LBM在多核并行编程模型中的应用

李彬彬,李 青
(上海大学 计算机工程与科学学院,上海 200072)

摘 要:LBGK(Lattice Bhatnagar-Gross-Krook)模型不仅是LBM(Lattice Boltzmann Method)理论及应用上的新突破,而且是一种非常新颖的数值计算方法,适合大规模并行计算。多线程并行编程接口库(Multi-Thread Interface, MTI)充分利用多核处理器的资源来提升计算的性能,为在多核环境下方便地开发高效的并行程序提供了一个接口,大大地减轻了开发人员的负担。MTI提供了使用cache块技术划分数据集实现单任务数据并行计算,以及采用任务密取调度策略实现多任务并行处理。应用MTI实现了LBGK模型模拟斑图形成的并行计算,并获得了较高的并行效率。

关键词:多核处理器;格子Boltzmann方法;并行计算;多线程接口库
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2011)07-0221-03

Application of LBM on Multi-Core Parallel Programming Model

LI Bin-bin, LI Qing
(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract: LBGK (Lattice Bhatnagar-Gross-Krook) model is not only the new ground on theory and application of LBM (Lattice Boltzmann Method), but also a very novel numerical method. It applys to the massively parallel processing. With management of threads, MTI (Multi-Thread Interface) provides two main methods for parallel coding on multicore processor computer. One is data parallelism based on cache blocking, the other is a tasks schedule with working stealing. MTI provides an interface for the development of multicore environment conveniently and efficiently, greatly reducing the burden on developers. An LBGK model for pattern formation is realized by MTI, and the numerical results show that MTI is efficient and easy to use.

Key words: multi-core processor; lattice Boltzmann method; parallel computing; multi-thread interface

0 引 言

LBGK方法具有边界条件容易处理、适合大规模并行计算等优点,受到物理学界、数学界及计算机界极大的重视,发展非常迅速。在流体力学中的流动模拟及其他相关领域都得到应用和发展。LBGK模型的分布式并行算法秉承着并行计算流体力学问题的一般方法,即采用物理区域分割的并行方法,并在Jacobi迭代的基础上进行。

当前多核处理器已经普及,然而专门针对多核的并行程序开发模型尚不多见。文献[1]提出了一个可用于多核处理器系统的基于多线程的并行程序开发接口库MTI(Multi-Thread Interface)。利用该接口库可将计算任务及数据划分并映射到多核上并行执行,通过优化调整有效减少存储竞争,达到负载均衡,利用多核资源来提高程序的性能[2]。具体地,MTI提供了使用cache块技术[3]划分数据集实现单任务数据并行计算,以及采用任务密取调度策略[4]实现多任务并行处理。文中利用该接口库实现了LBGK模型模拟斑图形成的并行计算,并获得了较高的并行效率。

1 LBGK模型

1.1 LBGK演化方程

LBM是一种模拟流体流动的数值方法,它将流体看成许多只有质量没有体积的微粒组成,这些微粒可以向空间的若干方向流动。其主要思想是以简单规则的微观粒子运动刻画复杂多变的宏观现象。粒子在每个时间步的运动由两个子步构成。

① 碰撞:当多个粒子到达同一网格结点时,它们按碰撞规则相互作用并改变各自的速度方向。根据LBGK模型,该碰撞操作已经简化成为一个松弛过程。

② 迁移:每个粒子按其速度方向移动到其最邻近的网格结点。LBGK模型的演化方程为如下的松弛过程。

收稿日期:2010-12-02;**修回日期**:2011-03-07
基金项目:上海市重点科学建设项目(J50103)
作者简介:李彬彬(1986-),女,湖南人,硕士研究生,研究方向为多核处理器体系结构、并行计算;李 青,博士,教授,博士生导师,研究方向为高性能计算及其应用。

$$f_i(x + ce_i\Delta t, t + \Delta t) = (1 - \omega) f_i(x,t) + \omega f_i^{eq}(x,t) \quad (1)$$

其中 $f_i(x,t)$ 是定义在离散速度集 e_i 上，t 时刻位于二维空间网格点 $x = (x_1, x_2)$ 处的粒子密度分布函数；$f_i^{eq}(x,t)$ 为平衡态分布函数，与流体的宏观速度和密度有关；ω 为松弛因子。

1.2 模拟反应-扩散系统的 D2Q7 模型

考虑有两种组分的反应-扩散方程，即

$$\begin{cases} \dfrac{\partial u}{\partial t} = D_1 \nabla^2 u + R_1(u,v) \\ \dfrac{\partial v}{\partial t} = D_2 \nabla^2 v + R_2(u,v) \end{cases} \quad (2)$$

其中 u 与 v 是参与反应的两种组分的密度（或浓度），这两种组分的扩散系数分别为 D_1 与 D_2；∇^2 为 Laplace 算子；R_1, R_2 为反应项。

1993 年 S P Dawson 和 Chen（陈十一）提出了模拟反应-扩散方程的 D2Q7 模型[5]。

2002 年 Li（李青）等提出了模拟反应-扩散方程的 D2Q5 模型并模拟了 CIMA 反应动力学系统中的图灵斑图形成[6]。

D2Q7 模型的粒子离散速度集为：

$$e_i = \begin{cases} 0 & i = 0 \\ \left(\cos\left[\dfrac{(i-1)\pi}{3}\right], \sin\left[\dfrac{(i-1)\pi}{3}\right]\right) & i = 1,2,3,4,5,6 \end{cases} \quad (3)$$

假定 $\dfrac{\partial^2}{\partial t} \propto \varepsilon^2, R_s \propto \varepsilon (s = 1,2)$，则 LBGK 模型的演化方程为：

$$\begin{cases} f_i(x + c\varepsilon e_i, t + \varepsilon) = (1 - \omega_1) f_i(x,t) + \omega_1 f_i^{eq}(x,t) + \dfrac{R_1(u,v)}{7} \\ g_i(x + c\varepsilon e_i, t + \varepsilon) = (1 - \omega_2) g_i(x,t) + \omega_2 g_i^{eq}(x,t) + \dfrac{R_2(u,v)}{7} \end{cases} \quad (4)$$

其中粒子分布函数 f 和 g 分别联系组分 u 和 v，$c = \Delta x/\Delta t$，$\varepsilon = \Delta t$。有

$$u(x,t) = \sum_{i=0}^{6} f_i(x,t), \quad v(x,t) = \sum_{i=0}^{6} g_i(x,t) \quad (5)$$

$$f_i^{eq}(x,t) = \frac{u(x,t)}{7}, \quad g_i^{eq}(x,t) = \frac{v(x,t)}{7} \quad (6)$$

扩散系数 D_1 和 D_2 与松弛因子 ω_1 和 ω_2 的关系分别为：

$$D_1 = \frac{2}{7} c^2 \varepsilon \left(\frac{1}{\omega_1} - \frac{1}{2}\right), \quad D_2 = \frac{2}{7} c^2 \varepsilon \left(\frac{1}{\omega_2} - \frac{1}{2}\right) \quad (7)$$

2 多线程接口库 MTI 程序设计

2.1 并行模型

文献[1]所提出的多线程接口库 MTI 提供了单任务多数据流和多任务多数据流两种并行计算模式。

在使用基于多任务并行的编程模型时，开发人员则不必考虑如何将一个具体的任务分配给底层的线程执行，只需要将任务定义为一个任务类，在任务类中定义任务执行的操作和相应数据集即可[7]。这种并行模式的程序框架如下：

```
class MyTask : public MTI_Task
{
private:
    //任务操作的数据变量
public:
MyTask();
~MyTask();
void Task_Operation(MTI_Range &range);
};
int main()
{
    MyTask *task =new MyTask();
    MTI_Parallel_Do(begin, end, task);
    delete task;
    return 0;
}
```

任务类进行的具体操作在 Task_Operation 中定义，Task_Operation 的参数 range 代表在某一个线程上操作的子数据集的范围，任务定义完成后，MTI_Parallel_Do 会自动将任务按照数据并行的方式分配到各个线程去执行。文中的计算采用了多任务多数据流并行计算模式。

2.2 LBGK 模型的多任务并行计算实现技术

CIMA 反应动力学系统是第一个证实图灵斑图（Turing Pattern）存在的化学实验[8]，其反应动力方程为[9-10]：

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{1}{\sigma}\left(a - u - 4\dfrac{uv}{1+u^2} + \nabla^2 u\right) \\ \dfrac{\partial v}{\partial t} = b\left(u - \dfrac{uv}{1+u^2}\right) + d\nabla^2 v \end{cases} \quad (8)$$

可知其反应项为 $R_1(u,v) = \dfrac{1}{\sigma}\left(a - u - 4\dfrac{uv}{1+u^2}\right)$ 和 $R_2(u,v) = b\left(u - \dfrac{uv}{1+u^2}\right)$，扩散系数分别为 $D_1 = 1/\sigma$，$D_2 = d$。

具体计算步骤如下：

第一步：取定网络规模（如 1000×1000），$\Delta x = 1$，$\Delta t = 0.1$，则 $c = 10$，读取参数 a, b, σ 的值。根据公式（7）计算松弛因子 ω_1 和 ω_2。

第二步:随机地选取约一半的网格点,赋值

$$u(x,0)=\frac{a}{5},\ v(x,0)=1+\frac{a^2}{25} \tag{9}$$

计算局部动态平衡分布函数的初值,作为分布函数的初值:

$$f_i(x,0)=f_i^{eq}(x,0)=\frac{u(x,0)}{7},g_i(x,0)=g_i^{eq}(x,0)=\frac{v(x,0)}{7} \tag{10}$$

第三步:每个演化步,根据公式(5)计算宏观量、根据公式(6)计算局部动态平衡态分布函数,再根据公式(4)进行碰撞和迁移的计算。

第四步:时间步增1,并转第三步或终止程序。

上述算法的第三步是计算量密集之处,对其中的碰撞及迁移操作按任务并行模式处理[11],相应的任务类定义如下:

```
// 碰撞操作类
class CollideTask : public MTI_Task
{
private:
CCimaDoc * doc;
public:
CollideTask(CCimaDoc * cdoc) : doc(cdoc) {};
~CollideTask() {};
void Task_Operation(MTI_Range &range);
};
//演化类(含迁移操作)
class EvoluteTask : public MTI_Task
{
private:
CCimaDoc * doc;
public:
EvoluteTask(CCimaDoc * cdoc) : doc(cdoc) {};
~EvoluteTask() {};
void Task_Operation(MTI_Range &range);
}
```

实现并行计算的两个函数为par_collide及par_evolute。在par_evolute函数定义中,首先执行碰撞步的并行计算,然后创建一个演化类,直接执行此任务的MTI_Parallel_Do函数来实现并行计算。

3　数值实验结果

数值实验在系统内存为4GB,CPU主频为2.67GHz的Intel Core2 Quad(4核)机器上进行,程序实现了LBGK模型模拟图灵斑图形成。程序执行时,每个时间步启动四个线程进行碰撞及迁移的操作。数值实验针对不同的网络规模、不同的时间步数进行了大量的实验,通过对实验结果的分析,也验证了在多核平台下应用多线程接口库MTI实现并行程序的易用性和有效性。

时间步固定($t=2000$)时,网格规模逐步扩大,启动4个线程时加速比及并行效率如表1所示。

表1　计算步数固定时LBGK模型的并行计算时间(毫秒)

规模	一个线程	四个线程	加速比	并行效率
100×100	24411	19453	1.254871	31.37%
200×200	37125	28250	1.314159	32.85%
500×500	226000	77969	2.898588	72.46%
800×800	606797	203812	2.977239	74.43%
1000×1000	980079	330578	2.964744	74.11%

网格规模固定(800×800)时不同时间步的情形,启动4个线程时加速比及并行效率如表2所示。

表2　计算规模固定时LBGK模型的并行计算时间(毫秒)

步数	一个线程	四个线程	加速比	并行效率
1000	304281	100797	3.018751	75.47%
2000	609312	200203	3.043471	76.09%
3000	915328	302640	3.024478	75.61%
4000	1235221	402953	3.065422	76.64%
5000	15341403	5055641	3.034512	75.86%

从上述数值实验结果可见,随着计算规模的增加,程序的并行效率为75%左右。

4　结束语

文中应用自行开发的多线程并行编程模型接口库,实现了多核系统下的并行计算,数据结果表明该多线程接口库易用并对LBGK计算可获得较高的并行加速比和并行效率。

参考文献:

[1] 曹折波,李　青.多核处理器并行编程模型的研究与设计[J].计算机工程与设计,2010,31:2999-3002.

[2] 施惠丰,袁道华.基于多核的多线程程序优化研究[J].计算机技术与发展,2010,20(6):70-73.

[3] Frigo M,Strumpen V. The Cache Complexity of Multithreaded Cache Oblivious Algorithms[C]//ACM Symposium on Parallelism in Algorithms and Architectures. [s. l.]:[s. n.], 2006:271-280.

[4] Christophe C,Koskas M. Work Stealing Technique and Scheduling on the Critical Path[C]//The 3rd International Conference on Grid and Pervasive Computing. [s. l]:[s. n.],2008.

[5] Dawson S P,Chen Shiyi,Doolen G D. Lattice Boltzmann computations for reaction-diffusion equations[J]. J. Chem. Phys.,

(下转第228页)

与请求方的连接尝试并在相应端口启动 TCP 监听服务，等待请求方的连接，实现代码如下所示：

```
TCPConnector * mConnector=new TCPConnector();
mConnector->ConnectToServer(recvbuf.iStringLen, recvbuf.Port); //连接到服务器 S
ACE_INET_Addr tmpAddr;
mConnector->getLocalAddr(tmpAddr); //取本地连接地址
CTOSMessage sendmsg;
sendmsg.iMessageType=CONNACK;//回复 CONNACK
sendmsg.message.loginmember.localip=tmpAddr.get_ip_address();//本地地址
sendmsg.message.loginmember.localport=tmpAddr.get_port_number();//本地端口
mConnector->sendMsg(sendmsg);
unsigned int iLocalIP = sendmsg.message.loginmember.localip;//保存本地地址，稍后用作监听服务的地址
u_short iLocalPort = sendmsg.message.loginmember.localport;//保存本地端口，稍后用作监听服务的端口
STOCMessage serverMsg;
mConnector->recvMsg(serverMsg,1);//取服务器 S 应答
mConnector->Disconnect();//关闭与服务器 S 连接
if(serverMsg.iMessageType== DIRECTCON){//在 iLocalIP、iLocalPort 地址启动 TCP 监听器
if(serverMsg.iMessageType== CONDETA){
//1. 启动报文窥探线程，把即将发起的与请求方连接的握手信息发送给服务器 S
//2. 使用地址 iLocalIP、iLocalPort 向请求端发起 TCP 连接（将超时），打通我方到请求方的 NAT
//3. 在 iLocalIP、iLocalPort 地址启动 TCP 监听器
//4. 再向服务器 S 发起 TCP 连接并发送 CONNRES，告知 TCP 监听服务已准备好}
```

至此，接受方将在 iLocalIP、iLocalPort 地址上监听请求方的连接请求，并完成实际的数据传输。本程序代码在 VS2010+ACE 5.8.1 环境下编写，因篇幅关系只分析了核心代码。

4 结束语

借助 UDP 协议及第三方服务器，在原本无法通信的两个私网主机之间建立起一条 TCP 通信通道，解决 P2P 类应用程序所遇到的通信问题，同时借助于简单高效、面向对象的 ACE 框架，方便、快捷地实现了 TCP 穿透 NAT 的原理代码。目前，笔者已把该技术应用在广东省教育厅的《英语网络考试系统》项目中，实现了基于 TCP 协议的 P2P 大规模考试平台。实践表明，由于私网主机间存在通信通道，彼此可共享数据、提供服务，大大减轻了服务器的负担，使因访问量问题引发的服务器故障大幅度减少，取得了较好的效果。

参考文献：

[1] 冯 昊，黄治虎，伍技祥. 交换机/路由器的配置与管理[M]. 北京：清华大学出版社，2005.

[2] 万川龙，桑 军，向 宏，等. 基于 P2P 网络的视频点播系统设计[J]. 计算机技术与发展，2009，19(6)：17-21.

[3] 范会波，张新有. 基于 P2P 的文件共享系统的设计与实现[J]. 计算机技术与发展，2010，20(3)：48-49.

[4] 谢希仁. 计算机网络[M]. 北京：电子工业出版社，2008.

[5] 冯 坚，王书田，林日光. 一种 TCP 博弈模型的 Nash 均衡存在性分析与仿真[J]. 计算机技术与发展，2009，19(11)：76-79.

[6] 王 南，孙保锁，王月平. P2PSIP 系统中 NAT 穿越方案的研究与设计[J]. 计算机技术与发展，2009，19(10)：66-69.

[7] 黄道颖，李祖鹏，庄 雷，等. 分布式 Peer to Peer 网络 Gnutella 模型研究[J]. 计算机工程与应用，2003(5)：60-63.

[8] Eppinger J L. Tcp Connections for P2P Apps：A Software Approach to Solving the NAT Problem[R]. Pittsburgh：Carnegie Mellon University，2005.

[9] Guha S，Francis P. Characterization and Measurement of TCP Traversal through NATs and Firewalls[C]//Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement. Berkeley：USENIX Association，2005.

[10] Stevens W R. TCP/IP Illustrated Volume 1：The Protocols[M]. [s.l.]：Addison-Wesley，1994.

[11] Schmidt D C，Huston S D. C++Network Programing，Volume 1[M]. 武汉：华中科技大学出版社，2003.

[12] Schmidt D C，Huston S D. C++Network Programing，Volume 2[M]. 北京：电子工业出版社，2004.

（上接第 223 页）

1993，98：1514-1523.

[6] Li Q，Zheng C G，Wang N C，et al. LBGK simulations of Turing patterns in CIMA model[J]. J. Scientific Computing，2001，16(2)：121-134.

[7] 陈国良. 并行计算——结构·算法·编程[M]. 北京：高等教育出版社，2004：104-109.

[8] Ouyang Q，Swinney H L. Transition from a uniform state to hexagonal and striped Turing patterns[J]. Nature，1991，352：610-612.

[9] Lengyel I，Epstein I R. Modeling of Turing structure in the chlorite-iodide-malonic acid-starch reaction system[J]. Science，1991，251：650-652.

[10] Rudovice B，Barillot E，Davies P W，et al. Experimental studies and quantitative modeling of Turing patterns in the (chlorine dioxide，iodine，malonic acid) reaction[J]. J Chem. Phys. A，1991，103：790-1800.

[11] 郭 静，祝永志，王延玲. 基于 MPI 的动态负载平衡算法[J]. 计算机技术与发展，2009，19(5)：150-153.