POPULATION ANNEALING METHOD AND HYBRID SUPERCOMPUTER ARCHITECTURE

L.N. Shchur [1,2]

[1] *Landau Institute for Theoretical Physics, 142432, Chernogolovka, Russia*

[2] *HSE University, 101000, Moscow, Russia*

E-mail: lev@landau.ac.ru

A population annealing method is a universal algorithm applicable to statistical mechanics systems and optimization problems. It is potentially scalable on any parallel architecture. We review recent developments in the area, emphasizing the implementation of the algorithm on a hybrid parallel program architecture combining CUDA and MPI. The problem is to keep all general-purpose graphics processing unit devices as busy as possible by efficiently redistributing replicas. We provide testing details on hardware-based Intel Skylake/Nvidia V100, running more than two million replicas of the Ising model samples in parallel. As the complexity of the simulated system increases, the acceleration grows toward perfect scalability.
This work was done under Grant No. 19-11-00286 from the Russian Science Foundation and was supported in part through computational resources of HPC facilities at HSE University [1].

Keywords: population annealing, GPU, CUDA, MPI

Lev Shchur

1. Previous work

Population annealing algorithm proposed by Hukushima and Iba [2] to simulate statistical mechanics systems with the complex energy landscape. The central idea is to simulate the enormous number of system replicas, splitting simulation into two steps, resampling replicas at each cooling step, and equilibrating replicas independently at the current temperature. The exciting feature of the algorithm is that it estimates the free energy at each cooling step using the average over the number of replicas. It was shown by Machta [3] that it is possible to estimate the behavior of statistical errors and systematic errors using weighted averages as a function of the number of replicas R. The statistical errors decay as $1/R^{1/2}$, and the systematic errors decay as $1/R$ for a large enough number of replicas R.

There are successful applications of the method to the spin glasses [4,5], molecular dynamics [6], first-order phase transitions [7], and optimization problems [8]. A detailed description of the method and the analysis of the accuracy dependence on the essential parameters of the simulation can be found in the recent publication [9].

The algorithm was successfully implemented using CUDA [10], and it was found that the optimal number of replicas per one GPU V100 node should be about ten times larger than the number of threads. It gives the possibility to run 20 thousand replicas on one GPU node.

Recently, we extended the range of population annealing (PA) simulations up to more than two million replicas running in parallel on the HSE University supercomputer cHARISMa with 104 GPU Nvidia V100 [11].

2. CUDA/MPI realization of PA algorithm

The main question in the multi-node realization of the PA algorithm is how to keep the distribution number of replicas at each node as flat as possible? It can happen that at some nodes, the number of replicas will grow while cooling, and at some nodes, the number of replicas can become very small. In such a case, the computing time at nodes becomes very different, and simulation could be inefficient.

The possible solving of the problem consists of the grouping of replicas in the blocks with the moderate size 1024 and using twenty blocks per GPU [11]. Blocks redistribute replicas. Before the redistribution step, the algorithm calculates the excess value of blocks at each node as the difference between the optimal number of blocks and the allowed excess number or the shortage number of blocks, depending on which one has a positive value. The decision is performed at the master node, and the master starts the redistribution of replicas depending on the information on the excess and shortage values.

The number of replicas during the simulations is relatively flat with the block algorithm and fluctuates within the allowed window, which is one or two excess/shortage blocks [11].

3. Testing at HSE supercomputing facilities

The block PA algorithm was tested in [11] on the example of Ising model with square lattice with 64^2 spins running GPU code published in [7] on the HSE cluster with 26 nodes, with 104 GPUs available. The program environment was OPENMP 4.0.1, CUDA version 10.2, and NVIDIA driver version 440.33.01. The scalability is shown in Figure 1, each GPU simulates an average of 20 blocks with 1024 replicas, and the volume of computations grows with the number of GPUs from 1 to 104. Therefore, the HSE computer simulated up to 1024x20x104 = 2 129 920 replicas of the Ising model in parallel. More than 50 percent of possible simulation power was used. We have to note that the value of the relaxation parameter was relatively moderate in simulations, and with the large value of the relaxation be deduced for the simulation of systems that takes more time for equilibration or need more time for the calculation at each step.

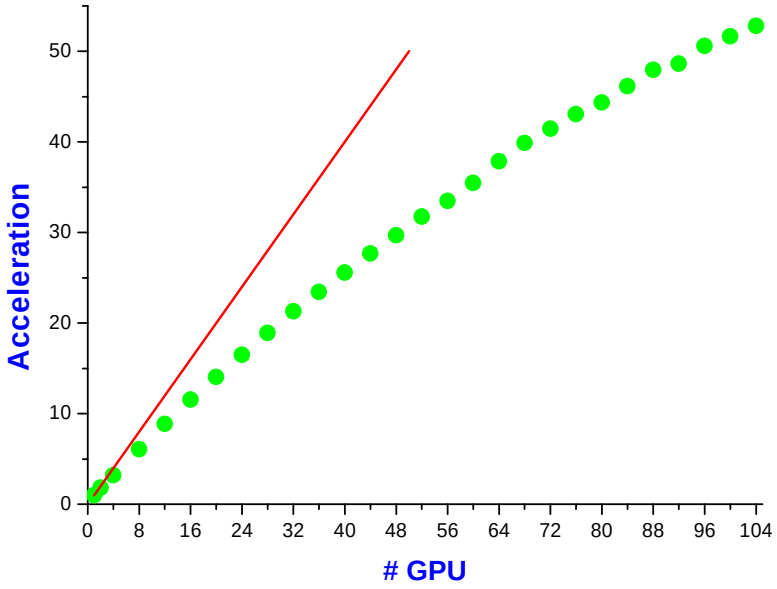

Figure 1. Acceleration of simulations as function of number of GPU used. The red line is the maximum possible acceleration.

4. Future plans

Nowadays, the HSE supercomputer cHARISMa has been upgraded with the 184 GPU available. We plan to use the block PA algorithm to simulate the statistical mechanic's complex behavior at low temperatures.

In addition, there is an extension of the PA algorithm in which the variable parameter temperature is replaced with the variable parameter energy [12]. This algorithm is based on the program code for GPU from paper [10] and demonstrates the possibility of catching the models' non-equilibrium properties. We are working on the combined GPU/MPI implementation with the spirit of the block PA algorithm.

References

[1] P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev, J. Phys. Conf. Ser. 1740, 012050 (2021)

[2] K. Hukushima, Y. Iba, AIP Conf. Proc. 690, 200 (2003)

[3] J. Machta, Phys. Rev. E 82, 026704 (2010)

[4] W. Wang, J. Machta, and H.G. Katzgraber, Phys. Rev. E 92, 063307 (2015)

[5] A. Barzegar, C. Pattison, W. Wang, and H.G. Katzgraber, Phys. Rev. E 98, 053308 (2018)

[6] H. Christiansen, M. Weigel, and W. Janke, Phys. Rev. Lett. 122, 060602 (2019)

[7] L. Yu. Barash, M. Weigel, L.N. Shchur, and W. Janke, Eur. Phys. J. Spec. Top. 226, 595 (2017)

[8] A. Askarzadeh, L. Coelho, C.E. Klein, V.C. Mariani, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), DOI: 10.1109/SMC.2016.7844961

[9] M. Weigel, L. Barash, L. Shchur, and W. Janke, Phys. Rev. E 103, 053301 (2021)

[10] L.Yu. Barash, M. Weigel, M Borovsky, W. Janke, and L. Shchur, Comp. Phys. Comm. 220, 341 (2017)

[11] A. Russkov, R. Chulkevich, L. Shchur, Comp. Phys. Comm. 261, 107786 (2021)

[12] N. Rose and J. Machta, Phys. Rev. E 100, 063304 (2019)