

文章编号: 1004-4574(2012)05-0016-07

基于高性能 GPU 计算的城市建筑群震害模拟

韩 博¹, 陆新征¹, 许 镇¹, 李 易^{1, 2}

(1. 清华大学土木工程系, 土木工程安全与耐久教育部重点实验室, 北京 100084;
2. 北京工业大学 城市与工程安全减灾教育部重点实验室, 北京 100124)

摘 要: 为适应不断提高的震害预测需求, 精细化模型已经成为城市区域震害预测的主要发展方向。然而传统的基于 CPU 平台的计算方法成本过高, 使得精细化模型的应用受限。近年来, GPU 技术由于其强大的并行计算能力和较为低廉的价格优势, 在通用计算领域得到了快速的发展和应用。基于 GPU-CPU 协同计算技术, 建立了城市区域震害的计算模拟方法, 显著缩短了城市区域震害预测的时间。并采用该方法对某中型城市的真实震害进行了模拟应用, 展示了 GPU 技术在大区域城市群震害模拟中的独特优势。

关键词: 城市区域; 震害模拟; GPU; 并行计算

中图分类号: TU984 文献标志码: A

Seismic damage simulation of urban buildings based on high performance GPU computing

HAN Bo¹, LU Xinzhen¹, XU Zhen¹, LI Yi^{1, 2}

(1. China Key Laboratory of Civil Engineering Safety and Durability of China Education Ministry, Department of Civil Engineering, Tsinghua University, Beijing 100084, China;
2. Key Laboratory of Urban Security and Disaster Engineering of Ministry of Education, Beijing University of Technology, Beijing 100124, China)

Abstract: To meet the continuously increased needs for seismic damage prediction, refined models have become an important development trend in urban regional seismic damage prediction. However, the application of refined models is limited by the high cost of computation if it is implemented on traditional CPU platform. In recent years, a technology with powerful parallel computing capability and lower cost, GPU technology, has been developed and applied rapidly. In this paper, a simulating method for urban regional seismic damage that can significantly reduce the computation time, was developed based on GPU-CPU cooperative computing technology. The proposed method was then applied to the seismic damage simulation of a medium-sized city to illustrate the unique advantages of GPU technology in large-scale regional seismic damage simulation.

Key words: urban region; seismic damage simulation; GPU; parallel computing

我国是世界上地震灾害最为严重的国家之一, 科学预测地震灾害对防震减灾有着极其重要的意义。目前世界各国都在不断改进传统震害预测方法, 而其中的一个重要方面就是通过一定的建模方法, 将城市中的所有建筑都建立起相应的计算模型, 从而得到城市区域震害预测精细化模型, 以满足震害预测准确、高效和

收稿日期: 2011-12-14; 修回日期: 2012-03-02

基金项目: 国家自然科学基金重点项目(51178249); 城市与工程安全减灾教育部重点实验室开放基金项目; 北京市重点实验室开放基金项目(EESR2010-03); 清华大学自主科研项目(2010Z01001)

作者简介: 韩博(1989-), 男, 硕士研究生, 主要从事城市区域抗震研究。E-mail: b-han11@mails.tsing.edu.cn

多样化的需求。如日本东京大学地震研究所开展的“Full computation of earthquake hazards and disasters”项目,通过建立整个东京市区建筑群的精细化地震模型,利用日本的超级计算机,完成了整个东京市地震震害预测的模拟,取得了很多重要成果。但是,计算平台是这种方法的一个重要制约,特别是需要迅速得到模拟结果时,必须依赖非常强大的计算机系统。现有城市区域震害精细化模拟很多都是基于超级计算机平台,其价格昂贵、体积庞大、维护代价高昂,难以得到广泛推广。因而目前城市区域震害精细化模拟还大都只停留在“可行”而非“可用”的阶段^[1-4]。

近年来,随着图形处理器(graphic processing unit ,GPU) 技术的飞速发展,为城市区域震害精细化模拟的应用打开了一个全新的天地。传统的计算都是基于中央处理器(central processing unit ,CPU) 的,而 CPU 发展遵循摩尔定律,单核性能越来越强劲,但核心数较少,且价格昂贵。而对于 GPU,虽然每个计算核心的能力都不是很强,但是 GPU 里面集成了成百上千个计算核心。这样一来,其总浮点计算能力就远远超过了 CPU,且价格低廉,其价格只有同等传统 CPU 计算机的百分之几。特别是 NVIDIA 公司推出了统一计算架构(compute unified device architecture ,CUDA) 之后,大大降低了 GPU 通用计算程序的开发难度。目前 GPU 计算已经在生物、电磁、地球等学科中发挥了越来越重要的作用^[5-7]。

当然,GPU 计算并非十全十美,由于 GPU 中每个核心的计算能力有限,所以一个计算任务中子任务的数量越多,且单个子任务的工作量大小适中,子任务间数据交互越少,就越能充分发挥 GPU 计算强大的能力。而城市区域建筑群震害模拟就具有这个特点:一个城市区域中建筑物的数量成千上万,但对于一般单体建筑而言,层模型等宏观模型已经可以满足其震害预测的精度需要,每个单体非线性模型的自由度不多,且不同建筑物之间在地震下的相互影响不大。这正是最能充分发挥 GPU 核心能力的计算领域。对于动辄几百个核心的 GPU 而言,由每个 GPU 核心负责一个单体建筑的分析,这样即便是一座城市有成千上万栋建筑,也只需要数次任务分配就可以批量完成,效率极高。

GPU/CPU 协同计算为城市区域震害预测提供了良好的硬件条件,但要真正实现基于 GPU/CPU 协同计算的城市区域震害精细化模拟,尚有很多问题有待解决。本文对基于 GPU 的大区域城市群震害模拟进行了初步研究,其手段和成果可供有关研究人员参考。

1 计算流程与模型选择

本研究采用显式动力弹塑性时程分析来计算结构的动力响应,其计算程序流程如图 1

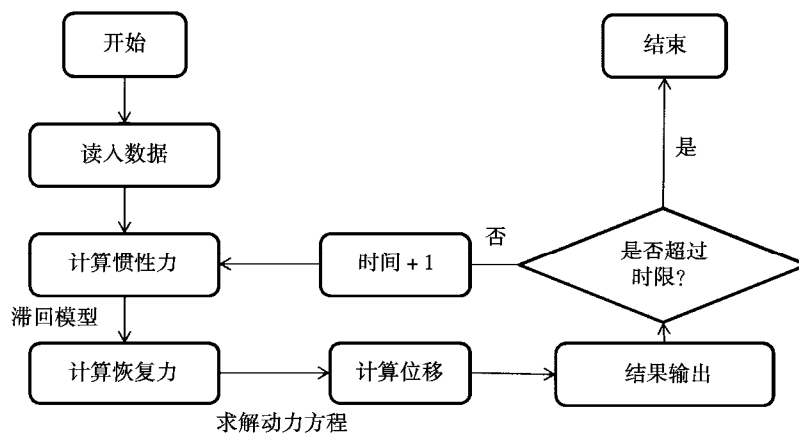


图 1 单座建筑地震反应分析流程

Fig. 1 Seismic response analysis process for a single building

在大多数情况中,城市群中多为普通多层建筑,因此在本研究中,采用剪切层模型进行模拟。层间剪力-层间位移滞回模型可采用平行四边形弹塑性模型或陆新征-曲哲 10 参数滞回模型^[8]。平行四边形模型计算最为简便,且不同建筑的不同受力阶段计算量相当,故选取其用来进行计算效率的比较。陆新征-曲哲 10 参数滞回模型可以模拟屈服、强化、软化等比较复杂的层间行为,具有较强的通用性,通过调整参数,可以得到比较精确的模拟结果,因此用来进行真实城市群倒塌震害的模拟。

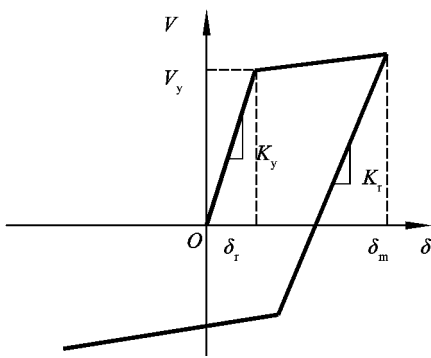


图 2 平行四边形滞回模型示意图

Fig.2 Schematic diagram of the bilinear hysteretic model

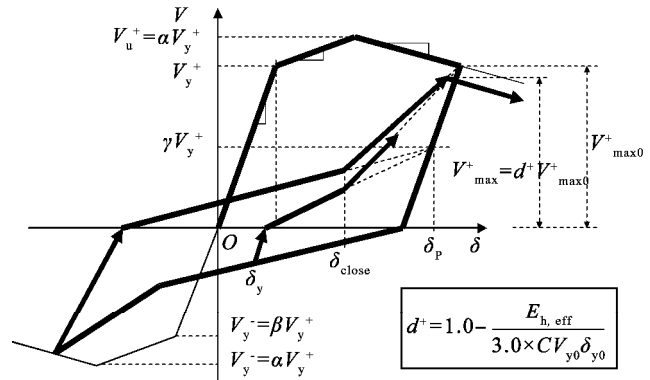


图 3 陆新征-曲哲滞回模型示意图

Fig.3 Schematic diagram of Lu-Qu hysteretic model

在 GPU 计算中,如果存在大量的逻辑分支运算,则计算效率会显著下降。且本文的震害模拟需要考虑倒塌问题,隐式计算方法存在非线性迭代不收敛的问题,综合上述两点考虑,本文采用中心差分法^[10]进行动力方程求解。其计算公式为

$$\left(\frac{m}{\Delta t^2} + \frac{c}{2\Delta dt}\right) u_{n+1} = P_n - \left(k - \frac{2m}{\Delta t^2}\right) u_n - \left(\frac{m}{\Delta t^2} - \frac{c}{2\Delta dt}\right) u_{n-1}, \quad (1)$$

其中阻尼矩阵采用经典瑞雷阻尼,式中参数意义可参见文献[10]

2 计算程序架构

2.1 设计原则

- (1) 利用 CPU 强大的逻辑分析功能完成数据的读取和计算任务分配,利用 GPU 强大的并行处理能力完成单体建筑物的非线性时程计算。
- (2) 通过合理的程序架构设计,减少因 CPU、GPU 数据交换等导致的计算效率下降。

2.2 CPU 端功能

- (1) 读取地震动数据和结构参数数据,并将其存储于系统内存中;
- (2) 将 CPU 和 GPU 之间需要交换的数据在系统内存和显存上分别进行存储空间分配;
- (3) 将数据在系统内存和显存之间进行拷贝,以供 CPU 或 GPU 使用;
- (4) 调用 GPU 端 CUDA 语言的 global 函数,并对 GPU 资源分配进行调度;
- (5) 输出计算结果。

2.3 GPU 端功能

- (1) 在 GPU 端为临时数据分配显存空间;
- (2) 每条 GPU 线程读取一座建筑物的数据,进行弹塑性时程计算,并将计算结果写入 CPU 端分配好的用于拷贝的显存中。

2.4 CPU-GPU 通讯模式

- (1) 在 CPU 端 Host 函数中调用 GPU 函数时,对 CUDA 的运行参数进行设置,包括 grid、block、thread 的维度等;
- (2) 采用 cudaMemcpy() 函数将数据在内存和显存之间拷贝。

3 程序实现与测试

3.1 并行化实现

按照以下方法实现程序并行化:

- (1) 完成单个结构层剪切模型弹塑性时程分析 C++ 程序的编写,为将其移植到 CUDA C 做准备。

(2) 将移植好的 C++ 程序改写为 CUDA C 设备端程序(即 GPU 程序)。

由于每一个结构都作为一个并行的计算任务,因此单个结构的时程分析都是一个并行的内核程序,其整个计算过程均要在 GPU 上进行,因此要对其进行整体改写。

(3) 编写主机端程序(即 CPU 端程序),包括数据读入、拷贝、输出函数,以及主控制程序。

3.2 程序并行化测试

程序编写完成之后,为验证并行执行的效果以及相关参数的选取,本文对其性能进行了测试,并将 CPU/GPU 协同计算的结果和 CPU 单独计算的结果进行对比。

需要说明的是:在本程序中,所有的计算模块都是在 GPU 上完成的,CPU 只是起到调度作用,对性能影响很小,可不必考虑 CPU 对 GPU 计算性能的提高。因此后文中的对比均采用 GPU 和 CPU 作为对比对象。

3.2.1 测试数据

(1) 采用 1000 座结构作为测试数据;

(2) 结构均为 6 层,每层质量相同;

(3) 地震波长度 40 s,步长 0.005 s,共 8 000 步;

(4) 层间剪力-层间位移采用最简单的平行四边形滞回模型,且仅输出了其中某一座结构的位移计算结果,以减少硬盘写入速度对测试结果的影响。

3.2.2 测试平台

CPU: Intel Core i3 CPU 530 @ 2.93GHz,内存: DDR3 4G 1333MHz

CPU: Intel 赛扬双核 E3200 @ 2.4GHz + NVIDIA GeForce GTX 460 1GB

两种测试平台价格相当(约 1 500 元左右),因此可以用来比较其性价比。

测试中 GPU 程序运行参数设置为 grid=1,block=4,编译参数中,GPU_Architecture 取为 sm_10,即采用计算能力为 1.0 的计算架构运行程序^[11]。

3.2.3 测试结果

首先进行了建筑个数与计算时间关系的测试,其结果如图 4。

图中 GPU 计算时间明显低于 CPU 的计算时间。由于图中 GPU 计算时间曲线形状不太清楚,将其进行放大之后,如图 5 所示。

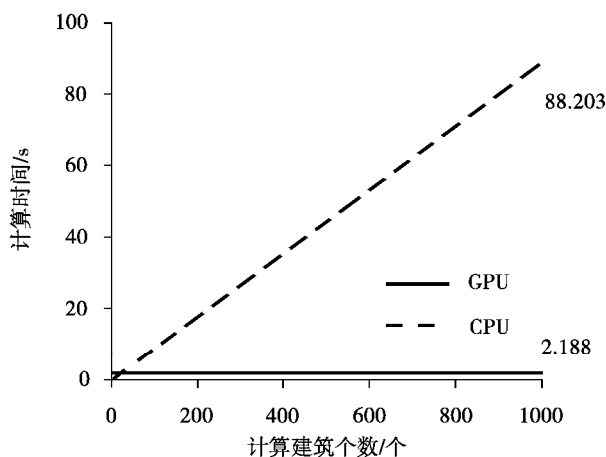


图4 CPU/GPU 计算时间与计算个数的关系

Fig. 4 Relationship between computing quantity and time in use of GPU and CPU

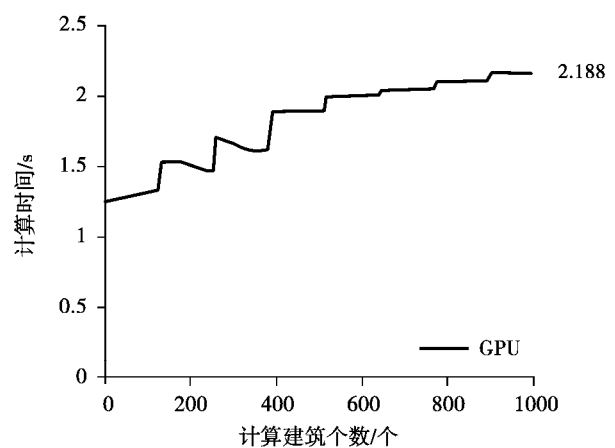


图5 GPU 计算时间与计算个数关系

Fig. 5 Relationship between computing quantity and time using GPU

从图中可以看出,CPU 程序的计算曲线呈现直线型,即随着计算建筑个数的上升,其计算时间与计算个数成正比。这也符合串行计算理论“单线程、单任务”的理念。而 GPU 的计算结果呈现折线状,计算建筑个数增加,时间增加并不是很多,这是因为 CUDA 在计算时,在同一个 block 内会以 32 条线程作为一个线程束,同一个线程束内,程序并行效率最高,如果超过 32 条线程,就会每 32 条线程构成一个线程束,在线程束之间会产生线程分配延迟。而本文在定义 CUDA 计算时,使用了 4 个 block,因此,当计算线程数超过 $4 \times 32 = 128$ 条时,线程就需要多分配一次,因此计算时间就会显著增长。但是这种效应随着计算时间的增长

而逐渐减小,建筑物数量小于 384 栋时,每次额外的线程分配计算时间增长很多,当建筑数量多于 384 栋后,其时间增量很小。这是由于在线程数较多时,CUDA 自身可以对线程分配进行优化,将部分线程分配延时隐藏于计算之中,从而令线程分配延时降低。

通过对比 GPU 和 CPU 的计算时间,可以看出,GPU 在并行计算大量弹塑性时程分析时速度是非常理想的,其计算时间比如图 6。

当计算建筑个数为 1 时,采用 CPU 计算的时间仅为 GPU 计算的 0.07 倍,这是由于 CPU 单个核心的计算能力要比 GPU 好很多,然而,随着计算个数的增多,CPU 计算时间显著变长,而 GPU 的计算时间由于并行的优势,变化不大,在计算 1000 座建筑时,采用 GPU 计算的时间仅为 CPU 计算的 1/40。

表 1 和表 2 给出了 CPU 和 GPU 的计算结果误差的比较,从中可以看出,在 0.005 s(第一个计算时间点)时,CPU 和 GPU 的计算结果是相同的,当到达 40.000 s(最后一个计算时间点)时,CPU 和 GPU

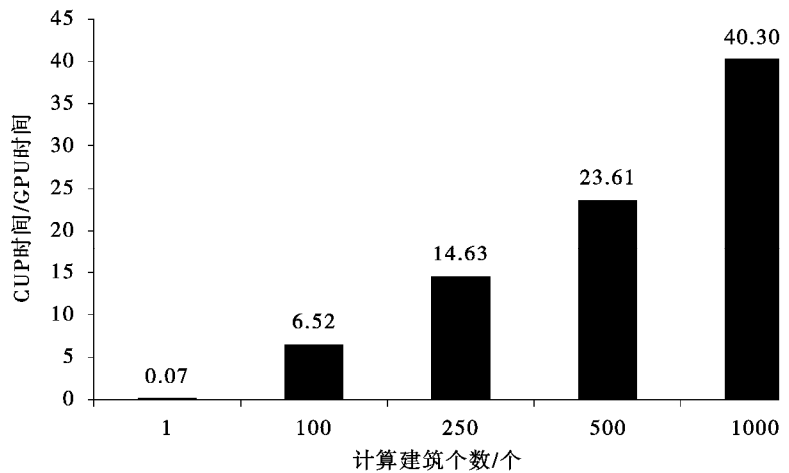


图 6 CPU/GPU 计算时间比

Fig. 6 Ratio of computing time between CPU and GPU

的误差为 1/1000 左右。这种累积误差是由于 CPU 和 GPU 中对于浮点计算规则的差别导致的,在可接受的范围内,可以忽略。

表 1 CPU/GPU 计算结果对比(0.005 s 时)

Table 1 Results of computation using CPU and GPU (at 0.005 s)

| 位移(10 ⁻⁸) | 1 层 | 2 层 | 3 层 | 4 层 | 5 层 | 6 层 | GPU |
|-----------------------|---------|---------|---------|---------|---------|---------|-----|
| CPU | 0.55608 | 0.55608 | 0.55608 | 0.55608 | 0.55608 | 0.55608 | |
| CPU | 0.55296 | 0.55608 | 0.55608 | 0.55608 | 0.55608 | 0.55608 | |
| 误差 | 0% | 0% | 0% | 0% | 0% | 0% | |

表 2 CPU/GPU 计算结果对比(40.000 s 时)

Table 2 Results of computation using CPU and GPU (at 40.000 s)

| 位移(10 ⁻²) | 1 层 | 2 层 | 3 层 | 4 层 | 5 层 | 6 层 |
|-----------------------|---------|---------|---------|---------|---------|---------|
| GPU | 0.52284 | 0.70995 | 0.24754 | 0.19642 | 0.19633 | 0.19628 |
| CPU | 0.52272 | 0.70970 | 0.24729 | 0.19616 | 0.19607 | 0.19602 |
| 误差 | 0.02% | 0.04% | 0.10% | 0.13% | 0.13% | 0.13% |

4 实际应用

以某中型城市(约 7 000 多座建筑)为例,采用本文方法对其建筑的地震响应进行了预测

(1) 将城市的 GIS 数据库内的建筑信息数据(包括建筑面积,高度,层数,坐标,结构类型等信息)转换为程序的数据输入文件,将结果读入程序中,共 7449 栋建筑物;

(2) 采用陆新征一曲哲滞回模型作为层间滞回模型,采用文献[9]的方法估算滞回模型参数;

(3) 进行时程计算,得到结构位移和倒塌数据;

(4) 采用层模型的方式对城市建筑物进行三维建模,结构位移映射到三维模型的坐标上,进行顶点和三角面片生成,得到动态展示结果。

采用上述步骤,本文首先将整个城市的建筑物进行了模型化,得到的效果如图 7,8,9 所示。



图 7 静态场景整体平面图

Fig. 7 General plan of static scene



图 8 静态场景整体斜视图

Fig. 8 General oblique view of static scene

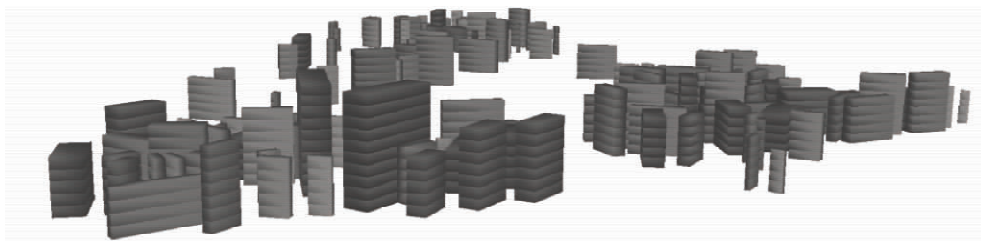


图 9 静态场景局部斜视图

Fig. 9 Partial oblique view of static scene

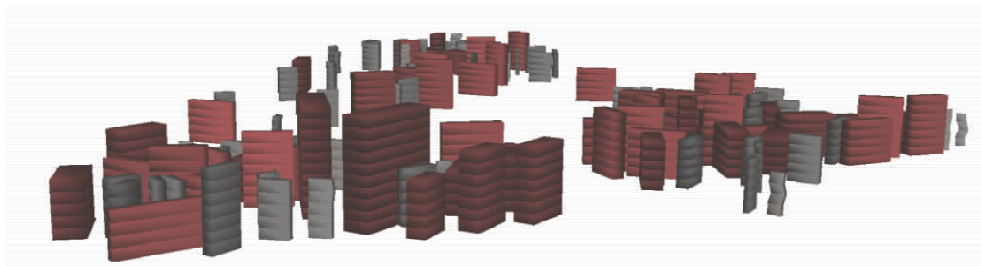


图 10 动态场景局部斜视图(位移放大 10 倍)

Fig. 10 Partial oblique view of dynamic scene (the displacements were enlarged to 10 times)

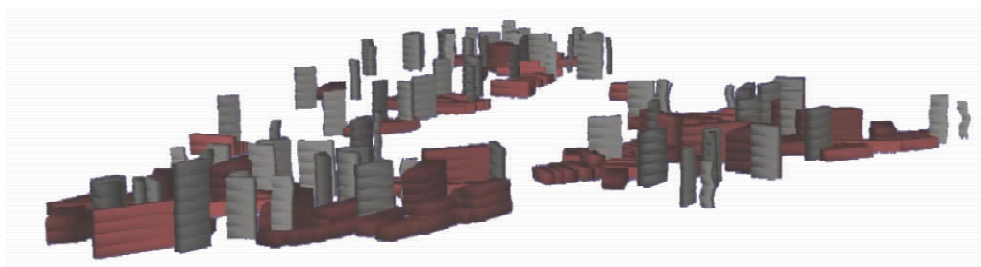


图 11 倒塌模拟效果

Fig. 11 Simulation effect of collapse

模型共 7 449 栋建筑物, 172 064 个结点, 314 220 个三角形面片。静态模型效率很好, 能够达到满帧率 (每秒 60 帧), 可以实时游览。

之后,采用本文建议方法完成建筑群的弹塑性模拟,地震波选用 Northridge 地震动记录,所有 7449 栋建筑物在前述测试所用的 GPU 平台上完成 40 s 动力弹塑性计算共耗时 216 s,其中输出结果用时为 194 s,动力弹塑性计算用时仅为 22 s,满足高效廉价震害预测的要求。预测得到的建筑震害分布如图 10 所示。红色代表可能发生倒塌的建筑物,灰色代表没有倒塌的建筑物。如果引入倒塌特效算法,移除发生倒塌的楼层,则可得到倒塌效果模拟如图 11 所示,可以较真实的模拟震害场景。

5 结论

本文建立了 GPU-CPU 协同建筑群震害模拟的架构,编制了计算程序,实现了基于 GPU 技术的高性能大区域城市震害模拟。结果表明,采用 GPU 的计算效率同比可以达到同价位 CPU 的 40 多倍,满足低成本高性能计算的要求。并通过对某中型城市进行实际震害模拟,得到了较理想的震害模拟效果,证明了采用 CPU-GPU 协同进行震害模拟是具有可操作性的。未来可以进一步提高其计算效率和通用性,优化计算模型,并提升渲染和场景展示效果,最终实现高效、低成本的城市大区域震害模拟。

参考文献:

- [1] 乔亚玲,闫维明. 建筑结构震害预测方法研究评述[J]. 工业建筑,2005,35(6): 1-6.
QIAO Yaling, YAN Weiming. Review of predictive method of seismic damages to buildings[J]. Industrial Construction, 2005, 35(6): 1-6.
- [2] 胡宝生,徐祥文. 展望建筑物震害预测的发展方向[J]. 世界地震工程,1995,(3): 33-38.
HU Baosheng, XU Xiangwen. Look forward to developmental direction of predicting earthquake damage to buildings[J]. World Information On Earthquake Engineering, 1995(3): 33-38.
- [3] Hori M, Ichimura T. Current state of integrated earthquake simulation for earthquake hazard and disaster[J]. Journal of Seismology, 2008(2): 307-321.
- [4] 陈序平. 基于 GIS CAD 和 VR 技术的城市地震仿真系统[D]. 北京: 清华大学,2007.
CHEN Xuping. An Urban Earthquake Simulation System Based on GIS CAD and VR Technology[D]. Beijing: Tsinghua University, 2007.
- [5] 李波,赵华成,张敏芳. CUDA 高性能计算并行编程[J]. 微型电脑应用,2009(9): 55-57,64-69.
LI Bo, ZHAO Huacheng, ZHANG Minfang. Parallel programming for high-performance computing on CUDA[J]. Microcomputer Applications, 2009(9): 55-57,64-69.
- [6] 吴恩华,柳有权. 基于图形处理器(GPU)的通用计算[J]. 计算机辅助设计与图形学学报,2004,16(5): 601-612.
WU Enhua, LIU Youquan. General purpose computation on GPU[J]. Journal of Computer Aided Design & Computer Graphics, 2004, 16(5): 601-612.
- [7] 张朝晖,刘俊起,徐勤建. GPU 并行计算技术分析与应用[J]. 信息技术,2009(11): 86-89.
ZHANG Chaohui, LIU Junqi, XU Qinjian. Analysis and application of the GPU parallel computing technology[J]. Information Technology, 2009(11): 86-89.
- [8] 陆新征,叶列平,缪志伟,等. 建筑抗震弹性分析——原理、模型与在 ABAQUS, MSC. MARC 和 SAP2000 上的实践[M]. 北京: 中国建筑工业出版社,2009: 18-25.
LU Xinzhen, YE Lieping, LIAO Zhiwei, et al. Elasto-plastic Analysis of Buildings Against Earthquake—theory, Model and Implementation on ABAQUS, MARC AND SAP2000. [M]. Beijing: China Architecture & Building Press, 2009: 18-25.
- [9] Xu F, Chen X P, Ren A Z, et al. Earthquake disaster simulation for an urban area, with GIS, CAD, FEA and VR integration[J]. Tsinghua Science and Technology, 2008(S1): 311-316.
- [10] Chopra A K. Dynamics of Structures: Theory and Applications to Earthquake Engineering[M]. New Jersey: Prentice-Hall, 2001: 171-174.
- [11] NVIDIA, NVIDIA CUDA Programming Guide[EB/OL]. (2011-11-18) [2011-12-01]. http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf.