Journal of Computational Physics 231 (2012) 7932–7958

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Large-scale simulations on multiple Graphics Processing Units (GPUs) for the direct simulation Monte Carlo method

C.-C. Su [a], M.R. Smith [b], F.-A. Kuo [a,c], J.-S. Wu [a,c,*], C.-W. Hsieh [c], K.-C. Tseng [d]

[a] Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan
[b] Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan
[c] National Center for High-Performance Computing, National Applied Research Laboratories, Hsinchu, Taiwan
[d] National Space Organization, National Applied Research Laboratories, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 26 November 2011
Received in revised form 21 May 2012
Accepted 24 July 2012
Available online 8 August 2012

Keywords:
Rarefied gas dynamics
Parallel direct simulation Monte Carlo
Graphics Processing Unit (GPU)
MPI-CUDA
Very large-scale simulation

ABSTRACT

In this study, the application of the two-dimensional direct simulation Monte Carlo (DSMC) method using an MPI-CUDA parallelization paradigm on Graphics Processing Units (GPUs) clusters is presented. An all-device (i.e. GPU) computational approach is adopted where the entire computation is performed on the GPU device, leaving the CPU idle during all stages of the computation, including particle moving, indexing, particle collisions and state sampling. Communication between the GPU and host is only performed to enable multiple-GPU computation. Results show that the computational expense can be reduced by 15 and 185 times when using a single GPU and 16 GPUs respectively when compared to a single core of an Intel Xeon X5670 CPU. The demonstrated parallel efficiency is 75% when using 16 GPUs as compared to a single GPU for simulations using 30 million simulated particles. Finally, several very large-scale simulations in the near-continuum regime are employed to demonstrate the excellent capability of the current parallel DSMC method.

1. Introduction

The direct simulation Monte Carlo (DSMC) method [1] is a computational tool for simulating flows in which effects at the molecular scale become significant. The Boltzmann equation [2], which is appropriate for modeling these rarefied flows, is extremely difficult to solve numerically due to its high dimensionality and the complexity of the collision term. DSMC provides a particle based alternative for obtaining realistic numerical solutions. In DSMC, the movement and collision behavior of a large number of representative "simulation particles" within the flow field are decoupled over a time step which is a small fraction of the local mean collision time. The computational domain itself is divided into either a structured or unstructured grid of cells which are then used to select particles for collisions on a probabilistic basis and also are used for sampling the macroscopic flow properties. The method has been shown to provide a solution to the Boltzmann equation statistically when the number of simulated particles is large enough [3]. However, high computational cost has hindered further applications of DSMC to some practical problems, especially in the near-continuum (collision-dominated) regime, while the non-equilibrium effect may be important. Hence, it is important to increase the computational efficiency to extend the applicability of the DSMC method.

The general wisdom for accelerating the DMSC computation is to parallelize the code using the massage passing interface (MPI) protocol running on clusters with large numbers of processors [4–7]. Such implementations rely upon the multiple

* Corresponding author at: Department of Mechanical Engineering, National Chiao Tung University, Hsinchu, Taiwan. Tel.: +886 3 573 1693; fax: +886 3 611 0023.
E-mail address: chongsin@faculty.nctu.edu.tw (J.-S. Wu).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.07.038

instructions on multiple data (MIMD) parallelization philosophy. Recently, Graphics Processing Units (GPUs) have become an alternative platform for parallelization, employing a single instruction on multiple data sets (SIMD) parallelization philosophy. The resulting parallelization is much more efficient at the cost of flexibility – as a result, the computational time of several scientific computations, especially those which are optimally applied to vectorized computation strategies, have been demonstrated to reduce significantly, together with power and equipment costs. The DSMC method is essentially a highly local particle method; however, reports on DSMC using GPU computing were very limited. Especially, there is no related study in hybrid MPI-CUDA implementation for DSMC method. A parallel DSMC-accelerated in SIMD architecture had been presented by Gladkov et al. [8]. Although they obtained speedup of 65 times on a single GPU by simulating several 3D small-scale simulations (0.1–1 million particles), they have not demonstrated its capability by simulating some challenging large-scale real flow problems. In addition, several important features, such as particle removal on a single GPU and realistic boundary conditions (fully diffusive wall and inlet/outlet boundary conditions), were not included in the proposed algorithm. Each component of DSMC-accelerated in SIMD architecture and interaction between GPU computing and dimensionless parameters in gas dynamics were not discussed in detail. Recently, we had presented a parallel DSMC method in SIMD on a single GPU [9]. We demonstrated a modest speedup of 3–10 times in the simulation of supersonic flow over a flat plate and a supersonic lid-cavity flow problem with 1–10 million particles caused by a low speedup due to smaller problems and worse GPU machines.

Graphics Processing Units (GPUs) are co-processors originally designed to assist in the computations required for displaying graphics on monitor. In recent years, the GPU has been demonstrated as an effective tool in the computation of scientific and engineering problems. In the past, researchers performing General Purpose computing using Graphics Processing Units (GPGPU) [10] were forced to use graphics-related API's (like OpenGL, and later OpenCL) until CUDA (Compute Unified Device Architecture) [11] was introduced in 2007 by NVIDIA Corp. NVIDIA's CUDA is a general-purpose parallel computing architecture with a new parallel programming model and instruction set architecture. Nowadays, we can employ GPU co-processors to accelerate our computation using CUDA. And also CUDA has been developed to work in many popular computing languages such as C/C++, FORTRAN and even Java.

Thus, in this paper we intend to study the parallel performance of the DSMC method using a hybrid MPI-CUDA parallelization paradigm and demonstrate its capability in simulating several large-scale near-continuum gas flow problems.

This paper is organized as follows. The governing equation and the standard DSMC method is first introduced. Then, the DSMC method using hybrid MPI-CUDA parallelization is described in detail. The speedup and parallel performance on multiple GPUs using CUDA is investigated through subsonic/supersonic lid-driven cavity benchmarks. Following this, we apply the parallel DMSC implementation to several very large-scale simulations in near-continuum flows. Finally, the paper is summarized with some important findings.

2. Numerical method

2.1. The Boltzmann Equation

The Boltzmann equation [2] is employed to describe molecular transport phenomena based on gas-kinetics theory and statistical mechanics. It is an integral-differential equation which is valid for all flow regimes, which ranges from highly rarefied to continuum flows. The Boltzmann equation based on the assumption of binary collision is written as

$$\frac{\partial(nf)}{\partial t}+\mathbf{c}\cdot\frac{\partial(nf)}{\partial \mathbf{r}}+\mathbf{F}\cdot\frac{\partial(nf)}{\partial \mathbf{c}}=\int_{-\infty}^{\infty}\int_{0}^{4\pi}n^2(f^*f_1^*-ff_1)c_r\sigma d\Omega d\mathbf{c}_1 \tag{1}$$

where f and n are the velocity distribution function and number density, respectively. $\mathbf{c}$ and c_r are the molecular velocity and the relative molecular speed between two particles, respectively. And $\mathbf{F}$ is an external force, t is the time, σ is the collision cross section, and $d\Omega$ is the solid angle. f and f_1 denote two different types of molecules, and the superscript $*$ denotes the post-collision quantities.

In general, it is very difficult to solve the Boltzmann equation directly using conventional numerical method because the difficulty of correctly and efficiently modeling the integral collision term (right-hand term), in addition to the high numbers of dimensionality of the Boltzmann equation (configuration and velocity spaces). Instead, the DSMC method, developed by Bird during 1960s [1], has been widely used to solve the Boltzmann equation when the flow is rarefied. Until recently, the DSMC method was shown statistically to equivalent to solving the Boltzmann equation as the number of particles is large [3].

2.2. The standard DSMC method

The DSMC method is a particle-based method for solving the Boltzmann equation and it is widely used to simulate gas flows in the rarefied gas regime. The central idea of DSMC is to reproduce real flow properties with no more than collision mechanics of gas molecules through a large number of pseudo particles that are used to represent real gas molecules. Each pseudo particle represents a fairly large number of real gas molecules. In the DSMC simulation, there is an important feature that the motions and the collisions of pseudo particles are uncoupled over a time interval. The time interval (timestep) should be kept much smaller than the mean collision time.