x1	x2	x3	No	类	x1	x2	x3
1	0	1	3	1	21	0	1	1	5
2	0	1	5	2	22	0	1	3	4
3	0	1	1	3	23	0	1	5	3
4	0	1	3	4	24	0	1	1	2
5	0	1	5	5	25	0	1	3	1
6	1	5	1	4	26	1	5	5	2
7	1	5	3	3	27	1	5	1	3
8	1	5	5	2	28	1	5	3	4
9	1	5	1	1	29	1	5	5	5
10	1	5	3	2	30	1	5	1	4
11	0	1	5	3	31	0	1	3	3
12	0	1	1	4	32	0	1	5	2
13	0	1	3	5	33	0	1	1	1
14	0	1	5	4	34	0	1	3	2
15	0	1	1	3	35	0	1	5	3
16	1	5	3	2	36	1	5	1	4
17	1	5	5	1	37	1	5	3	5
18	1	5	1	2	38	1	5	5	4
19	1	5	3	3	39	1	5	1	3
20	1	5	5	4	40	1	5	3	2

表 2 测试结果

类型	基于 CC 模型的神经网络	BP 神经网络
环境	Intel PII366/64RAM	
参数设置	3 个输入节点	3 个输入节点，2 个隐节点，1 个输出节点，最大允许误差 0.2
学习时间	0.02 秒	0.96 秒
学习结果	学习的结果是 CC 神经元层有两个节点，其参数为：节点 1：u = (1.0, 3.2, 3.4) r = 1.99 Class=0 节点 1：u = (5.0, 3.0, 2.4) r = 2.14 Class=1	得到 8 个连接参数和 3 个阈值参数
测试结果	正确率 100%	正确率 100%(输出大于 0.9 是 1，小于 0.1 是 0)

参 考 文 献

1 McCulloch G A, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematics Biophysics, 1943, 5: 115～133

2 张玲，等. 多层前馈神经网络的学习和综合算法. 软件学报，1995，6(7)：440～448

3 Tomas H. Modular Learning in Neural Networks. John Wiley & Sons, Inc., 1992

4 张玲，等. M-P 神经元的几何意义及其应用. 软件学报，1998，5：334～338

5 Takagi H, Hayashi I. NN-driven fuzzy reasoning. Int. Journal of Approximate Reasoning, 1991, 5: 191～213