

文章编号: 1006-2467(2023)08-0981-07

DOI: 10.16183/j.cnki.jsjtu.2022.209

一种改进 GPU 加速策略在光滑粒子 流体动力学方法中的应用

管延敏¹, 杨彩虹¹, 康庄², 周利¹

(1. 江苏科技大学 船舶与海洋工程学院, 江苏 镇江 212003;

2. 哈尔滨工程大学 船舶工程学院, 哈尔滨 150001)

摘要: 为解决粒子的无序化易引起的图形处理器(GPU)内存访问冲突问题和提高计算效率, 通过建立粒子重排序技术提出了一种改进的 GPU 加速策略. 将该加速策略应用于光滑粒子流体动力学(SPH)方法中对三维带障碍物溃坝进行模拟, 并与实验结果对比对算法进行验证, 获得了较高的计算精度. 基于此算例, 通过在不同硬件设施上进行模拟分别对粒子重新编号的效果和算法的求解效率比较研究. 结果表明, 粒子重新编号技术可以保证稳定的单步运行时间, 能够有效解决 GPU-SPH 算法显存访问冲突问题; 该 GPU 加速的 SPH 并行算法能够大幅提高 SPH 方法求解效率, 随着粒子数量的增加, 其大幅缩短计算时间的优势愈发明显, 为扩大 SPH 方法解决三维数值模拟的适用性提供了可能.

关键词: 光滑粒子流体动力学; 并行计算; 溃坝问题; 计算效率

中图分类号: U 661.1

文献标志码: A

Application of an Improved GPU Acceleration Strategy for the Smoothed Particle Hydrodynamics Method

GUAN Yanmin¹, YANG Caihong¹, KANG Zhuang², ZHOU Li¹

(1. School of Naval Architecture and Ocean Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, Jiangsu, China; 2. College of Shipbuilding Engineering,

Harbin Engineering University, Harbin 150001, China)

Abstract: In order to solve the problem of graphics processing unit (GPU) memory access conflicts possibly caused by the disorder of particles and enhance the computation efficiency, an improved GPU acceleration strategy is proposed by establishing particle reorder technology. The acceleration strategy is applied to the smoothed particle hydrodynamics (SPH) method to simulate the dam breaking with obstacles in three dimensions, and the algorithm is verified by comparing with the experimental results, which obtained a high calculation accuracy. Based on this benchmark example of the SPH, the studies on the effect of particle renumbering and the solution efficiency of the algorithm are conducted by comparing the simulations of different hardware facilities. The results indicate that the particle reorder technology can ensure a stable single-step running time, and can effectively solve the problem of graphic card memory access conflicts that commonly exist in the GPU-SPH algorithm. Furthermore, the GPU parallel algorithm

收稿日期: 2022-06-13 修回日期: 2022-08-09 录用日期: 2022-08-26

基金项目: 国家重点研发计划(2022YFE0107000), 国家自然科学基金面上项目(52171259)

作者简介: 管延敏(1983-), 高级工程师, 从事船舶水动力学研究.

通信作者: 杨彩虹, 博士, 讲师, 电话(Tel.): 0511-84401178; E-mail: ychnihao@126.com.

can greatly improve the solution efficiency of the SPH method, and with the increase of particle number, the advantage of drastically reducing the computation time becomes more obvious. The method proposed in this paper provides the possibility to expand the application of the SPH method to solve 3D numerical simulations.

Key words: smoothed particle hydrodynamics (SPH); parallel computing; dam breaking; computational efficiency

光滑粒子流体动力学 (Smoothed Particle Hydrodynamics, SPH) 对模拟自由表面流动问题具有先天优势,但随着粒子数量的增加,尤其是三维流动问题,计算效率急剧降低,极大限制了该方法在大规模模拟计算中的适用性。利用图形处理器 (GPU) 众核架构开展多核并行计算的计算机统一设备架构 (CUDA) 技术,其强大的并行特性非常适合解决大规模的高级计算问题^[1],尤其是计算密集型问题。SPH 计算能力密集的特性使其能容易地在 GPU 上实现并行运算,因此一些学者开始尝试将 GPU 技术应用于 SPH 方法。Harada 等^[2]首次描述了 SPH 方法在 GPU 上的实现,Crespo 等^[3]将 GPU 加速的 SPH 方法应用于模拟复杂的自由表面流动。他们的研究均表明在单个 GPU 上的 SPH 模拟比在单核中央处理器 (CPU) 上进行的 SPH 模拟要快两个数量级。目前 SPH 方法在 GPU 平台下的计算研究发展迅速,GPU 加速的 SPH 模型已应用于颗粒流体流动的模拟和浅水方程的求解^[4-5],并通过结合自适应粒子技术^[6]实现了高效计算。国内,徐锋^[7]在实现了基于 GPU 众核架构的并行 SPH 算法的同时,结合所使用的 GPU 硬件上的特点,对并行算法进行了优化,将计算效率提高了 20 倍以上;金善勤等^[8]提出一种基于粒子对的并行计算方法并将其与改进的 SPH 方法结合,实现了超过 10 的加速比;杨志国等^[9]将 GPU 算法应用于二维溃坝模拟中也获得了数量级的加速效果。

但是,GPU 的系统设计理念与 CPU 正好相反,GPU 面对的是类型一致、互不关联的大量数据,其显存的读写对算法求解效率影响很大。随着流场的不断演变,SPH 粒子的无序化很容易导致多个线程同时读写同一地址而引起访问冲突,极大地影响了计算效率的稳定性。针对此问题,并行索引排序方法^[3, 10]、Z 索引排序方法^[11]及其改进方法^[12]相继被提出,这些方法通过优化 SPH 粒子索引存储方式使同单元粒子在 GPU 显存中尽可能相邻,一定程度上改善了 GPU 显存访问的不连续问题,但没有从实质上改变单元粒子的无序化。为此,本文提出了一

种粒子重新编号技术,实现了 SPH 粒子的有序排列和 GPU 显存的连续访问,并将该方法应用于三维带障碍物溃坝模拟,通过与实验数据比较以及不同硬件设施上不同算法求解效率的对比,验证了本文方法的精确性和高效性。

1 SPH 的控制方程

SPH 控制方程的离散形式可表示为

$$\frac{d\rho_i}{dt} = - \sum_j \rho_j (\mathbf{u}_j - \mathbf{u}_i) \cdot \nabla \mathbf{W}_{ij} V_j \quad (1)$$

$$\frac{d\mathbf{u}_i}{dt} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \prod_{ij} \right) \nabla \mathbf{W}_{ij} + \mathbf{g} \quad (2)$$

式中: ρ_i 、 p_i 、 \mathbf{u}_i 、 V_i 分别为流体粒子 i 的密度、压力、速度和体积; t 为时间; m_j 和 ρ_j 为粒子 i 支持域中粒子 j 的质量和密度; \mathbf{g} 为重力加速度; \mathbf{W}_{ij} 为核函数,本文采用 Wendland^[13] 的 C2 核函数:

$$W(r, h) = \alpha_d \left(1 - \frac{q}{2} \right)^4 (2q + 1) \quad (3)$$

$$(0 \leq q \leq 2)$$

式中: r 为两粒子间距离; h 为光滑长度; q 为粒子间相对距离, $q = r/h$; α_d 为常数,对于三维问题, $\alpha_d = 21/8\pi h^3$ 。

人工黏性为

$$\prod_{ij} = \begin{cases} -\frac{\alpha \bar{c}_{ij} \mu_{ij}}{\bar{\rho}_{ij}}, & \mathbf{u}_{ij} \cdot \mathbf{r}_{ij} < 0 \\ 0, & \mathbf{u}_{ij} \cdot \mathbf{r}_{ij} > 0 \end{cases} \quad (4)$$

式中:

$$\mu_{ij} = \frac{h \mathbf{u}_{ij} \cdot \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + (0.01 h)^2} \quad (5)$$

α 为控制速度耗散强度的系数,通常为 $0 \sim 0.5$,取 0.015 ;平均声速 $\bar{c}_{ij} = 0.5(c_i + c_j)$; $\bar{\rho}_{ij}$ 为平均密度;粒子速度差 $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$;粒子间距 $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$; $(0.01 h)^2$ 为添加项,避免分母为 0 导致数值发散。

假定流体弱可压缩,采用 Monaghan 等^[14]提出的人工压缩法求解压强:

$$p = \frac{c_0^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] + p_{\text{atm}} \quad (6)$$

式中: ρ_0 为参考密度,取 $\rho_0 = 1\,000 \text{ kg/m}^3$; p_{atm} 为大

气压力; γ 为常数, $\gamma=7$; c_0 为人工声速,为确保密度波动低于 $1\%\rho_0$,本文取 $c_0=10\sqrt{gH}$, H 为水深.

2 CUDA-GPU 架构下的 SPH 模型

2.1 并行算法的实现流程

针对现代 CUDA-GPU 架构设计了一种高效的 SPH 流体数据,以提高 SPH 流体模拟在 GPU 上的数据处理速度.并行代码使用 C# 语言和 CUDA 编程语言进行开发,大多数源代码都是 CPU 和 GPU 所共有的,可以在 CPU 或 GPU 上执行,也可以在没有启用 GPU 的工作站上运行,只使用 CPU 实现.并行算法实现的流程如图 1 所示,粒子信息的初始数据存储在 CPU 中,将数据传输到 GPU 中之后,后续的所有计算都在 GPU 中进行,即所有涉及粒子循环的任务执行都是由 GPU 的并行结构来完成的.在完成控制方程求解和粒子物理量更新之后,将更新后的粒子物理量从 GPU 再次传输到 CPU 进行保存和输出.

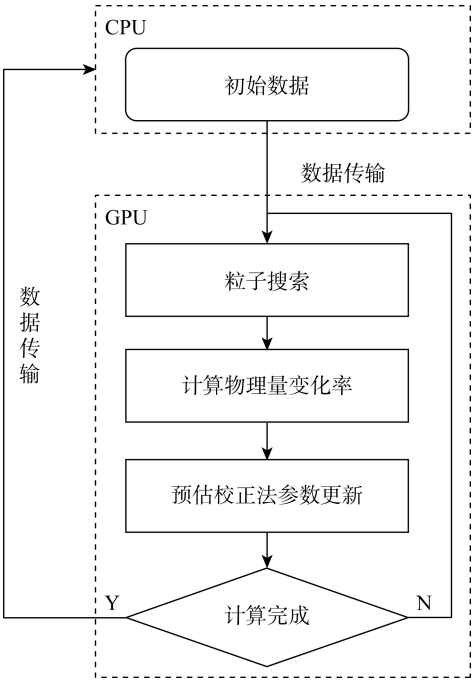


图 1 SPH 并行算法的实现流程
Fig. 1 Parallel arithmetic mode of SPH

2.2 粒子搜索

SPH 方法由于核函数的紧支性,只有粒子支持域内的相邻粒子才会相互作用,因粒子间的相对位置不断变化,每个时刻都需要进行最近相邻粒子的搜索,所以搜索算法极大地影响着 SPH 的计算效率.本文采用 Monahan^[15] 的链表搜索算法,通过使用控制网格搜索粒子的相邻质点.其主要思想为:在

计算域建立临时网格,网格大小和粒子支持域大小一致,然后建立每个粒子与所处网格的关联形成关联链表.对于三维问题,计算域被划分为立方单元,每个单元与其周围的单元(共 27 个)组成紧邻单元,如图 2 所示.粒子根据它们在区域中的位置被放置在单元格中,在搜索粒子时只需搜索粒子所在的紧邻单元即可,形成最近相邻粒子对.

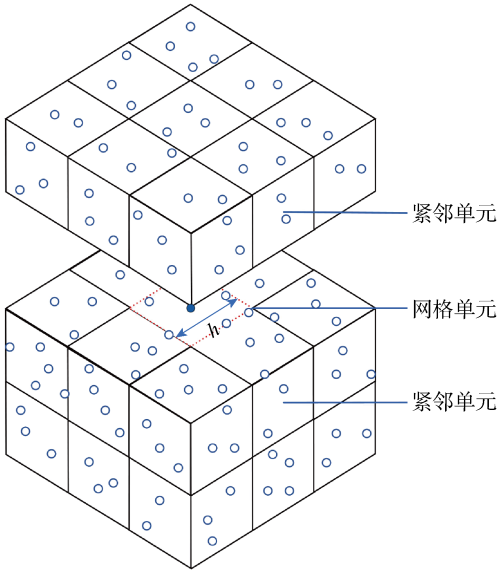


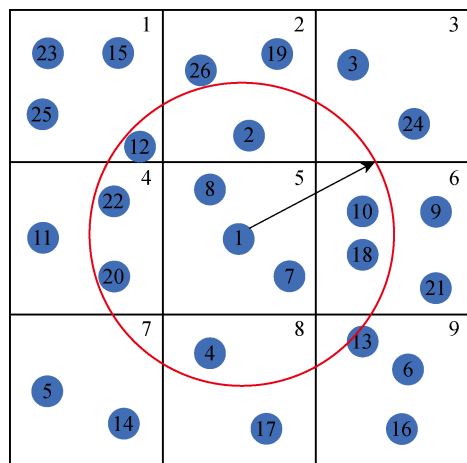
图 2 三维相邻粒子链表搜索法示意图
Fig. 2 Sketch for three-dimensional neighbor list search method

2.3 粒子重新编号技术

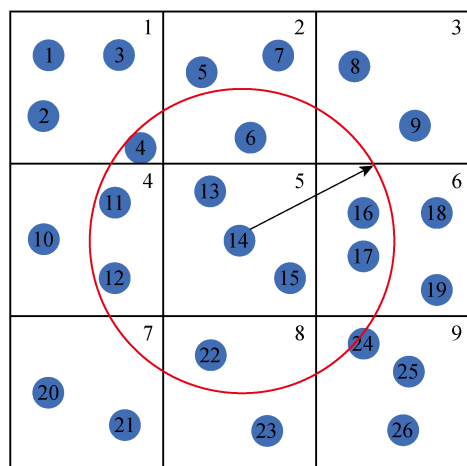
邻近粒子搜索的 GPU 并行算法的构建与在 CPU 上使用的过程有所不同,在 GPU 上需要建立线程与粒子之间的关联,并为每个粒子单独分配一个计算线程.所有粒子按编号排入一个连续的标签数组,并给固定数目的连续粒子分配一个线程块.在流场模拟过程中,随着时间步不断推进,SPH 粒子分布与初始粒子分布差异越来越大,链表搜索法中 SPH 粒子不再是顺序分布,如图 3(a)所示,这就导致线程结束并行访问内存时,极易导致多个线程同时对同一内存地址进行读写操作而产生访问冲突,降低并行算法的求解效率.为保证算法效率稳定性,提出了一种粒子重新编号方法,图 3 显示了用于生成按单元格重新编号法的粒子标签数组简单示意图.

该方法主要包含以下步骤:

- (1) 遍历所有 SPH 粒子,计算每个粒子所在网格索引,统计该索引所在网格存储 SPH 粒子数量,存储于数组 IDCount 中.
- (2) 创建数组 IDBegin,记录新粒子编号中每个网格中首粒子编号,对于网格 m 有 $IDBegin[m] =$



(a) 未重新编号



(b) 重新编号后

图 3 粒子重新编号算法示例

Fig. 3 Example of neighbor list reorder

IDBegin[m-1] + IDCount[m-1];

(3) 清空数组 IDCount;

(4) 遍历所有 SPH 粒子, 重新统计该索引所在网格存储 SPH 粒子数量, 根据粒子所在网格索引, 对其重新编号, 如对于某 SPH 粒子, 其对于网格索引为 n , 其新的粒子编号为 $IDBegin[n] + IDCount[n]$, 其中 $IDCount[n]$ 为统计至该粒子时网格 n 中存储的 SPH 粒子数量, 最终得到的重新粒子编号效果如图 3(b) 所示。

3 结果验证与分析

以 SPH 法验证自由表面流动问题的基准测试案例——溃坝问题为研究对象, 对三维带障碍物的溃坝进行数值模拟, 通过与实验数据进行对比验证本文并行算法的可靠性和有效性。

3.1 溃坝模型

荷兰海洋研究所(Marin)的溃坝试验^[16]被广泛认为是验证 SPH 自由表面流动的基准算例。试验包

含一个溃坝流与障碍物的碰撞, 如图 4 所示。水箱长 3.22 m、宽 1 m、高 1 m, 水柱被储存在水箱一端, 长 1.228 m、宽 1 m、高 0.55 m, 并在试验开始瞬间释放。障碍物设置在水流下游, 随着挡水墙的拆除, 由于重力作用, 流体逐渐淹没水箱干床并与障碍物发生碰撞。试验通过设置 3 个垂直高度探头(H1、H2 和 H3)测量不同位置的水深, 具体位置如图 4 所示。

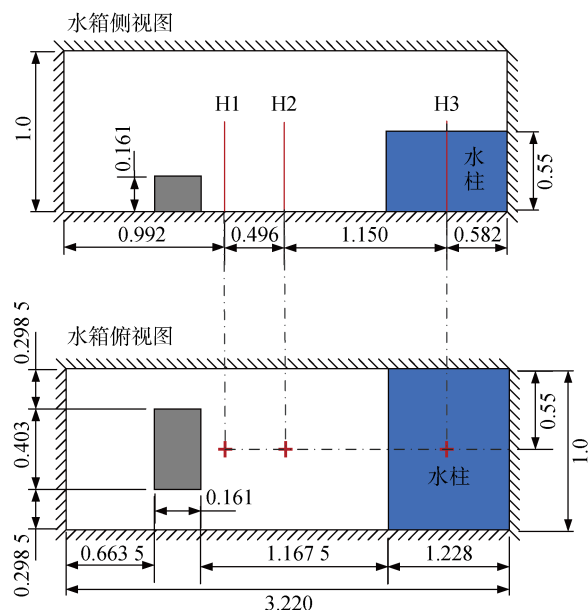


图 4 试验配置和试验数据的测量位置(m)

Fig. 4 Experimental configuration and measurement position of experimental data (m)

3.2 结果对比

数值模拟选取初始粒子间距为 0.01 m、流场实际粒子总数为 676 500, 时间步长为 0.1 ms。为验证本文粒子重新编号方法的准确性, 对不同位置处的水深展开比较研究。图 5 所示为 H1、H2 和 H3 位置处不同时刻的水深粒子重新编号前后计算值和实验值的比较。图中: h_w 为水柱高度。可见粒子重新编号前后计算值基本一致, 不会对精度造成影响; 在初始水柱中间(H3)、箱体中部(H2)、障碍物前缘附近(H1), 本文计算值与实验值的趋势和大小都很接近。图中 H3 探头清晰地再现了溃坝的整个过程, 在最初的 2 s 内水柱坍塌, 相应地, 这段时间的水位也不断下降, 而在其他探头中, 水流依次到达 H2 和 H1。在 1.75 s 后, 反射的水波撞击左墙后反向移动, 第 2 次撞击障碍物以及右墙, 同时, 水深达到第 2 个峰值(H1 的水深峰值时刻大概为 4.8 s, H2 为 4.6 s, H3 为 3.8 s)。本文计算的水深与实验水深随时间的变化趋势大致相同, 表明本文方法具有良好的计算精度。

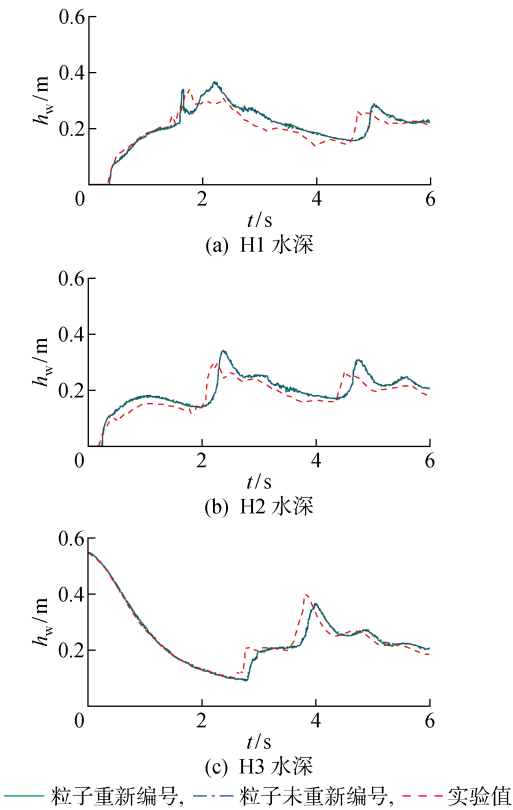


图 5 实验测量和本文模拟的探测点处的垂直水柱高度比较

Fig. 5 Comparison of vertical water heights at the detection point measured experimentally and simulated in this paper

4 效率测试结果与分析

4.1 测试平台

通过比较 CPU 串行、CPU 并行、GPU 并行 SPH 算法三维带障碍物溃坝数值模拟计算耗时,验证本文所采用 GPU 算法求解效率. 为保证计算结果的适用性,分别在不同的 CPU、GPU 硬件上运行 SPH 算法,具体配置如表 1 所示.

表 1 CPU 和 GPU 配置

Tab. 1 Hardware configurations of CPU and GPU

设备类型	型号	核数	主频/GHz
CPU	Intel Core i9-10900F	10	2.8
	Xeon® Gold 6268CL	48	2.8
GPU	GTX 1660 super	1 408	1.785
	GeForce RTX 2080	2 944	1.4
	GeForce RTX 3080Ti	10 240	1.67

4.2 GPU 算法粒子重新编号效果对比

图 6 为实粒子总数 676 500 时不同硬件条件下

粒子重新编号和未重新编号 GPU 加速 SPH 算法单步运行时间(t_s)对比. 图中: s_n 为计算步数. 由图可见,本文采用的粒子重新编号算法在不同硬件上都获得了稳定的单步运行时间,而粒子未重新编号时,随着流场中粒子的无序化导致 GPU 显存访问冲突,其单步运行时间呈对数增长,算例表明本文所采用的粒子重新编号方法可以保证稳定的单步运行时间,是有效的.

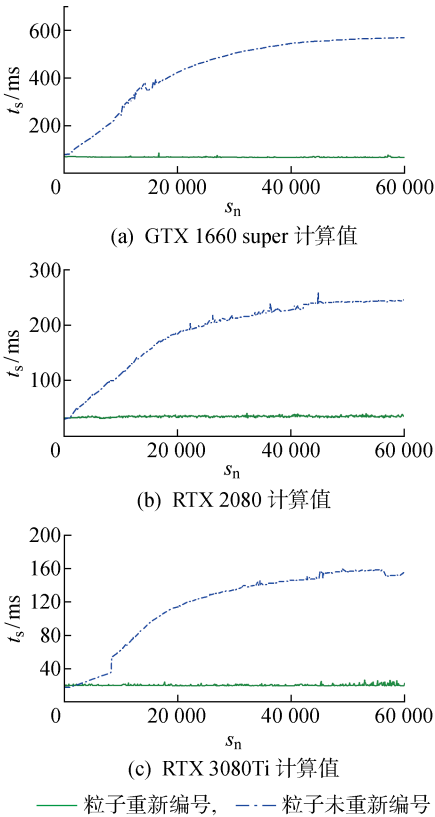


图 6 粒子重新编号和未重新编号单步运行时间对比

Fig. 6 Comparison of step running time between re-order and non-reorder method

4.3 算法求解效率对比

图 7 为实粒子数为 676 500 时 CPU 并行和 GPU 并行算法迭代 60 000 步单步运行时间对比,可见 GPU 并行算法都有良好的计算效率,而求解效率稳定性弱于 CPU 并行算法. 受 SPH 方法部分算法、函数间存在串行关系影响,计算效率未能随核数增加而线性增加,以 Intel Core i9-10900F 为参照,各硬件核数、效率比如表 2 所示,可见相对计算成本(核数效率比)随核数的增加而增大. 图 8 为不同实粒子数下 CPU 并行、GPU 并行 SPH 算法单步平均用时(t_m)对比. 图中: n_p 为实粒子数. 可见随着粒子数量的增加,CPU 并行算法运行时间显著增加,GPU 并行算法大幅缩短计算时间的优势愈发明显.

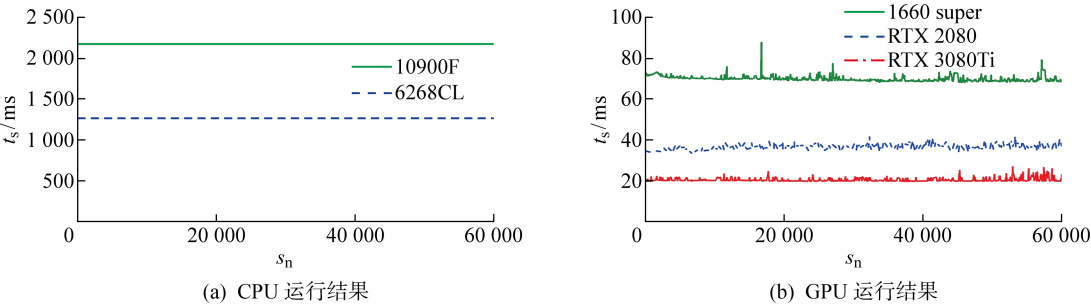


图 7 CPU 并行和 GPU 算法单步运行时间对比
Fig. 7 Comparison of step running time between CPU parallel and GPU parallel

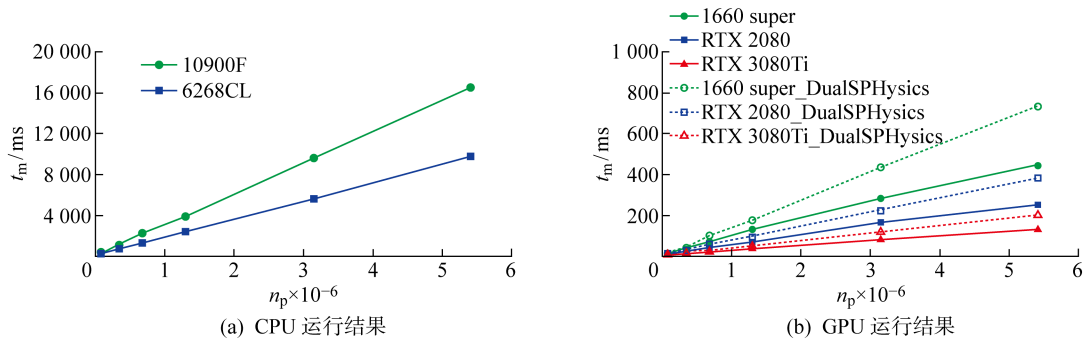


图 8 不同实粒子总数下 CPU 与 GPU 运行时间的对比
Fig. 8 Comparison of running time between CPU and GPU at different number of particles

表 2 各硬件核数和效率比

Tab. 2 Number of cores and efficiency ratio of different hardwares

硬件	核数比	效率比	核数/效率
Intel Core i9-10900F	1	1	1
Xeon® Gold 6268CL	4.8	1.73	2.78
GTX 1660 super	140.8	31.05	4.53
GeForce RTX 2080	294.4	59.04	4.99
GeForce RTX 3080Ti	1 024	104.8	9.76

为进一步验证本文粒子重新编号算法的有效性,对不同实粒子数下 GPU 算法并行效率与开源软件 DualSPHysics 进行了对比,如图 8(b)所示,可见同实粒子总数、同硬件设备条件下,本文方法平均单步运行时间均小于 DualSPHysics 软件,算例表明本文粒子重新编号方法具有良好的效率优势。

5 结论

运用粒子重新编号技术开发了一套高效 GPU-SPH 并行算法,将该算法应用于三维带障碍物溃坝问题,并对算法求解效率进行了比较研究,得到以下结论:

(1) 粒子重新编号前后计算值基本一致,不会

对精度造成影响,与试验值的对比表明本文所采用的方法精确有效。

(2) 粒子重新编号技术能够有效解决 GPU-SPH 算法中的显存访问冲突问题。

(3) GPU 并行算法能够大幅提高 SPH 方法求解效率,随着粒子数量的增加,其大幅缩短计算时间的优势愈发明显。

参考文献:

[1] 刘肃肃, 胡祎乐, 余音. 基于 GPU 的近场动力学模拟的并行化方法[J]. 上海交通大学学报, 2016, 50 (9): 1362-1367.
LIU Susu, HU Yile, YU Yin. Parallel computing method of peridynamic models based on GPU [J]. Journal of Shanghai Jiao Tong University, 2016, 50 (9): 1362-1367.
[2] HARADA T, KOSHIZUKA S, KAWAGUCHI Y. Smoothed particle hydrodynamics on GPUs [C] // Computer Graphics International. Petropolis, Brazil: Computer Graphics Society, 2007, 40: 63-70.
[3] CRESPO A C, DOMINGUEZ J M, BARREIRO A, et al. GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods[J]. PloS One, 2011, 6(6): 1-13.

- [4] HE Y, BAYLY A E, HASSANPOUR A, *et al.* A GPU-based coupled SPH-DEM method for particle-fluid flow with free surfaces[J]. **Powder Technology**, 2018, 338: 548-562.
- [5] XIA X, LIANG Q. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations[J]. **Environmental Modelling & Software**, 2016, 75: 28-43.
- [6] XIONG Q, LI B, XU J. GPU-accelerated adaptive particle splitting and merging in SPH[J]. **Computer Physics Communication**, 2013, 184 (7): 1701-1707.
- [7] 徐锋. 基于众核架构的并行 SPH 算法的研究与实现[D]. 上海: 上海交通大学, 2013.
- XU Feng. Research and implementation of the smoothed particle hydrodynamics algorithm based on multi-core architecture[D]. Shanghai: Shanghai Jiao Tong University, 2013.
- [8] 金善勤, 郑兴, 段文洋. 基于 GPU 并行的改进 SPH 方法对黏性流场的模拟[J]. **哈尔滨工程大学学报**, 2015, 36(8): 1011-1018.
- JIN Shanqin, ZHENG Xing, DUAN Wenyang. Viscosity flow simulation using improved SPH method based on GPU parallel calculation [J]. **Journal of Harbin Engineering University**, 2015, 36(8): 1011-1018.
- [9] 杨志国, 黄兴, 郑兴, 等. GPU 在 SPH 方法模拟溃坝问题的应用研究[J]. **哈尔滨工程大学学报**, 2014, 35(6): 661-666.
- YANG Zhiguo, HUANG Xing, ZHENG Xing, *et al.* The application research of GPU in the SPH method to simulate the dam breaking problem[J]. **Journal of Harbin Engineering University**, 2014, 35(6): 661-666.
- [10] 车庆首, 李传文, 张轶, 等. GAPI: GPU 加速的移动对象并行索引方法[J]. **计算机科学与探索**, 2017, 11(11): 1713-1722.
- CHE Qingshou, LI Chuanwen, ZHANG Yi, *et al.* GAPI: GPU accelerated parallel method for indexing moving objects[J]. **Journal of Frontiers of Computer Science and Technology**, 2017, 11(11): 1713-1722.
- [11] IHMSEN M, AKINCI N, BECKER M, *et al.* A parallel SPH implementation on multi-core CPUs[J]. **Computer Graphics Forum**, 2011, 30: 99-112.
- [12] 聂霄. 不可压缩 SPH 流体的真实感模拟及其加速技术研究[D]. 成都: 电子科技大学, 2015.
- NIE Xiao. Study on realistic simulation and acceleration techniques of incompressible SPH fluids[D]. Chengdu: University of Electronic Science and Technology of China, 2015.
- [13] WENDLAND H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree[J]. **Advances in computational Mathematics**, 1995, 4(1): 389-396.
- [14] MONAGHAN J J, GINGOLD R A. Shock simulation by the particle method SPH[J]. **Journal of Computational Physics**, 1983, 52(2): 374-389.
- [15] MONAGHAN J J. Particle methods for hydrodynamics[J]. **Computer Physics Report**, 1985, 3(2): 71-124.
- [16] KLEEFSMAN K M T, FEKKEN G, VELDMAN A E P, *et al.* A volume-of-fluid based simulation method for wave impact problems[J]. **Journal of Computational Physics**, 2005, 206: 363-393.

(本文编辑: 王一凡)