Computers & Fluids 110 (2015) 19–26

Contents lists available at ScienceDirect

Computers & Fluids

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

Performance analysis of parallel algorithms in physics simulation for molecular dynamics simulation liquid metals solidification processes

Kenli Li [a,c], Dapu Li [a,*], Jie Liang [a], Yu Ye [a], Yingqiang Liao [a], Rangsu Liu [b], Yunfei Mo [a]

[a] School of Information Science and Engineering, Hunan University, Changsha 410082, China
[b] School of Physics and Micro Electronic, Hunan University, Changsha 410082, China
[c] National Supercomputing Center in Changsha, 410082, China

ARTICLE INFO

Article history:
Received 19 November 2013
Received in revised form 8 August 2014
Accepted 16 December 2014
Available online 26 December 2014

Keywords:
Liquid metals solidification processes
Molecular dynamics
Computer simulation
Parallel algorithm
OpenMP + MPI model

ABSTRACT

In this work, a parallel arithmetic program for the MD (molecular dynamics) simulation study of a large-sized system is proposed and reformed, based on the previous program with PVM (Parallel Virtual Machine) model applied in a small-sized system for liquid metals. Our methodology is combined with the MPI (message passing interface) and OpenMP (Open Multiple Processing) programming model using the spatial domain decomposition method, conquering the problems occurred in PVM model and enlarging the scale of simulated system. Comparing with the previous small-sized system consisting of 50,000–100,000 atoms, the large-sized system is consist of 5,000,000–10,000,000 atoms and the simulation results are more closely to the real situation of the simulated system. In this paper, the performance of parallel program using MPI + OpenMP model is analyzed, showing better speedup, parallel efficiency, and scalability. Finally, we adopt many physical evaluation methods to verify the validity of the simulation results, including pair distribution function, bond-type index analysis, atomic clusters analysis and visualizing analysis. From these physical results, it is clear that the simulation results are in good agreement with the experimental results, which confirm the correctness of the program in simulation.

1. Introduction

MD simulation has become a new research method for atomic-level theoretical analysis and experimental research in various fields, such as chemistry, materials science, physics and engineering after that Alder and Wainwright first proposed the molecular dynamics (MD) method for ideal hard ball systems in 1957 [1]. Moreover, the simulation theory, method and application have rapidly developed [2]. Given that the serial program for MD calculation is at low speeds, only a small number of atoms are involved in the simulation and the result is significantly different from the actual scenarios, especially in our study of liquid metal solidification processes simulations. However, the parallelization of MD simulations is an effective method for large-scale computation [3–7], and the supercomputer is the excellent support condition for parallel computing. Thus, the parallel algorithm for MD simulations is very important to realize the large-scale, complicated system simulations.

For our study of liquid metal cooling processes, a lot of related work has been made in the past [8–13]. In the whole simulations, MD method is adopted to observe the transient change of their microstructures at the electronic and atomic level. Given the initial position (coordinate r) and speed (v) of each atom in the systems under consideration, the interacting potential, the forces acting on each atom and the new position and speed of each atom can be calculated in sequence using Newton's classical equations of motion. During the whole calculation processes, the computing of forces is one of the most expensive parts. We had developed the initial cascade program of MD simulation by a PVM parallel program to enlarge the number of atoms in the simulating system from 5000 to 50,000 atoms [14]. At present, for enlarging the amount of the simulating system up to 5,000,000–10,000,000 atoms, the MPI + OpenMP parallel computating model is adopted based on our existing parallel program of PVM and research on parallel algorithms in other fields [15–22]. Under this circumstance, the simulation results will be closer to the realities of simulation system. In the MPI + OpenMP parallel computing architecture, we used MPI [23] to implement the parallel computing and communication for CPU heterogeneous computing platforms and the OpenMP [24] to implement the parallel implementation for multi-core platforms in MD simulations. The

* Corresponding author.
E-mail addresses: lkl@hnu.edu.cn (K. Li), lidapu000@163.com (D. Li), liangjie1988@hnu.edu.cn (J. Liang), yeyu.hnu@gmail.com (Y. Ye), lyq698828@hnu.edu.cn (Y. Liao).

http://dx.doi.org/10.1016/j.compfluid.2014.12.016
0045-7930/© 2014 Elsevier Ltd. All rights reserved.

decomposition method is based on spatial domain decomposition method.

The paper is organized as follows. In Section 3 the performance of the PVM algorithm is analyzed. The MD parallel model with MPI + OpenMP is described in Section 4, and the performance results for our implementations are evaluated in Section 5. In Section 6, the simulation results analysis with physic evaluation methods are presented. Finally, Section 7 concludes the paper.

2. Molecular dynamics simulation

Molecular Dynamics method is an important one to simulate the solidification processes of liquid metals. Given the initial position (coordinate r) and speed (v) of each atom in the system under consideration, the interacting potential between atoms are known [25,26], and then the forces acting on each atom can also be calculated, and the new position and speed can be obtained. Thus, we can simulate and get the dynamic situations and the instantaneous microstructures of the simulation system through circular computations step by step. From these results, the macroscopic thermodynamics properties of the system can be clearly acquired.

In MD simulation, we adopted the VERLET [27] algorithm, the VERLET algorithm is classical method to solve the motion equations of simulation systems, which is rather maturely and widely used. On the basis of VERLET algorithm, given that the coordinates and speed of each atom at a certain time (t), at time (t + dt), the displacement is displayed by Eq. (1):

$$r(t+dt) = 2r(t) - r(t-dt) + dt^2 r(t) \tag{1}$$

where the speed is not necessary to be calculated, but if we want to estimate the kinetic energy, the speed v should be got by Eq. (2):

$$v(t) = (r(t+dt) - r(t-dt))/2dt \tag{2}$$

In Eqs. (1) and (2), the calculation of the acceleration of each atom is equivalent to the calculation of the force f acting on the atom. We adopt the pair potential approximation to calculate the force f. If we limit our attention to the cases of pair interactions, the force on atom i can be calculated by the following Eq. (3).

$$\overrightarrow{F_j} = \sum_{i,j\neq i} \overrightarrow{f_{ij}} \tag{3}$$

3. The PVM parallel algorithm

In the past, the PVM program has been run on the YH23M supercomputer designed and produced by China. The PVM algorithm adopted the VERLET [28] algorithm. The VERLET algorithm is classic method to solve the classical motion equations of particle systems by computer. We adopts Master–Slave parallel model, and it is easy to maintain and adjust the number of Slave machines (see [14]). From the Fig. 1, it is clear that the Master run on the front workstation of YH23M and the Slave will be given to the processing machines of YH23M. The algorithm has parallel parts and has better solved the memory problem.

All the atoms in the simulation system are divided into each Slave machines in average, including coordinates, neighbor list of atoms. When the main program calculates the forces, the current coordinates of atoms will be broadcasted to every Slave machines. Then each Slave machine calculates the forces acting on each atom. When each Slave node finish it's computation task, they will send the results back to Master node, and the next step will be continued. In this way, the calculation of forces can be paralleled and the speed can be greatly accelerated. The simulation results would be more closely to the real situation of the system comparing with the previous small-sized simulation system. During the calculation

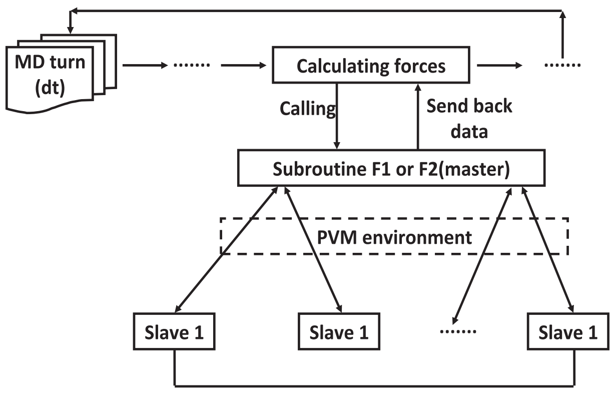

Fig. 1. Schematic diagram of PVM parallel algorithm.

processes, the gigantic neighbor list array has been divided into each Slave machine, thus this array is handled by each Slave but not by Master.

However, the MD simulation system using PVM architecture can only involve 5000–100,000 atoms of liquid metals, and when the number of atoms in simulation system is up to 100,000, the performance of PVM program is unsatisfactory. In the previous program, only the parallel parts were involved in the calculation with PVM, meanwhile, communication among processors using messages passing MPI interface occupy a certain amount of calculation, resulting in the decrease of computational efficiency. As is well known that the PVM has high performance in dynamic, fault tolerance and scalability, but not in communication. Moreover, the larger clusters of physical system can not be acquired and observed in the simulating system consisting of 5000–100,000 atoms with PVM model. So we change this existing parallel program of PVM to parallel program of MPI + OpenMP model in the following.

4. Design and realization of parallel algorithm

This new algorithm adopts the MPI + OpenMP programming model. The nodes are parallelized by MPI, whereas parallelization within each node is realized using OpenMP. Because spatial domain decomposition method has better scalability, and been widely used in some related software (for example LAMMPS, NAMD and so on), our algorithm is based on spatial domain decomposition method. The entire domain is divided into P sub-domain; each sub-domain has correspondence to a computing node. This assignment method can achieve a good load balancing. Once each node is computing force, we will send neighbor list and related data to each node. If the atoms run out from the processor, we will update the corresponding neighbor list and related data through the MPI_Send and MPI_Recv, and are ready to calculate the next step. Under this model, the storage requirements will ascend, but this method can reduce time of communication and computing, and computational efficiency has been improved by fully utilizing the heterogeneity of the multi-core system among nodes and the shared memory of the multi-core system in a single node.

In order to make each sub-task of balancing work, a dynamic load balancing method [33] is adopted to adjust the workload across processes. At each cycle of the MD simulation, the tasks running on different processors should accomplish their part of the job synchronously. This makes it possible to take full advantage of the parallel algorithm and to have the least possible idle time. To achieve this goal in the computational side the workload should be divided as equally as possible among the processors. In