

基于多层油藏问题负载均衡的并行任务划分*

舒继武^{1,2} 赵金熙^{1,2} 周维四³ 张德富^{1,2}

¹(南京大学计算机软件新技术国家重点实验室 南京 210093)

²(南京大学计算机科学与技术系 南京 210093)

³(胜利油田地质科学研究院 东营 257000)

摘要 该文基于分布式并行计算机系统,对一类多层二维二相流油藏数值模拟问题给出了 3 种任务划分策略——“卷帘”方式、区域分解方式和“卷帘”与区域分解结合的方式,对它们进行了比较,提出了减少求解时间、利于负载均衡和提高并行性能的任务划分方法,并实际应用于有多达 72 万个网格节点的大规模油藏模拟问题.实算结果表明,该策略划分产生的并行求解任务均衡,有利于加速比的提高.该方法也适用于区域或数据并行的任务划分问题.

关键词 并行计算,负载均衡,任务划分,油藏数值模拟.

中图法分类号 TP319

油藏模拟需要大量的计算时间,并行计算机的使用对其产生了强有力的推动作用^[1].国外的许多石油公司用并行处理来降低生产成本,提高生产率,国内的一些石油公司也正在投资使用并行计算机进行油藏模拟,以提高处理速度,缩短处理时间.国内外已有相关的工作报道.文献[2]利用分布式并行计算机(ipcs/860)求解了一个两种组分的油藏模拟问题.文献[3]给出了将区域分解法和多重网格技术相结合来解决油藏模拟计算问题的一种方法.文献[4]基于 Transputer 并行系统实现了多层二维二相流油藏模拟的并行试验.在油藏数值模拟中,多层二维二相流油藏数值模拟应用非常广泛.本文以此类问题的并行计算作为讨论背景.它具有在不同时间步任务大小不同的特殊性.图 1 表示它在一个时间步的串行计算流程(KC 表示需计算的油层数).流程中的主要计算时间是花在解压力方程组上.在每一时间步一般有 1 个或多个油层需要计算;在不同的时间步,需计算的油层数 KC 有时是变化的,有时甚至相差较大.目前,此类问题的并行处理一般都采用这种按油层并行的策略,也有的按拟三维问题采用区域分解方法.文献[4]利用每个油层压力方程组求解的相互独立性,按“卷帘”方式将油层映射到多处理机并行环境中并行求解,但是由于在不同的时间步,层数是变化的,有时层数小于或远小于处理机数时,处理机负载很不均衡,处理机得不到充分利用,也未能充分挖掘问题的并行性,其加速比不高;另外,在每一时间步解饱和度方程组需各层压力值,问题求解的完成是等待最慢结点机上的压力数据,这就导致整体并行性能退化到与最慢结点机相同.本文通过比较几种任务划分策略,提出了一种减少求解时间、有利于负载均衡和提高处理机性能的任务划分策略(并行方案 3).该策略划分的并行任务比文献[3,4]的两种并行方案负载均衡,并行加速比高,并且在大规模油藏模拟实际生产

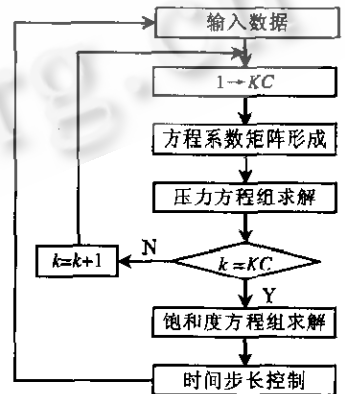


图 1 压力串行计算过程

* 本文研究得到国家 863 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,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,博士生导师,主要研究领域为计算技术,油藏数值模拟.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,并行处理技术,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 1998-05-12 收到原稿,1998-10-09 收到修改稿

应用中取得了良好的效果.

1 任务划分对并行性能的影响

在并行计算中,任务的划分应与求解问题的规模和处理机的结构密切相关,它极大地影响到能否有效地发挥并行机的性能和减少求解时间,尤其是对求解任务大小发生变化的问题.对于一个给定的并行环境,其并行计算机结构类型和处理机数是确定的,加速比将取决于:① 任务的均衡;② 各处理机之间的同步和通信耗费;③ 任务的执行速度.这3点是相互关联的.

减少通信时间有两方面的途径:一方面是提高通信速率,减少通信延迟;另一方面是减少通信数据量.前者与系统的硬件结构、性能及操作系统有关,后者与并行粒度有关.并行粒度也是影响负载均衡的一个因素,选择合适的并行粒度是设计并行处理程序、改善并行处理性能的重要途径,它可以大大减少问题的求解时间.

根据 Amdahl 定理可知,高并行度可以提高并行性能.在决定并行加速实际可达到的并行度的诸多因素中,数据规模和数据本身取决于具体的问题,是我们所不能控制的.从改进代码结构这一角度来讲,细化计算粒度可以做到提高算法的并行度,从而提高并行系统的性能.

细化计算粒度使得系统中实际参与分配的任务数变多,易使负载均衡.但在细化计算粒度的同时,将引起通信量的增加.通信对并行计算来说是一种额外开销,对性能将产生负面的影响.在大规模并行系统中,当采用细化计算粒度的并行时,系统性能有时会因为计算粒度细化而受益,同时又因为通信开销随之增大而被部分或全部抵销,甚至反而会导致性能下降,因此必须在并行度与粒度间充分地进行折衷并尽可能地使负载均衡.

并行求解时间一般为参与并行计算的处理机中运行时间的最大值.这里定义该问题的一个时间步并行求解的加速比 S_p 为

$$S_p = \frac{T_1}{T_p} = \frac{T_1}{t_{calc} + t_{comm} + t_d} \tag{1}$$

其中 T_p 表示所讨论的并行算法使用 p 台处理机的运行时间, T_1 表示该算法在单处理机上的运行时间, t_d 表示分配数据给结点机和从结点机接收结果数据所需的时间, t_{comm} 表示处理机之间通信和同步所需时间, t_{calc} 表示在一个时间步最慢结点机的计算时间.

2 划分策略

2.1 并行方案1及其实现

利用该问题“每一时间步解每层压力方程组是相互独立的”这一特性,按层“卷帘”方式分配层数据到各处理机.设有 p 个处理机,在某一时间步内,有 k 层需要计算,那么在第 i 个处理机上分配的层集为 $R_i = \{k|(k \bmod p) = i\} (i=0, \dots, p-1)$.如图 2 所示为 p 个处理机上的并行求解过程.

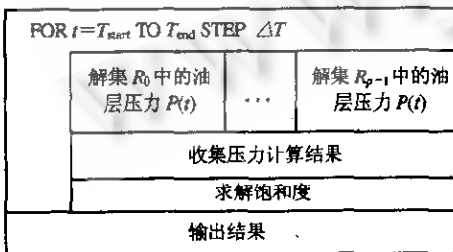


图 2 p 个处理机并行计算过程

在这种并行方案中,是以油层为并行计算粒度,是粗粒度并行.其主要优点是每一时间步解压力方程组没有处理机之间的通信和同步开销,只有分配数据给结点机和从结点机接收结果数据所需的时间.其主要不足是:在每一时间步,需计算的层数不一定相同,容易使处理机的负载不均衡,并且在每一次时间步,饱和度的求解需要等待最慢结点机上的压力数据,这样可导致整体并行性能退化到与最慢结点机相同,从而使加速比下降.尤其是在某些时间步,当计算层数为 1 时退化为串行处理.对该并行方案,式(1)可变为

$$S_p = \frac{T_1}{T_p} = \frac{\sum_{j=1}^k t_j}{\max_{i=0, \dots, p-1} \left(\sum_{j \in R_i, j=1, \dots, k} t_j \right) + t_d} \quad (2)$$

其中 t_j 为第 j 层的计算时间,在确定的时间步,分配数据给结点机和从结点机接收结果数据所需的时间 t_d 是一定的,因此要使其 S_p 最大,必须有

$$\min \left\{ \max_{i=0, \dots, p-1} \left(\sum_{j \in R_i, j=1, \dots, k} t_j \right) \right\} \quad (3)$$

对于这种粒度划分方法,并行粒度大,对于在不同时间步计算层数变化或在某些时间步计算层数小于处理机数的问题,任务分布难以均匀,处理机也未得到充分利用,加速比不高,并行效率低.对有些模型,加速比甚至很低.目前采用的多是这种以层为并行粒度^[4]的方法.

2.2 并行方案2及其实现

设有 p 个处理机,在某一时间步,有 k 层需计算,每一层采用区域分解法,将每一层对应的计算区域分为 p 个子区域(任务),仍按“卷帘”方式分配子区域数据到各处理机,则在第 i 个处理机上分配的子区域集为所有 k 层的第 i 个子区域组成的集合,即 $Q_i = \{q | (q \bmod p) = i\}$,其中 $i=0, \dots, p-1, q=1, \dots, k \times p$.

该方法细化了方法 1 的并行粒度,它是以油层的子区域为并行计算粒度,有利于处理机负载均衡,特别是当每层划分的子区域数与处理机数相等时,每一处理机分配的子区域数相等,负载易于均衡.对于实际的油藏模拟问题,由于计算的区域一般都是大范围的,以子区域作为并行计算粒度仍是粗粒度并行,可以认为是方案 1 的粒度的细化.图 3 表示,在某一时间步,每一层计算区域划分为 p 个子区域,在 p 个处理机上并行求解的过程.

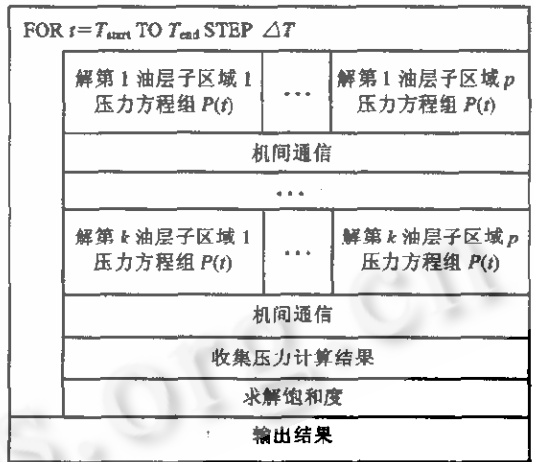


图 3 p 个处理机并行计算过程

对于实际问题,区域分解法的通信主要是相邻边界数据交换,其通信量不大.特别是对大规模问题,细化并行粒度改善负载均衡时缩短的时间大于通信增加的时间.对于某一时间步,当计算层数 $k <$ 处理机数 p 时,尤其是 $k \ll p$ 时,该方法与第 1 种方法比较, t_{comm} 的开销一般要小于由于负载不平衡所引起的同步等待开销时间,此时细化第 1 种方法的并行粒度可以获得更高的加速比.但在 $k \gg p$ 时, t_{comm} 随着计算层数的增大而增大,会成为影响加速比的重要因素,此时可能没有第 1 种方法的加速比高.对于该并行方案,式(1)可变为

$$S_p = \frac{T_1}{T_p} = \frac{\sum_{j=1}^k t_j}{\max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=1, \dots, k \times p} t_q \right) + \sum_{i=1}^k t_s^i + t_d} \quad (4)$$

其中 t_q 为子区域 q 的计算时间, t_s^i 为第 i 层区域间通信同步时间.要使式(4)中的 S_p 最大,需有

$$\min \left\{ \max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=1, \dots, k \times p} t_q \right) + \sum_{i=1}^k t_s^i \right\} \quad (5)$$

2.3 并行方案3及其实现

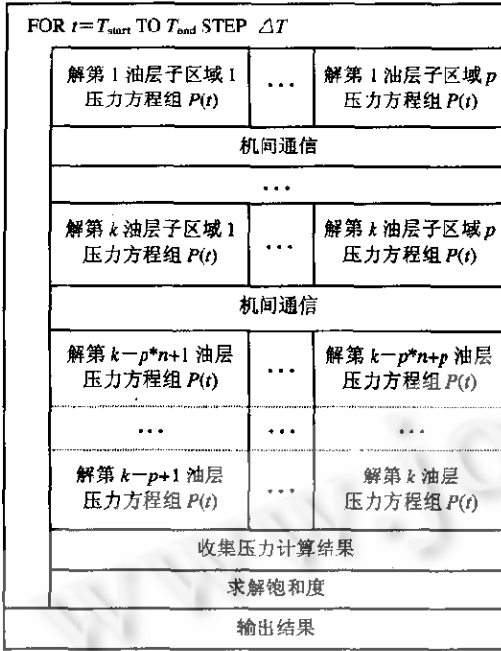


图 4 p 个处理机并行计算过程

令 $n = \lfloor \frac{k}{p} \rfloor$, 第 1 层~第 $k-p*n$ 层, 采用方法 2 划分任

务, 每层划分为 p 个子区域(任务), 共有 $p*(k-p*n)$ 个任务, 有 $k-p*n$ 次机间通信同步. 按子区域“卷帘”方式分配数据块到 p 个处理机, 即 $Q_i = \{q | (q \bmod p) = i, i=0, \dots, p-1, q=1, \dots, (k-p*n) \times p\}$, 这是以子区域为并行粒度. 第 $k-p*n+1$ 层~第 k 层, 采用方法 1 来划分任务, 即以层为粒度的共有 $p*n$ 个任务, 按层“卷帘”方式分配层数据到 p 个处理机, 即 $R_i = \{j | (j \bmod p) = i, i=0, \dots, p-1, j=k-p*n+1, \dots, k\}$. 如图 4 所示为在 p 个处理机上的并行实现过程. 该方法综合了方法 1 和方法 2, 同时采用了两种不同的并行计算粒度——油层和油层的子区域, 既易于负载平衡, 又减少了通信同步, 它克服了上述两种方法的不足, 而保留了它们的优点.

当 $k \ll p$ 时, 方法 3 是采用区域分解方法来分割区域并均匀地映射到处理机上, 各处理机的负载也基本平衡. 由于 k 很小, 通信同步也开销不大. 方法 2 可看成是此种情况的特例. 当 $k \gg p$ 时, 方法 3 综合了方法 1 与方法 2, 各处理机的负载基本平衡.

这种方法也存在通信开销的问题, 只是它比方法 2 的通信开销要小, 有时甚至小很多, 因此与方法 2 相比有较高的加速比. 对该并行方案, 式(1)可变为

$$S_p \approx \frac{T_1}{T_p} \approx \frac{\sum_{j=1}^k t_j}{\max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=1, \dots, (k-p*n) \times p} t_q + \sum_{j \in R_i, j=k-p*n+1, \dots, k} t_j \right) + \sum_{i=1}^{k-p*n} t_s^i + t_d} \quad (6)$$

对于在不同的时间步计算层数变化的问题, 该方法总能使各处理机的负载基本平衡, 因此比方法 1 和方法 2 更有效. 要使式(6)中 S_p 最大, 必须有

$$\min \left\{ \max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=1, \dots, (k-p*n) \times p} t_q + \sum_{j \in R_i, j=k-p*n+1, \dots, k} t_j \right) + \sum_{i=1}^{k-p*n} t_s^i \right\} \quad (7)$$

比较式(2)和式(6)可知, 当

$$\max_{i=0, \dots, p-1} \left(\sum_{j \in R_i, j=1, \dots, k-p*n} t_j \right) - \max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=1, \dots, (k-p*n) \times p} t_q \right) > \sum_{i=1}^{k-p*n} t_s^i \quad (8)$$

时, 方法 3 比方法 1 有效. 这表明采用细化一部分任务的计算粒度使得任务均衡, 系统性能因计算粒度的细化而提高. 比较式(4)和式(6)可得, 当

$$\max_{i=0, \dots, p-1} \left(\sum_{q \in Q_i, q=(k-p*n) \times p+1, \dots, k \times p} t_q \right) - \max_{i=0, \dots, p-1} \left(\sum_{j \in R_i, j=k-p*n+1, \dots, k} t_j \right) > \sum_{i=k-p*n+1}^k t_s^i \quad (9)$$

时, 方法 3 比方法 2 有效. 这表明完全细化计算粒度引起通信量的增加, 增大的通信开销导致性能下降, 因此必须在并行度与粒度间进行充分折衷并尽可能使负载平衡.

在实际应用中使用 PVM 并行环境,为了减少通信开销,可采用集中通信方式,减少通信次数,从而减少通信启动时间,缩短 t_{comm} 。另外,每一结点机的通信开销可以分成两部分:一部分是接收来自其他处理机的相邻边界数据;另一部分是发送本结点机的相邻边界数据的过程,PVM 发送采用异步过程,基本可以与计算重叠进行,因此,通信开销主要体现在接收方面。

3 数值结果

取压力方程组的收敛判别为 $\|p^{i+1} - p^i\|_2 \leq 5 \times 10^{-3}$, p^{i+1} , p^i 分别为第 $i+1$ 和第 i 迭代步的压力。设 N_x, N_y 分别表示油层 X, Y 方向上的网格划分, N_z 表示油层数。选取的实际油藏模型有以下几个。模型 1: $N_x \times N_y \times N_z$ 为 $142 \times 75 \times 12$, 节点数为 127 800, 每个阶段 N 有 $MP=8$ 个时间步。模型 2: $N_x \times N_y \times N_z$ 为 $146 \times 125 \times 16$, 节点数为 292 000, 每个阶段 N 有 $MP=8$ 个时间步。模型 3: $N_x \times N_y \times N_z$ 为 $211 \times 203 \times 17$, 节点数为 728 161, 每个阶段 N 有 $MP=23$ 个时间步。这 3 种模型在 $N=1 \sim 17$ 个阶段的计算层数 k 如表 1 所示。在实际应用中,为了提高并行效率,采用压缩通信技术,并行计算环境是通过 PVM 用以太网连接两个基于共享主存的 SGI Challenge 并行机组成的,每个 Challenge 有两个处理器。采用上述 3 种粒度划分方法对这 3 种模型分别求解,方法 2 与方法 3 中每层划分的子区域数为 4,运行加速比结果见表 2。

表 1

时间阶段 N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
模型 1: 计算层数 k	5	5	5	6	6	7	7	9	10	10	10	11	11	12	12	12	12
模型 2: 计算层数 k	1	1	1	2	7	7	9	11	11	11	11	12	12	12	12	12	12
模型 3: 计算层数 k	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17

表 2

	模型 1	模型 2	模型 3
S_p^1 (方法 1)	3.0	2.45	2.30
S_p^2 (方法 2)	2.74	2.03	1.70
S_p^3 (方法 3)	3.16	2.74	2.72

采用方法 2 对这 3 个模型的效果都不理想,主要是由于在整个运行阶段,时间步及计算层数较多,时间步分别总计为 136,136,391,计算层数分别总计为 1 200,1 152 和 6 647,而每一时间步都有层间通信同步,细化并行粒度引起的通信同步开销大大增加了。方法 3 对这 3 种模型的 S_p 都有改善,我们求解有多达 72 万个网格节点的大规模油藏模拟生产实际问题 3 的实践表明,方法 3 划分产生的并行求解任务均衡,有利于进一步提高加速比。

参考文献

- 1 Kendall R P *et al.* Large Scale Reservoir Simulation in the Concurrent Processing Milieu. Los Alamitos, CA: IEEE Computer Society Press, 1991
- 2 Killough J E *et al.* Simulation of compositional reservoir phenomena on a distribute memory parallel computer. In: Killough J E *et al.* eds. The 1991 SPE Symposium on Reservoir Simulation. Anaheim: University of Anaheim Press, 1991. 69-82
- 3 Bhogeswara Rao, Killough J E. Parallel linear solvers for reservoir simulation: a generic approach for existing and emerging computer architectures. In: Killough J E *et al.* eds. Proceedings of the 12th SPE Symposium on Reservoir Simulation. Houston: University of Houston Press, 1993. 71-82
- 4 徐向明,孙家昶等.多层二维二相油藏模拟的并行试验.见:李晓梅等编.全国第 4 届并行算法学术会议论文集.北京:航空工业出版社,1993.255-260

(Xu Xiang-ming, Sun Jia-chang *et al.* Parallel experiment for multiple layers two-dimension two-phases flow reservoir simulation. In: Li Xiao-mei *et al.* eds. Proceedings of the 4th Parallel Algorithms Conference. Beijing: Aviation Industry Press, 1993. 255-260)

Parallel Tasks Partitioning of Load-balancing for a Multiple Layers Numerical Simulation of Reservoir

SHU Ji-wu^{1,2} ZHAO Jin-xi^{1,2} ZHOU Wei-si³ ZHANG De-fu^{1,2}

¹(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

²(Department of Computer Science and Technology Nanjing University Nanjing 210093)

³(The Academy of Geological Sciences Shengli Oil-Field Dongying 257000)

Abstract In this paper, three tasks division strategies of load-balancing are given, which include a round-robin fashion, a domain partitioning fashion and a fashion of round-robin combining with domain partitioning, and a comparison is made. Then tasks division methods are proposed which can reduce solving times and benefit load-balancing and increases parallel efficiency for a simulation of reservoir of multiple layers two-dimension two-phase flow numerical problems on distributed memory parallel systems. This approach is applied to a large scale reservoir simulation which has over 720 000 grid points. The practical results show that the tasks generated by the strategy are well-balanced and benefit to improve speedup. This approach is apt to the problems for domain parallel or data parallel.

Key words Parallel computing, load-balancing, task dividing, numerical simulation of reservoir.