Sensors & Transducers, Vol. 159, Issue 11, November 2013, pp. 242-248

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

Parallel Computing of Polymer Chains Based on Monte Carlo Method

[1, 2] Hong Li, [1] Bin Gong, [2] Chang-Ji Qian, [*] He-Bei Gao

[1] School of Computer Science and Technology, Shandong University, Jinan, 250100, China
[2] Department of Physics, Wenzhou University, Wenzhou, 325035, China
[*] Department of Computer Science, Wenzhou Vocational and Technical College, Wenzhou, 325035, China
[*]E-mail: bogolyx@163.com

Received: 27 August 2013 /Accepted: 25 October 2013 /Published: 30 November 2013

Abstract: Computer simulation of polymer chain is carried out by Monte Carlo method in this paper. The properties of polymer chain are simulated by self avoiding walk and bond fluctuation model. The parallel computing model is established to solve the physical problems of the polymer chains. We analyze the algorithms for polymer chains to parallel computing at high performance computing center. Gustafson law is analyzed and is applied to obtain linear speedup. *Copyright © 2013 IFSA.*

Keywords: Computer simulation, Parallel computing, Polymer chain, Monte Carlo method, Speedup.

1. Introduction

Recently polymers have attracted much interest in the areas of sensors [1-4]. Polymeric sensor consists of a sensor layer with chemical or physical interactions to the analyte and transducer. Both conducting polymers and non-conducting polymers have tremendous technological potential for the development of sensors. There are many unresolved mechanisms in the experiment. Computational Science is a multidisciplinary interdisciplinary science, including high-performance computing, computer networks, physics, chemistry, mathematics, etc. Different application software was developed to solve computational problems of physics, chemistry, molecular dynamics, sensors design and other fields. High-performance computing is a research tool for modern science and technology together with the traditional theory and laboratory experiments, and it is one factor of key technologies to promote the major scientific discoveries and technological developments. How to effectively develop a variety of high-performance computing software is becoming a hot issue of the application of computational science.

Polymer is everywhere, and all living can be regarded as a collection of polymer. To study the adsorption between polymer and surface has great significance [5-6]. Computer simulation is particularly suitable for using to study the properties of a block copolymer with complex structure due to characteristics of polymer chain topology and sequence distribution. We carry a variety of computer simulation work for properties of polymer, predict the nature of the polymer conformation and provide evidence basis as experimental and theoretical studies. Parallel computing technology is need in these areas of numerical calculations. It is very necessary and meaningful for self-developed high-performance computing software to study polymeric nature.

Article number P_1566

Polymer has been studied by a lot of parallel computing, such as MPI, OpenMP and GPU and other applications [7-10]. There are still many researchers whose program is computed to use only one single CPU alone in the field of molecular, lacking high performance computing applications in these areas according to the specific areas. The reason, why parallel computing is necessary, is mainly due to that the performance of current single-processor cannot meet the needs of large-scale scientific computing for polymer. Parallel computing is currently the only way to meet the actual needs of large-scale computing supporting technology. The reason why parallel computing is feasible, is mainly due to that concurrency is a universal property of the material world. The scientific computing problems of polymer aspects can be decomposed into multiple sub-tasks which can be used for parallel computing.

2. Physical Models

2.1. Monte Carlo Method

Numerical simulation of polymer is often used in molecular dynamics, Monte Carlo and other methods. The basic idea of Monte Carlo method is: to create a probability model, to set the parameter such as probability distributions or mathematical expectation which is equal to the known problem solution, to calculate the statistical characteristics of the required parameters through a large number of samples, and finally to get the numerical solution of target parameters as the approximate solution. Monte Carlo method is the combination product between mathematical statistics and computer science. Since the polymer chain is made up of a large number of repeating unit, there is a random in polymerization. The size distribution of the molecular weight, the sequence distribution of the copolymer, the molecular conformation all exhibit randomness. Therefore, Monte Carlo method will undoubtedly become one of the best methods to study the polymer problems. Since the computer produced, Monte Carlo method has been applied in the field of polymers.

The conformation of polymer chain is completely random. The partition function of self-avoiding walk (SAW) chain has no precise function expression which can be directly resolved. Monte Carlo simulation is usually to study static and dynamic properties of SAW polymer chains [11-18]. Monte Carlo sampling algorithm is shown in Fig. 1.

The schematic view of a two-dimensional SAW chain is shown in Fig. 2, in which Fig. 2 (a) is an initial state after the growth chain by self-avoiding walk (chain length N = 20). Polymer chain movement is simulated by the bond length fluctuation model. After a certain time to perform the movement of polymer, the schematic diagram is shown in Fig. 2 (b). The static and dynamic properties of polymer chains are related with interaction intensity between the chain and the surface. The monomer can be obtained interaction energy E = -1 when the monomer is located at the position with the unity distance between the monomer and the surface. There is a mutual attraction between polymer and surface.

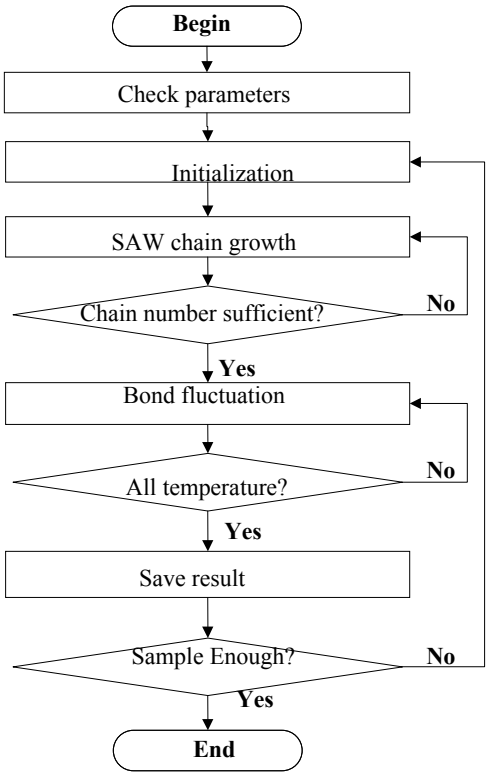
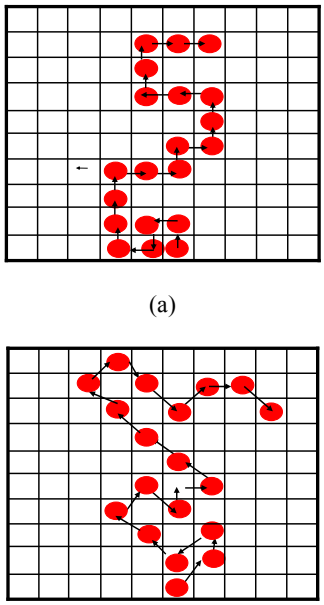


Fig. 1. The flowchart of Monte Carlo sampling algorithm.



(a)

(b)

Fig. 2. Two-dimensional SAW chain, (a) the initial state of chain growth, (b) the state after the chain motion.

2.2. Bond Fluctuation Method

In the simulation, the polymer chain does Brownian motion continuously as long as the chain generated. In the calculation model, one monomer is selected randomly and tries to walk one step. The direction is selected randomly from six coordinate axis. The polymer chain's motion is simulated by the bond fluctuation method. The bond length in simple cubic lattice model is allowed as three cases: 1, $\sqrt{2}$ and $\sqrt{3}$. In the bond fluctuations method, each lattice has the coordination number z = 8 in two-dimensional space, and z = 26 in three-dimensional space. The monomer located at z = 1 which is in contact with the surface can gain an interaction energy E.

Using the bond fluctuation method [2, 17], we simulate polymer chain of random motion, as shown in Fig. 3. Mean each monomer of chain attempts to move (success or failure) once, the time is called as a Monte Carlo time step (MCS). Monte Carlo time step is an important unit of time to analyze the polymer computing algorithm. One attempt to move is to meet the following five conditions:

1) One monomer is randomly picked from monomers of polymer chain.

2) Check the nearest neighbor (NN) position of the monomer and form motion direction set.

3) The new position is located above the surface.

4) The new bond length meets the allowed conditions (1, $\sqrt{2}$ and $\sqrt{3}$).

5) The monomer can move with probability exp ($-\Delta E/k_BT$) which is the Boltzmann factor, where ΔE is the energy shift before and after the motion. The Boltzmann constant k_B is set unity in the simulation.

If all the conditions (1) - (5) are met, the monomer moves successfully one step.

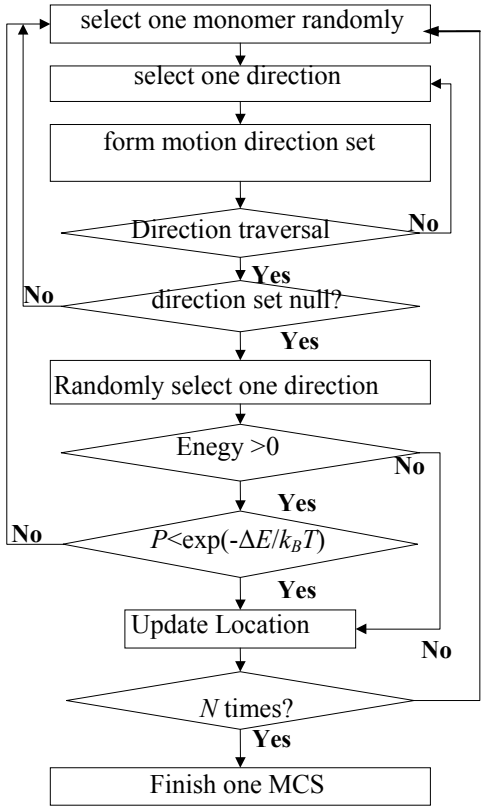


Fig. 3. The flowchart of bond fluctuation movement method.

3. Parallel Computing Technology and Analysis

3.1. Technology Roadmap

The overall objective to solve problems by parallel computing conformational properties of polymer is to map problems to the parallel machine, through different levels of abstraction mapping to achieve. First, we must establish the parallel solution model to calculate the physical problems of the polymer chains. Second, we design efficient parallel algorithm for the model, accurately describe and quantitative analyze for solving the problem, and compare the different algorithms performance. Finally, through the parallel programming, the problem to be solved in parallel need changed into parallel algorithm for a specific model to compute in a parallel manner. The polymer computing is achieved to map the parallel machine in high performance computing center, as shown in Fig. 4.

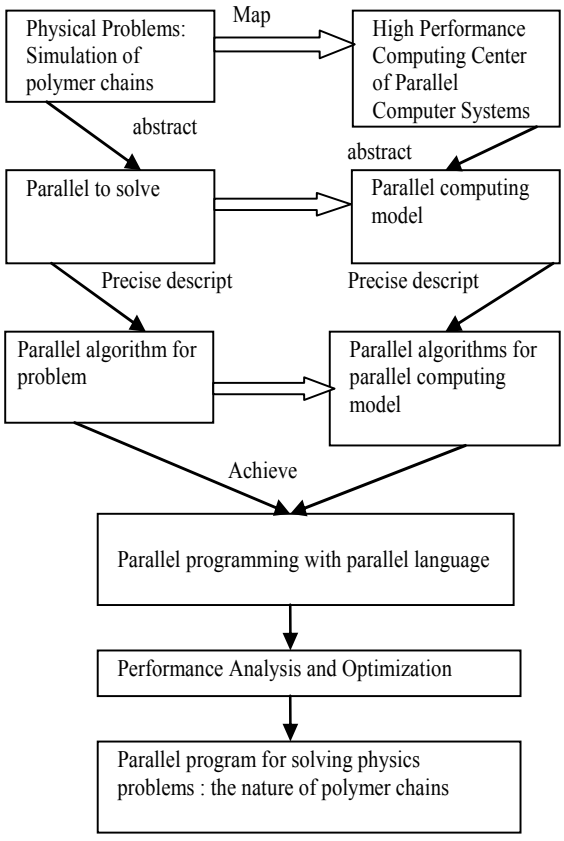


Fig. 4. The parallel computing for simulating polymer properties by self-avoiding walks.

3.2. Parallel Computer Architecture

Based on the High Performance Computing Center of Shandong University, experimental environment, we introduce the mainly used cluster system architecture. Cluster system is a complete set of commercial nodes connected together via commercial fleet switch system. Each node in the cluster is used for the unified Linux operating system, GNU (or Intel) build system and job management system, to provide users with a single system image. A single node in the cluster contains two six-core processors shared memory, as shown in Fig. 5. Cluster system primarily supports hybrid programming model include message passing and storage shared. You can only use message passing programming model to achieve the application of parallel computing.

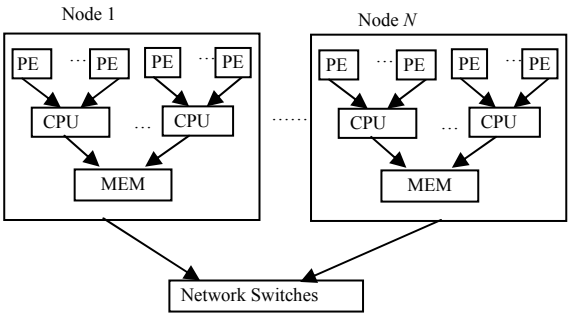


Fig. 5. Cluster system architecture diagram.

Numerical simulation of self-avoiding chain is carried out to calculate and to test on the cluster of high-performance computing centers. Test environment: high performance cluster systems blade-based cluster wave Tissot TS10000 Computer, 80 NX560T blades, using 32nm process technology Intel Xeon 5600 series processors, a total of 960 computing cores with its peak performance reaching 10 trillion times per second. 40 Gb/s InfiniBand switches are used to connect the server. The advanced water cooling system provides a stable and efficient operating environment.

3.3. Analysis of Parallel Programming Techniques

The parallel cluster machine, which is built up a number of SMP, can be carried by the programming model with MPI and OpenMP: communication using MPI between nodes, communication using OpenMP multithreaded shared memory for internal node.

4. Analysis of Scalable Parallel Computing

Speedup of parallel programs, is an index to evaluate acceleration performance. Suppose α represents the proportion of the parallel part running time, T represents the total running time, N represents the number of CPU core (or computer nodes), S represents the speedup. We introduce two performance models based on two speedup indicators, and then discuss two scalable methods.

4.1. Amdahl's Law

When the total amount of computation and simulation parameters is fixed constant, the main goal is to accelerate the computing speed. Pursuit of the ideal relationship between S and N is linear speedup curve. The calculation process of S is described as follows:

1) When not considering the cost,

$$S = \frac{T}{\frac{\alpha \cdot T}{N} + (1-\alpha)\cdot T} = \frac{N}{N-(N-1)\cdot\alpha} \xrightarrow{N\to\infty} \frac{1}{1-\alpha} \quad (1)$$

This equation (1), which is called Amdahl's law [19], is used to study one of the basic laws for parallel system. It has the following meanings:

- For a given workload, the maximum upper limit of speedup is $\frac{1}{1-\alpha}$. The sequential part of program is the bottleneck;
- In order to obtain a good speedup, we should let α as large as possible;
- The software hotspot is tested. If the above two parts of the problem meet, you consider the hot part of the program parallelization, and optimize the most commonly used parts.

2) When considering the cost of T_0,

$$S = \frac{T}{(1-\alpha)\cdot T + \alpha\cdot\frac{T}{N} + T_0} = \frac{1}{1+\frac{T_0}{T} - \alpha\cdot\frac{N-1}{N}} \xrightarrow{N\to\infty} \frac{1}{1+\frac{T_0}{T}-\alpha} \quad (2)$$

The extension of Amdahl's law indicates that we can not only reduce bottlenecks in the sequential part (1-α), but also increase the average particle size in order to reduce the harmful effects of overhead. The performance of the parallel program is not only limited by sequential bottlenecks, but also limited by the average overhead.

4.2. Gustafson's Law

In the application area, if the time spent to solve the small problems in the small-scale machine is about the same as that spent to solve the large problem in the major machine, we can consider Gustafson law [20]. For example, polymer size has increased tenfold, and the task of parallel part increases at the same speed and the sequential part has a constant execution time, so that the parallel work is more scalable. It means that we can apply more cores or larger cluster system.

Therefore, when the machine increases the size, we have increased the workload. Assuming the serial calculation time of the original problem is W, where α is the percentage of the load section can be parallel, the serial load section is 1-α. We will expand the workload as:

$$W' = \alpha \cdot N \cdot W + (1-\alpha) \cdot W \tag{3}$$

1) When not considering the cost, the parallel time after the expansion load on the N nodes:

$$T_p = T_{ParParts} + T_{SeqParts} = \frac{\alpha \cdot N \cdot W}{N} + (1-\alpha) \cdot W = W \tag{4}$$

However, the serial time to expand the load becomes:

$$T_s = \alpha \cdot N \cdot W + (1-\alpha) \cdot W \tag{5}$$

So there is a fixed sequential time, the speedup is:

$$\begin{aligned} S &= \frac{T_s}{T_p} = \frac{\alpha \cdot N \cdot W + (1-\alpha) \cdot W}{W} \\ &= \alpha \cdot N + (1-\alpha) \end{aligned} \tag{6}$$

The equation (6) becomes Gustafson's law. If the workload can be extended in order to maintain a fixed sequential execution time, the fixed-time speedup is a linear function of N. When the question of the expansion has been extended to the available computing power to match, the sequential part will no longer be the bottleneck. It should be noted, to make Gustafson's law to be established, the key is to only make a parallelizable workloads from αW expanded into αNW, sequential part (1-α)W should remain unchanged.

2) When considering the cost, if the overhead time T_0 is a function of N, the parallel time is expressed as:

$$T_p = T_0 + W \tag{7}$$

So there is expansion speedup:

$$S = \frac{T_s}{T_p} = \frac{\alpha \cdot N \cdot W + (1-\alpha) \cdot W}{W + T_0} = \frac{\alpha \cdot N + (1-\alpha)}{1 + \frac{T_0}{W}}$$

$$\begin{cases} = \alpha \cdot N + (1-\alpha), & W >> T_0 \\ = \dfrac{\alpha \cdot N + (1-\alpha)}{1 + \dfrac{T_0(N)}{W}}, & T_0 = T_0(\mathrm{N}) \end{cases} \tag{8}$$

Seen from (8), T_0 contains the part increase decreases with the increase in N, or remains constant part, so long as the control overhead of a non-increasing function of N, or a upper bound function, the Gustafson law can be obtained a linear speedup.

5. Parallel Design and Implementation

5.1. Parallel Strategies of Polymer Simulation

Before the application migration to high performance cluster platform, we should first analyze application mode suitable for high performance cluster computing. High performance cluster uses multiple blades nodes interconnected through an Ethernet switch, each blade is an SMP parallel machine. Therefore, parallel modes are available: message passing mode, single-node shared memory system model, mixed mode between message passing and shared memory.

This design pattern is first selected to play parallel Monte Carlo method in use characteristics. However, parallel Monte Carlo method also has limitations. When the particle size allocated to each MPI process one sample, the parallel calculating performance is maximized. In order to increase computing scale, the mixed mode between MPI and OpenMP is selected to use the process communicate among different computing nodes and to use OpenMP multi-threaded parallel on a single node.

5.2. Parallel Algorithms of Polymer Simulation

Algorithm is designed to solve the problem by a computer. The nature of the serial algorithm is to describe using a serial computer to solve a sequential problem or sequence steps. However, parallel algorithm is to describe for a plurality of processors. Two key steps for parallel algorithm design are described as:

- The computing problem divided into more smaller computing ones;
- The computing parts would be assigned to different processors to be executed.

The parallel computing of polymer simulation application is achieved at high-performance cluster by using the parallel Monte Carlo algorithms and domain decomposition algorithms, as well as MPI and OpenMP programming.

Different samples are computed at different nodes, and polymer chain is divided to segments in a sample for parallel computing, respectively, using MPI and OpenMP implementation. The optimization parallel computing model is using MPI + OpenMP optimization parallel computing. The applications are improved scalability speedup between processors with little communication between nodes. The parallel grogram flowchart is described in Fig. 6.

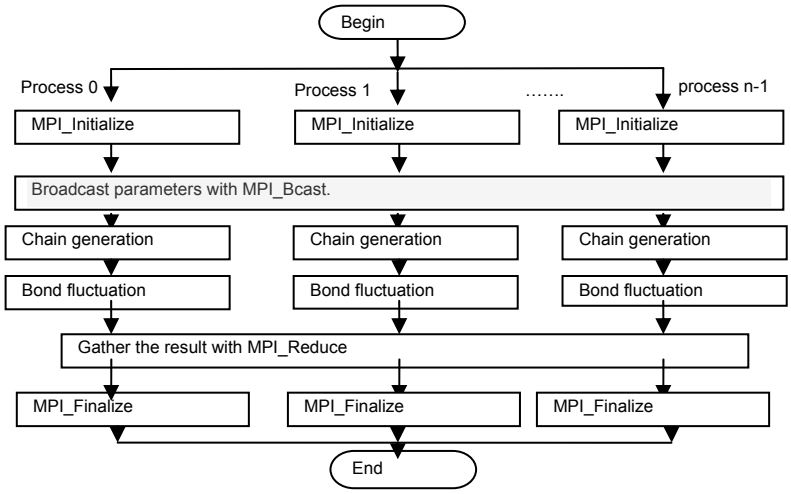


Fig. 6. Parallel program flowchart with MPI.

6. System Test

6.1. System Benchmarks and Analysis

The bond fluctuation motion is the most time-consuming in the calculation of polymer adsorption simulation. The main reason is to complete a calculation of each Monte Carlo step, which is necessary to traversal for the whole chain once. Each traversal must try the current monomer's six directions movement. Finally whether the movement should be accepted is based on the conditions judgment. With the expansion of the size calculation, the entire application program running bond-fluctuation takes up 95 % of the system over time. Therefore, for each hotspot function bond_fluctuation optimization process on the final performance results will have a qualitative impact.

6.2. MPI Parallel Performance

Samples are often independence performed by parallel Monte Carlo method. MPI message passing programming model is often applied to Monte Carlo method. The parallelism characteristics mainly reflected that MPI communication time scale is very small. Fig. 7 is in MPI mode, use the tool IPM measure MPI statement used in the frequency and time for cluster with 20 nodes, 240 cores, 480 samples, a total of 240 processes executed in parallel. You can see the MPI communication hotspots in two function calls MPI_Reduce and MPI_Barrier, but the walls of MPI takes time less than 1 %, to meet the conditions of Gustafson's law.

We calculate the polymer adsorption at a high-performance cluster. The speedup with MPI parallel acceleration on the computing problems expansion with the increase of statistics interval time is shown in Fig. 8. The results show that the increase in computing statistics interval requires to increase cores to maintain parallel computing time unchanged. When the statistics interval is small, the proportion of MPI communication increases with the increase of cores. Therefore, the speedup increases slowly. With the increases of the statistics interval, the task of parallel computing work increases and sequential computing work reduces proportionally. Then the proportion of MPI communication reduces, the speedup tends to increase linearly. According to Gustafson's law, the statistics interval continues to increase, linear speedup can be obtained.

command: unknown			
codename:	unknown	state:	running
username:	user38	group:	gos
host:	c04b04 (x86_64_Linux)	mpi_tasks:	240 on 20 hosts
start:	05/17/12/21:49:52	wallclock:	1.13302e+02 sec
stop:	05/17/12/21:51:45	%comm:	11.6737820033333
total memory:	0 gbytes	total gflop/sec:	-2.11823268786076
switch(send):	0 gbytes	switch(recv):	0 gbytes

Communication Event Statistics (100.00% detail, -2.3903e-05 error)

	Buffer Size	Ncalls	Total Time	Min Time	Max Time	%MPI	%Wall
MPI_Reduce	144	6000	18.990	1.555e-06	1.801e-01	71.49	0.76
MPI_Reduce	8	2280000	3.873	1.054e-06	1.667e-01	14.58	0.16
MPI_Barrier	0	160	3.542	1.123e-02	4.450e-02	13.33	0.14

Fig. 7. Testing parallel MPI program of polymer simulation at high-performance clusters.

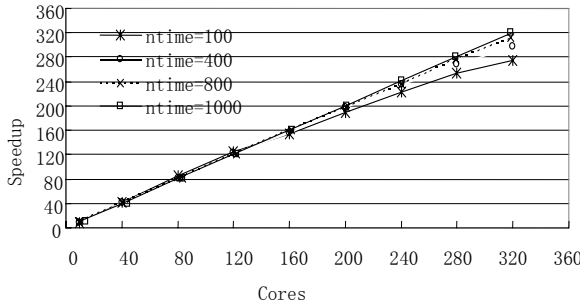


Fig. 8. Performance test of parallel MPI program of polymer simulation.

7. Conclusions

In This paper, Monte Carlo method is carried out for polymer simulation. We simulate polymer chain growth and movement. The initial state of the polymer chain is generated by self-avoiding walk method to simulate. The polymer chain movement is simulated by bond fluctuation model. The static and dynamic properties of polymer chains often need a large number of samples to calculate statistical sampling. Monte Carlo methods are using different independent samples for parallel computing to achieve the desired speedup. Based on the MPI message passing, parallel programming software is analyzed and designed to ensure program portability. Scientific calculation time can be reduced to an acceptable time. In order to further reduce the computation time or to calculate longer polymer chain, MPI + GPU CUDA is a good choice.

Acknowledgment

The authors would like to thank M. B. Luo and J. H. Huang for their useful discussions. The authors would like to thank Z. G. Sun for his assistance in the simulation and test work. The authors would like to thank Shandong University High Performance Computing Center and Wenzhou University High Performance Computing Center.

This work was supported by the National Science Foundation of China under Grants No. 11304231, No. 21174132 and No. 21171145.

References

[1]. S. K. Jha and R. D. S. Yadava, Data Mining Approach to Polymer Selection for Making SAW Sensor Array Based Electronic Nose, *Sensors & Transducers*, Vol. 147, Issue 12, December 2012, pp. 108-128.

[2]. Y. Wang, H. Xu, Z. G. Wang, R. F. Hu, Z. Y. Luo, Z. K. Xu, G. Li, A Glucose Sensor Based on Glucose Oxidase Immobilized by Electrospinning Nanofibrous Polymer Membranes Modified with Carbon Nanotubes, *Sensors & Transducers*, Vol. 152, Issue 5, May 2013, pp. 180-185.

[3]. S. V. Patil, P. R. Deshmukh, C. D. Lokhande, Room Temperature Liquefied Petroleum Gas Sensing Polymer (n-polypyrrole/p-polyaniline) Based Heterojunction, *Sensors & Transducers*, Vol. 137, Issue 2, February 2012, pp. 104-114.

[4]. M. M. Radhi, Electrocoductivity Studies of Grafted Polymer Thin Film, *Sensors & Transducers*, Vol. 125,Issue 2, February 2011, pp.168-176.

[5]. H. Li, C. J. Qian, M. B. Luo, *et al.*, Conformational properties of a polymer tethered to an interacting flat surface, *Polymer J.*, Vol. 42, 2010, pp. 383.

[6]. H. Li, C. J. Qian, L. Z. Sun, *et al.*, Simulation of a flexible polymer tethered to a flat adsorbing surface, *Journal of Applied Polymer Science*, 124, 2012, pp. 282.

[7]. A. Uhlherr, Parallel Monte Carlo simulations by asynchronous domain decomposition, *Comput. Phys. Comm.*, 155, 2003, pp. 31–41.

[8]. S. Nedelcu, M. Werner, M. Lang, *et al.*, GPU implementation of the bond fluctuation model, *Journal of Computational Physics*, 231, 2012, pp. 2811.

[9]. J. H. Lee, S. Y. Kim, J. Lee, Parallel algorithm for calculation of the exact partition function of a lattice polymer, *Comput. Phys. Comm.*, 182, 2011, pp. 1027.

[10]. J. Gross, W. Janke, M. Bachmann, Massively parallelized replica-exchange simulations of polymers on GPUs, *Comput. Phys. Comm.*, 182, 2011, pp. 1638.

[11]. H. Li, C. J. Qian, C. Wang, M. B. Luo, Critical adsorption of a flexible polymerconfined between two parallel interacting surfaces, *Phys. Rev. E*, 87, 2013, p. 012602.

[12]. H. Li, C. J. Qian, M. B. Luo, Simulation of a Flexible Polymer Confined between Two Parallel Surfaces, *Polymers & Polymer Composites*, 20, 2012, pp. 107-110.

[13]. H. B. Gao, H. Li, C. J. Qian, Parallel Monte Carlo Simulation of Single Polymer Chain, *Applied Mechanics and Materials*, 263–266, 2013, pp. 3317-3320.

[14]. H. B. Gao, Computer Simulation of a Flexible Polymer Chain in a Slit, Communications in Computer and Information Science, 163, 2011, pp. 417-424.

[15]. H. Li, B. Gong, H. B. Gao, C.-J. Qian, Parallel Computing Properties of Tail Copolymer Chain, *TELKOMNIKA*, 11, 2013, pp. 4344-4350.

[16]. M. B. Luo, The critical adsorption point of self-avoiding walks: A finite-size scaling approach, *Journal Chem. Phys.,* 128, 2008, pp. 044912.

[17]. M. B. Luo, J. H. Huang, Monte Carlo simulation of polymer chain with ferromagnetic Ising interaction, *Journal. Chem. Phys.*, 119, 2003, pp. 2439.

[18]. M. B. Luo, J. H. Huang, Y. C. Chen, J. M. Xu, Correlation between shape and size of linear polymer chains, *Eur. Polym. Journal.,* 37, 2001, pp. 1587.

[19]. G. M. Amdahl, Validity of the single-processor approach to achieving large scale computing capabilities, in *Proceedings of the AFIPS Spring Joint Computer Conference*, 1967, pp. 483.

[20]. J. Gustafson, Reevaluating Amdahl's Law, *Communication of ACM*, 31, 5, 1988, pp. 532.

[21]. M. Snir, D. Bader, A framework for measuring supercomputer productivity, *International Journal of High Performance Computing Applications*, 18, 4, 2004, pp. 417.