

# 非均匀区域油藏模拟负载平衡的分区并行算法\*

舒继武<sup>1,2</sup> 赵金熙<sup>1,2</sup> 周维四<sup>3</sup> 张德富<sup>1,2</sup>

<sup>1</sup>(南京大学计算机软件新技术国家重点实验室 南京 210093)

<sup>2</sup>(南京大学计算机科学与技术系 南京 210093)

<sup>3</sup>(胜利油田地质科学研究所 东营 257000)

**摘要** 基于分布式并行计算机系统,对一类非均匀区域的油藏数值模拟问题,采用了区域分解方法并行求解,给出了并行求解的负载平衡模型及区域负载平衡的一种有效分区算法,从而将这类油藏数值模拟问题均衡映射到并行环境中,高效地并行求解.在黑油油藏数值模拟并行软件的研究中,实验结果表明,该算法有利于提高加速比.

**关键词** 油藏数值模拟,区域分解,负载平衡,并行算法.

**中图法分类号** TP301

油藏数值模拟问题,无不归结为求解大型偏微分方程的问题.这类问题的求解需要大量的计算时间,因此,如何提高求解这类问题的计算速度成为人们非常关心的问题.区域分解法是应运而生的、适应并行计算机工作原理的偏微分方程数值算法.它体现了“分而治之”的思想,受到计算数学界和工程界的热切关注.该方法从问题的“模型级”发掘并行性,将计算区域分割成若干独立的规模较小的子区域,将原问题的求解转化为各子区域上子问题的求解.一个或若干个子问题用一个处理机来处理,当所有的子问题均衡地映射到多个处理机上时,通过并行求解子问题而获得整个区域上大问题的解,它有利于问题的并行求解,提高了解题速度,具有广阔的应用前景.

区域分解并行计算,在结合并行机的特有体系结构时,要考虑区域的划分及负载平衡.文献[1]基于二维网格连接的分布式并行处理机结构,在给出几种区域分解方法的计算复杂性和通信耗费的基础上对其加速比和效率进行了分析.文献[2]对区域的网格划分与并行计算的效率进行了讨论,并寻找其最优的网格分裂.它们都假设区域均匀,区域划分为等大小的子区域,认为子区域有完全相同的计算量和通信要求.但在实际的问题中,如油藏模拟,油区经常是不均匀的,而在选定区域时,为了便于处理,常选定一规则区域,如长方形区域.若将区域划分成与处理机数目相同的等大小的子区域,各子区域的收敛速度就不同,对应分配到处理机上将引起处理机的负载不均衡.因此,要维持一种相对稳定的高效并行处理,需采用负载平衡的分区并行计算方法.本文以在油藏数值模拟中,运用区域分解方法并行求解非均匀区域的黑油压力问题为背景,给出了并行求解的负载平衡模型及区域负载平衡的一种分区算法.

## 1 区域分解并行算法

区域分解法主要有3类:迭代子结构类型,子区域间无重叠;Schwarz类型,子区域间有重叠;迭代子结构和Schwarz的结合类型.

下面以 Schwarz 类型的区域分解算法来说明其并行求解问题的基本步骤.设有一个模型为

\* 本文研究得到国家 853 高科技项目基金资助.作者舒继武,1969年生,博士生,主要研究领域为并行处理和分布式计算.赵金熙,1950年生,博士,副教授,主要研究领域为计算数学,数值计算.周维四,1942年生,教授级高工,主要研究领域为计算技术和油藏模拟.张德富,1937年生,教授,博士生导师,主要研究领域为计算机软件,并行处理,分布式计算.

本文通讯联系人:舒继武,北京 100084,清华大学计算机科学与技术系

本文 1997-11-13 收到原稿,1998-03-03 收到修改稿

$$\begin{cases} LP=C \frac{\partial P}{\alpha} & (\Omega) \\ L_1 P=g & (\partial\Omega) \end{cases}, \quad (1)$$

对区域  $\Omega$  进行分解:令  $\Omega$  有分解  $\Omega=\bigcup_{i=1}^m \Omega_i$ ,  $\Omega_i$  是开集,且对每个  $i$  都可找到  $j \neq i$ , 有  $\Omega_i \cap \Omega_j \neq \emptyset$ . 对问题(1)用平均法构造的并行 Schwarz 方法如下<sup>[3]</sup>:

步骤 1: 选择初始近似  $P^0$ ,  $n=0$ .

步骤 2: 并行计算子区域上边值问题

$$\begin{cases} LP_i^{n+1}=C \frac{\partial P_i^{n+1}}{\alpha}, & (\Omega_i) \\ L_1 P_i^{n+1}=g, & (\partial\Omega_i) \end{cases} \quad (2)$$

步骤 3: 延拓  $P_i^{n+1}$  的定义到式(3),即定义

$$\tilde{P}_i^{n+1}=\begin{cases} P_i^{n+1} & (\Omega_i) \\ P^n & (\Omega \setminus \Omega_i) \end{cases} \quad (3)$$

并取平均值

$$P_i^{n+1}=\frac{1}{m} \sum_{i=1}^m \tilde{P}_i^{n+1}, \quad (4)$$

置  $n=n+1$ ,若达到问题精度,则终止;否则,转步骤 2.

因此,对于一般的数值计算问题,应用区域分解法并行求解问题的基本过程为:

- 将给定的实际问题的求解区域,分割为几个子区域.
- 将各个子区域的求解,归结于解线性代数方程组.
- 给各个处理机分配数据和任务.
- 整个区域迭代循环:

do{

    处理机  $P_i$  上子区域的迭代循环(与其他处理机并行);

do{

    子区域迭代求解

  }while(未达到对应子区域的收敛条件)

    求子区域间边界节点(无重叠区域分解)或公共区域(重叠区域分解)

  }while(整个区域未达到其收敛条件)

- 整个区域收敛,得到问题的解.

## 2 负载均衡的分区算法

用区域分解方法求解,任务的完成是等待最慢结点机上的消息,这样,导致整体并行性能退化到与最慢结点机相同.为了解决这个问题,需要做到:① 要提高网络传输性能;② 要采用高效的算法,使得各子区域收敛速度最快;③ 要采用负载均衡的分区方法,使得每个结点机上完成的时间大致相等.本文主要考虑区域分解方法并行求解非均匀问题的负载均衡.

### 2.1 非均匀区域的划分

在非均匀区域的油藏数值模拟中,有的子区域由于包含更多的“死节点(即不需迭代计算的节点)”,达到收敛条件时迭代次数很少,而有的子区域由于包含很少的“死节点”,达到收敛条件时迭代次数很多.若划分的子区域数目大于处理机的个数,由于在开始迭代时,不能估计出其迭代次数,为了能较好地使负载均衡,子区域的数目必须远大于处理机的个数,这样,由于划分的子区域太多,整个区域收敛速度减慢.这里划分的子区域数目等于处理机的个数,并在区域迭代求解过程中,根据本处理机上的局部状态信息和整个区域的全局状态来动态地改变区域的划分,从而达到任务最终能均衡地运行的目的.

设处理机集为:  $P = \{P_1, P_2, P_3, \dots, P_m\}$ ,  $P_i$  为处理机  $i$ , 处理机数为  $m$ .  $\Omega$  为一油藏模拟区域, 分解为  $m$  个子区域, 令  $\Omega$  有分解  $\Omega = \bigcup_{i=1}^m \Omega_i$ , 即子区域集  $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_m\}$ ,  $\Omega_i$  为第  $i$  个子域.

并行求解的每一迭代涉及到区域的并行迭代和负载平衡算法, 在  $P$  个处理机上完成  $\max\_iter$  次迭代的总的执行时间<sup>[4]</sup>为

$$T_{total}^{(k)} = \sum_{k=1}^{\max\_iter} \max_{\text{all } i} [T_{app,i}^{(k)}(k) + T_{bal,i}^{(k)}(k)], \quad (5)$$

其中  $k$  为迭代步,  $i$  为处理机  $P_i$  ( $i=1, 2, \dots, P$ ),  $T_{app,i}^{(k)}$  表示处理机  $P_i$  在第  $k$  次迭代步的执行时间,  $T_{bal,i}^{(k)}$  表示处理机  $P_i$  在第  $k$  次迭代后负载平衡算法的时间. 并行迭代过程中, 计算和同步通信是交替进行的, 而处理机  $P_i$  一次迭代的时间受 3 个因素的影响.

$$T_{app,i}^{(k)}(k) = f(t_{comp}, t_{comm}, t_{id}), \quad (6)$$

其中  $t_{comp}$  是一次迭代的计算时间,  $t_{comm}$  是迭代通信时间,  $t_{id}$  是同步等待时间. 在迭代收敛的情况下, 区域的划分应以总的执行时间尽量短为目标.

## 2.2 并行计算的几点说明

根据上述分析, 对油藏数值模拟中区域分解并行计算作如下几点说明:

(1) 油区是不均匀的, 而选定的计算域是一规则的区域, 如长方形区域. 子区域网格离散化相同, 各子域的求解方法可以不同, 各子区域的收敛条件也可以不同.

(2) 在并行计算环境中, 网络的带宽会由于环境中其他用户的介入或因本身任务而发生动态的变化, 结果是消息通过网络延迟发生变化, 这里不考虑网络传输性能及结点性能的变化. 同一处理机上的子区域之间的通信代价为 0, 不在同一处理机上的子区域之间的通信代价只与其相邻区域的边界大小或公共区域大小有关.

(3) 处理机上计算和通信不重叠.

(4) 区域分解并行迭代时, 整个区域的每一迭代步包括所有子区域的多次迭代, 直至分别达到各子区域的收敛条件. 整个区域的迭代步相对于各子区域的迭代步少得多. 这里将每一个子区域的每一次收敛到该子区域的收敛条件的迭代过程视为一个任务, 任务迁移使得相邻子区域间的边界移动, 引起子区域大小的变化, 从而将一部分迭代过程迁移到其他处理机上. 本文将不加区分地使用子区域的每一次迭代收敛过程和任务.

(5) 在实际应用中, 处理机的数目一般较少, 而油藏模拟的区域一般很大, 划分后的子区域的计算量和收敛速度与该子区域需计算的节点数、每次的迭代初值、所用算法等有关.

(6) 处理机间的连接是线性连接, 计算区域是长方形(体)区域, 区域的划分与处理机连接一一对应, 子区域的数目与处理机的数目相等. 每个处理机只与相邻的处理机交换数据.

## 2.3 负载平衡模型及分区算法

设整个区域的迭代步为  $k=1, 2, \dots$ , 在每一迭代步后, 可根据处理机的信息来启动负载平衡算法, 负载平衡算法涉及的时刻可以用  $k$  表示.

定义 1.  $w_i(k)$ : 并行系统在  $k$  时刻的状态-任务的分布向量.  $w_i(k)$ : 处理机  $P_i$  在区域第  $k$  步迭代后的任务量, 则时刻  $k+1$  的任务分布包括:  $k$  时刻的任务分布、负载平衡引起的任务传递. 处理机  $P_i$  的负载平衡模型可表述为

$$w_i(k+1) = w_i(k) - w_{i,t,d_s}(k) + w_{i,t,d_r}(k), \quad (7)$$

其中  $w_{i,t,d_s}(k)$ ,  $w_{i,t,d_r}(k)$  分别表示负载平衡引起的发送和接收的任务.

定义 2.  $A(k)$ : 处理机参与矩阵, 对于  $P$  个处理机的并行系统, 它是一个  $p \times p$  的矩阵, 每个处理机只能与其相邻的处理机有任务的重新分配. 如果处理机  $P_i$  和  $P_j$  参与任务迁移, 则对应元素置为“1”, 否则, 对应元素置为“0”.

定义 3.  $M(k)$ : 描述任务的迁移, 它是一个  $p \times p$  的矩阵. 每个处理机只能与其相邻的处理机有任务的迁移.  $M_{ij}$ : 处理机  $P_i$  向  $P_j$  迁移任务的因子, 如  $P_i$  送给  $P_j$  的任务可表示为  $M_{ij}(k) \cdot w_i(k)$ .

定义 4.  $H$ : 允许任务迁移的路径矩阵.  $H_{ij}$ : 如果处理机  $P_i$  允许向  $P_j$  迁移任务, 则对应元素置为“1”, 否则,

对应元素置为“0”. 因此,  $M_{ij} \neq 0$  仅当  $H_{ij} = 1$ .

根据上述的定义, 式(7)表述为

$$w_i(k+1) = w_i(k) - \sum_{j \neq i} M_{ij}(k) A_{ij}(k) w_j(k) + \sum_{j \neq i} M_{ji}(k) A_{ji}(k) w_j(k). \tag{8}$$

上式中, 第1个和项表示从处理机  $P_i$  到其相邻处理机的任务迁移; 第2个和项表示从  $P_i$  的相邻处理机到  $P_i$  的任务迁移. 定义:  $M_{ii}(k) = -\sum_{j \neq i} M_{ij}(k)$ , 则负载平衡模型可以表述为

$$w(k+1) = w(k) + [M(k)]^T A(k) w(k). \tag{9}$$

定义 5.  $\bar{w}$ : 处理机的理想任务分布.

$$\bar{w} = \frac{1}{m} \sum_{all p} w_i(k).$$

定义 6.  $\tilde{w}$ : 处理机  $P_i$  的相邻处理机任务分布的平均信息.

$$\tilde{w} = \frac{1}{d} \sum_{i \in R_i} w_i(k),$$

其中  $d$  为  $P_i$  的相邻处理机数,  $R_i$  为  $P_i$  的相邻处理机集.

$\bar{w}$  和  $\tilde{w}$  可分别作为处理机任务分布的全局信息和局部信息.

根据上述定义, 将处理机  $P_i$  第  $k$  次迭代完成的时间  $T_{app,i}^{(k)}$ , 记为其在  $k$  时刻的任务量  $w_i(k)$ , 采用负载平衡的分区并行算法可表述为:

- (1) 将区域均分为子区域  $\Omega_1, \dots, \Omega_m$ , 并对应地分配到处理机  $P_1, \dots, P_m$  上. 令初始集为  $S = \{1, 2, \dots, m\}$ .
- (2) 应用区域分解并行算法, 各子区域在处理机上并行执行, 直至达到各子区域的收敛条件; 分别统计其迭代完成时间  $T_{app,i}^{(k)}$ , 记为  $w_i(k)$ .
- (3) 求解子区域间的边界节点或公共子区域.
- (4) 如果整个区域达到其收敛条件, 则终止; 否则, 转(5).
- (5) 从  $S$  中选择满足  $\tilde{w}_i(k) = \max\{\tilde{w}_p(k), p \in S\}$  的处理机  $P_i$  及处理机  $P_j, P_j$  满足  $\tilde{w}_j(k) = \min\{\tilde{w}_p(k), p \in S\}$ . 如果  $\tilde{w}_i - \tilde{w}_j \leq \alpha$  ( $\alpha$  为处理机间容许的负载差), 转(2); 否则, 转(6).
- (6) 在  $P_i$  的相邻处理机集中选择  $P_j$ , 满足  $\tilde{w}_j(k) = \min\{w_p(k), p \in R_i\}$  ( $\Omega_i$  的相邻区域集记为  $\Omega_R$ ), 如果  $\tilde{w}_i - \tilde{w}_j > \alpha$ , 处理机  $P_i$  的部分任务需迁移到  $P_j$  上, 即需在域  $\Omega_i$  与其邻域  $\Omega_j$  之间重新划分区域.  $P_i$  迁移到  $P_j$  的任务量约为:  $w_i(k) - \bar{w}_i(k)$ ; 否则, 转(7).
- (7)  $S = S - \{i\}$ , 如果  $S \neq \emptyset$ , 转(5); 否则, 重置  $S = \{1, 2, \dots, m\}$ , 转(2).

### 3 计算实例

在黑油数值模拟并行化软件开发的研究中, 使用了区域分解并行计算, 获得了较好的效果. 下面是一油藏模拟中黑油压力问题模型.

$$\begin{cases} (B_o - R_{oc} B_g) [\nabla T_o (\nabla P_o - \nabla \gamma_o Z) - \frac{q_w}{\rho_{osc}}] + (B_w - R_{ow} B_g) [\nabla T_w (\nabla P_o - \nabla \gamma_w Z - \nabla P_{cow}) - \frac{q_w}{\rho_{osc}}] \\ + B_g [\nabla T_g (\nabla P_u - \nabla \gamma_g Z - \nabla P_{cgo}) + \nabla R_{so} T_u (\nabla P_o - \nabla \gamma_o Z) + \nabla R_{sw} T_w (\nabla P_o - \nabla \gamma_w Z - \nabla P_{cow}) - \frac{q_s}{\rho_{osc}}] \\ = \Phi C, \frac{\partial P_o}{\partial t} \quad (\Omega) \\ P(x, y, z, 0) = P^{(0)} \\ L_i P_o = f \quad (\partial\Omega) \end{cases} \tag{10}$$

其中  $C_o = -\frac{1}{B_o} \frac{\partial B_o}{\partial P_o} + \frac{B_g}{B_o} \frac{\partial R_{so}}{\partial P_o}, \quad C_w = -\frac{1}{B_w} \frac{\partial B_w}{\partial P_o} + \frac{B_g}{B_w} \frac{\partial R_{sw}}{\partial P_o}, \quad C_g = -\frac{1}{B_g} \frac{\partial B_g}{\partial P_o}, \quad C_r = \frac{1}{\Phi} \frac{\partial \Phi}{\partial P_o}.$

$$C_i = C_o + C_w S_o + C_g S_g, \quad P_{cow} = P_o - P_w, \quad P_{cgo} = P_g - P_o, \quad S_o + S_w + S_g = 1$$

这里,下标  $o, w, g$  分别为油、水、气相,下标  $sc$  为地面标准条件,  $B$  为体积系数,  $R_{so}$  为溶解气油比,  $q$  为注入或采出的体积流量,  $S$  为饱和度,  $P$  为压力,  $P_i$  为毛管压力,  $\Phi$  为孔隙度,  $\rho$  为密度,  $L_i$  为边界条件算子,  $t$  为时间,  $f$  是关于时间的压力函数。

对于上述模型的这种大型边值问题,如应用迭代子结构类型的区域分解方法,其离散模型从总体上说都可化为一个系数矩阵加边对角块的线性方程组。将区域  $\Omega$  分成互不相交的  $m$  个子区域  $\Omega_i, i=1, 2, \dots, m$ , 若用步长为  $h$  的正方形网格,用 7 点差分格式离散方程,可得到如下线性方程组<sup>[5]</sup>。

$$Ap = \begin{bmatrix} A_{1,1} & & 0 & & A_{1,m+1} \\ & \ddots & & & \vdots \\ & & 0 & & A_{m,m+1} \\ A_{m+1,1} & \dots & A_{m+1,m} & & A_{m+1,m+1} \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_m \\ p_{m+1} \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_m \\ f_{m+1} \end{bmatrix}, \quad (11)$$

式中  $A_{1,1}, A_{2,2}, \dots, A_{m,m}$  分别对应于子区域  $\Omega_1, \Omega_2, \dots, \Omega_m$  上的矩阵。式(10)的求解可运用迭代法,以前一阶段的近似解作为下一阶段的初始量,使整个问题收敛。对每一迭代步中的每个子区域问题  $A_{j,j} p_j = f_j - A_{j,m+1} p_{m+1}$  ( $j=1, 2, \dots, m$ ) 的求解,可以对应地分配到  $m$  个处理机上并行求解,在每一迭代步后,可以根据负载平衡算法,重新划分区域,形成新的方程组

$$A'p = \begin{bmatrix} A'_{1,1} & & 0 & & A'_{1,m+1} \\ & \ddots & & & \vdots \\ & & 0 & & A'_{m,m+1} \\ A'_{m+1,1} & \dots & A'_{m+1,m} & & A'_{m+1,m+1} \end{bmatrix} \begin{bmatrix} p'_1 \\ \vdots \\ p'_m \\ p'_{m+1} \end{bmatrix} = \begin{bmatrix} f'_1 \\ \vdots \\ f'_m \\ f'_{m+1} \end{bmatrix}. \quad (12)$$

对新形成的子区域问题:  $A'_{j,j} p'_j = f'_j - A'_{j,m+1} p'_{m+1}$  ( $j=1, 2, \dots, m$ ), 仍对应在  $m$  个处理机上并行求解。这样,经过负载平衡算法,各处理机的任务较先前的均衡。计算实例为:区域  $x, y, z$  方向网格数分别为 170、50、5, 节点规模为 42 500 的上述黑油模型,沿平行于  $z$  方向剖分区域,该区域中非均匀地分布着许多“死节点”。运用区域分解法并行计算 30 个阶段(时间段)。区域收敛判别为:  $\|P_{i+1} - P_i\|_{\infty} \leq 10^{-4}$ ,  $P_{i+1}, P_i$  分别为整个区域的第  $i+1$  迭代步和第  $i$  迭代步的压力。各子区域采用 LU 预处理广义极小剩余法(GMRES( $m$ ))求解,取 GMRES( $m$ )算法中的截断参数  $m$  为 10, 各子区域收敛判别为  $\|p_{m+1} - p_m\|_{\infty} \leq m * 10^{-4}$ ,  $p_m, p_1$  分别为解子区域的 GMRES( $m$ )算法的第  $m$  迭代步和第 1 迭代步压力。计算环境是用 PVM 通过以太网分别连接 2 个、3 个、4 个、5 个 SGI indy 构成几种分布式并行计算系统。图 1 给出了该黑油数值模型采用负载平衡方法和无负载平衡时的加速比的比较。实验结果表明,采用负载平衡方法能使系统并行加速比提高 10%~40%。如对 4 个处理机的并行系统在无负载平衡时,上述模型计算 30 个阶段的加速比为 2.51,采用本文的负载平衡分区算法后其加速比为 3.25,加速比提高 29.5%。

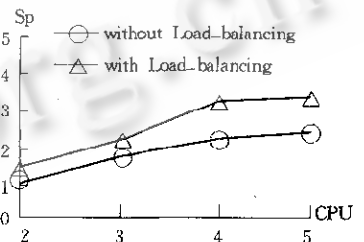


图1 加速比比较

### 参考文献

- Hoffmann K H, Zou Jun. Parallel efficiency of domain decomposition methods. *Parallel Computing*, 1993, 19(12): 1375~1391
- Chan T F, Shao J P. Parallel complexity of domain decomposition methods and optimal coarse grid size. *Parallel Computing*, 1995, 21(7): 1033~1049
- 吕涛, 石济民, 林振宝等. 区域分解法——偏微分方程数值解新技术. 北京: 北京科学技术出版社, 1992 (Lu Tao, Shi Ji-min, Lin Zhen-bao et al. *Domain Decomposition Methods—New Technology for Numerical Solver of Partial Differential Equations*. Beijing: Beijing Science and Technology Press, 1992)
- Franklin M A, Govindan V. A general matrix iterative model for dynamic load balancing. *Parallel Computing*, 1995, 22(7): 969~989

- 5 赵金熙, 舒继武, 张德富等. 大型对称不定箭形线性方程组的分解方法. 高等学校计算数学学报, 1997, 19(1): 83~92  
(Zhao Jin-xi, Shu Ji-wu, Zhang De-fu *et al.* A direct method for solving symmetric indefinite arrow linear system. Numerical Mathematics, A Journal of Chinese Universities, 1997, 19(1): 83~92)

### Zonal Parallel Algorithm of Load-balancing on Nonuniform Domain Simulation of Reservoir

SHU Ji-wu<sup>1,2</sup> ZHAO Jin-xi<sup>1,2</sup> ZHOU Wei-si<sup>3</sup> ZHANG De-fu<sup>1,2</sup>

<sup>1</sup>(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

<sup>2</sup>(Department of Computer Science and Technology Nanjing University Nanjing 210093)

<sup>3</sup>(The Academy of Geological Sciences of Shengli Oil-field Dongying 257000)

**Abstract** Based on the distributed parallel environment, for one of the nonuniform domain numerical simulation problems of reservoir, the authors give a load-balance model of parallel computing and a zonal parallel algorithm of load-balance using domain decomposition method, so some well and effective solutions which balance map the nonuniform domain numerical simulation of reservoir to parallel environment are found. In the study of block oil reservoir numerical simulation parallel software, the practical results show that the zonal parallel algorithm can improve the performance in speedup.

**Key words** Numerical simulation of reservoir, domain decomposition, load-balancing, parallel algorithm  
© 中国科学院软件研究所 <http://www.jos.org.cn>