DOI:10.13409/j.cnki.jdpme.20240912002

基于国产加速卡的地震模拟计算性能分析与优化*

周 辉[1]，朱虎明[2,3,4]，高天琦[2,4]，董西淼[2,3]，张凌云[2,3]，
刘卉杰[2,3]，陈志鹏[2,3]

（1.中国地震局第二监测中心，陕西 西安 710054；2.西安电子科技大学智能感知与图像理解教育部重点实验室，陕西 西安 710071；3.西安电子科技大学人工智能学院，陕西 西安 710071；4.西安电子科技大学杭州研究院，浙江 杭州 311231）

摘要：AWP-ODC是基于有限差分数值方法来实现大规模三维地震模拟的软件。随着国外对我国高性能计算芯片的出口限制，我国急需发展自己的高性能计算芯片及其软件生态。早期的AWP-ODC加速主要基于NVIDIA GPU软硬件架构来设计优化，近年来，多种异构计算平台迅猛发展，如何基于新的异构计算软硬件平台来加速AWP-ODC具有重要研究价值。为此，本文在一种国产加速卡上对AWP-ODC进行移植。针对耗时较多的核函数dstrqc，通过GPU访存优化和网格参数优化等方式缩短了其运行时间。最后分别在国产类GPU单卡和双卡上，利用Fréchet Kernels地震和8•3鲁甸地震数据集进行性能测试。实验结果表明，在单卡计算环境下，两个数据集的FLOPS分别提高了30.51%和25.21%；在双卡计算环境下，两个数据集的FLOPS分别提高了9.42%和23.6%。

关键词：地震模拟；国产加速卡；AWP-ODC；异构计算；性能优化
中图分类号：P315　　**文献标识码：**A　　**文章编号：**1672-2132(2025)01-0021-13

Performance Analysis and Optimization of Seismic Simulation Computation Based on Domestic Accelerator Cards

ZHOU Hui[1], ZHU Huming[2,3,4], GAO Tianqi[2,4], DONG Ximiao[2,3], ZHANG Lingyun[2,3], LIU Huijie[2,3], CHEN Zhipeng[2,3]

(1. The Second Monitoring and Application Center, CEA, Xi′an 710054, China;
2. Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, XiDian University, Xi′an 710071, China; 3. School of Artificial Intelligence, XiDian University, Xi′an 710071, China;
4. Hangzhou Institute of Technology, XiDian University, Hangzhou 311231, China)

Abstract: AWP-ODC is a software for large-scale 3D seismic simulation based on the finite difference numerical method. Due to foreign export restrictions on high-performance computing chips to China, there is an urgent need to develop China′s own high-performance computing chips and software ecosystem. The early acceleration of AWP-ODC was primarily designed and optimized based on the NVID-

* **收稿日期：**2024-09-12；**修回日期：**2025-01-10
基金项目：陕西省重点研发计划(2022ZDLGY01-09)、光合基金(202302019674)、陕西省自然科学基础研究计划(2023-JC-YB-242)资助
作者简介：周辉(1981—)，男，高级工程师，硕士。主要从事地震软件研发与测试。E-mail: hui@smac.ac.cn
通信作者：朱虎明(1978—)，男，副教授，博导，博士。主要从事高性能计算及其应用和人工智能计算系统及其并行加速等方向的研究。E-mail:zhuhum@mail.xidian.edu.cn

IA GPU software and hardware architecture. In recent years, various heterogeneous computing platforms developed rapidly. How to accelerate AWP-ODC based on new heterogeneous computing software and hardware platforms showed significant research value. To this end, AWP-ODC was ported to a domestic accelerator card. By optimizing GPU memory access and grid parameters, the execution time of the time-consuming kernel function dstrqc was reduced. Finally, performance tests were conducted on a domestic GPU single-card and dual-card setup using the Fréchet Kernels seismic dataset and the 8·3 Ludian earthquake dataset. Experimental results showed that, under a single-card computing environment, the FLOPS for the two datasets increased by 30.51% and 25.21%, respectively. Under a dual-card computing environment, the FLOPS for the two datasets increased by 9.42% and 23.6%, respectively.

Keywords: seismic simulation; domestic accelerator card; AWP-ODC; heterogeneous computing; performance optimization

0 引 言

自21世纪初以来,地震学研究领域发生了深刻的变革。科研人员现在能够从多维学术视角,如板块构造动力学,借助计算机模拟技术深入分析地震问题。然而,三维地震模拟分析产生的数据量巨大,往往达到千万亿字节。因此,为了加快处理海量数据,迫切需要一种适用于大规模三维地震模拟的并行计算方法。

加利福尼亚大学的K. B. Olsen等[1-3]率先开发了用于地震模拟的有限差分法代码。在此基础上,Y. Cui[4]进一步研发了高度可扩展的AWP-ODC并行有限差分软件,这一进展使得大规模三维地震模拟成为现实。2012年,M. Christen等[5]开发了Patus框架,这是一个专为现代多核和众核处理器设计的模板计算的代码生成和自动调优框架。他们结合AWP-ODC在Cray XE6平台上使用波动方程离散化进行了千万亿次计算的评估。2016年,D. Roten等[6]对AWP-ODC地震模拟程序中的CUDA内核进行了优化,更充分地利用了GPU的内存带宽,显著提升了并行效率。目前,国外对AWP-ODC和地震模拟的加速主要基于NVIDIA GPU技术。

国产计算平台在近十年间经历了迅猛的发展,我国在超级计算领域已达到世界先进水平,其中包括鲲鹏ARM处理器、类GPU加速器、神威超算以及天河系列超算等。然而,由于国产计算平台的架构与NVIDIA GPU的计算平台存在显著差异,因此将AWP-ODC地震模拟软件移植到国产计算平台上面临诸多技术挑战。

近年来,在国产化计算平台移植AWP-ODC上取得了很大的进展。2017年,清华大学的H. H. Fu等[7]等基于“神威·太湖之光”超级计算机开发了高度可扩展的“非线性地震模拟”应用。2023年,田浩东[8]在付昊桓研究的基础上,针对新一代神威超级计算平台,通过多层级划分方案、增大通信带宽以及增大DMA通信带宽等方式对AWP-ODC进行了优化。这两项工作均基于神威计算平台,而我们则基于国产加速卡实现地震模拟,与神威、天河计算平台和NVIDIA GPU的软硬件架构存在较大区别。因此,探索新的移植优化方法具有重要价值。

本文主要在单节点异构计算环境下开展工作,具体贡献如下:

(1) 采用国产加速卡的兼容方案,成功将AWP-ODC地震模拟软件移植至国产加速卡上,并完成了正确性验证。

(2) 针对国产加速卡的存储架构,对程序进行了类GPU的访存优化和网格参数优化。

(3) 还对国产加速卡与NVIDIA GPU的性能进行了对比分析。

1 相关工作

1.1 异构计算与异构编程模型

异构计算是指在一个系统内,CPU与其他处理器(如GPU、国产加速卡、FPGA、ASIC)共同协作进行计算的模式。这种模式的采用是由其硬件架构决定的。CPU通常适用于通用计算任务,具有复杂的控制逻辑和高速缓存,擅长处理串行任务。而其他协同处理器,如GPU,则拥有大量的计算单元,

适合处理大规模数据并行计算。

目前主要的GPU有NVIDIA的A100、H100和AMD的Instinct系列，这些芯片均提供了非常强的计算能力，它们在软硬件架构上有较大差异。NVIDIA GPU使用CUDA Cores作为基本的计算单元，其通常集成在Streaming Multiprocessors（SMs），多个SMs又构成了图形处理集群GPC（Graphics Processing Cluster）。每个SMs包含CUDA Cores的数量随着架构的不同而不同。而AMD GPU使用Stream Processors（SPs）作为基本的计算单元，通常集成在Compute Units（CUs）。以NVIDIA在2020年发布的基于Ampere架构的A100 GPU为例，A100内部包含128个流多处理器（SMs），每个SM有64个FP32 CUDA Cores，共计8 192个FP32 CUDA Cores，其FP32的计算能力为19.5 TFLOPS。2022年发布了基于Hopper架构的H100 GPU，H100内部包含144个流多处理器（SMs），每个SM有128个FP32 CUDA Cores，共计18 432个FP32 CUDA Cores，其FP32的计算能力为67 TFLOPS。AMD在2023年发布了基于CNDA3架构的Instinct MI300X，集成了304个计算单元（CUs），每个CU包含64个流处理器SP，共计19 456个流处理器（SPs），其FP32的计算能力为163.4 TFLOPS。综上所述，NVIDIA CUDA核心聚于SM，AMD SP整合于CU，他们的性能均较强。

此外，国内计算平台如新一代神威超级计算机采用了自主研发的SW26010P CPU，该CPU使用64位申威指令系统。SW26010P处理器采用主从异构架构，由主核和从核组成。每个SW26010P处理器包含6个核簇，采用片上融合异构架构，每个核簇包括一个主核和64个从核[9]。每个SW26010P的浮点运算量的理论峰值达到3.06 TFLOPS。新一代天河超级计算机则使用自主研发的Matrix-3000处理器，同样采用片上异构架构。Matrix-3000处理器包含16个CPU、96个控制核和1 536个加速器核，架构分为通用区和加速区。通用区由16个CPU构成，而加速区则划分为四个集群，每个集群包含24个控制核和384个加速核[10]。每个Matrix-3000处理器的浮点运算量可以达到11.6 TFLOPS。

国产加速卡采用类似GPU的架构，如图1所示。国产加速卡有64个独立计算单元，每个单元内含16个向量ALU单元，每个ALU单元可以运行一个线程。线程束（warp）是国产加速卡的基本执行单元，包含64个线程同时执行相同指令，从而提高计算效率[11]。

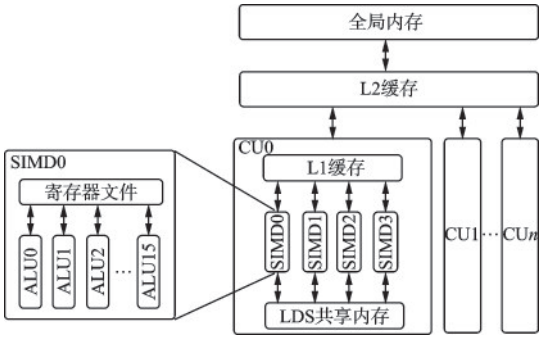

图1　国产加速卡系统结构图

Fig.1　System architecture of domestic accelerator card

以此可以得出国产加速卡与神威和天河超级计算机的不同和优势。首先，在架构设计方面，国产加速卡采用类似GPU的架构，具有64个独立计算单元，每个计算单元内含16个向量ALU单元。这种设计使得每个ALU单元能够同时运行一个线程，从而在并行计算能力上表现出色。线程束（warp）作为基本执行单元，允许64个线程同时执行相同指令，极大地提高了计算效率。

相较之下，神威和天河的处理器架构更强调主从异构设计。神威的SW26010P处理器通过主核与从核的组合，适用于科学计算和数据密集型任务；而天河的Matrix-3000处理器则结合了通用区和加速区的设计，旨在提高通用计算性能和灵活性。

其次，在应用场景上，国产加速卡通常针对高并发和高吞吐量的计算任务，如深度学习、大数据分析和复杂模拟，而神威和天河架构则更加适合传统的科学计算。

NVIDIA和AMD均推出了自己的异构编程模型。CUDA（Compute Unified Device Architecture）是由NVIDIA推出的一种异构编程模型[12-14]，它可以让CPU和GPU协同地完成计算任务。HIP（Heterogeneous-compute Interface for Portability）是由AMD推出的开源的异构编程模型[15-16]。而国产加速卡主要基于HIP异构编程模型来完成CPU和GPU的协同计算任务。为了方便开发者对程序的移植和使用，HIP设计的语法等与CUDA类似，但其基于的底层框架差异较大，不能直接将CUDA代码进行移植，往往需要开发者手动调试。其中CUDA、HIP和国产计算平台的DTK在线程束上略有差异（表1）。

表1　各个平台的线程数限制

Table 1　Thread count limits on various platforms

	CUDA	HIP	DTK(国产化HIP)
线程束	32	64	64
线程块中最大线程数	1 024	256	256

其中DTK(DCU Toolkit)能兼容CentOS、Ubuntu等主流的Linux发行版,还能兼容麒麟、方德等国产操作系统。DTK支持开发者在C++语言中使用HIP、OpenCL等编程接口开发DCU应用,还支持在C++或Fortran(部分支持)语言中使用OpenMP、OpenACC等编译器导语的方式进行开发。DTK中提供了编译器、性能分析工具、调试工具、系统状态监控等完整的开发工具链以保证便捷地开发。DTK中还包含了强大的运行时系统模块,以支持DCU应用的高效执行(图2)。

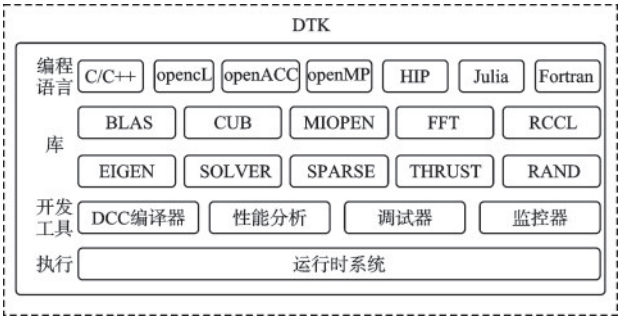

图2　DTK结构图

Fig.2　DTK structure

目前,国内已有很多研究将基于NVIDIA CUDA的程序移植到国产加速卡上,并进行了优化。如黄聪伟等[17]将MPS算法程序移植到国产加速卡上,并开发了MPSDCU-SJTU求解器,用于模拟三维溃坝流动。J. Niu等[18]提出一种基于多面体模型(polyhedral model)的DCU代码生成和优化方法。李冰洋[19]将供水管网仿真程序移植到国产加速卡上,来预测管道内的数值状态。杨思驰等[20]在国产加速卡上通过LDS访存向量化和共享内存向量化实现程序的优化。

1.2　地震模拟方法与AWP-ODC软件

三维地震模拟主要使用弹性动力学计算方法,而弹性动力学计算方法可以分为解析和半解析法、数值方法和混合方法(前两种方法的结合)[21]。此外数值法主要包括有限差分法、隐式有限元法和谱元法等多种方法[22]。尽管这些方法均可用于模拟地震断层的自发破裂及波动的传播过程,但有限差分法以其易于在大规模并行超级计算机上部署而突出,能够在保证高精度的同时实现显著的计算效率。因此其得到了广泛的应用[23-24]。

本次移植的AWP-ODC地震模拟软件使用的方法属于数值方法,后续也有不少学者在此研究的基础上进行扩展。例如,2017年,J.Tobin等[25]基于KNL(Intel Xeon phi Knights Landing)处理器开发出AWP-ODC-OS,使其在KNL处理器上的性能达到P100 GPU的98.5%。2019年,D. Mu等[26]基于AWP-ODC-OS开发了awp-odc-insitu,实现了地震模拟的可视化。2021年,Q. Zhou等[27]通过优化MPI通信库,使得AWP-ODC在GPU上的浮点运算量得到提高。2023年,Y. Cui等[28]综述了AWP-ODC软件在新一代高性能计算架构上的移植进展,以及其在TACC Frontera超级计算机上进行的4-Hz Iwan型非线性动态模拟ShakeOut场景的应用,展示了其在地震模拟领域的先进性能和潜力。同年,H. Zhou[29]通过构建前兆观测数据检测与异常自动识别系统,实现了短临地震的预测。

AWP-ODC地震模拟软件采用显式交错网格有限差分格式来求解三维速度应力波方程,该格式在空间维度上达到四阶精确度,在时间维度上实现二阶精确度。经过多次迭代更新和功能增强,该软件已广泛应用于多个重要的SCEC模拟,包括TeraShake、TeraShake-2、ShakeOut-K、ShakeOut-D、Chino Hills、Wall2Wall和M8[30]。

2　AWP-ODC算法数学模型及其软件结构

AWP-ODC通过结合控制方程、交错网格有限差分方程和外部边界条件精确模拟地震波传播。其中控制方程描述地震波的弹性与非弹性传播,交错网格有限差分方程用于离散化求解速度和应力分量,外部边界条件通过完全匹配层技术消除波反射,优化模拟精度。

2.1　AWP-ODC算法数学模型

2.1.1　控制方程

利用有限差分方法求解一个偏微分方程耦合系统。令:

$$\boldsymbol{v}=(v_x,v_y,v_z) \tag{1}$$

表示粒子速度张量，

$$\boldsymbol{\sigma}=\begin{pmatrix}\sigma_{xx}\ \sigma_{xy}\ \sigma_{xz}\\ \sigma_{yx}\ \sigma_{yy}\ \sigma_{yz}\\ \sigma_{zx}\ \sigma_{zy}\ \sigma_{zz}\end{pmatrix} \tag{2}$$

为对称应力张量。则弹性动力学方程表示为：

$$\left.\begin{aligned}\partial_t\boldsymbol{v}&=\frac{1}{\rho}\nabla\cdot\sigma\\ \partial_t\boldsymbol{\sigma}&=\lambda(\nabla\cdot\boldsymbol{v})I+\mu(\nabla\cdot\boldsymbol{v}+\nabla\cdot\boldsymbol{v}^{\mathrm{T}})\end{aligned}\right\} \tag{3}$$

式中，λ和μ为*lamé*系数；ρ为介质密度。

按分量方向分解可得到速度张量分量的三个标量值方程和应力张量分量的六个标量值方程。

地震波在地球上实际传播时会出现非弹性效应，所以这种效应也须包含在实际模拟的波传播中。于是可以用Drucker-Prager塑性方法[31]描述高频地震波中的非线性效应：

$$Y(\sigma)=\max(0,c\cos\phi-(\sigma_m+P_f)\sin\phi) \tag{4}$$

式中，$Y(\sigma)$为屈服应力；c为凝聚力；ϕ为摩擦角；σ_m为平均压力；P_f为流体压力。

用屈服应力可以更新应力：

$$\sigma_{ij}=\sigma_m^{trial}\delta_{ij}+rs_{ij}^{trial},\quad(i,j=x,y,z) \tag{5}$$

式中，r由屈服应力计算所得；s_{ij}为应力偏导。

2.1.2 交错网格有限差分方程

控制标量方程在时间和空间上通过交错网格上的有限差分[32]来近似。时间导数可以近似为：

$$\begin{aligned}\partial_t v(t)&\approx\frac{v(t+\frac{\Delta t}{2})-v(t-\frac{\Delta t}{2})}{\Delta t}\\ \partial_t\sigma(t+\frac{\Delta t}{2})&\approx\frac{\sigma(t+\Delta t)-\sigma(t)}{\Delta t}\end{aligned} \tag{6}$$

对于空间导数，Φ为速度或应力分量，h为等距网格尺寸，$\partial_x\Phi$网格点(i,j,k)的有限差分近似为：

$$\partial_x\Phi_{i,j,k}\approx D_x^4(\Phi)_{i,j,k}=\frac{c_1(\Phi_{i+\frac{1}{2},j,k}-\Phi_{i-\frac{1}{2},j,k})+c_2(\Phi_{i+\frac{3}{2},j,k}-\Phi_{i-\frac{3}{2},j,k})}{h} \tag{7}$$

该方程用于近似每个速度和应力分量的每个空间导数，其中$c_1=9/8$，$c_2=-1/24$。

2.1.3 外部边界条件

在对三维建模域进行网格划分时，常常会出现非物理性的波反射现象。为了最大程度消除这些反射，设计吸收边界条件成为一种迫切需求，其中完全匹配层（PML，Perfect Match Layer）技术是公认的最有效解决方案之一[33-34]。完全匹配层可以通过一个简洁的时域分解过程来实现，该过程将每个波动方程分解为横向和纵向的分量，并在横向部分引入特定的波阻抗项。该波阻抗能够与边界介质完美匹配，从而优化边界处的波吸收，显著提高模拟精度。

2.2 AWP-ODC软件结构

AWP-ODC地震模拟软件运行时，首先进行变量初始化，它从文件读入由密度和*lamé*参数构成的速度模型以及源项（源项通常表示地震波在震源处的初始条件或激发信号，可以看作是震源处向地下介质注入的能量），并为所需数据结构分配和初始化空间。AWP-ODC总共使用了26个常规三维网格：每个x、y和z分量的速度网格；六个应力网格，表示应力张量的上三角分量；一个用于密度和*lamé*参数的网格；一个用于Cerjan海绵层的网格；最后是13个用于非弹性衰减的网格，由记忆矩阵的六个上三角分量和七个用于粗粒度衰减的网格组成。

随着时间的推移，AWP-ODC地震模拟软件不断更新计算的时间步，使地震波从震源处逐步传播和扩散。最初，地震波以高强度从震源区域向外发射，能量集中，传播速度较快。随着时间步的增加，地震波逐渐传播到更远的区域，能量也随之分散在更大的范围内。模拟过程中，速度模板、应力模板、源项及潜在的输出内容都会随时间步同步更新。速度模板和应力模板是一组预定义的计算公式集，用于指导如何在网格或数据点上应用有限差分法来近似求解偏微分方程，这些方程描述了地震波在地球介质中传播的物理过程。此外，对于包含自由表面边界的区域，在速度和应力相关变量更新完成后，还会在这些表面边界上分别计算满足自由界面条件的速度和应力，以确保模拟的准确性。

在软件的文件架构中，系统由多个专门组件构成，其中核心文件pmcl3d.c扮演主导角色，而其他文件负责各自的特定功能。如图3所示，command.c定义并传递所需参数给pmcl3d.c，cerjan.c实现吸收边界条件，io.c负责数据输入输出管理，swap.c调用cerjan.c进行边界数据传输，grid.c负责动态内存管理，mesh.c专注于初始化和管理网格结构及介质属性。上述五个文件作为工具文件，被source.c调用以处理地震源数据。在主程序pmcl3d.c启动后，它

接收这些处理过的数据，并通过kernel.cu并行计算地震波传播，实现地震模拟。

当模拟达到预设的结束时间，主程序pmcl3d.c结束循环，并完成所有剩余数据的输出和资源的释放。在地震模拟完成后，AWP-ODC软件会将模拟结果输出为x、y、z三个方向的地震波形数据，并分别存储为.bin文件格式。这些输出数据包含了地震波在整个模拟区域内各个网格点随时间变化的位移或速度信息，可用于后续的地震分析、可视化和研究工作。

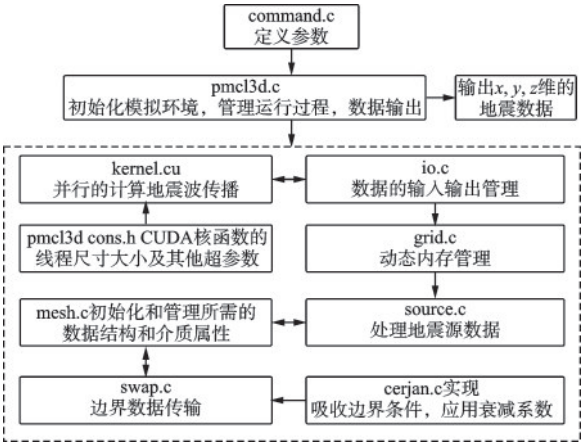

图3　AWP-ODC地震模拟软件的文件构成及作用

Fig.3　File structure and functions of AWP-ODC seismic simulation software

3　AWP-ODC软件性能分析与优化

（1）移植过程概述：本文在移植AWP-ODC时，首先使用计算平台提供的自动适配技术进行编译，然后针对平台编译中出现的错误进行手动修正，如图4所示。其中，自动适配技术是通过DTK提供的类CUDA环境并设置相关环境变量以及调用DTK提供的相关库来实现的。编译中出现register声明的报错，这是因为在国产加速卡中不需要

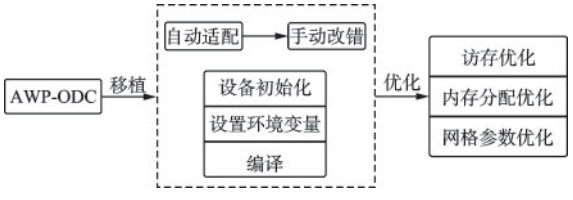

图4　AWP-ODC地震模拟软件的移植与优化过程

Fig.4　Porting and optimization process of AWP-ODC seismic simulation software

单独对寄存器变量进行声明，故将register声明删除，这样不仅确保了代码的可移植性，还简化了代码维护，最终软件成功编译。

（2）性能分析结果：利用nvprof性能分析工具，获取了各个核函数的时间，其时间占比如图5所示，用于指导下一步的程序优化。

由图5可知，核函数时间占比最多的是dstrqc，它负责计算地震波的传播和场量的变化，是地震模拟中计算量最大的部分之一。笔者深入剖析了该核函数的代码，发现其中使用了大量的register声明。虽然register变量是寄存器变量，但它仍需从全局内存中获取数据并存入寄存器。由于寄存器内存有限，这会导致数据读取速度变慢。

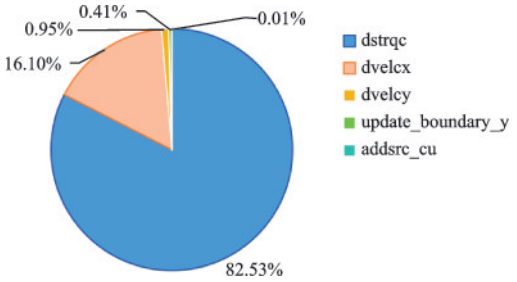

图5　核函数时间占比

Fig.5　Time proportion of kernel functions

（3）具体优化步骤：针对（2）中的分析结果，首先定义了LDS（共享内存）存储，将全局内存中的数据先存入LDS，再从LDS中读取到寄存器中，并使用共享内存进行规约。具体优化措施为：首先，使用__shared__修饰内存变量，并将其长度设置为线程块大小；其次，将全局内存中的数据复制到共享内存中，使每个线程块都能拥有一个共享内存变量副本；最后，将共享内存中的数据读取到寄存器中。这些优化措施可以减少全局内存访问，提高数据访问的局部性，从而提升访存命中率和计算效率。

此外，又发现在该软件中有许多动态内存管理的代码，由此判断内存管理也是其时间比重大的原因之一。由于动态内存管理的函数是在grid.c文件中定义的，因此对grid.c文件中的函数进行了内存分配的优化，以加快动态内存管理的速度。其中被优化的函数如Alloc3D函数，其主要负责动态分配三维的浮点型数组，如图6和7所示。通过这一系列的优化措施，不仅提升了软件的运行效率，还为后续的地震模拟研究提供了更为坚实的技术基础。

```
Grid3D U = (Grid3D)malloc(sizeof(float**)*nx + sizeof(float
*)*nx*ny +sizeof(float)*nx*ny*nz);//使用malloc进行内存分配

for(i=0;i<nx;i++){
      U[i] = ((float**) U) + nx + i*ny;
 }//初始化指针
float *Ustart = (float *) (U[nx-1] + ny);
//指针偏移和数据初始化
for(i=0;i<nx;i++)
      for(j=0;j<ny;j++)
            U[i][j] = Ustart + i*ny*nz + j*nz;
for(i=0;i<nx;i++)
      for(j=0;j<ny;j++)
            for(k=0;k<nz;k++)
                U[i][j][k] = 0.0f;
```

图6　Alloc3D函数未优化前

Fig.6　Unoptimized Alloc3D function

```
size_t size=sizeof(float**)* nx+sizeof(float*)*
nx*ny+ sizeof(float)*nx*ny*nz;
Grid3D U =(Grid3D)calloc(1,size);//使用calloc进行内存分配和初
始化
//分配和初始化第二层指针
float**U2D=(float**)(U+nx);
float*Ustart=(float*)(U2D+nx* ny);
//设置指针偏移
for(i=0:i<nx:i++)
    U[i]=U2D+i*ny;
        for(j=0:j<ny:j++)
            v[i][j]=Ustart+(i*ny+j)*nz;
```

图7　Alloc3D函数优化后

Fig.7　Optimized Alloc3D function

优化后的代码通过使用calloc来分配内存[35]，不仅分配了所需的内存，还将所有字节初始化为零，从而缩短了多重循环初始化内存的时间，使代码更高效且简洁。

其次，通过使用两层指针来分配三维数组的内存，并直接在内存块中设置指针偏移，避免了不必要的重复分配和初始化。此外，优化了指针偏移，通过计算偏移量在初始化阶段将三维数组的指针结构设置为连续的内存块，这不仅减少了内存碎片，还提高了内存访问的局部性和缓存命中率。

最后，对于类GPU的网格参数优化（初始值为(256,2,2)，采用贪心的策略进行搜索。如表1所示，首先依据平台的一个warp中的线程数的倍数来设置搜索范围。其中block的X方向设置为[64－256]，Y方向设置为[1-4]，Z方向设置为[1-4]，且设置约束条件为X、Y和Z的乘积需小于等于1 024。之后利用贪心策略在搜索空间中进行搜索，得出的最佳block为(160,2,2)。

此外还可以通过__launch_bounds__修饰符来优化线程数。默认情况下是按照每个线程块1 024个线程来分配的。当线程块中的线程数量小于1 024，使用__launch_bounds__修饰符可以指定实际使用的线程数量[36]。

4　实验评估

4.1　实验环境与数据集

本次实验选取的与国产加速卡对比的GPU有NVIDIA GTX 1080、NVIDIA RTX 2080Ti、NVIDIA RTX 3080和NVIDIA A100。其与国产加速卡的部分信息见表2。

表2　计算平台参数

Table 2　Computing platform parameters

GPU型号	显存大小/GB	SM线程数/个	显存带宽/(GB·s^{-1})	FP32/TFLOPS	FP64/TFLOPS
国产加速卡	16	4 096	800	11.8	5.9
1080	8	3 584	320.3	11.3	0.35
2080Ti	11	4 350	616	13.5	0.42
3080	10	8 704	760.3	29.8	0.46
A100	40	8 192	1 555	19.49	9.746

其中国产计算平台是使用了由我国制造的SIMT加速器组成的集群。该集群包括许多节点，每个节点包含1个32核CPU和4个加速器。CPU内有四个NUMA节点，每个NUMA节点有8个X86核。加速器采用类似GPU的架构，由16 GB HBM2设备内存和许多计算单元组成。加速器通过PCI-E与CPU连接，主内存与设备显存之间的数据传输峰值带宽为16 GB/s。

本次实验使用了两个案例。Fréchet Kernels地震案例是由Z. Li等[37]设计Fréchet Kernels地震波传播的计算例子，整个计算区域为200*100*160个格点，空间网格大小为200 m，时间步长为0.01 s，模拟计算了15 s的波场传播过程。计算用到的介质模型是半空间(half space)，主要参数为：密度：3 000 kg/m，P波速：6 500 m/sec，S波速：3 500 m/sec。这是一个虚拟的地震模拟案例。另一个真实的地震模拟案例是发生在云南省昭通市鲁甸县[38]，波传播数据由距震源200 km内的40个触发台站采集，地震模拟的过程中，整个计算区域为500*500*100个格点，空间网格大小为400 m，时间步长为0.02 s，模拟计算了40 s的波场传播过程。

在计算平台的GPU上分别测试了Fréchet Kernels地震和8•3鲁甸地震案例的单卡和两卡的程序，并对比该程序在不同GPU上的性能差异。其次对上述结果进行了正确性验证，并对其进行性能优化。

4.2 正确性验证

通过分别在NVIDIA 1080和国产加速卡上运行实际的地震模拟案例，并对比得到的结果，来验证结果的正确性，判断的标准是两个结果之间的误差小于十的负六次方。经对比后所有结果均小于十的负六次方，由此判断在国产加速卡上的地震模拟结果正确。

4.3 在Fréchet Kernels地震模拟案例上的实验结果及计算性能分析

利用matlab软件进行结果可视化如图8所示（其中最左侧一栏是地震模拟迭代的步数，图8中横轴代表东西方向的距离，纵轴代表南北方向的距离，其中颜色的深浅表示地震波的振幅大小，红色表示一个波峰，蓝色表示一个波谷，绿色表示平静或没有明显变化的区域。(a)，(c)，(e)，(g)表示在国产加速卡上的波传播可视化，(b)，(d)，(f)，(h)表示在NVIDIA GPU 1080上的波传播可视化）。其中可以观察出在389步时由震源发出地震波随着步数的增加，地震波逐渐由强变弱的扩散开来。

在评估Fréchet Kernels地震波传播模拟的性能时，对比了NVIDIA GPU 1080、2080Ti、3080、A100，以及国产加速卡在单卡和双卡配置下的浮点运算量、运行总时间、加速比与并行效率，如表3和图9所示。

国产加速卡在浮点运算量和运行总时间上仅超过了NVIDIA 1080，弱于NVIDIA 2080Ti和3080。国产加速卡的加速比为1.89，并行效率为94.51%，相对而言，NVIDIA 1080的加速比仅为1.05，而并行效率为52.50%。NVIDIA 2080Ti和3080则显示出了较强的并行计算能力，其加速比分别为1.76和1.71，而并行效率分别为88.00%和85.50%，与国产加速卡的性能相媲美。而A100均强于上述显卡，其加速比和并行效率分别为1.79和89.58%。于是下文着眼于在加速比和并行效率变化不大的情况下优化AWP-ODC软件，在国产加速卡上使其浮点运算量得到提升，运行总时间得到下降。

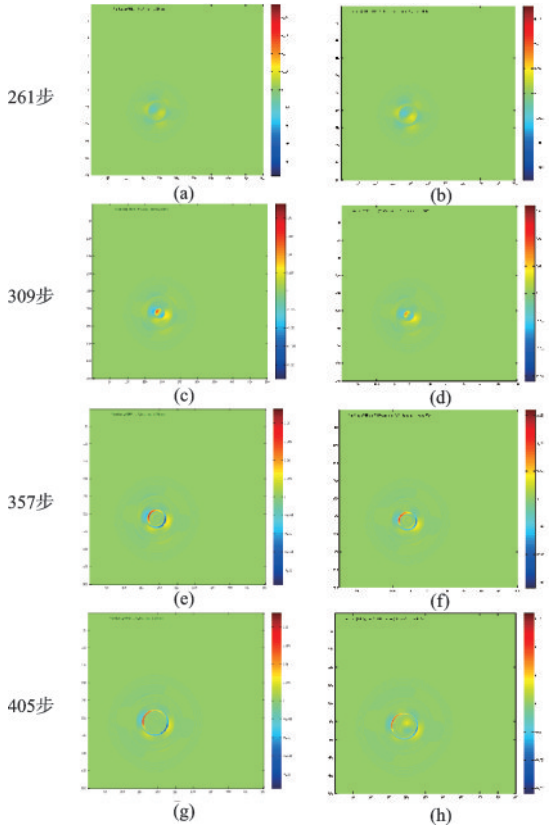

图8 Fréchet Kernels地震模拟波传播图

Fig.8 Fréchet Kernels earthquake simulation wave propagation

表3 Fréchet Kernels在不同GPU上的性能数据

Table 3 Performance data of Fréchet Kernels on different GPUs

GPU型号	卡数	浮点运算量/GFLOPS	运行总时间/秒
1080	1	48.974	59.618
2080Ti	1	66.714	43.765
3080	1	71.092	41.069
A100	1	88.195	33.693
国产加速卡	1	64.066	48.820
1080	2	52.105	56.973
2080Ti	2	119.578	24.871
3080	2	122.983	24.151
A100	2	160.852	18.805
国产加速卡	2	115.596	25.863

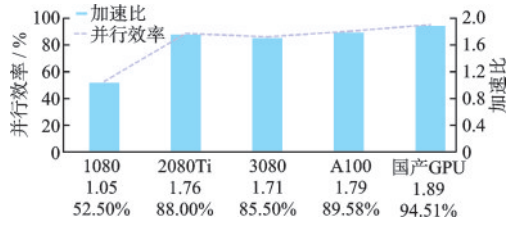

图9 Fréchet Kernels在不同GPU上的性能数据

Fig.9 Performance data of Fréchet Kernels on different GPUs

4.4 在8·3鲁甸地震模拟案例上的实验结果及计算性能分析

2014年8月3日下午4点30分，云南省昭通市鲁甸县（位于北纬27.1度、东经103.3度）遭遇里氏6.5级强震[38]，如图10所示。

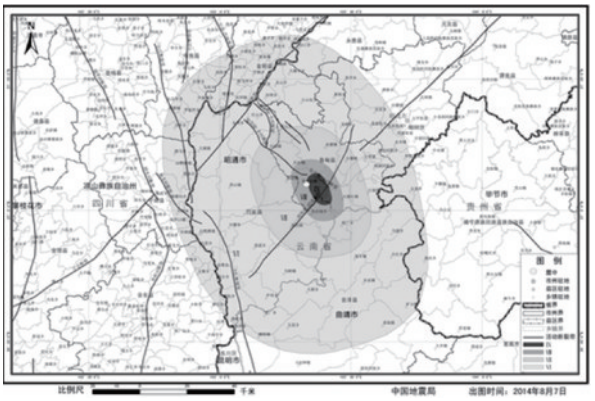

图10　云南鲁甸6.5级地震烈度图

Fig.10　Intensity map of the 6.5 magnitude Ludian earthquake, Yunnan

由于Fréchet Kernels地震波传播模拟的规模较小、运行时间较短，并且并行损耗较大，这里决定增大问题的规模，并选择了一个更大的8•3鲁甸地震实例进行模拟。

利用AWP-ODC软件仿真的结果如图11所示（其中最左侧一栏是地震模拟迭代的步数。(a)，(c)，(e)，(g)表示在国产加速卡上的波传播可视化，(b)，(d)，(f)，(h)表示在NVIDIA GPU 1080上的波传播可视化）。其中可以观察出在261步时由震源发出地震波随着步数的增加，地震波逐渐由强变弱的扩散开来。

同样在8•3鲁甸地震的地震波传播模拟任务中选用了NVIDIA GTX 1080、NVIDIA RTX 2080Ti、NVIDIA RTX 3080、NVIDIA A100和国产加速卡进行性能对比（表4）。

结合图12和表4来看，国产加速卡的浮点运算量、加速比和并行效率低于NVIDIA 2080Ti、3080和A100。

综合此前对Fréchet Kernels地震模拟的结果，国产加速卡首要仍是提高浮点运算量，以保证地震模拟的效率。

4.5 性能优化结果分析

首先运用第3节的程序优化方法，得出每个优化方法在单卡Fréchet Kernels地震模拟下提升的效率，如图13所示。

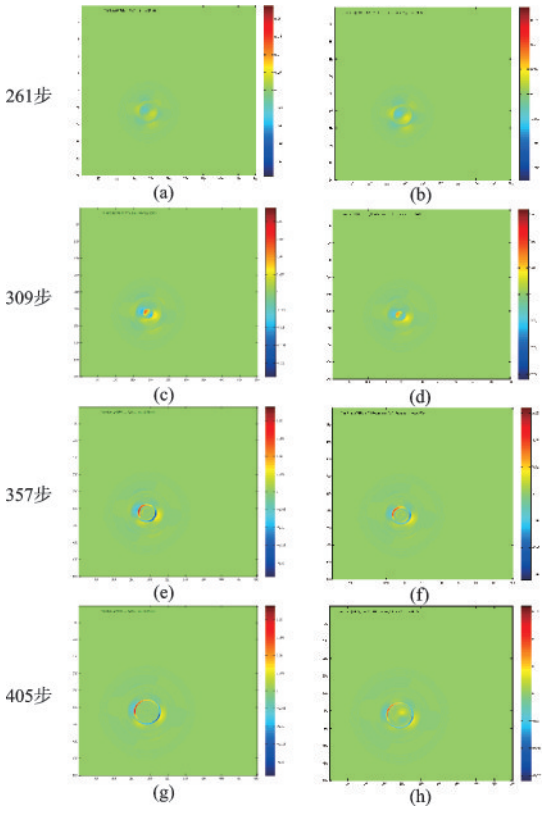

图11　8•3鲁甸地震模拟波传播图

Fig.11　Simulation wave propagation of the 8•3 Ludian earthquake

表4　鲁甸地震模拟在不同GPU上的性能数据

Table 4　Performance data of Ludian earthquake simulation on different GPUs

GPU型号	卡数	浮点运算量/GFLOPS	运行总时间/秒
1080	1	32.248	474.320
2080Ti	1	53.007	288.570
3080	1	57.591	265.600
A100	1	85.967	178.842
国产加速卡	1	59.482	257.156
1080	2	26.143	588.574
2080Ti	2	110.346	146.073
3080	2	105.208	146.379
A100	2	159.951	99.508
国产加速卡	2	94.452	162.740

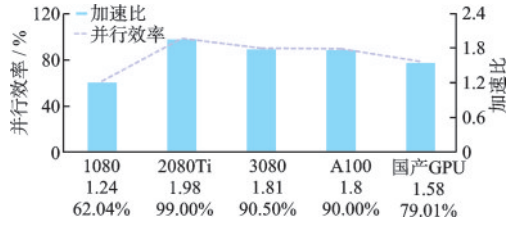

图12　鲁甸地震模拟在不同GPU上的性能数据

Fig.12　Performance data of Ludian earthquake simulation on different GPUs

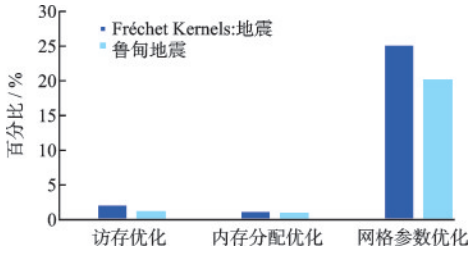

图13　单卡Fréchet Kernels地震模拟和鲁甸地震模拟下各方法提升的效率

Fig.13　Efficiency improvement of various methods under single-card Fréchet Kernel and Ludian earthquake simulations

在Fréchet Kernels地震波传播模拟中，访存优化、内存分配优化和网格参数优化分别提高了性能2.12%、1.23%和25.12%。这三种优化方法叠加后，总体提升达到30.51%，超出了各项优化提升的总和。在鲁甸地震仿真模拟中，访存优化、内存分配优化和网格参数优化的提升幅度分别为1.21%、1%和20.21%，叠加后的总体提升为25.21%，同样超过了各项优化提升的总和。

此外由图14可知，通过整体优化，单卡国产加速卡在Fréchet Kernels地震波传播模拟和鲁甸地震仿真模拟中的FLOPS分别提升至83.615 GFLOPS和74.475 GFLOPS，较未优化前分别提高了30.51%和25.21%。此外，运行总时间也分别缩短了28.47%和20.13%。

在双卡国产加速卡环境下，Fréchet Kernels和鲁甸地震模拟的FLOPS分别提升至126.489 GFLOPS和116.887 GFLOPS，较未优化前分别提高了9.42%和23.6%。运行时间也分别缩短了9.39%和19.11%。在加速比和并行效率上，Fréchet Kernels地震波传播模拟达到1.50和75%，8•3鲁甸地震仿真模拟中达到1.56和78%。

与未优化前相比，Fréchet Kernels地震波传播模拟的加速比略有下降，但是8•3鲁甸地震仿真模拟加速比基本不变。认为这与模拟的地震规模有关。具体而言，Fréchet Kernels地震波传播模拟规模较小，因此其加速比更容易受到优化措施的影响而产生浮动。相比之下，8•3鲁甸地震仿真模拟规模较大，其加速比相对稳定，不易受到优化措施的波动影响。

由图14可知，在Fréchet Kernels地震波传播模拟中，优化后的国产加速卡单卡浮点运算量较1080、2080ti、3080和国产加速卡分别提升了70.73%、25.33%、17.62%和30.51%，双卡浮点运算量分别提升了142.76%、5.78%、2.85%和9.42%，优于1080、2080ti、3080和未优化前的国产加速卡。但是仍弱于NVIDIA A100。

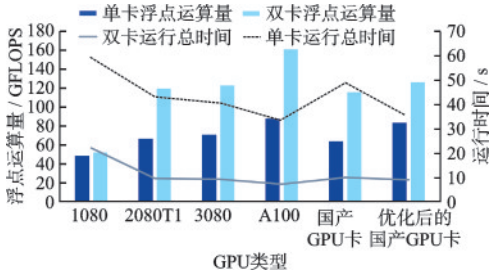

图14　Fréchet Kernels在单卡和双卡GPU上的性能数据

Fig.14　Performance data of Fréchet Kernels on single-card and dual-card GPUs

由图15可知，在8•3鲁甸地震仿真模拟中，优化后的国产加速卡单卡浮点运算量较1080、2080ti、3080和国产加速卡分别提升了130.94%、40.50%、29.23%和25.21%，双卡浮点运算量分别提升了347.11%、5.93%、11.10%和23.75%，远超1080，优于3080、2080Ti和未优化前的国产加速卡。但是仍弱于NVIDIA A100。

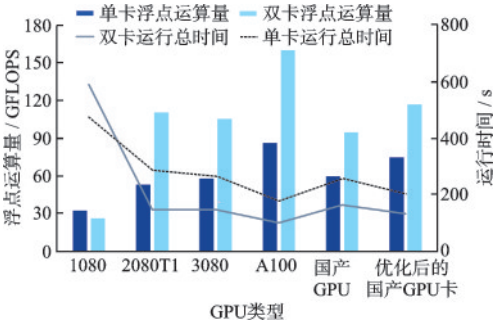

图15　鲁甸地震模拟在单卡和双卡GPU上的性能数据

Fig.15　Performance data of Ludian earthquake simulation on single-card and dual-card GPUs

5　总结和展望

5.1　总结

本文总结了近年来国内外在地震模拟加速技术领域的研究进展，并指出国内在GPU加速地震模拟技术方面仍存在一些不足，例如软件兼容性问题。为此，将AWP-ODC地震模拟软件移植至国产

加速卡，并以Fréchet Kernels地震事件和2014年8月3日发生的鲁甸地震为例进行了仿真模拟。通过与NVIDIA GTX 1080、NVIDIA RTX 2080Ti、NVIDIA RTX 3080和NVIDIA A100的性能比较，结果发现：

在Fréchet Kernels地震模拟中，国产加速卡表现出卓越的性能，其加速比分别为NVIDIA GTX 1080的1.8倍、NVIDIA RTX 2080Ti的1.07倍、NVIDIA RTX 3080的1.11倍和NVIDIA A100的1.05倍，体现出显著的性能优势。而在鲁甸地震模拟中，国产加速卡的加速比为NVIDIA GTX 1080的1.27倍，但相较于NVIDIA RTX 2080Ti、NVIDIA RTX 3080和NVIDIA A100，则分别为0.79倍、0.87倍和0.88倍，仍存在差距。总体表现出在不同地震场景下的综合性能。

之后，针对国产加速卡计算平台对程序进行了优化，在加速比和并行效率变化较小的情况下，使国产加速卡在浮点运算能力（FLOPS）和运行总时间上超过了NVIDIA GTX 1080、NVIDIA RTX 2080Ti和NVIDIA RTX 3080，但仍弱于NVIDIA A100。优化后，单卡国产加速卡在Fréchet Kernels地震波传播模拟和鲁甸地震仿真模拟中的FLOPS分别提升至83.615 GFLOPS和74.475 GFLOPS，较未优化前分别提高了30.51%和25.21%。运行总时间也分别缩短了28.47%和20.13%。在双卡国产加速卡环境下，Fréchet Kernels和鲁甸地震模拟的FLOPS分别提升至126.489 GFLOPS和116.887 GFLOPS，较未优化前分别提高了9.42%和23.6%。运行时间也分别缩短了9.39%和19.11%。

文中展示了国产加速卡在特定地震模拟任务中的潜在性能优势，尤其是在进行并行计算时。通过程序的优化和平台的适配，国产加速卡在部分应用场景下超越了国外GPU的性能指标。然而，在某些复杂模拟任务中，仍需进一步优化和提升，以缩小与高端NVIDIA GPU之间的性能差距。未来的研究可以进一步探索硬件优化与软件适配的结合，为地震模拟技术的发展提供更为强大的计算支持。

5.2 展望

本文主要在单节点上进行测试和优化，下一步可以在多节点对程序进行测试和优化。此外地震模拟软件传统上依赖于数值计算方法来模拟地震波在地质结构中的传播。然而，随着深度学习技术的快速发展，一些研究者开始探索利用深度学习的网络架构来加速地震模拟过程，并已经取得了显著成果[39-40]。鉴于地震模拟涉及大量数据处理和高计算成本，预计未来深度学习与地震模拟的融合将变得更为紧密。目前流行的深度学习模型，如Transformer和扩散模型（Diffusion model），在地震模拟领域展现出巨大潜力。Transformer能够处理和理解复杂的时空数据关系，从海量地震数据中提取有用信息，从而提升模型对地震活动的预测能力。而扩散模型则能够生成高质量的合成数据，帮助填补实际观测数据的不足，进而增强模拟结果的可靠性和准确性。此外，可以利用AWP-ODC模拟出的地震波扩散图像，将其作为扩散模型的训练集，以预测未来的波传播图像。综上所述，深度学习模型预期将在地震模拟领域发挥重要作用，为模拟的准确性和效率带来新的增长点。

参考文献：

[1] Olsen K B, Archuleta R J, Matarese J R. Three-dimensional simulation of a magnitude 7.75 earthquake on the San Andreas fault [J]. Science, 1995, 270 (5242): 1628-1632.

[2] Olsen K B, Day S M, Bradley C R. Estimation of Q for long-period (> 2 sec) waves in the Los Angeles basin[J]. Bulletin of the Seismological Society of America, 2003, 93(2): 627-638.

[3] Olsen K B, Day S M, Minster J B, et al. Strong shaking in Los Angeles expected from southern San Andreas earthquake[J]. Geophysical Research Letters, 2006, 33 (7): 1-4.

[4] Cui Y, Olsen K B, Jordan T H, et al. Scalable earthquake simulation on petascale supercomputers [C]// SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2010: 1-20.

[5] Christen M, Schenk O, Cui Y. Patus for convenient high-performance stencils: Evaluation in earthquake simulations[C]//SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE, 2012: 1-10.

[6] Roten D, Cui Y, Olsen K B, et al. High-frequency nonlinear earthquake simulations on petascale heteroge-

neous supercomputers[C]//SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2016: 957-968.

[7] Fu H H, He C H, Chen B W, et al.18.9-Pflops nonlinear earthquake simulation on Sunway TaihuLight: enabling depiction of 18-Hz and 8-meter scenarios[C]// Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2017: 1-12. http://doi.org/10.1145/3126908.3126910.

[8] 田浩东.基于新一代神威超级计算机的地震模拟并行优化方法研究[D].济南:山东大学, 2023.
Tian H D. Parallelizing and optimizing seismic simulations on the new-generation sunway[D]. Ji'nan: Shandong University, 2023. (in Chinese)

[9] 范国炜,吴涛,刘壮.基于新一代神威天气和气候预测系统并行优化[J].计算机仿真,2023,40(12):353-358.
Fan G W,Wu T,Liu Z.Parallel optimization of weather and climate prediction system based on new generation of sunway[J].Computer Simulation,2023,40(12):353-358.(in Chinese)

[10] 胡文娇.SM2算法在天河新一代超级计算机上的实现和优化[D].长沙:湖南大学,2023.
Hu W J.The implemention and optimization of the SM2 algorithm on tianhe new generation supercomputers[D]. Changsha: Hunan University,2023. (in Chinese)

[11] 郝萌, 田雪洋, 鲁刚钊, 等.基于国产DCU异构平台的图匹配算法移植与优化[J].计算机科学, 2024, 51(4): 67-77.
Hao M, Tian X Y, Lu G Z, et al. Transplanation and optimization of graph matching algorithm based on domestic DCU heterogeneous platform[J]. Computer Science, 2024, 51(4): 67-77. (in Chinese)

[12] NVIDIA Corporation. NVIDIA Developer Zone [EB/OL]. (n. d.) [2024-07-30]. https://developer. nvidia.com.

[13] Sanders J, Kandrot E. CUDA by example: an introduction to general-purpose GPU programming[M]. USA: Addison-Wesley Professional, 2010.

[14] Cook S. CUDA programming: a developer's guide to parallel computing with GPUs [M]. USA: Morgan Kaufmann Publishers Inc., 2012.

[15] IncAMD. ROCm Documentation: HIP Programming Guide [EB/OL]. (n. d.) [2024-07-30]. https://rocmdocs.amd.com/en/latest/Programming_Guides/HIP-GUIDE.html.

[16] Jin Z, Vetter J S. Evaluating Unified Memory Performance in HIP[C]//2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2022: 562-568.

[17] 黄聪祎, 赵伟文, 万德成.国产DCU加速卡与MPS方法结合高效模拟带障碍物溃坝流动问题[J].水动力学研究与进展A辑, 2024, 39(2): 187-195.
Huang C Y, Zhao W W, Wan D C. Efficient simulation of obstructed dam-break flows using the MPS method on domestic DCU accelerator card[J]. Chinese Journel of Hydrodynamics, 2024, 39(2): 187-195.(in Chinese)

[18] Niu J, Gao W, Han L, et al. A DCU code generation and optimization method based on polyhedral model[J]. In International Conference on Cloud Computing Performance Computing and Deep Learning , 2023, 12712: 416-428.

[19] 李冰洋.面向"嵩山"超级计算机系统的供水管网仿真计算移植与优化[D].郑州:郑州大学, 2022.
Li B Y. Transplantation and optimization of water supply network simulation calculation for "Songshan" supercomputer system[D].Zhengzhou: Zhengzhou University, 2022. (in Chinese)

[20] 杨思驰, 赵荣彩, 韩林, 等.面向DCU的LDS访存向量化优化[J].计算机工程, 2024, 50(2): 206-213.
Yang S C, Zhao R C, Han L, et al. Vectorization optimization of LDS memory access for DCU[J]. Computer Engineering, 2024, 50(2): 206-213. (in Chinese)

[21] Poursartip B, Fathi A, Tassoulas J L. Large-scale simulation of seismic wave motion: A review[J]. Soil Dynamics and Earthquake Engineering, 2020, 129: 105909.

[22] Xiaolin H, Xiaofeng J. High-order dynamic lattice method for seismic simulation in anisotropic media[J]. Geophysical Journal International, 2018, 212(3): 1868-1889.

[23] Tessmer E. Seismic finite-difference modeling with spatially varying time steps[J]. Geophysics, 2000, 65(4): 1290-1293.

[24] Virieux J, Calandra H, Plessix R É. A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging[J]. Geophysical Prospecting, 2011,59: 794-813.

[25] Tobin J, Breuer A, Heinecke A, et al. Accelerating seismic simulations using the intel xeon phi knights landing processor[C]//International Conference on High Performance Computing. Cham: Springer International

Publishing, 2017: 139-157.

[26] Mu D, Moran J, Zhou H, et al. In-situ analysis and visualization of earthquake simulation [C]//Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning). PEARC, 2019: 1-5.

[27] Zhou Q, Chu C, Kumar N S, et al. Designing high-performance mpi libraries with on-the-fly compression for modern gpu clusters[C]//2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2021: 444-453.

[28] Cui Y, Roten D, Palla A, et al. Progress of porting AWP-ODC to next generation HPC architectures and a 4-Hz Iwan-type nonlinear dynamic simulation of the ShakeOut scenario on TACC Frontera[J]. SCEC Publications, 2023: 1.

[29] Zhou H. Application of intelligent optimization algorithms to the design of automatic generation of software tests for data anomaly identification[J]. Applied Mathematics and Nonlinear Sciences, 2023, 9(1): 1-17.

[30] SCEC. AWP-ODC - SCECpedia [EB/OL]. (2024-11-06) [2024-11-06]. https://scec. usc. edu/scecpedia/AWP-ODC

[31] Wang G, Ji S, Lv H, et al. Drucker-prager yield criteria in viscoelastic-plastic constitutive model for the study of sea ice dynamics[J]. Journal of Hydrodynamics, 2006, 18(6): 714-722.

[32] 杨庆节, 刘财, 耿美霞, 等. 交错网格任意阶导数有限差分格式及差分系数推导[J]. 吉林大学学报(地球科学版), 2014, 44(1): 375-385.
Yang Q J, Liu C, Geng M X, et al. Staggered grid finite difference scheme and coefficients deduction of any number of derivatives[J]. Journal of Jilin University (Earth Science Edition), 2014, 44(1): 375-385. (in Chinese)

[33] Berenger J P. A perfectly matched layer for the absorption of electromagnetic waves[J]. Journal of Computational Physics, 1994, 114(2): 185-200.

[34] Komatitsch D, Martin R. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation[J]. Geophysics, 2007, 72(5): SM155-SM167

[35] cppreference. com. C++ Reference: C++17 Features [EB/OL]. (n. d.) [2024-07-30]. https://en. cppreference.com/w/cpp/17.

[36] 商建东, 熊威, 华浩波, 等. 面向DCU的流固耦合浸没边界算法异构实现[J]. 计算机工程, 2024, doi: 10-19678/j.issn.1000-3428.0068818.
Shang J D, Xiong W, Hua H B, et al. Heterogeneous implementation of fluid-structure interaction immersed boundary method for DCU[J]. Computer Engineering, 2024, doi: 10-19678/j. issn. 1000-3428.0068818. (in Chinese)

[37] Zhao L, Jorden T H, Olsen K B, et al. Fréchet Kernels for imaging regional earth structure based on three-dimensional reference models[J]. Bulletin of the Seismological Society of America, 2005, 95: 2066-2080.

[38] 冀昆, 温瑞智, 崔建文, 等. 鲁甸M_S6.5级地震强震动记录及震害分析[J]. 震灾防御技术, 2014, 9(3): 325-339.
Ji K, Wen R Z, Cui J W, et al. Ludian M_S6.5 earthquake earthquake real vibration record and earthquake analysis[J]. Technology for Earthquake Disaster Prevention, 2014, 9(3): 325-339. (in Chinese)

[39] Moseley B, Nissen-Meyer T, Markham A. Deep learning for fast simulation of seismic waves in complex media[J]. Solid Earth, 2020, 11(4): 1527-1549.

[40] Ali Najah nori, Walid fahs. Seismic waves near oil reservoir prediction using deep learning[J]. Humanitarian and Natural Sciences Journal, 2023, 4(9): 177-187.

(本文编辑:池营营)