МРНТИ 28.29.53

A.S. Kussainov,[1,2] *A.K. Beisekov*[1,2]

[1]Physics And Technology Department, al-Farabi Kazakh National University, Republic of Kazakhstan, Almaty

[2]National Nanotechnology Laboratory of Open Type, al-farabi Kazakh National University, Republic of Kazakhstan, Almaty

QUICK AND EASY PARALLELIZATION TECHNIQUE FOR THE ISING 2D MODEL IN OPEN MPI

Abstract. We have demonstrated a quick and easy compartmentalization method of the 2D Ising model and studied its efficiency and data produced. To optimize optional distributed computations, the intercompartmental communication was kept at minimum level. Boundary data between the compartments were updated only with each Monte Carlo step, that is, only after annealing has taken place within each individual compartment with number of steps bigger than the number of sites in compartment. Open MPI package implementing Message Passing Interface (MPI) for Debian Linux operational system was chosen to provide parallelization environment. The suggested method is straightforward, easy to use, produces correct results and is seamlessly scaled. Simulated ferromagnetic domains beyond the compartment size are clearly observed, and freely develop across the simulation grid. Exact results from the existing publications have been reproduced with greater efficiency. Significant speedup on the octacore desktop computer has been demonstrated.

Key words: spin gas, Monte Carlo Method, Ising model, parallel programming, compartmentalization, Open MPI.

Аннотация. Предложена быстрая и эффективная схема распараллеливания двухмерной модели Изинга на вычислительные блоки. Приведен анализ эффективности и результатов работы схемы. Для оптимизации работы схемы, в случае распределенных вычислений по сети необходимость обмена информацией между блоками сведена к минимуму. Значения граничных условий для каждого блока обновлялись после каждого шага Монте-Карло, т.е. только после того, как симулированный отжиг был проведен для данного блока. Количество шагов в отжиге не меньше количества элементов блока. Пакет Open MPI для операционной системы Debian Linux был использован для создания параллельной среды программирования. Предложенная схема проста в использовании и интерпретации, выдает правильные результаты и легко масштабируется. Кластеры спонтанной намагниченности при переходе от ферромагнитной фазы к парамагнитной легко наблюдаются и свободно эволюционируют в пределах моделируемой решетки. Полученные данные совпадают с ранее опубликованными результатами. Достигнуто существенное увеличение производительности на компьютере с восьмиядерным процессором.

Ключевые слова: спиновое стекло, метод Монте-Карло, модель Изинга, параллельные вычисления, распараллеливание, Open MPI.

Түйіндеме: Біздің жұмысымызда екі өлшемді Изинг моделінің есептеуіш блоктарға эффективті әрі жылдам параллельдеу сұлбасы ұсынылып отыр. Сұлбаның талдау эффективтілігі және жұмыс нәтижесі келтірілді. Сұлба жұмысын оңтайландыру үшін желі бойынша таралған есептеулер жағдайында блоктар арасындағы ақпарат алмасудың минимумға қажеттілігін енгіздік. Шеттік шарттар мәні әр блок үшін Монте Карлоның әр қадамы сайын жаңартылып отырды, яғни тек қана жасанды босаңдату берілген блок үшін жасалған кезде жаңартылды. Босаңдату кадам саны блок элементтері санынан аз болмайды. Debian Linux операциялық жүйесі үшін Open MPI пакеті параллельді программалау ортасын құру үшін пайдаланылды. Ұсынылған сұлба қарапайым және интерпретациялар дұрыс нәтижелер көрсетіп, оңай масштабталады. Оқыс магниттенгендер кластері ферромагнитті фазадан парамагниттіге ауысу кезінде жеңіл бақыланады, сондай-ақ моделі құрылатын тор шегінде еркін дамиды. Алынған ақпараттар ертерек жарияланған натижелермен сәйкес келеді. Сегіз ядролық процессорлы компьютерде өнімділіктің айтарлықтай өсуіне қол жеткізілді.

Түйінді сөздер: спинді шыны, Монте Карло әдісі, Изинг моделі, параллельді есептеу, параллельдеу, Open MPI.

Introduction

Powerful personal desktops or workstations, as well as clusters and high performance computing systems with remote access, are widely available nowadays. These are state-of-the-art and expensive machines maintained by numerous staff and capable of addressing the fundamental pure and applied science problems of today. Still, enormous amount of numerical simulations are done in a single thread. This fact seriously deteriorates the efficiency of computations and reduces the naturally parallel tasks to the bottleneck of a single thread application. Many scientists, though having in their possession the state-of-the-art multicore computing systems, are able to utilize only a small portion of their computational power. Multiple studies and helpful resources are published and circulated within the interested scientific and research communities to help them get familiar with parallel programming [1, 2].

Physicists have made many major advances in computational and mathematical physics. They clearly see the underlining physical processes and could tailor mathematics and programming algorithms to their specific tasks. For example, quantum computing is essentially a parallel multitasking at all levels. Blind use of an automated parsing software is unacceptable. Programming that uses the basic physics principles is required.

We address these and many other points by using the Open MPI library [3] that can handle multithread coding and feed it to a multicore CPU (central processing unit) or distribute the tasks across a network of computers connected in the computational cluster. Unlike its counterpart, the Open MP development of the message parsing protocols, [4] Open MPI is extensively documented in electronic resources and much easier to deploy.

We have implemented Monte Carlo method with importance sampling in 2D spin glasses for the Ising model [5] as an example of multithreading and performance optimization in scientific computing. Conventionally, performance could be gain by compartmentalization and deployment of custom-made communicator between compartments for the transient phenomena. The same results could be achieved by understanding the simple topology and physics of the problem, reducing communication time and instances to a minimum.

Methods

The Ising model we chose is described by the two-dimensional m x n grid of spins [6]. Each spin can take only two values, up or down, si={+1,-1}. Assuming that magnetic interaction strength is dropping as fast as 1/r3 with the distance, we will consider interactions only between the closest neighbors. These neighbors are forming a cross pattern, (see, for example, five spins' sites shaded in red in Fig.1(a)). Immediately, we introduce periodic boundary conditions. That is, if the left neighbor in the left corner of interaction pattern is missing, (see yellow-shaded areas), we assume that its place is taken by the spin across the whole grid on the right boundary. The same technique is valid for the right, upper and bottom boundary sites.

One could directly index such sites and use them as a precursor to a custom-made communicator between the processes in the parallel version of the program. Our choice is to increase the simulation grid by copying the right boundary column to the left and the left boundary column to the right, as well as the top row to the bottom and bottom row to the top, (see the gray-shaded columns and rows in Fig. 1 (b)). In Figure 1(c) this procedure will insure the communication between the compartments in parallel algorithm. However, in this case, the boundary will be provided by a neighboring compartment

Mathematical description of the model is presented below. The total interaction energy associated with all possible closest neighbor spins is given by

$$E = -J\sum_{ij} s_i s_j - H\sum_{i=1}^{m\times n} s_i \qquad (1)$$

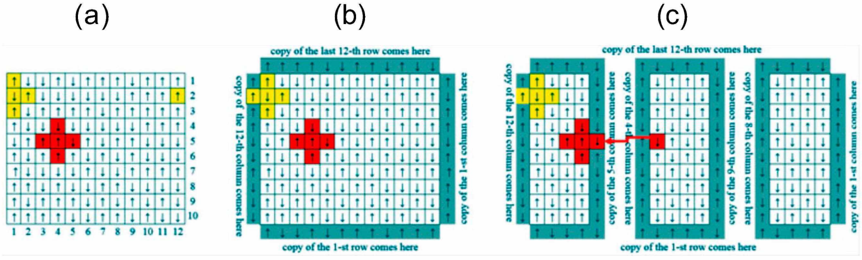
(a)
(b)
(c)
copy of the last 12-th row comes here
copy of the 12-th column comes here
copy of the 1-st column comes here
copy of the 1-st row comes here
copy of the 5-th column comes here
copy of the 4-th column comes here
copy of the 9-th column comes here
copy of the 8-th column comes here

Figure 1. Compartmentalization scheme for three processes

where J is the spin-spin interaction strength and H is the external magnetic field strength. If *J*>0 the system is ferromagnetic, otherwise, if J<0, it is paramagnetic. Indices i and j sample all the available pairs of neighboring spins on the grid, excluding the double count of ij and ji pairs. The upper bound for the first sum is determined by the range of interaction and number of spins within this range. If the spins are allowed to have Q ≥2 states, this will be the generalized Q-state Potts model [7]. Total magnetization value at the certain spins configurations is calculated as a sum

$$M = \sum_{i=1}^{m\times n} s_i \qquad (2)$$

If the external field strength *H* is set to zero then there are two distinct states at low and high temperatures. These are ferromagnetic and paramagnetic phases separated by the transition region around Currie temperature T_c.

According to Metropolis algorithm [8] instead of trying to calculate quantum-mechanical observables over all possible combinations of states

$$M = \frac{\sum_{conf.space} M e^{E/kT}}{\sum_{conf.space} e^{E/kT}} \qquad (3)$$

Temperature *T* is given in the units of [*E*/*k*], where *k* is the Boltzmann factor. *E* is dimensionless but in general has the units provided by expression (1). We should include only those configurations which are sampled according to the Boltzman factor, see algorithm below. For a sequence of N such states, magnetization for the particular temperature will be given by the formula

$$\langle M \rangle = 1/N \sum_{j=1}^{N} \sum_{i=1}^{m\times n} S_i \qquad (4)$$

The following steps should be taken repeatedly to achieve a desired distribution of states at the certain temperature T:

1. Choose at random any spin si and flip its sign, *si'*=-*si*.
2. Calculate change in the total energy according to formula

$$\Delta E = E_{S_i} - E_{S_i} \tag{5}$$

3. If $e^{\Delta E/kT} > X$, where $X \sim U([0,1])$ is a random variable uniformly distributed on [0,1], the change in sign is accepted.

4. Repeat the previous steps to achieve an equilibrium magnetization value for a given simulation grid at the temperature selected for the system.

These four steps, repeated multiple times but usually not less than the number of sites in simulation grid or compartment, represent one Monte Carlo step. To calculate any observable value for a given temperature, MC steps have to be repeated several times. Other properties, including density of states, could be calculated via similar algorithms [9, 10].

Convergence to the equilibrium values of observable parameters with one spin flip is slow. Changes introduced by a single local spin flip propagate diffusively. Various algorithms have been proposed to speed up the process by flipping the whole clusters of spins at once. Swendsen-Wang [11] and Wolff [12] Monte Carlo methods are among them. The Wolff algorithm is an improvement over the Swendsen-Wang algorithm since it has a larger probability of flipping bigger clusters. Alternatively, we could partition the simulation volume for the parallel processing [13, 14] as we did in this paper, see Figure 1.

One may also avoid numerous exponentiation steps by noticing, that for a two dimensional grid, ΔE takes a limited number of values depending on orientation of the four neighboring spins. We can easily show that this number is equal to five and it doubles if the external magnetic field H is switched on.

Results and Discussions

Configuration of our eight core desktop computer is listed as follows: Intel Core i7 4790K, 4.0GHz/LGA-1150/22nm/Haswell/8Mb L3 Cache, DDR-3 DIMM 16Gb/1866MHz PC14900, 2x8Gb Kit, CL10.

As we said before, we perform the boundary conditions exchange every time the equilibrium within each compartment is reached. That is the boundary conditions are renewed for all compartments every time after each Monte Carlo step in accordance with equilibrium spins configuration in the neighboring compartments. One should take care not to replicate the simple periodic boundary conditions for each compartment. This means that we have to make sure that compartments are communicating with each other and supplying each other with information about the boundary conditions along the interface line.

Fig.2(a) shows the spins orientation distribution computed in one thread without any compartmentalization.

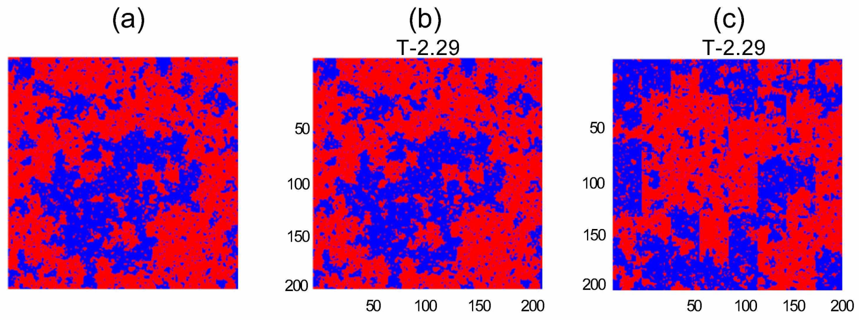

Figure 2. Sample magnetization distribution under different boundary conditions.

Simple periodic boundary conditions on four boundaries were used. In Fig. 2(b), we plotted the case of eight communicating compartments when data in the compartments are formatted according to our algorithm. The case when the closed toroidal boundary conditions were implemented for each compartment without the proper boundary information exchange is shown in Fig.2(c). Fig.2(c) shows the obvious signs of an erroneous dynamics, such as band structure, indicating that the compartments are not able to communicate with each other.

The next Fig. 3 gives a diagram for the simulated annealing of our system. Normalized magnetization M/M0, where M0 is the sum of all spins, changes its value from 1 at T=0, to 0 as the temperature rises beyond the critical value. Temperature of the phase transition Tc is about 2.3.

As one can see, the data from a single thread experiment with no compartmentalization, (represented by a green line and triangle markers), and data with proper compartmentalization, (blue line and diamond markers, are almost identical. The red line with pentagram markers stands for the incomplete implementation of the boundary conditions and exhibits a deviant behavior at phase transition. Nevertheless, all give about the same value of Tc.

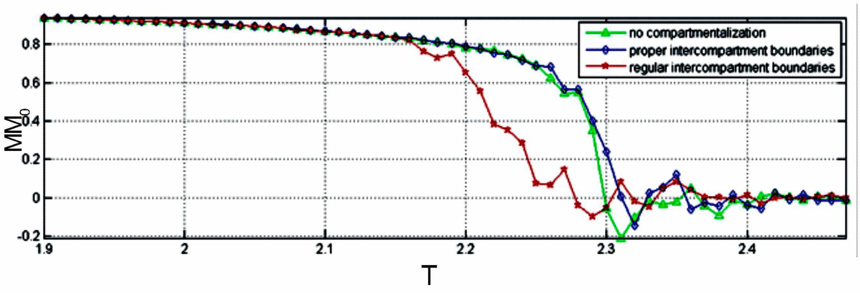

Figure 3. Simulated annealing for the 200 x 200 grid. Green line and triangle markers - no compartmentalization; Blue line and diamond markers - custom made compartmentalization and data exchange technique; Red line and pentagram markers - data produced with unadjusted boundary conditions between compartments

Next, Fig. 4 shows the timing data from our simulations when the number of threads goes up from 1 to 12, see the Y axis. Two sets of curves are plotted. The one on the left is for 200 by 200 data grid another one is for 400 by 400 grid. The obvious benefits of compartmentalization are visible. For the 200 by 200 grid, five thousands Monte Carlo steps and only one temperature value, CPU time is cut in half if we split the simulation volume between eight cores of a single processor. Normally, we simulate annealing for the range of T values and number of Monte Carlo steps is much higher than the number we used. Thus, the time savings are enormous.

Multiple parallel platforms and custom made algorithms make it difficult to compare performance for each case. For the formal performance evaluation and further optimization of the arbitrary code the LogP model of Culler is available [15].

Conclusions

We reported our studies of implementation and efficiency of the quick and robust method of compartmentalization for the 2D Ising

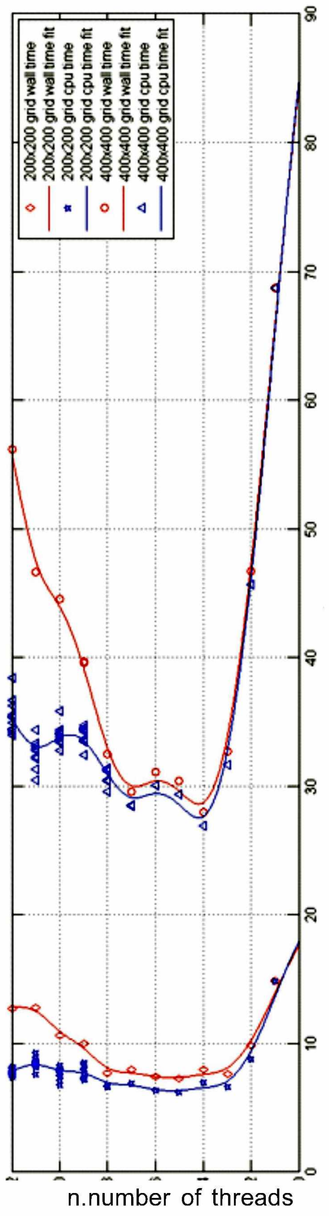

Figure 4. Parallelization algorithm performance for 200 x 200 and 400 x 400 grids. CPU, blue lines, and wall, red lines, times are given as a function of number of threads on the octacore processor.

model. Open MPI package has provided the parallel computation environment. Algorithm structure is optimized for optional distributed computations by updating the boundary data between compartments with each Monte Carlo step. Meanwhile annealing, within each Monte Carlo step and for each individual compartment, takes the same number of steps as the number of sites in a compartment.

This boundary information update is enough to couple the statistical processes within each individual compartment to its neighbors. Thus construction of the complicated blocking type communications through a message passing interface is avoided. In general, it allows us to keep the basic simple model of the spin glass intact and evenly distribute intensive computations between the available threads. These two facts about our method contribute to the clarity of the model and its data interpretation as well as to increased speed of calculations.

Our proposed method is straightforward, easy to use, and produces correct results identical to the previously published data. Simulated ferromagnetic clusters of spins, freely developing between individual compartments, are clearly observed. Significant speedup on the octacore desktop computer has been demonstrated.

The model and its development represent the computational basis for the whole generation of the quantum algorithms for approximating partition.

Acknowledgments

This research was supported by grant №3824/ГФ4 provided by the Science Committee at the Ministry of Science and Education of Republic of Kazakhstan to the principal investigator at the National Nanotechnology Laboratory of Open Type, Physics and Technology Department, al-Farabi Kazakh National University.

References

1 *Gergel' V.P.* Vysokoproizvoditel'nye vychisleniya dl mnogoyadernyh mnogoprocessornyh sistem. Uchebnoe posobie.- Nizhnij Novgorod:Izd-vo NNGU im.N.I.Lobachevskogo, 2010.- 421 p.

2 *Akhter S., Roberts J.* Multi-core Programming: Increasing Performance Through Software Multi-threading // Intel Press, 2006.- 336 p.

3 Open MPI project [web data base].-Information and resources.- http://www.open-mpi.org/

4 *Chapman B. Using OpenMP*: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation) // The MIT Press: Scientific and Engineering Computation edition, 2007. – 384 p.

5 *Ising E.* Beitrag zur Theorie des Ferromagnetismus // Zeitschrift fur Physik. – 1925. – № 31. – P. 253-258.

6 *Domb C., Green M.S.* Ising Model. Phase Transitions and Critical Phenomena // Academic Press. – 1974. – V. 3. – P. 357-484.

7 *Potts R.B.* Some Generalized Order-Disorder Transformations // Proceedings of the Cambridge Philosophy Society. – 1952. – № 48(1). – P. 106-109.

8 *Metropolis N., Rosenbluth A.W.* Equation of State Calculations by Fast Computing Machines // Journal of Chemical Physics. – 1953. – № 21.– P. 1087-1092.

9 *Wang F., Landau D.P.* Efficient, multiple-range random walk algorithm to calculate the density of states // Phys. Rev. Lett. – 2011. – № 86. – P. 2050-2053.

10 *Zhang C., Ma J.* Simulation via direct computation of partition functions // Phys. Rev. E. – 2007. – № 76. – P. 036708-15.

11 *Wang J.-S., Swendsen R.H.* Cluster Monte Carlo algorithms // Physica A: Statistical Mechanics and its Applications. – 1990. – 167(3). – P. 565-579.

12 *Wolff U.* Collective Monte Carlo Updating for Spin Systems // Physical Review Letters. – 1989. – № 62(4). – P. 361-364.

13 *Altevogt P., Linke A.* Parallelization of the two-dimensional Ising model on a cluster of IBM RISC system/6000 workstations // Parallel Computing. – 1993. – 19(9). – P. 1041-1052.

14 *Heermann D. W., Burkitt A.N.* Parallel Algorithms in Computational Science. Springer-Verlag New York, Inc. New York, NY, USA, 1991.

15 *Culler D., Karp R., Patterson D., Sahay A., Schauser K.E., Santos E., Subramonian R.* and Thorsten von Eicken. LogP: towards a realistic model of parallel computation // Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming (PPOPP '93). ACM, New York, NY, USA. – P. 1-12.