

## 多重网格格子 Boltzmann 方法的并行算法

刘智翔<sup>1</sup>, 宋安平<sup>1,2</sup>, 徐磊<sup>1</sup>, 郑汉垣<sup>1</sup>, 张武<sup>1,2\*</sup>

(1. 上海大学 计算机工程与科学学院, 上海 200444; 2. 上海大学 高性能计算中心, 上海 200444)

(\* 通信作者电子邮箱 wzhang@shu.edu.cn)

**摘要:**针对复杂流动数值模拟中的格子 Boltzmann 方法存在计算网格量大、收敛速度慢的缺点,提出了基于三维几何边界的多重笛卡儿网格并行生成算法,并基于该网格生成方法提出了多重网格并行格子 Boltzmann 方法(LBM)。该方法结合不同尺度网格间的耦合计算,有效减少了计算网格量,提高了收敛速度;而且测试结果也表明该并行算法具有良好的可扩展性。

**关键词:**格子 Boltzmann 方法; 多重网格; 并行算法; 可扩展性; OpenMP

**中图分类号:** TP338.6; O246 **文献标志码:** A

### Parallel algorithms for multi-grid lattice Boltzmann method

LIU Zhixiang<sup>1</sup>, SONG Anping<sup>1,2</sup>, XU Lei<sup>1</sup>, ZHENG Hanyuan<sup>1</sup>, ZHANG Wu<sup>1,2\*</sup>

(1. School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China;

2. High Performance Computing Center, Shanghai University, Shanghai 200444, China)

**Abstract:** Due to the shortcomings of large amount of computational grids and slow convergence rate in the numerical simulation of the complex flow, the parallel algorithm of multiple Cartesian grid generation based on three-dimensional geometry was proposed for Lattice Boltzmann Method (LBM). And multi-grid parallel LBM was developed based on the parallel algorithm of grid generation. The proposed algorithms can efficiently reduce the amount of computational grids and improve the convergence by coupling with the grids with different sizes. Numerical results also show that the proposed parallel algorithms have considerable scalability.

**Key words:** Lattice Boltzmann Method (LBM); multi-grid; parallel algorithm; scalability; OpenMP

## 0 引言

复杂流体运动的数值模拟一直是大规模科学与工程计算中最重要且具有挑战性的研究领域之一。与基于宏观纳维-斯托克斯(Navier-Stokes, N-S)方程的计算流体力学和基于微观分子动力学的流体力学模拟方法不同,基于 Boltzmann 方程的格子 Boltzmann 方法(Lattice Boltzmann Method, LBM)从介观尺度来描述流体系统<sup>[1]</sup>。由于 Boltzmann 方程自身本质的运动学特性,且能根据经典的 Chapman-Enskog 展开得到连续型的 N-S 方程,因此 LBM 比基于连续介质假设的 N-S 方程包含了更多的物理内涵<sup>[2]</sup>。同时, LBM 具有天然的并行性和方便处理边界条件等优点,使之适合处理复杂几何形状和复杂流动大规模计算问题<sup>[3]</sup>。由于 LBM 是将粒子速度分布离散成有限的几个值,通过格点间沿有限的速度方向迁移和碰撞的过程来模拟得到 N-S 方程的解,因此, LBM 通常采用的是统一尺度的笛卡儿网格,但在实际流动问题的计算中,存在计算网格量大、收敛速度慢等缺点<sup>[4-5]</sup>。

为了克服这些缺点,最早由 Filippova 等<sup>[6]</sup>提出了一种双向耦合的局部网格加密 LBM,但是在松弛因子接近 1 的情况下,该方法会发生不稳定现象,因此 Dupuis 等<sup>[7]</sup>在此基础上提出了一种基于分布函数和双向耦合的局部网格加细方法,

该方法能有效提高计算效率。最近, Toulil 等<sup>[8]</sup>也提出了一种结合大涡模拟的多尺度网格的 LBM,其实验结果进一步验证了多尺度网格在复杂流动问题中的可行性。另外, Eitel-Amor 等<sup>[9]</sup>采用中心点格式和局部网格加密的 LBM,并在圆球和圆柱绕流计算问题上证明了采用网格中心点格式在局部加密网格上的计算是可行的,且能较好地支持自适应网格方法。虽然以上的研究工作都能够有效提高计算效率,但是都增加了 LBM 计算的并行复杂度。

为了提高 LBM 的计算效率,本文基于网格中心点格式、八叉树结构和 OpenMP 编程模型,详细分析多重网格 LBM 中各部分的并行度,给出多重网格的并行生成算法和多重网格 LBM 并行算法两部分内容,并在高性能计算集群上进行测试验证。

## 1 单松弛格子 Boltzmann 方法

本章主要介绍三维情况下,基于 D3Q19 离散速度模型(见图 1)的单松弛格子 Boltzmann 模型。该模型的演化方程<sup>[10]</sup>为:

$$f_a(x + \xi_a \Delta t) - f_a(x, t) = -\frac{1}{\tau}(f_a(x, t) - f_a^{\text{eq}}(x, t)) \quad (1)$$

收稿日期: 2014-06-05; 修回日期: 2014-08-01。 基金项目: 国家自然科学基金资助项目(91330116)。

作者简介: 刘智翔(1986-), 男, 江西赣州人, 博士研究生, 主要研究方向: 高性能计算; 宋安平(1966-), 男, 上海人, 副教授, 博士, CCF 会员, 主要研究方向: 高性能计算、数据挖掘; 徐磊(1987-), 男, 江苏连云港人, 博士研究生, CCF 会员, 主要研究方向: 高性能计算; 郑汉垣(1965-), 男, 福建龙岩人, 博士研究生, 主要研究方向: 高性能计算; 张武(1957-), 男, 江西武宁人, 教授, 博士生导师, 博士, CCF 高级会员, 主要研究方向: 高性能计算、计算流体力学。

其中:  $\alpha$  表示方向,  $\xi_\alpha$  表示第  $\alpha$  个方向的离散速度,  $f_\alpha(x, t)$  表示在  $x$  处、时刻  $t$  时沿方向  $\alpha$  的粒子分布函数,  $\Delta t$  表示时间步长,  $\tau$  是松弛因子。宏观物理量密度和速度由式(2)计算:

$$\begin{cases} \rho = \sum_{\alpha} f_{\alpha} \\ \rho v = \sum_{\alpha} f_{\alpha} \xi_{\alpha} \end{cases} \quad (2)$$

本文采用 D3Q19 的离散速度(见图1),其速度方向值为:

$$\xi_{\alpha} = \begin{cases} (0, 0, 0), & \alpha = 0 \\ (\pm 1, 0, 0)c, (0, \pm 1, 0)c, (0, 0, \pm 1)c, & \alpha = 1, 2, \dots, 6 \\ (\pm 1, \pm 1, 0)c, (\pm 1, 0, \pm 1)c, (0, \pm 1, \pm 1)c, & \alpha = 7, 8, \dots, 18 \end{cases}$$

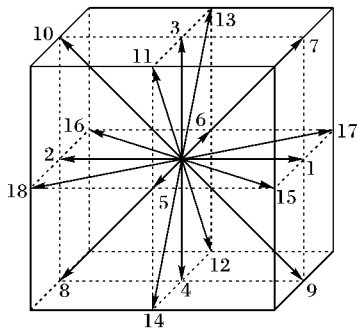


图1 D3Q19 离散速度模型

对应的平衡态分布函数为:

$$f_{\alpha}^{\text{eq}} = \rho \omega_{\alpha} \left( 1 + \frac{\xi_{\alpha} \cdot v}{c_s^2} + \frac{(\xi_{\alpha} \cdot v)^2}{2c_s^4} - \frac{v^2}{2c_s^2} \right)$$

其中:  $c_s$  表示声速,其值为  $c_s = c/\sqrt{3}$ 。权函数  $\omega_{\alpha}$  为:

$$\omega_{\alpha} = \begin{cases} 1/3, & \alpha = 0 \\ 1/18, & \alpha = 1, 2, \dots, 6 \\ 1/36, & \alpha = 7, 8, \dots, 18 \end{cases}$$

由于针对三维实际几何,所以涉及到曲面边界的处理,本文将采用 Yu 等<sup>[11]</sup>提出的统一插值格式(YMS Scheme)处理曲面边界。此外,采用非平衡外推格式处理计算区域包围盒<sup>[12]</sup>。具体的 LBM 算法流程见图2。本文将主要针对图2中的生成多重笛卡儿网络、初始化多重网络的计算信息和多重网络 LBM 计算进行研究,给出基于 LBM 的多重笛卡儿网络并行生成算法和多重网络 LBM 并行算法。

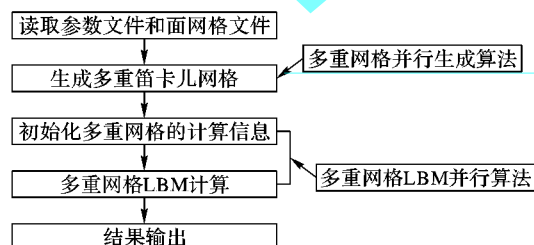


图2 LBM 算法流程

## 2 基于 LBM 的多重网格生成并行算法

根据第1章中的描述,LBM 采用的是单一尺度的笛卡儿网络。对于三维复杂几何,单一尺度的笛卡儿网络无法有效地描述实际的几何边界,因此,在实际几何边界处需要采用不同尺度的笛卡儿网络,即需要加密网格,以便更好地计算边界附近的流动。

由于三维立方体网格加密后将生成8个子立方体网格,所以本文使用八叉树数据结构来存储多重笛卡儿网络。为了减少存储量,每一个立方体网格都采用网格中心点坐标记录。

下面首先给出基于三维实际几何边界的多重网格生成算法,之后再对算法的并行性进行分析,给出基于 OpenMP 编程模型的多重网格并行生成策略。

步骤1 初始网格(第一重网格)的生成。

a) 根据计算域(长方体包围盒)和初始网格的尺寸,按照坐标轴的三个方向顺序生成初始网格。

b) 按照 D3Q19 离散速度模型中的19个方向,记录每一个网格点对应的19个速度方向的相邻网格点编号。

步骤2 第  $i$  重网格中所有网格点类型的判断。

a) 网格点类型分为三类:边界网格点、几何体内部网格点和几何体外部网格点。

b) 网格点类型的判断方法:射线法,即从网格中心点沿着某一方法(如  $x$  方向)发射一条射线,记录该射线与几何体相交的个数,如果交点个数为偶数,则网格点在几何体外部;如果交点个数为奇数,则网格点在几何体内部。

步骤3 对第  $i$  重网格中网格点进行加密,生成第  $i+1$  重网格。

a) 采用八叉树数据结构存储加密的网格。

b) 将所有边界网格点进行加密(即网格尺寸减少1/2,则第  $i$  重网格中的每一个边界网格点都存在第  $i+1$  重网格中的8个子网格)。

c) 根据实际几何体的形状,如曲率等,自适应加密计算域中的网格点,所有加密的网格点组成了第  $i+1$  重网格。

d) 查找并记录第  $i+1$  重网格中所有网格点沿着19个离散速度方向的相邻网格点编号。

步骤4 重复步骤2~3,完成第  $N$  重网格的生成。

步骤5 计算第  $N$  重网格中所有边界网格中心点沿着19个离散速度方向到实际几何边界的距离。

步骤1和3主要是进行网格生成,即将不同尺度的网格插入到八叉树数据结构中,这种插入操作只能按顺序串行执行。在步骤2和5的计算过程中,第  $i$  重网格中的网格点类型判断和边界距离的计算都是在网格点内部进行的,因此具有良好的并行性,可在步骤2和5中采用 OpenMP 进行加速。

针对三维实际几何,对比有限差分使用的多重笛卡儿网络和基于多重网格 LBM 的笛卡儿网络,后者的计算复杂度更高,所需要记录的信息量也更大(包含相邻的网格点编号以及不同尺度网格的相互关系)。本文提出的基于 LBM 的多重网格并行生成算法,采用了八叉树结构,最大限度地保留了不同尺度网格间的连接信息。从步骤1~5可看出:基于 LBM 的多重网格生成的主要开销是在网格点类型的判断与边界距离的计算上,而将这两部分并行处理将显著缩短多重网格生成的时间。

## 3 多重网格 LBM 的并行算法

为了方便描述,本文先以二维情况下的双重网格 LBM 计算为例,见图3。图3(a)中的黑色实心圆圈表示粗细网格(双重网格)边界中粗网格的中心点,灰色菱形表示粗细网格边界中细网格的中心。图3(b)是 D2Q9 的9个速度方向图。多重网格 LBM 计算的主要难点是处理粗细网格的交界面,该

部分需要保证宏观物理量的连续性。考虑图 3 中的粗网格中心点  $A$ , 其对应的三个离散速度方向 (1, 4, 8) 的分布函数值是由粗网格中对应的反方向相邻网格点碰撞后的分布函数迁移得到, 而离散速度方向 (2, 3, 5, 6, 7) 的分布函数值需要由细网格  $a, b, c, d$  四个子网格点上的分布函数插值得到。类似地, 细网格中的未知量也需要由相邻的粗网格插值得到。因此粗细网格在界面处都需要进行插值计算。

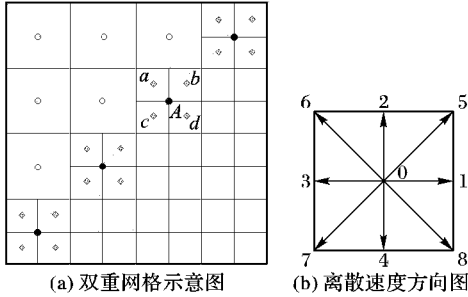


图 3 双重网格存储示意图

下面给出三维情况下的交界面处的计算公式。

假设  $C$  代表粗网格,  $F$  代表细网格, 则粗细网格交界面中的粒子分布函数的计算公式为:

$$\begin{cases} f_{\alpha,C}^* = f_{\alpha,F}^q + m \frac{\tau_C - 1}{\tau_F - 1} (f_{\alpha,F} - f_{\alpha,F}^q) \\ f_{\alpha,F}^* = f_{\alpha,C}^q + \frac{\tau_F - 1}{m(\tau_C - 1)} (f_{\alpha,C} - f_{\alpha,C}^q) \end{cases} \quad (3)$$

其中:  $m$  表示  $\Delta x_C / \Delta x_F$ , 即“粗网格尺寸/细网格尺寸”; 带上标“+”表示碰撞后的分布函数值。粗网格中的松弛时间  $\tau_C$  和细网格中的松弛时间  $\tau_F$  满足  $\tau_F = 0.5 + m(\tau_C - 0.5)$ 。

多重网格 LBM 计算与上述的双重网格 LBM 计算过程类似, 且多重网格的算法也是基于双重网格算法。限于篇幅, 本文仅给出三维情况下的双重网格 LBM 并行算法。

步骤 1 双重网格的初始化。

a) 根据不同的计算问题, 给每一重网格的网格点赋初值, 包括密度、速度、分布函数值;

b) 记录粗细网格交界面处的网格点编号。

步骤 2 粗网格点的碰撞迁移过程 (非交界面)。

a) 粗网格中除交界面网格点外的所有网格点根据式 (1) 进行碰撞和迁移计算, 得到  $t+1$  时刻的分布函数值;

b) 粗网格中除交界面网格点外所有网格点根据式 (2) 进行  $t+1$  时刻的宏观量计算。

步骤 3 细网格点的第一次碰撞迁移过程。

a) 细网格中除交界面网格点外的所有网格点根据式 (1) 进行碰撞和迁移计算, 得到  $t+0.5$  时刻的分布函数值;

b) 细网格中交界面网格点根据式 (1) 进行碰撞和迁移计算 (对已知的离散速度方向进行迁移), 得到部分已知方向的  $t+0.5$  时刻的分布函数值;

c) 细网格中交界面网格点根据式 (3) 和相邻粗网格上  $t$  时刻碰撞后的分布函数值插值得到未知离散速度方向上的分布函数, 从而得到全部方向的  $t+0.5$  时刻的分布函数值;

d) 细网格中所有网格点根据式 (2) 进行  $t+0.5$  时刻的宏观量计算。

步骤 4 细网格点的第二次碰撞迁移过程。

a) 细网格中除交界面网格点外的所有网格点根据式 (1)

进行碰撞和迁移计算, 得到  $t+1$  时刻的分布函数值;

b) 细网格中交界面网格点根据式 (1) 进行碰撞和迁移计算 (对已知的离散速度方向进行迁移), 得到部分已知方向的  $t+1$  时刻的分布函数值;

c) 细网格中交界面网格点根据式 (3) 和相邻粗网格上的碰撞后分布函数值 ( $t=1$  时刻) 插值得到未知离散速度方向上的分布函数, 从而得到全部方向的  $t+1$  时刻的分布函数值;

d) 细网格中所有网格点根据式 (2) 进行  $t+1$  时刻的宏观量计算。

步骤 5 粗网格点的碰撞迁移过程 (交界面)。

a) 粗网格中交界面网格点根据式 (1) 进行碰撞和迁移计算 (对已知的离散速度方向进行迁移), 得到部分已知方向的  $t+1$  时刻的分布函数值;

b) 粗网格中交界面网格点根据式 (3) 和相邻细网格上的碰撞后分布函数值 ( $t=1$  时刻) 插值得到未知离散速度方向上的分布函数, 从而得到全部方向的  $t+1$  时刻的分布函数值;

c) 粗网格中交界面网格点根据式 (2) 进行  $t+1$  时刻的宏观量计算。

步骤 6 重复步骤 2~5, 直到满足精度。

下面根据步骤 1~6 中的描述, 分析其对应的 OpenMP 并行算法。步骤 1 在整个计算过程中只需执行一次, 且初始化时是对每个网格点单独赋值, 具有良好的并行性, 因此可以采用 OpenMP 进行细粒度并行, 但是在记录不同尺度网格交界面处的网格点编号时, 为了避免线程间相互竞争, 需要设置 `#pragma omp critical` 操作。步骤 2~5 是整个多重网格 LBM 计算的核心步骤, 主要涉及到四个部分: 网格点的碰撞、网格点的迁移、网格点宏观量的计算和交界面上的数据传递。根据式 (3), 每个网格点的碰撞计算可以在各自网格点的内部进行, 而网格点的迁移需要涉及到相邻网格点, 且需要使用碰撞后的分布函数值, 因此需要单独存储每个网格点上碰撞后的分布函数值。虽然会导致内存使用量的增加, 但是能够提高网格点在迁移计算的并行性, 缩短计算时间。根据式 (2), 网格点的宏观量计算也仅需使用各自网格点迁移计算后的分布函数值。此外为了保证交界面上宏观量的连续性, 交界面上的数据传递需要严格按照时刻值进行。如果出现不同时刻的分布函数值, 则需要进行时间方向上的插值。此外空间上需要通过式 (3) 进行不同尺度网格间分布函数值的转换。基于上述分析, 步骤 2~5 中的每一步都可以只在网格点内部操作, 因此可以利用 OpenMP 编程模型对每一计算步骤进行加速, 且理论上, 该并行算法具有较好的可扩展性。

本文采用网格中心点格式的多重网格 LBM 并行算法, 与文献 [5] 采用网格顶点算法的主要优势在于中心点结构能够减少部分网格点的存储量, 并对自适应网格加密的 LBM 并行算法有良好的可移植性。

## 4 数值实验与分析

### 4.1 测验环境

本文使用上海大学“自强 4000”高性能计算集群中的胖节点进行测试, 其具体配置如下:

4 颗 Intel SandyBridge 架构的 E5-4650 CPU (2.7 GHz/8-



core), 总计 32 核; 512 GB 内存; 采用 Intel C/C++ 编译器, 且所有程序都使用“-O3”选项进行编译优化。

#### 4.2 测验算例

本文以圆球绕流为算例, 圆球的面网格 STL (STereoLithography) 文件的大小为 794 KB, 面网格三角形个数为 4032, 见图 4。

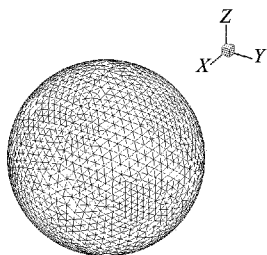


图4 圆球算例的面网格

本文将重点测试多重网格并行生成算法和多重网格 LBM 并行算法的可行性和加速比, 并分别从不同网格重数和不同计算规模两方面分析两种并行算法的性能。

#### 4.3 结果与分析

1) 验证多重网格并行生成算法和多重网格 LBM 并行算法的正确性。

取双重网格进行测试, 雷诺数  $Re = 100$ , 马赫数  $Ma = 0.1$ , 粗网格尺寸为 0.2, 细网格尺寸为 0.1, 其 XZ 截面的网格结果如图 5 所示。计算得到的阻力系数  $c_d = 1.08$ , 与参考文献[8]中的结果一致。

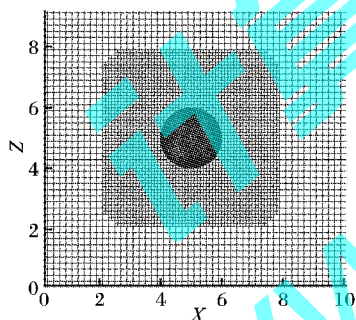


图5 绕圆球流动的双重网格

下面考虑单重网格下三种不同计算规模的加速比, 三种测试规模分别为: 375 000 (规模 1)、3 000 000 (规模 2)、24 000 000 (规模 3), 其对应的网格尺寸分别为 0.2、0.1 和 0.05, 即规模 1 到规模 3 的网格量都按照 8 倍线性增长。

图 6 为三种不同规模下, 单重网格生成并行算法和多重网格 LBM 并行算法的加速比。

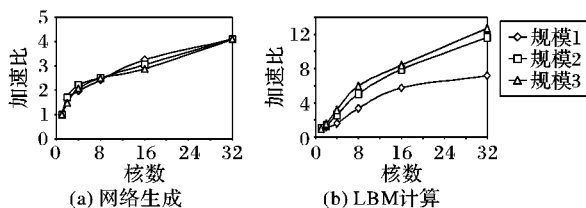


图6 单重网格 LBM 的加速比

从图 6 可看出: 随着计算核数的增加, 并行网格生成的加速比不断增大。根据第 2 章中的分析, 网格生成中存在部分步骤无法并行化, 所以加速比值不高。此外不同规模下网格生成的加速比相对稳定, 变化不明显。在图 6(b) 中, 随着计

算规模的增大, 加速比不断提高, 尤其从规模 1 到规模 2 的加速比增加明显, 因此本文后面的测试采用规模 2 中的网格量和网格尺寸作为基准。

2) 测试不同网格重数的加速比, 这里仅考虑双重网格和三重网格的情况。

根据图 6, 选定初始网格 (即第一重网格) 的尺寸为 0.1。在绕圆球流动问题的计算中, 双重网格的总网格数为 3 340 912, 三重网格的总网格数为 4 262 536。

图 7 为双重网格生成的加速比和双重网格 LBM 计算加速比。对比图 6(a) 和图 7(a), 双重网格生成的加速趋势与规模 2 的单重网格加速趋势基本一致, 双重网格并行生成的加速比值稍大些。对比图 6(b) 和图 7(b), 双重网格 LBM 计算的加速比值比单重网格的加速比值略低, 其主要是由于在同等计算规模下, 双重网格 LBM 并行算法提高了计算收敛速度, 缩短了计算总时间, 但是也更多次地启动 OpenMP, 其启动开销在一定程度上影响了加速效率。

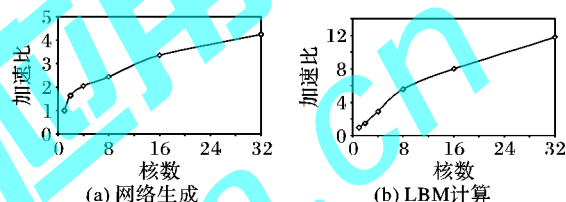


图7 双重网格 LBM 的加速比

图 8 为三重网格生成的加速比和三重网格 LBM 计算加速比。对比图 6(a)、图 7(a) 和图 8(a), 单重、双重和三重网格生成的加速趋势基本一致, 且三重网格生成的加速比值最大。因此随着网格重数的增加, 本文提出的多重网格并行生成算法具有良好的可扩展性, 能够适应于更多重数的网格生成, 且具有良好的加速比。对比图 6(b)、图 7(b) 和图 8(b), 三重网格 LBM 并行算法的计算加速比值最低, 但是其收敛速度最快, 计算总时间最短, 又保持了良好的加速比。

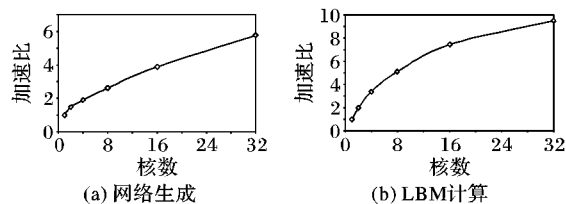


图8 三重网格 LBM 的加速比

根据以上的测试结果, 本文提出的多重网格并行生成算法和多重网格 LBM 并行算法都有良好的可扩展性和加速比, 且适用于三维真实问题的并行计算。同时, 从图 6~8 可看出 1~8 核间的并行效率最好, 其主要原因是由于 4 个 CPU 是共享内存的, 其通过总线相连, 存在带宽上的限制, 所以在跨多个 CPU 间执行 OpenMP 时将可能影响加速效率。

## 5 结语

本文给出了基于 LBM 的三维多重笛卡儿网格并行生成算法, 其采用网格中心点格式和八叉树数据结构进行存储, 最大限度地保留了不同尺度网格间的相关信息。此外本文还给出了基于该网格并行生成算法的多重网格 LBM 并行算法, 通过对圆球绕流问题的数值实验, 验证了所提算法的可行性, 证明了两种并行算法都具有良好的加速比。在网格重数增加或

(下转第 3072 页)

从实验结果可看出:本文提出的 RTS-THTC 算法在使用任务分层和任务完成时间约束的基础上,有效地减少了任务的完成时间,减缓了任务延迟情况的发生,保证任务能够在完成时间约束的条件下按时完成。

#### 4 结语

本文利用云计算环境下关联任务调度的特征,提出了一种基于任务分层和任务完成时间约束的关联任务调度策略。在使用 DAG 描述任务执行次序的基础上,对任务图进行分层以提高任务的并行性,同时对每层任务进行任务完成时间的约束以保证任务的完成时间。实验结果表明,本文提出的 RTS-THTC 任务调度算法在减少任务完成时间的同时,能够有效地减缓任务延迟情况的发生。需要说明的是本文提出的关联任务调度算法弱化了关联任务之间的通信情况,这与实际情况不一致,未来的工作将对这个因素加以考虑作进一步研究。

#### 参考文献:

- [1] CHEN Q, DENG Q. Cloud computing and its key techniques[J]. Journal of Computer Applications, 2009, 29(9): 2562-2567. (陈全, 邓倩妮. 云计算及其关键技术[J]. 计算机应用, 2009, 29(9): 2562-2567.)
- [2] ZHU F, HE Y. Parallel theory and design of scheduling algorithm in distributed computing[M]. Wuhan: Wuhan University Press, 2003. (朱福喜, 何炎祥. 并行分布计算中的调度算法理论与设计[M]. 武汉: 武汉大学出版社, 2003.)
- [3] WU X, DENG M, ZHANG M, *et al.* Task scheduling algorithm based on priority and cost constraint in cloud computing[J]. Journal of Computing Application, 2013, 33(8): 2147-2150. (武小年, 邓梦琴, 张明玲, 等. 云计算中基于优先级和费用约束的任务调度算法[J]. 计算机应用, 2013, 33(8): 2147-2150.)
- [4] YAN G, YU J, YANG X. Reliability-aware workflow scheduling strategy on cloud computing platform[J]. Journal of Computing Application, 2014, 34(3): 673-677. (闫歌, 于炯, 杨兴耀. 基于可靠性的云工作流调度策略[J]. 计算机应用, 2014, 34(3): 673-677.)
- [5] TOPCUOGLU H, HARIRI S, WU M. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 260-274.
- [6] DU X, JIANG C, XU G, *et al.* A grid DAG scheduling algorithm based on fuzzy clustering[J]. Journal of Software, 2006, 17(11): 2277-2288. (杜晓丽, 蒋昌俊, 徐国荣, 等. 一种基于模糊聚类的网格 DAG 任务图调度算法[J]. 软件学报, 2006, 17(11): 2277-2288.)
- [7] WILMER D, KLOS T, WILSON M. Distributing flexibility to enhance robustness in task scheduling problems[EB/OL]. [2014-02-01]. [http://bnaic2013.tudelft.nl/proceedings/papers/paper\\_88.pdf](http://bnaic2013.tudelft.nl/proceedings/papers/paper_88.pdf).
- [8] PATEL R, PATEL S. Survey on resource allocation strategies in cloud computing[J]. International Journal of Engineering Research and Technology, 2013, 2(2): 1-5.
- [9] LI J, SAIFULLAH A, AGRAWAL K, *et al.* Capacity augmentation bound of federated scheduling for parallel DAG tasks, WUCSE-2014-44[R]. St Louis: Washington University, 2014.
- [10] LIU Z, QU W, LIU W, *et al.* Resource preprocessing and optimal task scheduling in cloud computing environments [EB/OL]. [2014-02-10]. <http://www.chinacloud.cn/upload/2014-01/14013115059049.pdf>.
- [11] GUO L, ZHAO S, SHEN S, *et al.* Task scheduling optimization in cloud computing based on heuristic algorithm[J]. Journal of Networks, 2012, 7(3): 547-553.
- [12] HUANG J. The workflow task scheduling algorithm based on GA model in the cloud computing environment[J]. Journal of Software, 2014, 9(4): 873-880.

(上接第 3068 页)

网格规模增大的情况下,多重网格生成的并行算法加速比值将略有增加,两种并行算法都具有良好的可扩展性和加速比。结合 MPI 编程模型,本文提出的两种并行算法也可应用于大规模分布式集群。

#### 参考文献:

- [1] SUCCI S. The lattice Boltzmann equation for fluid dynamics and beyond [M]. New York: Oxford University Press, 2001: 3-15.
- [2] LALLEMAND P, LUO L. Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability [J]. Physical Review E, 2000, 61(6): 6546-6562.
- [3] AIDUN C K, CLAUSEN J R. Lattice-Boltzmann method for complex flows [J]. Annual Review of Fluid Mechanics, 2010, 42(1): 439-472.
- [4] WU J, SHU C. A solution-adaptive lattice Boltzmann method for two-dimensional incompressible viscous flows [J]. Journal of Computational Physics, 2011, 230(6): 2246-2269.
- [5] GUZIK S M, WEISGRABER T H, COLELLA P, *et al.* Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement [J]. Journal of Computational Physics, 2014, 259(20): 461-487.
- [6] FILIPPOVA O, HANEL D. Acceleration of lattice BGK schemes with grid refinement[J]. Journal of Computational Physics, 2000, 165(2): 407-427.
- [7] DUPUIS A, CHOPARD B. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm [J]. Physic Review E, 2003, 67(1): 66707.
- [8] TOUIL H, RICOT D, LEVEQUE E. Direct and large-eddy simulation of turbulent flows on composite multi-resolution grids by the lattice Boltzmann method[J]. Journal of Computational Physics, 2014, 256(1): 220-233.
- [9] EITEL-AMOR G, MEINKE M, SCHRODER W. A lattice-Boltzmann method with hierarchically refined meshes [J]. Computers and Fluids, 2013, 75(1): 127-139.
- [10] JAHANSHALOO L, POURIAZDANPANAH E, SIDIK N A C. A review on the application of the lattice Boltzmann method for turbulent flow simulation [J]. Numerical Heat Transfer, Part A: Applications: An International Journal of Computation and Methodology, 2013, 64(11): 938-953.
- [11] YU D, MEI R, SHYY W. A unified boundary treatment in lattice Boltzmann method [C]// Proceedings of the 41st Aerospace Sciences Meeting and Exhibit. Madison: AIAA Press, 2003.
- [12] GUO Z-L, ZHENG C-G, SHI B-C. Non-equilibrium extrapolation method for velocity and boundary conditions in the lattice Boltzmann method [J]. Chinese Physics, 2002, 11(4): 366-374.