

GPU 加速分子动力学模拟的热力学量提取*

刘丹, 赵广辉, 夏红霞, 胡磊

(武汉理工大学计算机科学与技术学院, 武汉 430070)

摘要: 近年来, 统一计算设备架构(CUDA)的提出和图形处理器(GPU)快速提升的并行处理能力和数据传输能力, 使得基于 CUDA 的 GPU 通用计算迅速成为一个研究热点。针对含有大规模分子动力学模拟的热力学量提取效率低下的问题, 提出了分子动力学模拟的热力学量提取的新方法, 利用 CUDA 设计了并行算法, 实现了利用 GPU 加速分子动力学模拟的热力学量提取。实验结果表明, 与基于 CPU 的算法相比, GPU 可以提高速度 500 倍左右。

关键词: 图形处理器; 统一计算设备架构; 分子动力学模拟; 热力学量

中图分类号: TP391.9 **文献标志码:** A **文章编号:** 1001-3695(2010)05-1820-03

doi: 10.3969/j.issn.1001-3695.2010.05.060

Extraction of thermodynamic quantities of MD simulation with GPU

LIU Dan, ZHAO Guang-hui, XIA Hong-xia, HU Lei

(School of Computer Science & Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract: Recently, the compute unified device architecture (CUDA) and the rapidly promoting ability of data-parallel processing and data transferring of graphic processing units (GPU) make an attractive area for general purpose computation based on CUDA. As the extraction of thermodynamic quantities in large scale molecular dynamics simulation was relative low, this paper proposed a new approach to the extraction of thermodynamics quantities of molecular dynamics simulations, and used the CUDA to design and implement parallel algorithm. The results indicate that GPU achieves very high speedup of up to near 500 folds compared with CPU-based algorithm.

Key words: GPU; CUDA; molecular dynamics simulation; thermodynamic quantity

0 引言

图形处理器 (graphic processing unit, GPU) 是目前通用计算机系统中专门从事图形计算的单元, 与常规 CPU 不同的是, GPU 中大部分晶体管被用来进行数据处理, 只有极少数用于数据缓存和指令流控制。GPU 特别适合用来处理单指令多数据流 (single instruction multiple data, SIMD) 任务, 适用于算法密集、高度并行、控制简单、分多个阶段执行且吞吐量比延迟重要的应用^[1]。由于 GPU 是专门为图形图像处理而设计的硬件芯片, 体系结构较 CPU 大大简化, 决定了其在数据流处理上可以获得较高的效率^[2]。最新的 GPU 的浮点计算能力和内存带宽已经大大超过最快的 CPU, 并且可以进行大规模的双精度浮点数并行计算。一个类 C 语言的开发环境——统一计算设备架构 (computer unified device architecture, CUDA) 技术的诞生开创性地改革了 GPU 的编程模式及存储模式, 使得 GPU 的并行计算能力得到极大的发挥。目前, CUDA 已经被用于分子动力学^[3]、生物信息学^[4]、地球物理学^[5]等需要将大量数据并行计算的领域中。

分子动力学 (MD) 模拟是指对于原子核和电子所构成的多体系统, 通过分析系统中各个粒子的受力情况, 用经典或量

子的方法求解系统中各个粒子的位置和速度以确定粒子的运动状态, 从而计算系统的结构与性质。MD 方法可以直接模拟许多宏观现象, 取得许多与实验相符合的实验数据, 比较准确地提取粒子轨迹、压力、温度、势能等统计宏观量。目前在物理、化学、材料、摩擦学等学科以及纳米机械加工中得到广泛而成功的应用。热力学量的提取是分子动力学模拟的关键环节之一, 对分子动力学模拟效率有非常重要的影响, 利用 CPU 实现热力学量提取计算量大, 计算时间长, 对 CPU 处理能力有较高的要求。

本文基于 CUDA 平台, 利用 GPU 强大的并行计算能力和高效率的数据传输能力^[6,7], 实现了基于 GPU 的分子动力学模拟的热力学量提取, 将耗时较多的力计算、势能计算交给 GPU 完成。实验结果表明, 基于 GPU 的热力学量提取方法与 CPU 方法相比, 执行效率明显提高, 可获取 500 倍左右的加速。

1 GPU 计算能力及 CUDA 概述

GPU 是一个具有非常强大的并行处理能力和高效率数据传输能力的多线程多核处理器。其中, 并行性主要体现在指令级、数据级和任务级三个层次。高效率的数据传输能力主要体现在两个方面: GPU 与显存之间的高带宽; 系统内存到显存的

收稿日期: 2009-10-10; **修回日期:** 2009-11-25 **基金项目:** 国家教育部高校行动计划智能科学与技术项目 (2004XD-03)

作者简介: 刘丹 (1983-), 男, 湖北武汉人, 硕士研究生, 主要研究方向为计算机模拟、高性能计算 (liudan@whut.edu.cn); 赵广辉 (1973-), 男, 讲师, 博士研究生, 主要研究方向为材料计算、计算机模拟、高性能计算; 夏红霞 (1960-), 女, 教授, 硕导, 主要研究方向为网络数据库、软件工程; 胡磊 (1986-), 男, 硕士研究生, 主要研究方向为智能计算。

高带宽。NVIDIA Fermi (GT300) 芯片的浮点运算能力将有望达到 2.46Tflops,且显存带宽峰值为 281.6 Gbps。从图 1 和 2 可看出,近年 GPU 的浮点处理能力和内存带宽的增长很快,均大大超过同级别的 CPU,高浮点运算能力和高带宽使 GPU 应用于高性能计算具有 CPU 无法比拟的先天优势。Fermi (GT300) 芯片拥有超过 30 亿的晶体管个数、512 个核心 (core)。由于 GPU 将绝大多数晶体管用于数据处理,而不是将更多的晶体管投入到数据缓存和流量控制,这使得它对于计算密集型的科学应用有非常大的吸引力,越来越成为通用计算的一个有效的并行平台^[8,9]。

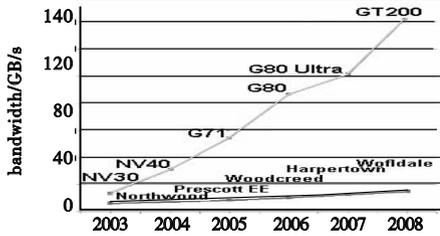


图1 CPU与GPU内存带宽比较

为了使 GPU 超强的计算能力在数据处理和科学计算等通用计算领域发挥优势,NVIDIA 于 2007 年推出了统一计算设备架构(CUDA)。CUDA 是一个全新的软硬件架构,可以将 GPU 视为一个并行数据计算的设备,有助于解决科学及工程方面复杂计算问题。CUDA 提供了对 G80/G9x/GT200/Fermi 核心显卡的支持,同时也能够在 Red Hat Enterprise Linux、OpenSUSE、SUSE Linux Enterprise Desktop 以及 32 bit/64 bit Windows XP 下进行 CUDA 的显卡编程。

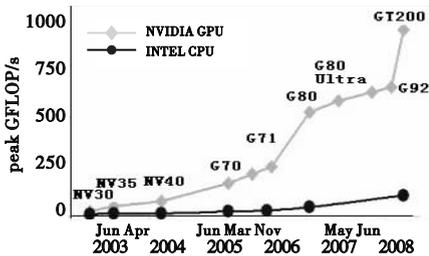


图2 CPU与GPU浮点运算能力比较

2 热力学量

微正则系综(NVE)是孤立、保守的系统,即所模拟的系统中分子数 N 、体积 V 、能量 E 都不变。本文选用微正则系综(NVE)作为平衡态体系来进行感兴趣的热力学量的提取。

2.1 压力

$$P = \frac{1}{V}NK_B T + \frac{1}{DV} \left(\sum_i r_i F_i \right)$$

这是著名的 Virial 方程。其中: P 为压力; V 为体系体积; T 为系统温度; N 为系统总的粒子数目; D 系统维数; K_B 为 Boltzmann 常数; F_i 为原子 i 所受的原子间作用力; r_i 为原子 i 的位置。

2.2 单原子平均势能

若计算的微正则系综(NVE)包含 N 个原子,且原子间的作用为成对加成(pair-wise additive),则系综的单原子平均势能为

$$\bar{E} = \frac{1}{N} \sum_i \sum_j U_{ij}(r_{ij})$$

即对系统中每个原子间的势能加和平均,以便于一些理论获得的结果对比、对照。

3 基准问题

分子动力学模拟通常由时间来推进体系演化。时间被离散成许多个时间点,两个相邻时间点的距离称为时间步长,分子动力学算法一般采用等时间步长。分子沿着这些时间点集合运动,每次依据时间点的顺序往前推进一个时间步长到下一个时间点,更新分子位置和速度。

由于 Velocity-Verlet 分子运动路径积分方法更新位置和速度的误差非常小,在较长时间能保持能量守恒,本实验采用 Velocity-Verlet 分子运动路径积分方法来更新分子位置和速度,具体实现为

$$\begin{cases} v(t + \Delta t/2) = v(t) + f(t) \Delta t / (2m) \\ r(t + \Delta t) = r(t) + \Delta t v(t + \Delta t/2) \\ v(t + \Delta t) = v(t + \Delta t/2) + f(t + \Delta t) \Delta t / (2m) \end{cases}$$

Velocity-Verlet 路径积分在分子动力学中需要分成两部分完成:a)将 t 时刻分子速度 $v(t)$ 根据 t 时刻的分子受力 $f(t)$ 更新到 $v(t + \Delta t/2)$,从而把分子位置 $r(t)$ 更新到 $r(t + \Delta t)$;b)由 $t + \Delta t$ 时刻的各分子位置计算受力 $f(t + \Delta t)$,把分子速度由 $v(t + \Delta t/2)$ 更新到 $v(t + \Delta t)$,完成分子位置和速度的同步。

选择合适势函数来描述系统内分子间相互作用是进行分子动力学模拟的基础,其决定了计算的工作量和计算结果与实际物理的接近程度。本实验选用分子动力学常采用的 L-J 势(Lennard-Jones),表达式为

$$u(r_{ij}) = 4\epsilon \left[a \left(\frac{\delta}{r_{ij}} \right)^{12} - b \left(\frac{\delta}{r_{ij}} \right)^6 \right]$$

其中: ϵ 为作用势能; δ 为作用特征长度(一般为分子直径); r_{ij} 为分子 i 到分子 j 的距离;系数 a, b 表示斥力项和引力项的大小,选定 $a = b = 1$ 。由此分子 i 到分子 j 的作用力 F_{ij} 为

$$F_{ij} = -\nabla u(r_{ij})$$

这种作用方式被称为软球作用,分子间的力作用是可叠加的,作用顺序对结果无影响。分子的所有受力计算完成后,就能更新分子的速度和位置信息,对分子的运动方程进行积分。同时,分子动力学模拟中分子运动遵循牛顿运动定律:

$$F_i = m_i a_i$$

由于受计算机硬件的限制,不可能对一个真正的宏观系统直接实施模拟,通常选取小体系作为研究对象。但由于位于表面和内部的分子受力差别较大,将会产生表面效应,为消除此效应,本实验引入了周期性边界条件。通过周期性边界条件的设置,不必计算巨大数目的分子来得到物质的宏观性质,只需有限的分子即可,大大降低了计算的工作量。

在计算粒子所受邻近粒子作用力时,本实验选用截断半径法来避免对所有粒子进行逐个计算。该方法假定存在一个球半径 r_c ,如果两个粒子间的距离 $r > r_c$,则两粒子间的势能为 0,忽略超过一个截断半径 r_c 的原子间的相互作用力,大大减少了计算时间。

4 性能测试及分析

本文的实验平台:处理器 CPU 为 Pentium® Dual-Core 2.5 GHz,4 GB 内存, Windows XP 操作系统, GPU 为 NVIDIA 公司的 GeForce GTX 260, GT200 核心,其核心频率为 576 MHz,显存频率为 999 MHz,显存位宽为 448 bit、显存带宽为 111.9 GB/s、显存容量为 896 MB。编译环境为 CUDA2.3 版本,在 VS2005 的环境下进行了程序的编译与执行。

内核配置情况为线程块内 128 个线程,由于在 GPU 上建立了线程与计算粒子之间的对应,并为每一个粒子单独分配一个 GPU 中的计算线程,则线程块数应为体系分子数除以块内线程数。设定积分步程为 100,进行基于 CPU 和基于 GPU 的不同分子数体系的分子动力学模拟的压力和单原子平均势能的提取。其时间开销如图 3 所示,加速比如图 4 所示。

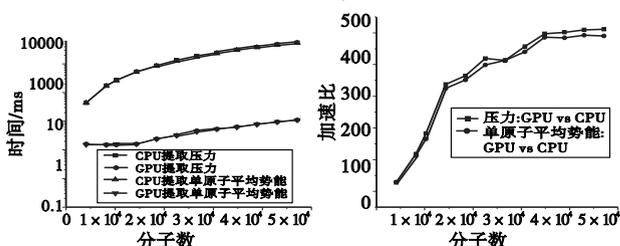


图3 提取时间随分子数的变化 图4 加速比随分子数的变化

从图 4 可以看出,利用 GPU 进行热力学量提取的加速比,随着体系规模的增大呈增大趋势,当分子达到一定数目时,计算效率不再有较大的提升,相对于 Pentium® 2.5 GHz 双核 CPU 可以获得 500 倍左右的速度提升。

图 5 和 6 分别是利用 CUDA 的 GPU 占用率计算器统计的内核函数 GPUGetPressure()、GPUGetPotEnergy() 使用不同大小线程块和不同大小线程占用寄存器的多处理器线程束占用率。

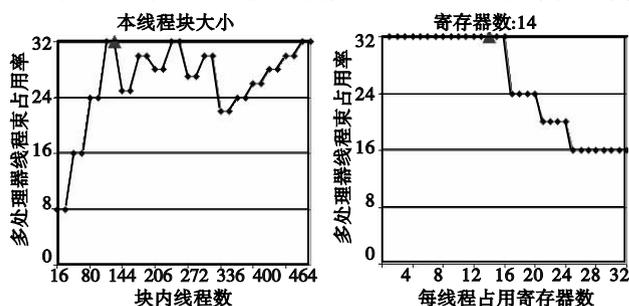


图5 GPU占用率随块内线程数的变化 图6 GPU占用率随每线程占用寄存器数的变化

从图 5 可以看出,当使用较大的线程块时, GPU 的占用率要比较小的线程块高,反映出了线程块大小对计算性能的影响。从图 6 可以看出,当使用较少的寄存器数时, GPU 的占用率要比较多寄存器数高,反映出了线程占用寄存器数多少对计算性能的影响。

5 结束语

为了加速大规模分子动力学模拟的热力学量提取速度,本文利用 CUDA 开发包实现了基于 GPU 的分子动力学模拟的热力学量提取,其方法的关键是将占热力学量提取的耗时较多的计算任务映射到 GPU 上去执行;将相同的热力学量提取的计算任务交给 CPU 和 GPU 完成,发现利用 GPU 完成的提取任务相对于 Pentium® 2.5 GHz 双核 CPU 可以获得 500 倍左右的速度提升。分析多处理器线程束占用率随线程块内线程数、每线程占用寄存器数的变化,得出利用 GPU 进行计算时, block 内线程数、线程占用寄存器数的选取应尽量适中。

参考文献:

- [1] 田绪红,江敏杰. GPU 加速的神经网络 BP 算法[J]. 计算机应用研究,2009,26(5):1679-1681.
- [2] 王磊,张春燕,基于图形处理器的通用计算模式[J]. 计算机应用研究,2009,26(6):2356-2358.
- [3] STONE J E, PHILLIPS J C, FREDDOLINO P L, et al. Accelerating molecular modeling applications with graphics processors [J]. Journal of Computational Chemistry,2007,28(16):2618-2640.
- [4] MANAVSKI S A, VALLE G. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment [J]. BMC Bioinformatics,2008,9(2):S10.
- [5] PORTEGIES ZWART S F, BELLEMAN R G, GELDOP P M. High performance direct gravitational N-body simulations on graphics processing units [J]. New Astronomy,2007,12(8):641-650.
- [6] NVIDIA. CUDA compute unified device architecture; programming guide, vrsion 2.3 [K]. 2009.
- [7] NVIDIA. Oficial site [EB/OL]. http://www.nvidia.com.
- [8] OWENS J D, LUEBKE D, GOVINDARAJU N, et al. A survey of general-purpose computation on graphics hardware [J]. Computer Graphics Forum,2007,26(1):80-113.
- [9] MARK W R, GLANVILLE R S, AKELEY K, et al. Cg: a system for programming graphics hardware in a C-like language [J]. ACM Trans on Graphics, 2003,22(3):896-907.

(上接第 1808 页)的构件分类管理以及平台集成框架下业务构件的开发调试等进行了研究。上述研究成果已经在汽车产业链协同工作平台的开发中得到应用验证,汽车产业链协同工作平台(http://auto.easp.cn)是一个以 SaaS 模式运行的第三方公共服务平台,平台支持以整车和零部件制造厂为核心的多个联盟开展整车协同销售、整车及零部件协同售后服务以及零部件的协同采购等,目前平台已应用企业 3 000 多家。平台构件库中集成了各类构件 800 多种,当企业业务需求增加或发生变化时,可以通过构件库快速匹配或集成开发需要的构件,在平台集成框架下进行注册和组装后,即可授权给企业使用^[1]。

参考文献:

- [1] 王淑莹.面向制造业产业链的协同商务平台集成框架[J]. 西南

- 交通大学学报,2008,43(5):643-647.
- [2] 王淑莹.一种面向制造业产业链协同商务系统的数据交换适配器研究[J]. 计算机应用研究,2009,26(1):189-191.
- [3] KIM S G. Designing a domain framework with component management model[J]. Journal of Software,2002,13(3):335-341.
- [4] 常志明,毛新军,齐治昌.基于 agent 的网构软件模型及其实现[J]. 软件学报,2008,19(5):1113-1124.
- [5] 张涌,王渊峰,钱乐秋.一个集成式的软件描述框架[J]. 计算机学报,2002,25(5):502-508.
- [6] 盛津芳,王斌,陈松乔.方面化构件模型及其组装方法[J]. 计算机工程,2006,32(5):39-40,45.
- [7] 王卓.模型驱动的软件分类及检索方法研究[D].哈尔滨:哈尔滨工程大学,2005.