第 41 卷 第 4 期 计 算 机 工 程 2015 年 4 月
Vol. 41 No. 4 Computer Engineering April 2015

·开发研究与工程应用· 文章编号：1000-3428(2015)04-0288-06 文献标识码：A 中图分类号：TP301.6

基于 GPU 的碳纳米管分子动力学并行仿真

孟小华[a]，覃大胜[a]，郑冬琴[b]，周玉宇[a]

（暨南大学 a. 计算机科学系；b. 物理系，广州 510632）

摘 要： 基于计算机的分子动力学仿真具有理论分析方法和实验方法无法比拟的优点，但分子动力学仿真算法计算量非常大，特别是在对碳纳米管的大规模粒子数进行仿真处理时，普通的基于 CPU 的串行算法执行效率低且耗时多。为此，提出基于统一计算设备架构的碳纳米管分子动力学的图形处理单元（GPU）并行算法，设计并实现仿真算法中适合 GPU 并行运算的分裂算法，将具有竞争资源的运算以非竞争方式运行。实验结果表明，与 CPU 串行仿真算法相比，分裂算法的运算速度较快，且在只有 16 个 GPU 流处理器显卡上可获得十多倍的加速比。

关键词： 碳纳米管；分子动力学；统一计算设备架构；并行计算；时间效率；加速比

中文引用格式： 孟小华，覃大胜，郑冬琴，等. 基于 GPU 的碳纳米管分子动力学并行仿真[J]. 计算机工程，2015，41(4)：288-293.

英文引用格式： Meng Xiaohua，Qin Dasheng，Zheng Dongqin，et al. Parallel Simulation of Carbon Nano Tube Molecular Dynamics Based on GPU[J]. Computer Engineering，2015，41(4)：288-293.

Parallel Simulation of Carbon Nano Tube Molecular Dynamics Based on GPU

MENG Xiaohua[a]，QIN Dasheng[a]，ZHENG Dongqin[b]，ZHOU Yuyu[a]

（a. Department of Compute Science；b. Department of Physics，Jinan University，Guangzhou 510632，China）

【Abstract】 Molecular dynamics simulation has advantages dramatically superior to both theoretical analysis and experiments. However，due to the extremely high cost of computation resources during the simulation of a large number of Carbon Nano Tube（CNT） particles，typical CPU serial algorithm implementation is non-efficient and slow. A Compute Unified Device Architecture（CUDA） based parallel algorithm of CNT molecular dynamics is proposed in this paper to take advantage of the data parallelism of Graphic Processing Unit （GPU）. A CNT is divided to several blocks and processed parallel in the GPU. Experimental results show that the algorithm can obtain a speed-up more than 10 times to the CPU serial algorithm in a low-configured graphics card that has only 16 GPU stream processors.

【Key words】 Carbon Nano Tube（CNT）；molecular dynamics；Compute Unified Device Architecture（CUDA）；parallel computation；time efficiency；speed-up ratio

DOI： 10.3969/j.issn.1000-3428.2015.04.055

1 概述

自从文献[1]首次应用分子动力学研究硬球系统以来，分子动力学广泛应用于晶体生长、压痕试验、摩擦学以及低压金刚石合成等领域。在新兴的纳米工程领域，建立在连续介质基础上的宏观机理很难解释纳米工程中出现的一些特殊现象，因而分子动力学方法成为重要的研究手段之一[2]。文献[3]通过分子动力学模拟了 Ni 针尖在 Au(001)表面的纳米压痕过程，其模拟结果发表在《科学》上，成为该领域的标志性成果。文献[4]通过分子动力学仿真算法，研究了不同晶粒尺寸的多晶铜塑性变形过程，从而发现晶粒尺寸与强度不完全满足 Hall-Petch 关系。

分子动力学仿真算法，有着理论分析方法和实验方法无法比拟的优点，但是其计算要求极高，往往受限于计算机的计算能力。为了提高仿真算法规模，国内外许多专家学者做了大量研究[5-7]。仿真算法已由最初提高单机规模的串行算法[8]，发展到如今通过将计算任务分配给多个 CPUs，从而提高模拟规模的并行算法[9]，规模也由最初的几千个原子提

基金项目： 国家“863”计划基金资助项目(2013AA040404)；广东省重点实验室建设基金资助项目(2011A060901026)。

作者简介： 孟小华(1965－)，男，副教授、硕士、CCF 会员，主研方向：并行分布式系统；覃大胜，硕士研究生；郑冬琴，副教授、博士；周玉宇，副研究员，博士。

收稿日期： 2014-04-09 **修回日期：** 2014-06-12 **E-mail：** xhmeng@163.com

高到上百万、上千万甚至到上亿个原子。由于碳纳米管[10]系统中含有大量粒子,结构和受力复杂[11],其分子动力学模拟需要极强的计算能力,对 CPU 处理能力也有较高的要求。而利用 CPU 实现仿真串行算法,计算量极大,运算时间长,并且模拟粒子数也受到很大的限制。

本文基于 CUDA 平台,通过调度 GPU 多个流处理单元[12-13],利用 GPU 强大的并行计算能力和高效的数据传输能力,对碳纳米管分子动力学的并行仿真算法进行研究和实现,通过把碳纳米管系统分成多层可并行处理的单元进行并行计算。

2　分子动力学 Verlet 仿真算法

分子动力学是一套分子模拟方法,是一种重要且有效的原子尺度计算机模拟手段。该方法主要依靠牛顿力学来模拟分子体系的运动,以在由分子体系的不同状态构成的系统中抽取样本,从而计算体系的构型积分,并以构型积分的结果为基础进一步计算体系的热力学量和其他宏观性质。因而,用分子动力学是碳纳米管仿真的重要途径。

在分子动力学 Verlet 仿真算法中,粒子位置的泰勒展开式如下:

(1)将 $\boldsymbol{x}(t+\Delta t)$ 和 $\boldsymbol{x}(t-\Delta t)$ 进行泰勒展开如下式所示:

$$\boldsymbol{x}(t+\Delta t)=\boldsymbol{x}(t)+\boldsymbol{v}(t)\Delta t+\frac{\boldsymbol{a}(t)\Delta t^2}{2}+\frac{\boldsymbol{b}(t)\Delta t^3}{2}+O(\Delta t^4) \tag{1}$$

$$\boldsymbol{x}(t-\Delta t)=\boldsymbol{x}(t)-\boldsymbol{v}(t)\Delta t+\frac{\boldsymbol{a}(t)\Delta t^2}{2}-\frac{\boldsymbol{b}(t)\Delta t^3}{2}+O(\Delta t^4) \tag{2}$$

其中,$\boldsymbol{x}(t-\Delta t)$ 表示为前一时刻的位置;$\boldsymbol{x}(t+\Delta t)$ 表示为后一时刻的位置。

(2)将式(1)和式(2)相加,得到位置表达式如下:

$$\boldsymbol{x}(t+\Delta t)=2\boldsymbol{x}(t)-\boldsymbol{x}(t-\Delta t)+\boldsymbol{a}(t)\Delta t^2+O(\Delta t^4) \tag{3}$$

(3)将式(1)和式(2)相减,再两边同时除以 $2\Delta t$,获得速度的表达式如下:

$$\boldsymbol{v}(t)=\frac{\boldsymbol{x}(t+\Delta t)-\boldsymbol{x}(t-\Delta t)}{2\Delta t}+O(\Delta t^2) \tag{4}$$

(4)由式(1)~式(4),在已知粒子 $t-2\Delta t$ 时刻的位置 $\boldsymbol{x}(t-2\Delta t)$、$t-\Delta t$ 时刻的位置 $\boldsymbol{x}(t-\Delta t)$ 和 $t-\Delta t$时刻的加速度 $\boldsymbol{a}(t-\Delta t)$ 的情况下,启动 Verlet 算法进行积分:根据 $\boldsymbol{x}(t-2\Delta t)$,$\boldsymbol{x}(t-\Delta t)$ 和 $\boldsymbol{a}(t-\Delta t)$,将 $t=t-\Delta t$ 代入式(3),获得 $\boldsymbol{x}(t)$;根据 $\boldsymbol{x}(t)$,基于一定的势函数更新 $a(t)$;同时,根据 $\boldsymbol{x}(t)$ 和 $\boldsymbol{x}(t-2\Delta t)$,将 $t=t-\Delta t$ 代入式(4),更新 $\boldsymbol{v}(t-\Delta t)$;即得到粒子 $\boldsymbol{x}(t)$,$\boldsymbol{v}(t-\Delta t)$ 和 $\boldsymbol{a}(t)$,重复执行该步骤。

由上,可利用 Verlet 列表法设计串行仿真算法的实现步骤如下:

(1)先构造如图 1 所示的碳纳米管结构模型,并设置各种参值。

(2)设计前端热浴和后端热浴。

(3)构造并调用计算碳纳米管结构内部的力的结构函数及计算碳纳米球 C60 与碳纳米管间范德华力的函数等。

(4)计算所有的力,解牛顿运动方程,分层计算并积分(算法的核心)。

(5)模拟出碳纳米球 C60 随着热浴时间的运行轨迹,根据运算结果,可画出和碳纳米管内随着能量传送的温度变化曲线图。

3　基于 CUDA 并行仿真算法的研究与实现

3.1　碳纳米管系统模型的建立

在数据库文件中读取碳纳米管系统的参数条件,如初始温度、粒子数、密度时间等,构造碳纳米管系统模型 MOD1 及参数模型 MOD2,模型 MOD2 可以设置各种参数在模型 MOD1 上使用,模型 MOD1 如图 1~图 3 所示,由碳纳米管和设置在碳纳米管内部的一个足球状 C60 分子(由 60 个碳原子构成的分子,形似足球,又名足球烯)组成,所述碳纳米管为层状中空结构,管身是准圆管结构,由多个六边形碳环结构单元组成,其直径一般在一到几十个纳米之间,长度则远远大于其直径,对该模型进行初始化,读取模型中所有粒子的速度和位置坐标。

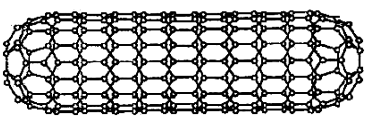


图 1　碳纳米管



图 2　C60 分子

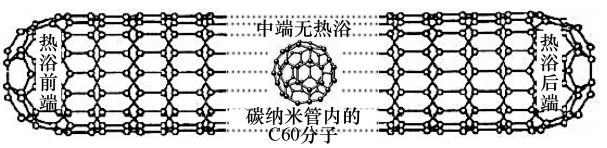


图 3　碳纳米管系统模型

在碳纳米管中,除管口层粒子外,其余粒子相互

组成正六边形。如图 4 所示,非管口粒子都有 3 个近邻粒子(ip,jp,kp 均为 idx 粒子的近邻粒子)。

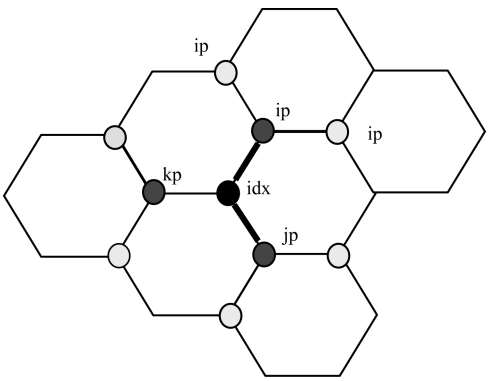


图 4 碳纳米结构

在碳纳米管系统中,因为 C60 分子受到碳纳米管管壁上碳粒子产生的力,在平衡位置上来回振动,当 C60 分子处于平衡状态时,采用 Gaussian 热浴法(约束温度调节方法),在碳纳米管两端同时分别设置不同温度的热浴条件,并设定热浴时间,使碳纳米管管壁上碳粒子和 C60 分子因平衡状态被打破而发生位置和速度的变化。而这些变化是相互影响的,具有传递性,伴随着这些粒子位置和速度的变化,能量在长管状的碳纳米管系统内发生传递。由于 C60 分子位于碳纳米管内,其受力会随着碳纳米管组成粒子的位置改变而改变,因此伴随着能量传递的过程,C60 分子的运动状态和能量也会发生改变;采用 Gaussian 热浴法的基本原理是在运动方程中加入摩擦力 f_i,并将其与粒子速度 v_i 联系起来,其受力公式为:

$$f_i = ma_i + \xi m v_i$$

当平衡态时,系统温度不变,因此有 $\mathrm{d}E_k/\mathrm{d}t = 0$。即有:$\sum_i v_i a_i = 0$,由此可得:$\xi = \dfrac{\sum_i f_i v_i}{m\sum_i v_i^2}$。

3.2 并行仿真算法研究

在 CUDA 平台上对碳纳米管进行分割,分割成多层大小适宜、相互独立的计算单元,分割原则是在避免过多重复计算的前提上提高并行度,即分割的计算单元必须合理粗细化。因为计算单元过大,则并行不明显,计算单元过小,会导致太多不必要的重复计算。由于粒子间的作用力间距影响较大,必须设定一个距离临界值,以判断其位置关系,当粒子间的间距小于距离临界值,为近邻粒子;当粒子间的间距大于距离临界值,为次邻粒子,则认为其相互间的分子作用力可以忽略不计。

可以预料到的是每个中心粒子最多只会对其 3 个近邻及 3 个近邻的各 2 个近邻,即 6 个次邻施力。可以这样来考虑,以中心粒子以及 3 个近邻 6 个次邻划分到一组计算。分批次并行计算。在某一批次计算中,未当成中心粒子计算的一个粒子加入可并行计算队列中,由于竞争关系,其 3 个近邻 6 个次邻在本批次中将不能加入队列,同样的,对这 3 个近邻6 个次邻施力的其他所有未计算粒子也不能加入队列中。这样,同一批次的粒子将可以并行计算而不产生竞争关系。

设计分裂算法,采用 CPU 遍历计算单元得到 n 个可并行运算队列:

(1)若所有粒子均已作为中心计算粒子,跳到步骤(8)。

(2)按次序找到第一个非竞争粒子并没有按中心计算的粒子,加入可并行运算队列。

(3)标记该粒子的所有近邻为本并行队列一度不可并行粒子,若近邻已经是二度不可并行粒子,则提升为一度不可计算。

(4)标记该粒子所有次邻为本并行队列二度不可并行粒子,若次邻为一度不可并行粒子,则不修改其度数。

(5)标记本粒子为已按中心计算粒子。

(6)判断是否已遍历到粒子队尾。若是继续执行,若否跳到步骤(1)。

(7)下一个可并行队列开始,返回步骤(2)。

(8)结束。

分裂算法在 CPU 预处理阶段完成,只计算一次得出的可并行队列可以在后面的循环计算中一直使用。当循环次数较多时,分裂算法所使用的时间占总时间比例将大大减少,对程序总体性能影响也降到比较低的程度。但是由于算法核心是批次计算,因此计算力时将多次启动内核计算函数,一定程度上加大了运行开销。

如图 5 和图 6 所示,调度 GPU 的流处理单元,采用 Verlet 算法进行并行运算和处理:

(1)按碳纳米管系统模型所有粒子的位置,计算近邻粒子以及次邻粒子间的键关系和角度关系。

(2)调度 GPU 的流处理单元,并行计算被分割在不同计算单元里的粒子,并积累计算碳纳米管管壁上每个粒子与其近邻粒子的相互作用力。

(3)根据 C60 分子所在区域,计算 C60 分子内每个粒子与其所在碳纳米管区域中的每个粒子的相互作用力(范德华力)。

(4)根据粒子所受到的力及其速度,更新粒子位置,再执行步骤(2)和步骤(3)。

(5)根据粒子所受到的力,计算粒子本次速度及碳纳米管前、后两端热浴的热流值。

(6)若达到循环频数,则计算结束;否则,在 CPU 端间隔保存粒子的数据,返回步骤(4)。

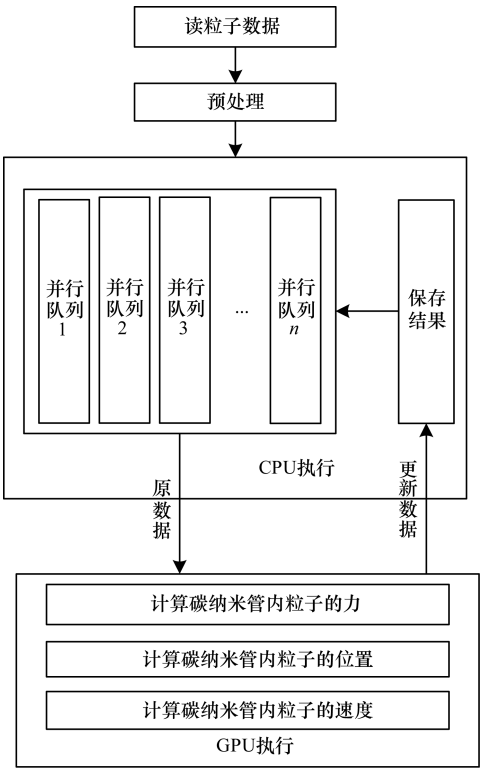


图 5　碳纳米管分子动力学并行仿真中的调度模式

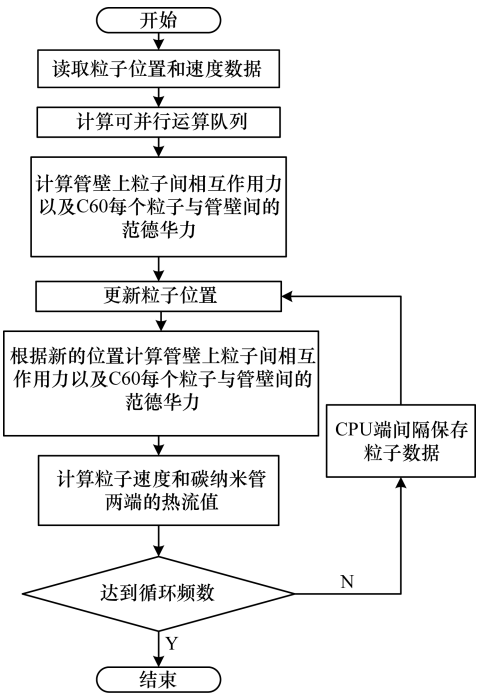


图 6　碳纳米管分子动力学并行仿真算法的流程

4　GPU 并行算法的优化

Verlet 仿真程序是一个计算力→计算位置→计算速度→计算力……的不断循环过程,每个粒子独立运行一个线程,然后循环成千上万次甚至更多,因此小部分的程序优化也会给整个程序带来极大的性能提升。对于 CUDA 的程序而言,优化主要集中在指令优化、存储器访问优化、网格优化以及资源均衡等。

4.1　指令优化

目前,GPU 单精度计算性能要远远超过双精度计算性能,因此,程序中对精度要求不高的部分可使用单精度运算代替双精度运算。在 GPU 中整数乘法、除法、求模运算的指令吞吐量较为有限,而控制流指令由于会影响指令发射器的并行发射,打断指令流开销巨大。此时,在可知整数范围中以经优化的库 24 位整数运算__mul24,__umul24 代替 32 位整数运算,以__sinf(x),__fdivide(x,y)等代替相应的运算。尽量避免整数除和模运算,或以(i&($n-1$))代替(i% n),除数是 2 的幂次方时以移位运算代替除法。

4.2　存储器访问优化

合理使用 share memory 资源,减少不必要的临时变量,使用精简函数以减少 register 使用。同时,将程序中的常量以及计算中可合并常量保存于 constant memory 常量存储器中。Global memory 是显存中容量最大的存储器,可被任意 GPU 线程读写存储器任意位置,但由于全局存储器没有缓存,具有较高的访存延迟。因此,程序中在减少不必要全局存储读写的同时,增加处理器占用率,即当一个 block 访存时,另一个 block 可切入运算隐藏延迟。此外,改变显存的存储结构,尽量满足合并访问存储器。

4.3　CPU + GPU 处理

考虑到 GPU 并行处理要发挥其强大的并行计算能力,要有足够多的线程并行运算。因此,把部分较小计算量的计算交给 CPU 执行。如计算 C60 与炭纳米管间范德华力函数,由于 C60 只有 60 个粒子,相对来说并行度小,将其计算放到 CPU 里。由于内核的启动是异步的,因此把 CPU 函数放到内核启动函数下面,CPU 启动内核执行 GPU 运算时可立即返回执行 CPU 程序,实现 CPU 与 GPU 的并行处理。

5　实验结果与分析

本文测试在 Widows 7 平台下进行,GPU1 和 GPU2 是 2 个不同型号的显卡,每次实验只选取其中一块显卡。为准备记录运行时间并方便 CPU 与 GPU 并行运行,测试时间以 CPU 时间为记录。部分硬件参数如表 1 所示。

表 1　测试平台部分硬件参数

器件	处理器	主频/MHz	核心数	内存或显存/GB
CPU	Intel Xeon W3550	3 066	4 核 8 线程	8
GPU 1	NVIDIA GeForce GT 430	1 400	96	2
GPU 2	NVIDIA Quadro 2000	1 251	192	1

分别使用 CPU 串行、GeForce GT 430 显卡和 Quadro 2000 显卡按不同粒子数计算，测试三者耗时，如表 2 所示。

表 2 串行并行程序运行于不同硬件平台时的数据

粒子层数	粒子数	CPU 串行	GPU1 耗时/ms	GPU 耗时/ms
50	1 000	15 912	4 462	4 134
100	2 000	29 172	6 412	5 632
200	4 000	55 210	9 625	7 379
300	6 000	85 172	12 168	9 282
600	12 000	163 544	21 656	14 352
900	18 000	238 150	30 919	19 828
1 200	24 000	322 000	40 763	25 178
1 500	30 000	400 878	50 552	30 420
3 000	60 000	835 889	98 155	56 855
6 000	120 000	1 675 152	194 038	109 996

5.1 不同 GPU 间并行算法性能对比

为使数据形象化，将 GPU1 与 GPU2 数据绘制成条状图。由图 7 可得，前期粒子数较少时，2 种显卡耗时相差不大，但随着粒子数的增加，GeForce GT 430 显卡耗时增长速度逐渐快于 Quadro 2 000 显卡。而当粒子数量增加到 120 000 个时，前者耗时几乎是后者的 2 倍。

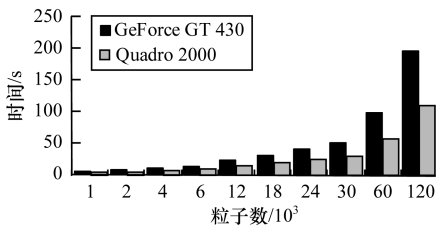


图 7 2 种并行显卡运行程序耗时对比

由表 1 可知，Quadro 2 000 有着 2 倍于 GeForce GT 430 的 CUDA Cores（即流处理器 stream processor）。结合 GPU 的线程运行方式，当 GPU 的处理单元增多，同时处于运行状态的线程也会相应增加。因此，当无其他硬件限制时，GPU 并行计算的性能几乎随着 GPU 处理器的增加而线性增加，使用拥有更多处理器的 GPU 能给 Verlet 带来更多的性能提升。

5.2 串并行算法性能对比

对比 CPU 串行程序与 GPU 并行程序的性能，计算不同粒子数下耗时，GPU 使用 Quadro 2000。

表格数据绘制成串并行程序执行时间对比如图 8所示。可以明显看出，当要计算的粒子数不断增加时，并行程序相比于串行程序的加速比有一定程度的增加。粒子数量在 1 000 ~4 000 时，由于线程数据不足，不能很好地隐藏 GPU 程序加载时间、数据传输时间以及访存时间。当粒子数量不断增加时，加速效果越来越好。

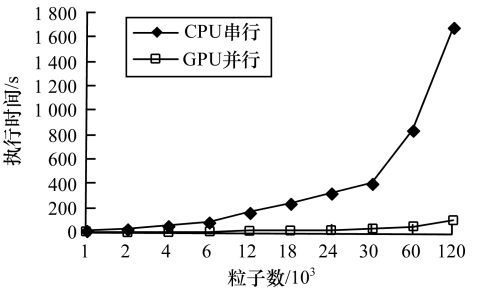


图 8 串并行程序执行时间对比

5.3 并行算法性能分析

为分析并行算法的加速比，将表 2 中串并行程序运算时间按公式：

$$\text{加速比} = \frac{\text{串行时间} - \text{并行时间}}{\text{并行时间}} \times 100\%$$

计算，GPU 使用 Quadro 2000，并将结果绘制成 GPU 并行计算加速比，如图 9 所示。

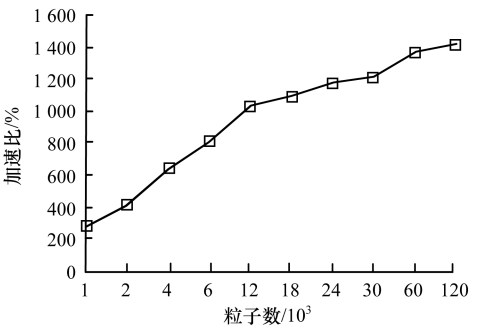


图 9 GPU 并行计算加速比

可以看出，随着粒子数的增加，GPU 并行算法相比于串行算法的加速比也随之增大。当粒子数到达 120 000 个时，加速比达到 1 400%，即 14 倍的加速效果。

为分析影响 GPU 并行程序性能因素，使用性能评估工具 NVIDIA Visual Profiler 对程序进行分析。目标程序计算 600 层 12 000 个粒子，每个 block 中有 128 个线程。首先分析各函数运行时间占总运算时间的比例。通过多次运算求平均值，如图 10 所示。

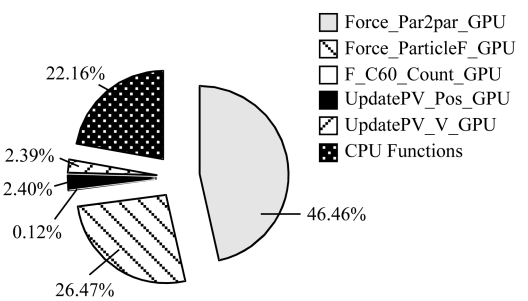


图 10 各个函数的比重

6　结束语

在对碳纳米管分子动力学进行仿真时,原来的分子动力学仿真算法有良好的效果,但因碳纳米管系统粒子数的规模太大、计算量多、运行时间长,缺乏实用性。而由用 Verlet 算法实现的分子动力学仿真算法具有易并行性,可以在具有强大图像处理和浮点计算能力的 GPU 上并行处理,从而大大提高了碳纳米管分子动力学仿真算法的效率。

实验结果表明,对小规模仅有 120 000 个粒子组成的碳纳米管系统,在拥有 192 个流处理器的 NVIDIA Quadro 2000 显卡上实现的 GPU 并行仿真算法,与在拥有 4 核 8 线程的 Intel Xeon W3550 上实现的基于 CPU 的串行仿真算法相比,其加速比即可达到十多倍,从而有效地解决了算法速度上的缺陷。

然而由于本次实验的硬件条件有限,所能计算的碳纳米管系统规模有限,因此无法深度发掘 CUDA 的能力。而由实验结果所示,碳纳米管系统的粒子数规模越大,GPU 并行算法对于串行算法的加速比就越明显。在实际运用中,碳纳米管系统的粒子数的规模都很大,超过千万乃至十亿,可以预估该基于 GPU 的分子动力学并行仿真算法,对碳纳米管的研究会发挥至关重要的作用。

参考文献

[1] Alder B J, Wainwright T E. Phase Transition for a Hard Sphere System[J]. The Journal of Chemical Physics, 1957, 27(5): 1208-1209.

[2] 唐玉兰,胡　适,王东旭,等. 纳米工程中大规模分子动力学仿真算法的研究进展[J]. 机械工程学报, 2008, 44(2): 8-15.

[3] Landman U, Luedtke W D, Burnham N, et al. Atomistic Mechanisms and Dynamics of Adhesion, Nanoindentation, and Fracture[J]. Science, 1990, 248(4954): 454-461.

[4] Schiøtz J, Jacobsen K W. A Maximum in the Strength of Nanocrystalline Copper[J]. Science, 2003, 301(5638): 1357-1359.

[5] 徐　伟,王　丽. MD 算法在集群系统中的并行实现研究[J]. 计算机工程, 2002, 28(3): 103-105.

[6] Proctor A J, Lipscomb T J, Zou Anqi, et al. Performance Analyses of a Parallel Verlet Neighbor List Algorithm for GPU-optimized MD Simulations[C]//Proceedings of International Conference on Biomedical Computing. Washington D. C., USA: IEEE Press, 2012: 14-19.

[7] Anderson J A, Lorenz C D, Travesset A. General Purpose Molecular Dynamics Simulations Fully Implemented on Graphics Processing Units[J]. Journal of Computational Physics, 2008, 227(10): 5342-5359.

[8] Verlet L. Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules[J]. Physical review, 1967, 159(1): 98-103.

[9] Tamayo P, Mesirov J P, Boghosian B M. Parallel Approaches to Short Range Molecular Dynamics Simulations[C]//Proceedings of Conference on Supercomputing. New York, USA: ACM Press, 1991: 462-470.

[10] 曹　伟,宋雪梅,王　波,等. 碳纳米管的研究进展[J]. 材料导报, 2007, 21(5): 77-82.

[11] 刘文志,李晓霞,余　翔,等. 分子动力学模拟中基于 GPU 的范德华非键作用计算[J]. 计算机与应用化学, 2010, 27(12): 1607-1612.

[12] 孟小华,刘坚强,区业祥,等. 基于 CUDA 的拉普拉斯边缘检测算法[J]. 计算机工程, 2012, 38(18): 190-193.

[13] 孟小华,黄丛珊,朱丽莎. 基于 CUDA 的热传导 GPU 并行算法研究[J]. 计算机工程, 2014, 40(5): 41-44, 48.

编辑　顾逸斐

(上接第 287 页)

[7] Cohen D M, Dalal S R, Fredman M L, et al. The AETG System: An Approach to Test Based on Combinatorial Design[J]. IEEE Transactions on Software Engineering, 1997, 23(7): 437-444.

[8] 惠战伟,黄　松,苏士瀚,等. 成对覆盖组合测试用例生成方法研究[C]//2010 亚太地区信息论学术会议论文集. [出版地不详]: 中国电子学会信息论分会, 2010: 264-267.

[9] Yu Lei, Tai K C. In-parameter-order: A Test Generation Strategy for Pairwise Testing[D]. Raleigh, USA: North Carolina State University, 2001.

[10] Tai K C, Yu Lei. A Test Generation Strategy for Pairwise Testing[J]. IEEE Transactions on Software Engineering, 2002, 28(1): 109-111.

[11] 严　俊,张　健. 组合测试:原理与方法[J]. 软件学报, 2009, 20(6): 1393-1405.

[12] 史　亮,聂长海,徐宝文. 基于解空间树的组合测试数据生成[J]. 计算机学报, 2006, 29(6): 849-857.

[13] 戴　丽. 组合测试用例生成技术的研究与应用[D]. 广州:华南理工大学, 2011.

编辑　顾逸斐