Available online at www.sciencedirect.com
ScienceDirect
Journal of Hydrodynamics
2017,29(4):679-690
DOI: 10.1016/S1001-6058(16)60780-1
www.sciencedirect.com/
science/journal/l0016058

A GPU accelerated finite volume coastal ocean model *

Xu-dong Zhao (赵旭东) [1], Shu-xiu Liang (梁书秀) [1], Zhao-chen Sun (孙昭晨) [1], Xi-zeng Zhao (赵西增) [2], Jia-wen Sun (孙家文) [3], Zhong-bo Liu (刘忠波) [4]

1. *State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116023, China, E-mail: zhaoxvdong@dlut.edu.cn*
2. *Ocean College, Zhejiang University, Zhoushan 316021, China*
3. *National Marine Environmental Monitoring Center, State Oceanic Administration, Dalian 116023, China*
4. *Transportation Management College, Dalian Maritime University, Dalian 116026, China*

(Received July 6 2016, Revised November 17, 2016)

Abstract: With the unstructured grid, the Finite Volume Coastal Ocean Model (FVCOM) is converted from its original FORTRAN code to a Compute Unified Device Architecture (CUDA) C code, and optimized on the Graphic Processor Unit (GPU). The proposed GPU-FVCOM is tested against analytical solutions for two standard cases in a rectangular basin, a tide induced flow and a wind induced circulation. It is then applied to the Ningbo's coastal water area to simulate the tidal motion and analyze the flow field and the vertical tide velocity structure. The simulation results agree with the measured data quite well. The accelerated performance of the proposed 3-D model reaches 30 times of that of a single thread program, and the GPU-FVCOM implemented on a Tesla k20 device is faster than on a workstation with 20 CPU cores, which shows that the GPU-FVCOM is efficient for solving large scale sea area and high resolution engineering problems.

Key words: Graphic Processor Unit (GPU), 3-D ocean model, unstructured grid, finite volume coastal ocean model (FVCOM)

Introduction

The parallel computational schemes and the large scale, high resolution climate and ocean simulation models see a rapid development. The traditional CPU parallel algorithms for large scale ocean simulations are generally performed by the domain decomposition, where the domain is divided into many sub-domains with each sub-domain dealt by a different processor using distributed or shared memory computing. These parallel computational ocean models are mostly based on the Message Passing Interface (MPI) library, requiring High Performance Computing (HPC) clusters to realize the high computational capacity. Therefore, the high performance is difficult to achieve without access to HPC computers.

In recent years, microprocessors based on a single Central Processing Unit (CPU) have greatly enhanced the performance and reduced the cost in computer applications[1,2]. However, the increase of the CPU computing capacity is limited due to heating and transistor density limitations. However, the CPU is not the only processing unit in the computer system. The Graphic Processor Unit (GPU) is initially used for rendering images, but is also a highly parallel device[3]. The GPU's main task remains rendering video games, which is achieved by a fine gained parallelism for pixels rendered with a large number of microprocessors. Multiple threaded processors, and particularly, the GPUs, have enhanced the floating point performance since 2003[4]. NVIDIA provided a GPU computing Software Development Kit (SDK) in 2006 that extended the C programming language to use GPUs for general purpose computing. The ratio between multiple core GPUs and CPUs in the floating point calculation throughput is approximately 10:1 in 2009, i.e., the multiple core GPUs could reach 1 teraflops (1 000 gigaflops), while CPUs could only

* Project supported by the National Natural Science Foundation of China (Grant No. 51279028, 51479175), the Public Science and Technology Research Funds Projects of Ocean (Grant No. 201405025).

Biography: Xu-dong Zhao (1986-), Male, Ph. D. Candidate
Corresponding author: Shu-xiu Liang,
E-mail: sxliang@dlut.edu.cn

reach 100 gigaflops. Figure 1 shows that with the GPU Computing Accelerator, NVIDIA Tesla K40M, the double precision floating point performance has now reached 1.68 Tflops.

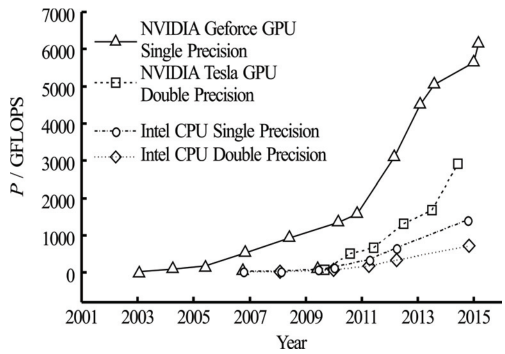

Fig.1(a) Floating point operation performance

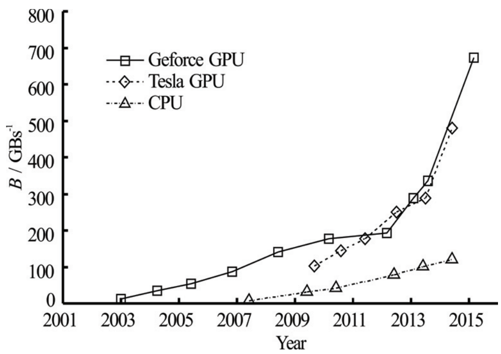

Fig.1(b) Memory bandwidth for the CPU and GPU

A number of programs and applications have been ported to the GPU, such as the lattice Boltzmann method[5], Ansys[6], and DHI MIKE. The use of the GPU devices was also explored for ocean and atmosphere predictions[7,8]. Michalakes and Vachharajani[9] proposed a Compute Unified Device Architecture (CUDA) C based weather prediction model. They achieved a twenty-fold speedup (by using NVIDIA 8800 GTX) compared to a single-thread Fortran program running on a 2.8 GHz Pentium CPU[9]. Most existing climate and ocean models are only accelerated for specific loops using the open accelerator application program interface (OpenACC-API) or CUDA Fortran, therefore these GPU accelerated models have achieved limited speedup. Horn implemented the GPU to a moist fully compressible atmospheric model [10]. Xu et al. [11] developed the Princeton Ocean Model POM.gpu v1.0, a full GPU solution based on MPI version of the Princeton Ocean Model (mpiPOM). POM.gpu v1.0 with four GPU devices can match the performance of the mpiPOM with 408 standard CPU cores. However, they were unable to resolve the complex irregular geometries of tidal creeks in an estuarine application[12]. Keller et al.[13] developed a GPU accelerated MIKE21 to solve 2-D hydrodynamics problems, and the latest DHI MIKE (2016) supports the GPU based 3-D hydrodynamic parallel computing.

The objective of this work is to reduce the computation time for the unstructured grid, Finite Volume Coastal Ocean Model (FVCOM) with a CUDA C parallel algorithm. The CUDA is a minimal extension of the C and C++ programming languages, and allows a program to be generated and executed on GPUs.

With the FVCOM parallel version, we demonstrate how to develop a GPU based ocean model (GPU-FVCOM) that runs efficiently on a professional GPU. The Fortran FVCOM is first converted to the CUDA C, and optimized for the GPU to further improve the performance.

In this work, the GPU-FVCOM performance is tested for two standard cases in rectangular basins with different grid numbers and devices, and then applied to the Ningbo coastal region. Particular attention is paid to the tidal current behavior, including the current velocity and the time history of sea levels.

1. Unstructured grid, finite volume coastal ocean model

The FVCOM is a prognostic, unstructured grid, finite volume, free surface, 3-D primitive equation coastal ocean circulation model jointly developed by UMASSD-WHOI. To solve the large scale sea area and high grid density problem, the METIS partitioning libraries are used to decompose the domain, and to implement the explicit Single Program Multiple Data (SPMD) parallelization method to the FVCOM[14]. The FVCOM has also been used in a wide range of engineering problems[15-19].

In the horizontal direction, the spatial derivatives are computed using the cell vertex and cell centroid (CV-CC) multigrid finite volume scheme. In the vertical direction, the FVCOM supports the terrain following sigma coordinates. The 3-D governing equations can be expressed as:

$$\frac{\partial \xi}{\partial t}+\frac{\partial Du}{\partial x}+\frac{\partial Dv}{\partial y}+\frac{\partial w}{\partial \sigma}=0 \tag{1}$$

$$\frac{\partial uD}{\partial t}+\frac{\partial u^2 D}{\partial x}+\frac{\partial uvD}{\partial y}+\frac{\partial uw}{\partial \sigma}-fvD=$$

$$-gD\frac{\partial \xi}{\partial x}-\frac{gD}{\rho_0}\left[\frac{\partial}{\partial x}(D\int_\sigma^0 \rho \mathrm{d}\sigma')+\sigma\rho\frac{\partial D}{\partial x}\right]+$$

$$\frac{1}{D}\frac{\partial}{\partial \sigma}\left(K_m\frac{\partial u}{\partial \sigma}\right)+DF_x \tag{2}$$