Computers & Fluids 110 (2015) 152–158

Contents lists available at ScienceDirect

Computers & Fluids

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

Large-scale parallelization based on CPU and GPU cluster for cosmological fluid simulations

Chen Meng [a,b], Long Wang [a,*], Zongyan Cao [c,a], Long-long Feng [d], Weishan Zhu [d]

[a] Supercomputing Center of Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China
[c] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China
[d] Purple Mountain Observatory, Chinese Academy of Sciences, Nanjing 210008, China

A R T I C L E I N F O

Article history:
Received 9 December 2013
Accepted 2 April 2014
Available online 21 April 2014

Keywords:
Cosmological hydrodynamics
WENO
GPU
Hierarchical memory
Heterogeneous
Large-scale

A B S T R A C T

We present our parallel implementation for large-scale cosmological simulations of 3D supersonic fluids based on CPU and GPU clusters. Our developments are based on a CPU code named WIGEON. It is shown that, compared to the original sequential Fortran code, a speedup of 19–31 (depending on the specific GPU card) can be achieved on single GPU. Furthermore, our results show that the pure MPI parallelization scales very well up to 10 thousand CPU cores. In addition, a hybrid CPU/GPU parallelization scheme is introduced and a detailed analysis of the speedup and the scaling on the different number of CPU/GPU units are presented (up to 256 GPU cards due to computing resource limitation). Our high scalability and speedup rely on the domain decomposition approach, optimization of the algorithm and a series of techniques to optimize the CUDA implementation, especially in the memory access pattern on GPU. We believe this hybrid MPI + CUDA code can be an excellent candidate for 10 Peta-scale computing and beyond.

1. Introduction

The advances in performance and architecture of supercomputers enable scientists and engineers to design more effective solutions for some challenging problems, especially the direct numerical simulations (DNS) in computational fluids dynamics. Here, we develop a software for the simulation of supersonic fluids derived by the self-gravity of dark matters in cosmological simulation. The observed luminous objects in the universe have existed in the form of baryonic matter, a typical Navier–Stokes fluid. To account for the observational features, it would be necessary to incorporate a variety of computational hydrodynamics algorithms into cosmological simulations. WIGEON [1] is a cosmological hydrodynamic/N-body simulation software, in which N-body simulation is used to involve the collisionless dark matter for computing self-gravity in the cosmological code. As a result of the high nonlinearity of gravitational clustering in the universe, there is extremely supersonic motion around the density peak, which is more challenging than the typical hydrodynamic simulation without self-gravity. So WIGEON uses high-order finite difference weighted essentially non-oscillatory (WENO) scheme to solve the computational fluid dynamic equations. WENO has been widely used in applications for high-resolution supersonic flow simulations, typically the cosmological hydrodynamics. The core of WENO scheme is the idea of adaptive stencils in the reconstruction procedure based on the local smoothness of the numerical solution to automatically achieve high-order accuracy and non-oscillatory property near discontinuities. It is extremely robust and stable for solutions containing strong shock and complex solution structures [2,3]. Here, we chose a uniform resolution over an Adaptive Mesh Refinement (AMR) for the discretization of the flow field. Because it is difficult to design a robust AMR-WENO scheme that is both conservative and 5th order (higher than second order), due to the mass inconsistency of coarse and fine grid solutions at the initial stage in a finite difference scheme.

Though a significant advantage of WENO is its ability to have high accuracy on coarser meshes, the meshes should be very large when we try to get a cosmological simulation to achieve acceptable resolution. Large-scale fluid computations based on CPU cluster have gained great success in the Tera-scale era [4,5]. From the list

* Corresponding author. Address: Supercomputing Center of Computer Network Information Center, Chinese Academy of Sciences, No. 4 South 4th Street, ZhongGuanCun, Beijing 100190, China. Tel.: +86 10 18611845735.

E-mail addresses: mengchen@sccas.cn (C. Meng), wangl@sccas.cn (L. Wang), zycao@sccas.cn (Z. Cao), fengll2000@gmail.com (L.-l. Feng), wszhu1985@gmail.com (W. Zhu).

http://dx.doi.org/10.1016/j.compfluid.2014.04.006
0045-7930/

of the world's Top500 fastest supercomputers [6], the Peta-scale era has arrived with heterogeneous systems involving CPU cores and accelerators or co-processors (GPU, MIC, etc.). And GPU has gained significant performance for computational intensive tasks in recent years [7,8], so designing hybrid codes is increasingly important for applications, which can utilize the computing power of both CPU clusters and GPU clusters.

In this paper, we present a heterogeneous parallel code for the large-scale 3D cosmological fluid dynamics in double precision on CPU/GPU cluster based on MPI and CUDA. We started our work with a massively CPU parallelization based on MPI. WIGEON is solving hydrodynamic problems on a uniform structured grid. The MPI-based parallelization uses domain decomposition assigning subdomains to different CPU cores, which are, then, calculated in parallel. However, several issues have to be taken into account: (1) The most efficient domain decomposition strategy for the best scalability is to be found. The 5th order WENO at least requires a five-point stencil for one point in each dimension and fifteen points in total in three-dimensional problems. When we decompose the computing domain, the nonlocal data of the stencil points are called as "ghost cells". The Euler equation solver solves a flow system with multi-iterations, each of which begins with a data exchange of ghost cells. That leads to a high amount of communication, since the points are a five-component vector and the ghost-cell layers are relatively thick. (2) Improving the efficiency of the collective communication is critical to extreme-scale parallel computing. The global communication in Lax-Friedrichs flux splitting will result in relatively bigger performance loss. So in this paper, we focused on designing an algorithm to extend the parallel scale to more than ten thousand CPU cores.

We can see that the traditional MPI-based parallel scheme can achieve very good scalability. However, there is always a ceiling because of the decreasing computation–communication ratio with the refinement of parallel granularity. Then on condition that the subdomain is big enough, we decided to make computations in one subdomain go through a second-level parallelization based on CUDA running on a shared memory system. Each simulation step involves three main procedures: RHS, DT, UPDATE. The DT procedure computes a time step from a global data reduction of the velocity. The UPDATE procedure updates the flow quantities, and it involves N-body based dark matter computations in WIGEON. RHS procedure computes the evaluation of Right Hand Side of the governing equations. We ported RHS procedure (about 88% of total time) involving WENO procedure in the subdomain to local GPU for further speedup and scaling [9–12]. In order to achieve a high speedup on the GPU, the following problems need to be solved: (1) WENO is not the type of procedure with high operational intensities (FLOP/Byte) derived from the adaptive-stencil calculation. However, memory access is the bottleneck compared with the computing power for GPUs. Ideally, we need many register resources for intermediate results, which is a very effective but limited on GPU; (2) GPUs have their own "cache locality". The cache locality on CPU does not work on GPU and even produces negative effects. In other words, WENO computation is a memory-bound, which seems not to be suitable for GPUs. So, we did a series of optimizations including making use of all levels of GPU on-chip memory, adjusting the data structure and the order of the instructions to make full use of GPU power.

This article is organized as follows. In Section 2, we introduce the algorithm of solving Euler equation based on the WENO scheme. Section 3 we outline the implementation and optimization details of MPI-parallelization on CPU cluster. In Section 4, we outline the implementation and optimization steps of GPU code on CPU/GPU cluster. In Section 5, we measure and analyze the performance of our implementation and in Section 6 the results are summarized and a short outlook is given.

2. Numerical algorithm

2.1. Governing equation

The observed luminous objects in the universe have existed in the form of baryonic matter. So we dictated them by the compressible fluid without any viscous and thermal conductivity terms. We use Euler equation for the cosmological fluid as the governing equation. It can be written in the compact form of hyperbolic conservation laws:

$$\frac{\partial U}{\partial t}+\frac{\partial f(U)}{\partial X}+\frac{\partial g(U)}{\partial Y}+\frac{\partial h(U)}{\partial Z}=F(t,U) \tag{1}$$

where the conserved quantity U and the fluxes $f(U)$, $g(U)$, and $h(U)$ are five-component column vectors:

$$U=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{pmatrix} f(U)=\begin{pmatrix}\rho u\\ \rho u^2+P\\ \rho uv\\ \rho uw\\ u(E+P)\end{pmatrix} g(U)=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+P\\ \rho vw\\ v(E+P)\end{pmatrix} h(U)=\begin{pmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+P\\ w(E+P)\end{pmatrix} \tag{2}$$

t is the time and (X, Y, Z) are the coordinates. ρ is the density, $V = (u, v, w)$ is the velocity vector, P is comoving pressure, and E is the total energy including kinetic and internal energies:

$$E=\frac{P}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2+w^2) \tag{3}$$

γ is the ratio of the specific heats of the baryon. Here $\gamma = 5/3$. The "force" term $F(t, U)$ on the right-hand side includes the contributions from the gravitation.

2.2. WENO scheme algorithm

As a result of the high nonlinearity of gravitational clustering in the universe, there can occur shock waves in the cosmological flows. So the discretization of the fluxes for solving the governing equation is based on the 5th order finite difference WENO [13,14]. As an example, $\frac{\partial f(u)}{\partial x}$ is discussed, keeping the values for Y and Z constant:

$$\frac{\partial f(u)}{\partial x}\Big|x=x_j\approx\frac{1}{\Delta x}(\hat{f}_{j+1/2}-\hat{f}_{j-1/2}) \tag{4}$$

here, $\hat{f}_{j+1/2}$ is the numerical flux; If $f'(u)\geqslant 0$, the 5th order finite difference WENO scheme has the flux given by:

$$\hat{f}_{j+1/2}=w_1\hat{f}^{(1)}_{j+1/2}+w_2\hat{f}^{(2)}_{j+1/2}+w_3\hat{f}^{(3)}_{j+1/2} \tag{5}$$

where $\hat{f}^{(i)}_{j+1/2}$ are fluxes on three different stencils given by:

$$\hat{f}^{(1)}_{j+1/2}=\frac{1}{3}f(u_{j-2})-\frac{7}{6}f(u_{j-1})+\frac{11}{6}f(u_j), \tag{6}$$

$$\hat{f}^{(2)}_{j+1/2}=\frac{1}{6}f(u_{j-1})+\frac{5}{6}f(u_j)+\frac{1}{3}f(u_{j+1}), \tag{7}$$

$$\hat{f}^{(3)}_{j+1/2}=\frac{1}{3}f(u_j)+\frac{5}{6}f(u_{j+1})-\frac{1}{6}f(u_{j+2}), \tag{8}$$

The key for the success of WENO scheme relies on the design of the nonlinear weights w_i, which are given by:

$$w_i=\frac{\tilde{w}_i}{\sum_{k=1}^{3}\tilde{w}_k},\quad \tilde{w}_k=\frac{\gamma_k}{(\varepsilon+\beta_k)^2} \tag{9}$$

where the linear weights γ_k are chosen to yield 5th order accuracy and are given by $\gamma_1=\frac{1}{10},\ \gamma_2=\frac{3}{5},\ \gamma_3=\frac{3}{10}$. The smoothness indicators β_k are given by: