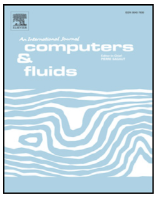
Computers and Fluids 167 (2018) 180–186

Contents lists available at ScienceDirect

Computers and Fluids

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



Parallelizing and optimizing a detonation combustion simulation application on heterogeneous platform

Yonggang Che [a,b,*], Chuanfu Xu [a,b], Tongtong Wang [b]

[a] *State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, China*
[b] *College of Computer, National University of Defense Technology, Changsha, China*

ARTICLE INFO

Article history:
Received 18 January 2017
Revised 7 February 2018
Accepted 1 March 2018
Available online 5 March 2018

Keywords:
Combustion simulation
Detonation
Heterogeneous computing
OpenMP
Offload
Vectorization

ABSTRACT

Initition3d is a detonation combustion simulation application developed based on the AMROC (Adaptive Mesh Refinement Object-oriented C++) framework, originally parallelized in pure MPI and runs only on CPUs. To port this application to the heterogeneous platform, we add OpenMP parallelization to the code, implement the heterogeneous algorithm using the OpenMP 4.0 programming interface, and perform a series of performance optimizations to optimize the parallelism and to reduce the overhead. We also solve some issues related to the framework based application development. Performance evaluation shows that the application exhibits good OpenMP parallel efficiency, and the hybrid MPI/OpenMP version is at best 1.79 × faster than the original pure MPI version on one server node. The heterogeneous version successfully utilized the MIC (Many Integrated Core) coprocessors to accelerate the performance by a factor of 2.69–2.92 on the Tianhe-2 supercomputer. Good parallel scalability across multiple nodes is also achieved.

1. Introduction

1.1. The Initition3d application

Scramjets are one of the first choices for hypersonic air-breathing propulsion systems due to their superior performance. Scramjet combustors adopt the Brayton circle design whose thermodynamic efficiency (27%) is far below than that (49%) in the detonation combustion [1]. In principle, detonation based engine will be a promising one of the future propulsion systems with supersonic cruise velocity. In the detonation research, detonation initiation has been one of the key problems, especially in the supersonic combustible mixtures. Reliable initiation methods are one of key issues in detonation investigations. Numerical simulation methods provide a viable approach to study the hot jet initiation process. For detonation combustion calculations, chemical reaction generally introduces additional temporal and spatial scales. The reactive case generally requires finer meshes than studies of non-reactive Euler equations alone. However, only a small area near the detonation front, where severe chemical reactions occur, needs a very fine mesh. Other areas with relatively mild flow behavior can be resolved with coarser grid. This kind of issue can be addressed by structured adaptive mesh refinement (SAMR) framework. In [2], Ralf Deiterding presented the algorithmic and mathematical details of a SAMR implementation for hyperbolic problems on distributed memory computers.

Initition3d [3,4] is a three-dimensional detonation combustion simulation application. It simulates the detonation combustion initiated with a hot jet in supersonic H2-O2-Ar mixtures to investigate reliable detonation initiation methods. Three dimensional inviscid reactive Euler equations are used as the governing equations with a mixture of different thermally perfect species. For convective flux discretization, a second-order accurate MUSCL-TVD is adopted using finite volume method. The numerical flux calculation and the reconstruction are used to solve the hydrodynamic process. A hybrid Roe-HLL Riemann solver is adopted to construct inter-cell numerical upwind fluxes, while the Van Albada limiter with MUSCL reconstruction is applied to construct a second-order method in space. As for the time integration, second-order MUSCL-Hancock technique is used. A series of cases of detonation initiation and propagation using a hot jet in supersonic H2-O2-Ar mixtures have been investigated using Initition3d.

Initition3d uses the AMROC (Adaptive Mesh Refinement Object-oriented C++) framework [5,6] to implement the parallel SAMR method. AMROC is a structured adaptive mesh refinement framework based on the DAGH (Distributive Adaptive Grid Hierarchies). AMROC defines three main abstraction levels. At the top level, the specific application is formulated. AMROC relies on standardized interface-objects to application specific components like initial and boundary conditions or numerical integration routines. Defining a

[*] Corresponding author.
E-mail address: ygche@nudt.edu.cn (Y. Che).

https://doi.org/10.1016/j.compfluid.2018.03.018
0045-7930/© 2018 Elsevier Ltd. All rights reserved.

new AMROC application does not require any knowledge of AMR. The second level, called the AMROC level, consists of the mere AMR-solver and its components for grid generation, error estimation, interpolation and flux correction. The base level is divided into elementary functionality for single grid patches and the implementation of various lists that store these patches hierarchically. The design of the base level involves a Box-class that specifies a single rectangular box in global integer index space. AMROC utilizes methods of the base level defined by the DAGH package, which supplies the necessary hierarchical data structures. As for the parallelization method, AMROC implements an effective parallelization strategy for distributed memory machines by employing the MPI library and the domain decomposition scheme [6].

1.2. Our work and contribution

It is well-known that combustion simulation is one of the grand challenge areas of High Performance Computing (HPC) [7]. Efficient utilization of current high performance computing architectures is crucial [8]. In this paper, we introduce our efforts parallelizing and optimizing Initition3d, targeting the heterogeneous parallel architecture consists of general-purpose CPUs and Intel MIC (Many Integrated Core) coprocessors. Such heterogeneous architecture is used in some most powerful supercomputers, including the Tianhe-2 (Milkyway-2) supercomputer [9]. To effectively utilize the computational resources of such heterogeneous platforms, the OpenMP thread-level parallelism, the collaboration and load balancing between the CPUs and the MIC coprocessors, are addressed. As the Initition3d code is developed based on the AMROC framework, we solve a series of issues that do not exist for non-framework based applications. We evaluate the performance on one server node and the Tianhe-2 supercomputer. The results show that our hybrid MPI/OpenMP version significantly outperforms the baseline MPI version, and the heterogeneous version also significantly improves the performance over the hybrid MPI/OpenMP version.

Our major contributions are as follows:

- We solve a series of issues associated with the framework based structure of Initition3d code and successfully implement the OpenMP parallelization and OpenMP 4.0 [10] based heterogeneous algorithm to the application.
- We apply a series of novel optimization techniques to Initition3d code, which enable utilizing the architecture features of the CPU and the MIC coprocessor at all level.
- We evaluate the performance on the Tianhe-2 heterogeneous platform and demonstrate large-scale heterogeneous computing capabilities for detonation combustion problems.

The rest of this paper is organized as follows: Section 2 presents a brief review of related work. Section 3 presents our methodologies parallelizing and optimizing the Initition3d application. Section 4 presents the performance evaluation results. Section 5 presents our conclusion and future plan.

2. Related work

Parallel combustion simulation are extensively studied in the literature. However, most of the previous work are performed on the traditional CPU based parallel platforms and only a few work are done on the heterogeneous parallel platforms.

Sankaran et al. ported S3D, a combustion simulation application, to the GPUs (Graphics Processing Units) using OpenACC [11]. They used a hierarchical parallelism method with three levels of parallelism, MPI between nodes, OpenMP on the node and vectorizable loops. They achieved 1.2 × speedup on Fermi-XK6 GPU vs CPU. The chemistry kernels ported to GPU runs on a Fermi GPU is 2 × −3 × faster than the CPU code on dual 6-core Opteron CPUs. In [12], Jacqueline Chen et al. restructured S3D to expose node-level parallelism, which resulted in hybrid MPI + OpenMP code and MPI + OpenACC code and achieved 6-fold performance improvement on Titan over Jaguar.

He et al. developed a high-order-accurate solver for the radiative transfer equation which uses the Discontinuous Galerkin method and implemented it for GPUs [13]. This code demonstrated good speed-up with increasing number of processors and the speed-up efficiency was around 95% in simulations of the Atlas II rocket plume. The GPU computations (each node has 2 NVIDIA Tesla cards) demonstrated a consistent speed-up by a factor of 3 over the CPU computations (each node has 2 Intel Xeon E5620 CPUs).

Hiremath et al. presented a combined dimension reduction and tabulation strategy for implementing chemistry in large scale parallel Large-Eddy Simulation (LES)/Probability Density Function (PDF) computations of turbulent reacting flows [14]. They performed full-scale simulations of the Sandia Flame D turbulent jet flame on up to 9216 cores, showing that the P-URAN parallel strategy achieves over 85% relative weak scaling efficiency and around 60% relative strong scaling efficiency on up to 9216 cores. The combination of ISATIRCCE and P-URAN algorithm enabled them to perform accurate and computationally-efficient large-scale LESIPDF simulations with real fuel chemistry involving hundreds of chemical species.

In [15], we ported LESAP (Large Eddy Simulation for Airbreathing Propulsion) [16], a numerical supersonic combustion simulator of the scramjet engine, to the heterogeneous platform consists of CPUs and MIC coprocessors. Our efforts included OpenMP threading, single thread performance optimization, and CPU / MIC collaborative computing. We evaluated the resulting codes for the case of turbulent combustion simulation in a supersonic combustor on the Tianhe-2 supercomputer. The results showed that the optimized CPU-only version significantly outperformed the baseline version, and CPU + MIC heterogeneous computing significantly accelerated the performance. The code exhibited excellent parallel scalability to scale to 5120 nodes.

However, porting Initition3d to the heterogeneous platform faces with some additional problems. As Initition3d is developed on based on the AMROC framework, most of the computational work is done with the framework code. The application level code is only in charge of defining the problem to be solved, selecting the solution schemes, and setting the initial conditions and the boundary conditions. Hence, the parallel algorithms and performance optimization must take into account of such code structure characteristics.

3. Methodology

3.1. Overall methodology

While most of the computational work is done with the AMROC framework code, parallelizing and optimizing the whole framework is infeasible as the code base is large. Further, parallelizing the Initition3d application at the framework code level may not be performance effective, as a coarse-grained parallelism cannot be achieved. Instead, we only consider the application level code and the framework level code that are used by the Initition3d application. We first isolate the source files used by the Initition3d application. Then we perform our parallelization and optimization on these files. Finally, we recompile the AMROC framework and the Initition3d code to get the parallelized and optimized executables.

Our efforts include three steps: (1) Adding OpenMP parallelization to the code to utilize the large numbers of cores on the CPUs and the coprocessors. This enables hybrid MPI/OpenMP parallel computing and is also an essential step for the heterogeneous