

基于CUDA高性能并行计算的边界面法研究

余 列祥*, 张 见明

湖南大学汽车车身先进设计制造国家重点实验室, 湖南长沙, 410082

摘要

随着图形处理器(GPU)的计算能力和可编程性的不断提高, 利用 GPU 进行通用计算(GPGPU)逐渐成为研究的热点^[1]。CUDA(Compute Unified Device Architect)^[2]是用于 GPU 计算的开发环境, 它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, 对所进行的计算进行分配和管理。本文首次在边界面法^[3-4]中研究规则单元积分计算的并行化, 已知正则化边界积分方程离散为关于 N 个节点未知量的方程为:

$$0 = - \sum_{j=1}^{N_E} \int_{\Gamma_j} q^s(s, y) \sum_{k=1}^N (N_k(s) - N_k(y)) u_k d\Gamma + \sum_{j=1}^{N_E} u^s(s, y) \sum_{k=1}^N N_k(s) q_k d\Gamma$$

组装成矩阵形式为:

$$Hu = Gq$$

其中,

$$H_{ik} = \sum_{j=1}^{Num} \int_{\Gamma_j} q^s(s, y_i) (N_k(s) - N_k(y_i)) d\Gamma$$

$$G_{ik} = \sum_{j=1}^{Num} \int_{\Gamma_j} u^s(s, y_i) N_k(s) d\Gamma$$

由于场点对规则单元源点的积分计算相互独立, 具有很好的并行性, 因此运用CUDA进行这一过程的计算会在一定程度上提高规则单元积分的计算效率。

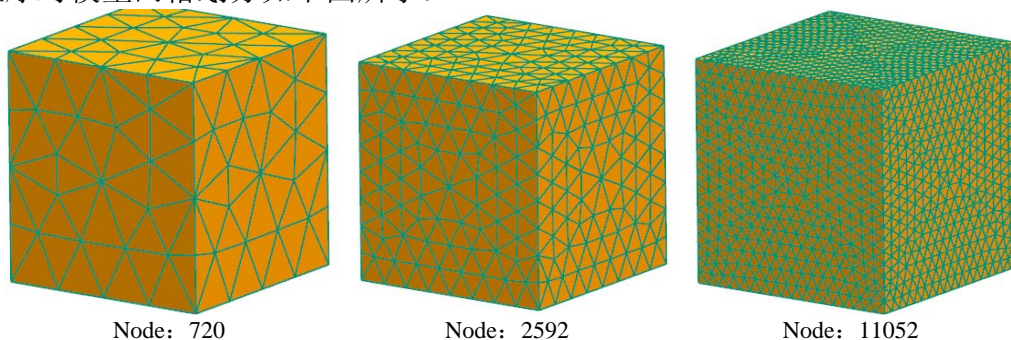
为验证CUDA对规则单元积分计算带来的效率提升, 本文分析了一个立方体模型, 对其进行3种不同网格尺寸的划分, 网格尺寸越密, 节点数越多, 得到的数据模型越大。用UG8.0建立分析模型, 结合UG二次开发进行CUDA并行代码开发。

计算平台为:

硬件平台: NVIDIA GTS450显卡, 2×PCI-E X16显卡插槽, Intel(R) Core(TM) i7-2600 CPU 3.40GHZ, 16GDDR3内存。

软件平台: Windows 7 64bit, Microsoft Visual Studio 2008, UG 8.0 64bit, CUDA4.0 64bit

运用边界面程序对模型网格划分如下图所示:



其中，Node为节点数，程序运行结果对比如下表所示：

Node	CPU_Time	GPU_Time			精度	加速比
	Regular_CPU	Regular_GPU	Translate_GPU	Compute_GPU		
720	0.047	0.078	0	0.078	0.624%	0.603
2592	0.600	0.156	0	0.14	0.2232%	3.846
11052	10.506	2.465	0.048	2.058	0.0826%	4.262

其中，Regular_CPU为CPU串行代码执行的时间，Regular_GPU为GPU并行代码执行的时间，Translate_GPU为数据在显存与内存之间传输的时间，Compute_GPU为内核函数计算时间。精度在CPU上和GPU上保持高度一致。

由表可以看出，在节点数Node较小时，即计算规模小，GPU未能达到加速的目的，反而比CPU计算得慢；在节点数Node逐渐变大，计算规模也逐步增加，GPU开始显示出其并行计算的优势，且计算规模越大，加速比将会越大，这是因为GPU只有在进行大规模计算的时候才会体现出其高性能并行计算能力。数据传输时间占GPU并行代码执行时间80%~90%以上，可见GPU的带宽成为了程序的瓶颈，通过升级硬件便可有效解决，加速比会有大幅度提升。

关键字：边界面法，GPU，CUDA，并行计算，UG二次开发。

参考文献：

1. 陈波. 基于 CPU-GPU 异构平台的性能优化及多核并行编程模型的研究. 合肥:中国科学技术大学, 2011.
2. 张舒. 褚艳丽. GPU 高性能运算之 CUDA. 北京: 中国水利水电出版社, 2009.
3. Zhang JM, Qin XY, Han X, Li GY. A boundary face method for potential problems in three dimensions. *Int. J. Num. Meth. Eng.*, 2008; **80**: 320-337.
4. 覃先云. 基于参数曲面的边界面法研究. 长沙:湖南大学, 2009.