Parallelization of simulations of the Monte Carlo type: 3D Ashkin-Teller Model

Grzegorz Musiał and Lech Dębski

Faculty of Physics, Adam Mickiewicz University
Ul. Umultowska 85, 61-614 Poznań, POLAND
E-mail: gmusial@amu.edu.pl
http://main2.amu.edu.pl/~g.musial/

G. Musiał, L. Dębski, GRID’2006
Outline

1. Motivation
2. Statistical mechanics and modeling
3. Monte Carlo simulations
4. Parallelization of simulations
5. Tests of our parallelized program
6. Conclusions
Motivation

Increasing availability of computing power, but atomized!

Physical problems in bigger and bigger systems.

Distributed Memory System – every processor has its own local memory which can be accessed directly only by its own CPU. Transfer of data from one processor to another is performed over a network.
Motivation

Modeling:

Numerical simulations

Nature observation

Practical verification (might be expensive!)

Theoretical explanation

The only true picture of a phase transition exist in the thermodynamic limit.

Thus the larger the samples considered, the better the analysis of the results.

Sometimes simulations are the only tool...

G. Musiał, L. Dębski, GRID’2006
Statistical mechanics and modeling

**Statistical mechanics** – description of the behavior (often irreversible) of macroscopic system ($10^{23}$ deg. of f.) deduced from **reversible** laws of molecular dynamics.

Chaos in microscopic scale → Order in macroscopic scale

In practice we have $N$ molecules in the volume $V$ and constant energy, i.e. between $E$ and $E+\Delta$, $\Delta \ll E$. Fulfilled by practically **infinite number of microstates**. In each of these states the system is with the same probab. These copies of our system form **microcanonical ensemble**.

The problem: it is not easy to control energy of a system!
Statistical mechanics and modeling

More practical: our system is in thermal equilibrium with a huge heat-bath at temperature $T$, then

$$p_\alpha = \frac{1}{Z} e^{-\beta E_\alpha}$$

with

$$Z = \sum_\alpha e^{-\beta E_\alpha}$$  

(partition function)

is the Gibbs distribution of the probability of the system being found in the microstate $\alpha$ (here $\beta=1/k_B T$).

Thus we can calculate the mean value of any property $X$:

$$\langle X \rangle = \sum_\alpha p_\alpha X_\alpha = \left(\frac{1}{Z}\right) \sum_\alpha X_\alpha e^{-\beta E_\alpha},$$

if we know $X_\alpha$ and $E_\alpha$. Canonical ensemble – simple and universal connection between laws in micro- and macro-world.
Statistical mechanics and modeling

The assumption that **thermal averages are equal to mean values calculated over a canonical ensemble** is always true, but ...

When we consider the **simple Ising model** on a square lattice $3 \times 3$ we have only $A=2^9=512$ microstates. But taking even very small lattice $10 \times 10$ we have $A=2^9 \approx 1.3 \cdot 10^{30}$ and calculation takes hundreds times the age of the Universe for the the fastest supercomputers.
Statistical mechanics and modeling

Of course: $A$ has to grow up exponentially, the entropy to be extensive quantity. So we have to pick up the method which allows to calculate thermodynamic quantities on the grounds of a small subset of microstates of a system.

To present and to test the parallelization method: the Ashkin-Teller (AT) model – nontrivial generalization of the Ising model:

$$-\frac{H}{k_B T} = \sum_{[i,j]} \{ K_2 (s_i s_j + \sigma_i \sigma_j) + K_4 s_i \sigma_i s_j \sigma_j \}$$

2 indep. Ising degrees of freedom $s, \sigma=\pm 1$ on each site, only nn. interact. with energy $J, K=-J/k_B T$. 

G. Musiał, L. Dębski, GRID’2006
The interesting and complicated phase diagram of the AT model.

+'s denote the testing points.

A, F, G, H, K and K' – the tricritical points, \( F_b \) – the bifurcation point.
Monte Carlo simulations

Picking up our small subset of microstates of a system, one should avoid the simple sampling of states:

in our simple square $3 \times 3$ of Ising spins only 7 from 512 states contribute 99.9% to $Z$ at $K=1.1$, and at $K=2.2$ only 2 states are enough. Importance sampling is necessary.

The method of parallelization will be presented using our program based on a single-flip Metropolis algorithm chosen because of its simplicity and typical structure.

More complicated algorithms reducting the critical slowing down problem can be found in [F. Wang, D.P. Landau, Phys. Rev. E64, 056101 (2001)].
Monte Carlo simulations

Using the Metropolis algorithm, we have generated equilibrium microstates (i.e. configurations of spins in the system) of the finite-size cubic spin samples of the size $L \times L \times L$ for fixed values of the model parameters. Periodic boundary conditions were imposed.

Thermalization of the initial microstates of the length of order $10^6$ Monte Carlo steps (MCS) was applied.

We calculate the change of the energy of a system at a single spin flip from the old state $\alpha$ to the new one $\alpha'$. We always accept this flip, when the change is negative, but when it is positive, it is accepted with the Gibbs probability $\exp[-\beta(E_{\alpha'} - E_{\alpha})]$ for what the 64-bit random number generator was used.
Monte Carlo simulations

It was sufficient to avoid about \( l=10 \) states before taking the next state for calculation of thermodynamic quantities to reduce enough the autocorrelation function along the path of changes of states.

A MC run was split into \( k \) segments consisting of about \( 10^6 \) MCS. From every segment one partial average of each measured quantity was calculated to estimate the statistical scattering of the results.
Parallelization of simulations

Several factors stimulate evolution of computing systems towards parallel ones:
– the finite speed of light
– effectiveness of heat dissipation
impose physical limits on the speed of a single computer. The cost of an advanced single-processor computer increases more rapidly than its power of computing. Step by step growth of wide-area networks.

MPI library to parallelize the computational process – effective and universal in every computer system with the distributed, shared or mixed type of a memory.
Parallelization of simulations

Some advantages of the message-passing model:
- *universality*, it fits well processors connected by fast or slow communication network;
- *expressive in parallel algorithms* and they are *debuggable*;
- using *more (distributed) memory* and cache this model gives the better performance of a system.

... and *MPI library advantages*:
- standardization when formulating its new versions;
- only a specification, not an implementation like PVM;
- portability for Fortran, C and C++ programs (by ordinary compilers and only linked with the MPI library);
- ensures both efficiency and functionality.
Parallelization of simulations

Crucial points in parallelization of processing:
– independent driving the system to the thermodynamic equilibrium on each of the $p$ parallel processes;
– different processes of the parallelized job calculate their parts of $k$ partial averages of measured quantities.

How many random numbers uses the algorithm when calculating 1 partial average?

The answer: $m*l*L^3$, where $m$ is the number of MCS used to calculate 1 partial average, $l$ is the number of MCS to be omitted before taking the next state for calculation of measured quantities, as mentioned above.
Parallelization of simulations

One more crucial point in parallelization of processing:
– every parallel processes should use completely different set of (pseudo)random numbers to obtain statistically independent partial averages of measured quantities.

Speedup $u$ of such a job is defined as $u = \frac{t_{ser}}{t_{par}}$.

Let us assume that $k = i*p$ – approximately the same amount of computational work for each parallel process, and SMP computer system (i.e. low latencies), then

$$u = \frac{t_0 + ipt_1}{t_0 + it_1},$$

$t_0$ – the computing time for leading the system to the thermod. equilibrium, $t_1$ – for one partial average.
Parallelization of simulations

Usually \( t_0 = t_1 \), then for \( u \) ... and for efficiency \( E = u/p \)

\[
  u = \frac{1 + ip}{1 + i}.
\]

\[
  E = \frac{1 + ip}{(1 + i)p}.
\]

When the number \( i \) of partial averages per process is \( i \gg 1 \), we obtain the ideal speedup \( u = p \) and efficiency \( E = 1 \).

Summarizing, the organization of the parallel program:

– produce \( r*i*m*l*L^3 \) random numbers, \( r \) – rank of a p. p.,
– each parallel process runs \( m*l*L^3 \) MCS (thermalization),
– each parallel process calculates \( i \) partial averages,
– 0\(^{th}\) process collects all partial averages, calculates, prints the results and their standard deviations.

G. Musiał, L. Dębski, GRID’2006
Tests of our parallelized program

Our program calculates: the Binder and Challa cumulants from partial averages of the 2\textsuperscript{nd} and 4\textsuperscript{th} moments of an order parameter and of the internal energy for spins $s$, $\sigma$ and $s\sigma$ [G. Musiał, Phys. Rev. \textbf{B69}, 024407 (2004)].

For a test, two points in the critical region of the 1\textsuperscript{st} order and continuous phase transitions were exploited. Parameters of the simulations were fixed: $L=16$ and $k=90$.

The test was carried out on SGI Origin 3800 multiprocessor of NUMAflex\textsuperscript{TM} architecture with R12000 processors.

A similar results on the SUN multicomputer (units with two dual-core AMD64 Opteron 2GHz CPUs), only times should be decreased by a factor 0.27.
Tests of our parallelized program

The results were obtained on SGI Origin 3800 multiprocessor.

Runtimes $t$ and total CPU times $t_{CPU}$ are in $10^3$s, SUN multicomputer – the factor 0.27.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$i$</th>
<th>$t$</th>
<th>$u$</th>
<th>$t_{CPU}$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>178.8</td>
<td>1.0</td>
<td>178.8</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>86.9</td>
<td>2.0</td>
<td>173.7</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>58.6</td>
<td>3.0</td>
<td>175.6</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>36.1</td>
<td>5.0</td>
<td>180.2</td>
<td>1.00</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>30.2</td>
<td>5.9</td>
<td>181.4</td>
<td>0.98</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>20.8</td>
<td>8.6</td>
<td>187.1</td>
<td>0.96</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>18.9</td>
<td>9.4</td>
<td>189.0</td>
<td>0.94</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>13.3</td>
<td>13.4</td>
<td>199.4</td>
<td>0.89</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>11.4</td>
<td>15.6</td>
<td>205.0</td>
<td>0.87</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>7.7</td>
<td>23.3</td>
<td>228.3</td>
<td>0.78</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>5.8</td>
<td>30.9</td>
<td>257.6</td>
<td>0.69</td>
</tr>
</tbody>
</table>
Tests of our parallelized program

Dashed line – the ideal speedup $u = p$

Amdahl's law? NO!

The reduction of the speedup – time used for leading the system to the thermodynamic equilibrium.
Conclusions

1. To have the best speedup and efficiency one should keep the number of partial averages per process $i \gg 1$.
2. Using the message-passing model and the MPI library ensures efficiency, full portability and functionality of an application in any distributed system, including grids and relatively cheap Ethernet networks of PCs.
3. The parallelization method may be applied to many types of simulations depending on the size of a system.
4. The method works well in completely different situations on the phase diagram.
5. Using more and more CPUs, one can consider larger physical systems, leading to increasingly credible results.
Conclusions

Acknowledgements:
The work was supported in part by the Polish Ministry of Science and Higher Education under Grant 4 T11F 014 24 and the European Commission under the project MAGMANet NMP3-CT-2005-515767.

Numerical calculations were mainly carried out on the platforms of the Poznań Supercomputing and Networking Center.

Thanks for your attention and patience!!!