DOI: 10.14529/jsfi220303

Scalability and Performance of a Program that Uses Domain Decomposition for Monte Carlo Simulation of Molecular Liquids

Alexander V. Teplukhin[1]

The main factors hindering the development of supercomputer programs for molecular simulation by the Monte Carlo method within the framework of classical physics are considered, and possible ways to eliminate the problems that arise in this case are discussed. Thus, the use of molecular models with moderate stiffness of covalent bonds between fragments makes it possible not only to increase the efficiency of scanning the configuration space of the model, but also to abandon the complex apparatus of kinematics with rigid links, which significantly limits the possibilities of domain decomposition. Based on the domain decomposition strategy and a simplified treatment of the deformation energy of covalent bonds and angles, an original parallel algorithm for calculating the properties of large all-atomic models of aqueous solutions of biopolymers by the Monte Carlo method was developed. To speed up computations within the framework of this approach, each domain is assigned its own group of processors/cores using local data replication and splitting the loop over the interacting partners. The article discusses the logical scheme of the computational algorithm and the main components of the software package (fortran77, MPI 1.2). Test calculations performed for water and n-hexane demonstrated the high performance and scalability of the program in which the proposed algorithm was implemented.

Keywords: parallel calculation, biopolymers, Monte Carlo, MPI.

Introduction

Stochastic Monte Carlo algorithms [19] are successfully used to solve a wide range of problems in many areas of science and engineering [9, 13, 16]. So, when studying the physicochemical and structural properties of molecular aggregates containing hundreds or more atoms, calculations are practiced within the framework of classical physics based on the mathematical apparatus of Markov chains [2, 6, 18, 34]. Due to the statistical nature of this type of computational experiments, it takes weeks, and sometimes months, to process huge amounts of data in order to achieve acceptable accuracy of the results. Worse, the size of the modeled object is strictly limited from above by the amount of RAM in a typical computer. It is clear that the study of models represented by millions of atoms (for example, biopolymers in an aqueous solution) should be carried out on a supercomputer using a program code that implements distributed computing by hundreds of processor cores.

At first glance, it may seem that, unlike the molecular dynamics method [2], algorithms that generate a Markov chain in the configuration space of the model under study (the Metropolis et al. procedure [2, 18]) have an extremely low potential for parallelization. Indeed, in the first case, all particles of the model simultaneously change their position at each iteration, while in the second, usually only one, and even then with a probability of no more than 50%[2]. Respectively, in the first case, parallelization can be organized by splitting the loops both over particles and over their interaction partners [22], and in the second case, only over interaction partners. Nevertheless, the success of using the Monte Carlo method in statistical physics [16] and polymer science [17] gives a serious stimulus to find solutions to this problem.

[1]Institute of Mathematical Problems of Biology of RAS - the Branch of Keldysh Institute of Applied Mathematics of RAS, Pushchino, Russian Federation

[2]Cluster and rejection free algorithms do not greatly improve the situation.

A necessary condition for the parallelizability of an algorithm is the presence of data-independent operations in it. It turned out that there is a very important class of models for which the steps of the procedure of Metropolis et al. [2, 18] will have this property if a restriction is introduced on the radius of interaction of their components. In this case, very often the connection of the previous and subsequent states of the generated Markov chain will have a formal character, since the values of the parameters characterizing one of these states do not affect the probability of transition to the second and vice versa. Using information about the interactions in the model, one can significantly reduce the time of calculations by organizing the simultaneous execution of calculations for independent groups of components. For example, in the 2D Ising model, the spins do not interact along the diagonals of the lattice. Consequently, the entire system of spins can be divided into two subgroups like a chessboard [21] and the states of all spins can be updated either in one or the other group in turn. Such a scheme is highly scalable and allows efficient use of computing resources [23].

The transition to continual (off-lattice) models of condensed phases requires the use of more complex grouping methods. Thus, in [10], a scheme was proposed according to which the space of the system being modeled (a rectangular cell with periodic boundary conditions) is divided into cubic domains, and each domain into eight more subdomains (cube octants). The length of the subdomain edge must be greater than the radius of the sphere that limits the interactions of particles in the model. This scheme ensures the independence of the movement of particles in similarly oriented domain octants. In the simplest case, the number of processors[3] used for calculations is equal to the number of domains, so that the processors perform the procedure of Metropolis et al. [2, 18] with particles of their domains selected from 'active' octants, looping through in the same sequence for all domains. Note that each processor 'sees' only particles in its own domain and in the boundary regions of 26 neighbors.

The domain decomposition strategy allows for a variety of practical implementations. In [10], the authors focused on optimizing the use of memory by proposing an algorithm with a very complex protocol of interprocessor communication, which adversely affected the speed of calculations. After Y2K, the problem of lack of memory lost its acuteness, which gave us the opportunity to develop a more advanced algorithm, significantly reducing the intensity of interprocessor data exchanges [14, 25].

Theoretically, spatial (domain) decomposition is capable of providing high scalability of computational algorithms by dividing a 'big problem' into several simultaneously soluble small subtasks. However, within the framework of this method, it is possible to reduce the size of domains only to a certain value (the cutoff radius of particle interactions). On the contrary, splitting the loop over particles interaction partners is very useful when studying objects of small and medium size, but it does not scale well and quickly exhausts the resources of the computer's RAM as the model size grows. The combination of these approaches makes it possible to 'neutralize' their limitations and develop efficient supercomputer programs capable of outperforming popular packages using the molecular dynamics method when studying models consisting of hundreds of millions of atoms. As a result of further modifications, we developed a new algorithm for calculating the properties of simulated systems in an isobaric-isothermal (NpT) ensemble [2], in which each domain is serviced by its own group of processors [26]. The program built on the basis of this algorithm was successfully used in the study of the structural and thermodynamic characteristics of water in a wide range of pressure [29, 30].

[3]Hereinafter, the word 'processor' will mean a separate processor core.

Another obstacle that generates skepticism about the Monte Carlo method and limits its application in the applied fields of materials science and biomedicine is the problem of taking into account intramolecular degrees of freedom. As it was shown in 1980 [20], the procedure of Metropolis et al. [2, 18], applied to individual atoms of a peptide molecule, gives 11 times less diffusion of atoms than the molecular dynamics method for the same amount of computer time. The reason for this inefficiency is that random monatomic displacements in a covalently bound system lead to large changes in the strain energy of bonds and angles, which significantly exceed the contributions of van der Waals and electrostatic interactions.

The standard way out of this situation is based on the idea of excluding from consideration the stiff degrees of freedom of molecules [8] by 'freezing' covalent bonds and the angles between them (partially or completely), as well as through the use of united atoms [12] and coarse-grained models of polymers. [24]. Unfortunately, the kinematics of displacements of atoms (groups of atoms) in such models is implemented through laborious procedures that use difficult mathematical calculations, sometimes requiring the solution of systems of nonlinear algebraic equations. Worse, some transformations [33] cannot be performed in parallel computations if the moved atoms are in different domains.

To increase the effectiveness of Monte Carlo trial moves, we [27] proposed to completely eliminate the contribution of the strain potentials of covalent bonds and angles to the energy increment, which is taken into account in the procedure of Metropolis et al. [2, 18]. Within this approach, a molecule must be considered either as one or several rigid fragments (compact, as a rule, π-conjugated groups of atoms) connected by single covalent bonds with zero stiffness and finite extensibility. Hydrogen and halogen atoms should be included in the fragment to which they are bonded. Some fragments may consist of a single atom, such as an ether oxygen atom or a quaternary carbon atom. From a mechanical point of view, fragments are solid bodies that perform rotational and translational motions as a whole, which position in space is determined by the Cartesian coordinates of their center and three Euler angles. The integrity of the chemical structure of the molecule is maintained by simply checking the bond lengths and angles for compliance with the standard ranges of values [1] for specific types of neighboring atoms. In the course of subsequent computational experiments, it turned out that such drastic measures with respect to strain potentials are not necessary at all [31]. In fact, the moderate stiffness of interfragment single bonds and angles not only improves the shape of the distribution of bond lengths and angles, but also makes it possible to increase the maximum amplitude of trial displacement and fragment rotation. It should be noted that this approach showed very good results in calculating the structural and thermodynamic characteristics of liquid hydrocarbons [31, 32].

The purpose of this work is to study the performance and scalability of the program we developed (fortran77, MPI 1.2), which uses domain decomposition together with the splitting of loop over fragments interaction partners for calculations based on a model with a simplified treatment of the strain energy of covalent bonds and angles [27, 31, 32], mentioned in the previous paragraph.

The article is organized as follows. In Section 1 the data on the molecular models used in the calculations will be presented. Section 2 is devoted to a description of a new algorithm for parallel computing by the Monte Carlo method, as well as the main components of a prototype[4] software package for studying aqueous solutions of biopolymers. The Section 3 discusses the

[4]In addition to the program code, it is necessary to prepare a set of force field parameters that is adequate to the model we use and allows us to accurately reproduce the properties of substances in the condensed phase.

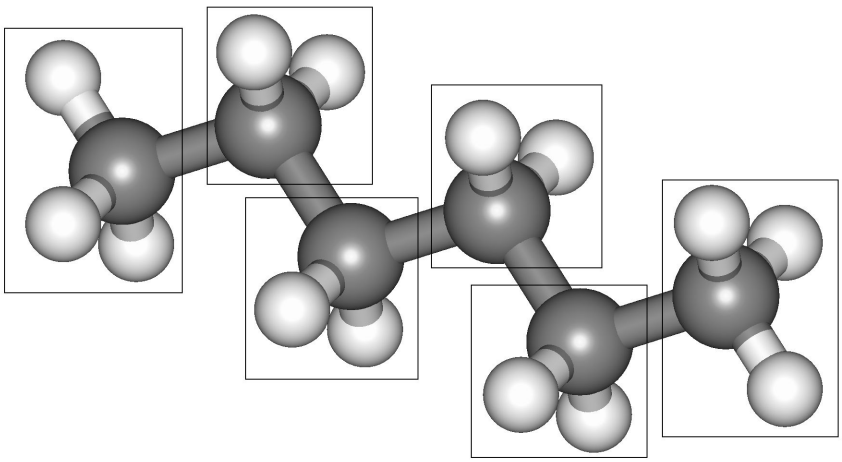

Figure 1. n-hexane molecule. Hydrogen atoms are light gray; carbon atoms are dark gray. Rigid fragments of the molecule are outlined by rectangles

results of computational experiments with all-atomic models of liquids of two kinds: n-hexane and water. Conclusion summarizes the study and points directions for further work.

1. Systems under Consideration

The model of the n-hexane molecule is a system of configurationally rigid fragments (CH_3–$(CH_2)_4$–CH_3: four CH_2 groups and two CH_3) connected by single bonds in accordance with the chemical structure of this molecule (Fig. 1). The CH bond length is 1.085 Å, and the HCH bond angles are 109.47°. The potential energy characterizing the conformation of the n-hexane molecule is acquired due to non-covalent interactions of atoms of its fragments (short-range approximations of the Lennard-Jones and Coulomb potentials, see below), the deformation (strain) energy of interfragment covalent bonds and bond angles adjacent to them, as well as torsion potentials, hindering the rotation of molecular fragments around these bonds. The parameters of the Lennard-Jones potential functions are the same as in [31], the partial charges on atoms and the parameters of torsion potentials are taken from [32]. The strain energy is proportional to the squared deviation of bond lengths and angles from their equilibrium values with coefficients of 100 kcal/mol/Å^2 and 25 kcal/mol/rad^2, respectively. For the equilibrium lengths of C–C chemical bonds, the value of 1.526 Å was taken, and for the angles, 109.47°. This choice was largely motivated by the parameters of the force fields presented in articles [5, 11]. Atoms of neighboring fragments, connected by a covalent bond with each other or through a third atom, do not participate in the interactions described by the Lennard-Jones potential.

After a trial move (displacement and rotation) of the selected fragment, new values of the lengths of covalent bonds and angles are calculated, in the formation of which this fragment participates. If any bond or angle is not within the allowed range, the remaining bonds and angles are not checked, the trial move is rejected, the state before the move is again taken as the new state of the Markov chain, and the program continues to the next fragment. If all stereochemical parameters of the proposed configuration remained within the acceptable limits (±0.12 Å for bond lengths and ±12° for angles), the energy change and then the transition probability are calculated according to the formula of Metropolis et al. [2, 18]. The maximum displacement and rotation angle (for each coordinate and angle) is determined in a preliminary computer experiment so that ∼38–43% of the trial configurations are accepted.

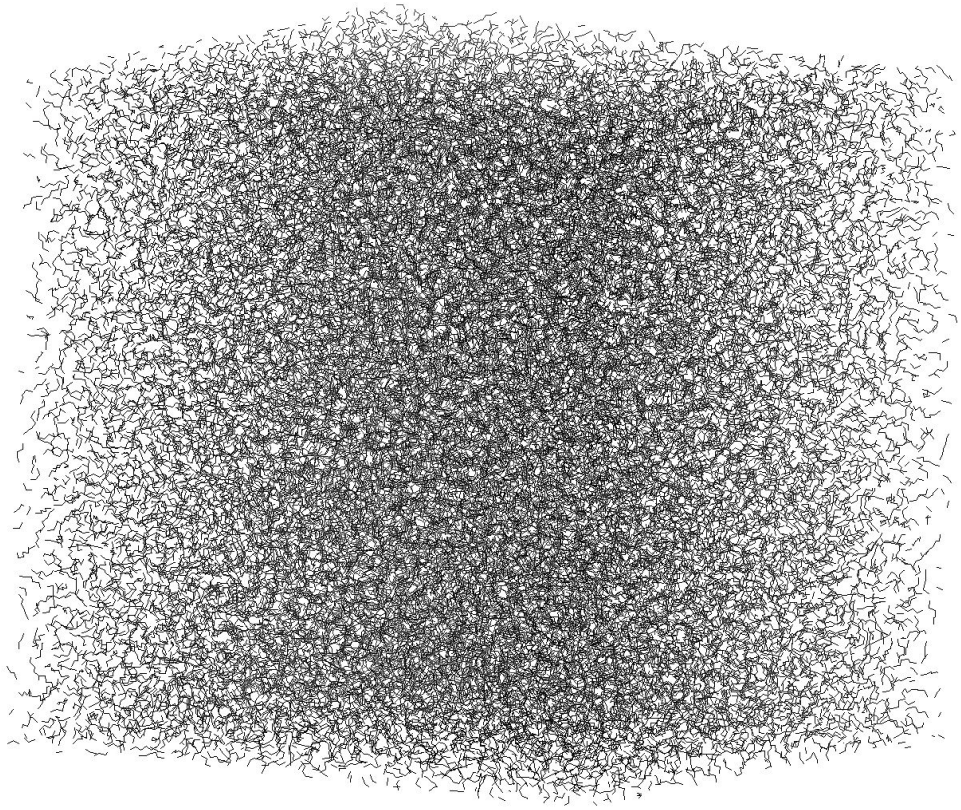

Figure 2. View of the main cell of the simulated system containing 19200 n-hexane molecules (384000 atoms). Hydrogen atoms not shown

To study water systems, we used the rigid three-center model of the water molecule SPC/E [4], which is an example of a molecule consisting of a single fragment.

Three simulated systems contained 2400, 19200 and 64800 n-hexane molecules each (48000, 384000 and 1296000 atoms, respectively), and three more – 33666, 269328 and 908982 water molecules (100998, 807984 and 2726946 atoms, respectively) in the main cubic cell with periodic boundary conditions (Fig. 2). The energy of intermolecular interactions was calculated using short-range pairwise additive atom-atom potential functions ($R_{cutoff} = 14$ Å, standard correction for the Lennard-Jones potential [2] and modification PF3 [28] for the Coulomb potential), which effectively take into account the contributions of all images of the main cell under periodic boundary conditions.

The initial configuration for each system is prepared using a special program that uses as many processors as there are domains in the simulated object. This program randomly places given amount of molecules within the boundaries of the main cell, taking into account periodic boundary conditions. The trial position and orientation of the molecule inserted is randomly set by the main processor (if the molecule is very large, the process is performed in batches). It also determines which domain a particular fragment falls into and sends the coordinates to the addressee. Upon completion of the distribution of fragments of the molecule intended for insertion, all processes determine, using a geometric criterion, whether there is an overlap of atoms with previously received fragments. The results of the check (0 or 1) are collected for all processes (MPI_ALLREDUCE, MPI_SUM). If 0 is received the attempt is accepted, otherwise it is retried. If the number of attempts exceeds the specified number, the program stops. To prevent this from happening on subsequent launches of the program, it is necessary to increase

the number of attempts or the volume of the main cell. If the program manages to fit all the molecules, each process writes all the information about its domain to a separate file on an external storage.

Obviously, the initial density of the simulated object will be much less than the real values corresponding to the given temperature and pressure. The relaxation of the studied models to the equilibrium state was carried out using the same program that is intended for calculating the structural and thermodynamic characteristics. Depending on the type of molecules, this requires 5–20 million trial moves per fragment. The main indicator in this process is the absence of a systematic trend in the batch-mean values of the density and heat of evaporation of the simulated liquids. To accelerate relaxation, the initial stages were performed at a pressure increased to 1 kbar.

The results were obtained using the equipment of Shared Resource Center of KIAM RAS (http://ckp.kiam.ru).

2. Computational Algorithm and Main Components of the Program

The simulated system occupies the main cell of a cubic shape, on which periodic boundary conditions are imposed. The volume of the main cell is divided into $N{=}n^3$ domains (n pieces along each coordinate axis). The computational task is submitted on $M{=}m{\cdot}n^3$ processors (processor cores), i.e. each domain is serviced by m processors (hereinafter referred to as the domain group of processors). Data about the simulated system are read from n^3 files on external media (each domain has its own data file) and written back to them after the completion of the next portion of the calculation. At startup, the head processor of each domain group reads data from the corresponding file and sends (MPI_BCAST) a copy of it to the rest of the processors in its group (replication). Processes of the same rank in a domain group are combined into corresponding communication groups with a 3D Cartesian topology (MPI_CART_CREATE). The algorithm described in [35] was used to generate pseudo-random numbers.

The actual computational procedure (see the logical scheme of the Algorithm 1) consists of two blocks of program code in the cyclic execution. The first block is responsible for performing an attempt, standard for the NpT ensemble [2], to randomly change the volume occupied by the model. To do this, one of the processors (rank 0 in MPI_COMM_WORLD) generates a new volume logarithm value and sends it (MPI_BCAST) to all processors. Then, domain group processors calculate trial values of atomic coordinates and exchange data with their topological neighbors (MPI_CART_SHIFT, MPI_SENDRECV). As a result, each processor receives information about the coordinates of atoms in the border zone from the side of the neighbor. Next, the processors calculate their portions in the change in the total energy of the system, and also, using simple geometric criteria, identify the partners in non-covalent and covalent interactions. All processors of each domain group, in parallel, carry out these operations by splitting of the corresponding loops over atoms. Upon completion of this, the processors of each domain group exchange data on the found pairs of neighbors (MPI_ALLGATHER), and all calculated energy increments are summed (MPI_REDUCE) with the result transferred to one of the processors (rank 0 in MPI_COMM_WORLD). It is this processor that decides (Metropolis et al. criterion [2, 18]) whether to accept the new volume value or keep the old one, and then sends the result to all M processors.

```
Initialization, input previously saved data;
while NStep ≠ 0 do
    //Begin block 1;
    Try to accept a new volume value;
    Generate random vector to shift domain-splitting grid;
    Generate random permutation for 8 subdomains selection order;
    //End of block 1;
    if Processor rank is 0 then
        Accumulate the data required to obtain averaged values of the density and the
         enthalpy of the simulated liquid;
    end
    //Begin block 2;
    I ← 2;
    while I ≠ 0 do
        if I = 2 then
            Set direct order of 8 subdomains selection;
        else
            Set the selection in reverse order;
        end
        Try to change the position of a randomly selected fragments in 'active'
         subdomains;
        I ← I − 1;
    end
    //End of block 2;
    NStep ← NStep − 1;
end
Output data to next run;
```

Algorithm 1. The logical scheme of the computational algorithm

The second block is responsible for performing an attempt to randomly change the position (shift and rotation) of a randomly selected rigid fragment. This operation is performed by the head processors of each domain group in parallel in their domain. To ensure the statistical independence of n^3 simultaneous trial moves, each domain is divided into 8 (2×2×2) subdomains and moved molecules are taken from the subdomains in the same octant for all domains. This construction ensures that the selected fragments do not turn out to be partners in interactions, since all potential functions have a limited range of action, not exceeding half the length of the edge of the domain cell. Next, each head processor calculates the increment in the strain energy of the covalent bonds, bond and dihedral angles, and sends trial coordinates to the domain group. The processors of each domain group calculate the increment in the energy of non-covalent interactions from their portion of the neighbors of the tested fragment and 'drop' the results to the head processor of the group. After that, the head processor, having added the previously calculated 'covalent' contribution, decides (Metropolis et al. criterion [2, 18]) whether to accept the new position of the fragment or keep the old one, and then sends the result to all m processors of its group and proceeds to select the next fragment in the same subdomain. In practice, the loop over subdomain fragments should be long enough that on average each

fragment is selected 25 times. In this case, the radius of the criterion for inclusion in the list of interaction partners should be increased (R_{neib}=R_{cutoff}+R_{fr}, where R_{fr} is the radius of the minimum sphere containing all atoms of the largest fragment) by 1.5–2.5 Å (buffer zone). Before moving on to the next subdomains, the processors send to the topological neighbors the coordinates of the fragments from the subdomains that have just completed their 'activity'. After fragments in all subdomains have been 'processed' in this way, the second block is completed, control is transferred to the beginning of the first block, and so on. To calculate the desired structural and thermodynamic characteristics, it is convenient to use the instantaneous configuration of molecular fragments formed before the start of the first block. Having performed the procedure described above for a specified number of times ($NStep$), the head processors write the information necessary for restart or further analysis to n^3 files on an external storage and the program stops.

An attentive reader may notice that in the above description of the logical scheme of the computational algorithm, some of the directly or indirectly mentioned episodes were left without discussion. Some of them have been omitted in order to reduce the size of the article. For example, these are procedures for transferring data between processors in neighboring domains, which can be implemented in different ways, depending on the preferences of the developer. The other, on the contrary, require separate consideration due to their importance. These include the problem of maintaining a detailed balance of transitions in the Markov chain while simultaneously performing the Metropolis et al. procedure [2, 18] in each domain. Recommendations on how to avoid violations of the detailed balance, as well as to improve the ergodicity of scanning the configuration space of the simulated systems, are given in the articles [3, 33].

Following these recommendations, our algorithm introduces a displacement of the boundaries of the system of domains by a random vector (the maximum displacement in each coordinate does not exceed half the length of the subdomain edge), as well as by random shuffling [7] of the order of selection of eight subdomains in the second block. The coordinates of this vector, as well as the sequence of subdomain numbers, are generated by the processor of rank 0 (in MPI_COMM_WORLD) and distributed (MPI_BCAST) along with the trial value of the volume logarithm to all processors at the very beginning of the first block. The second block is executed twice: first, in the forward order of the sequence of subdomain numbers, and then in the reverse order. The loop length over molecular fragments in a subdomain is equal to the number of fragments in it multiplied by 25. This ensures the (asymptotic) equality of the fragment selection frequencies regardless of local density fluctuations. Some unbalancing of the load of processors is compensated by a decrease in the number of potential interaction neighbors due to a decrease in the thickness of the buffer zone (R_{neib}). It should be noted that the fragments that crossed the subdomain boundary remain in the lists of the same subdomain in which they were at the time of the beginning of the second block. This guarantees the reversibility of the Markov chain transitions and ensures that the detailed balance is fulfilled [3, 33]. The determination of whether a fragment belongs to a particular subdomain occurs in the first block, after the boundaries of the entire system of domains have been moved.

Concluding this section, we note two important problems associated with the limited accuracy of the representation of floating point numbers in machine arithmetic. So, as a result of displacements, the length of the covalent bond between the atoms of neighboring fragments can be at the boundary established by the geometric criterion. In this case, it may turn out that the next time the criterion is applied the bond between them will be broken. To avoid this,

one should somewhat narrow (for example, by 0.0001 Å) the corridor of covalent bond lengths admissible as a result of trial moves. It is also necessary to reject trial configurations in which atoms that did not participate in the covalent bond are located at distances near the threshold of this criterion.

Another dangerous situation arises in parallel computing when it is required to determine which subdomain a fragment belongs to, i.e. to solve the problem of the position of its geometric center. In the case when the center is on the border of subdomains, the processors related to this fragment can find mutually contradictory solutions, which will result in the program crashing. To avoid this, the calculation algorithm should be organized in such a way that only the head processors of domain groups are involved in making such decisions, acting only on fragments in their subdomains. As a result of the trial move, the fragment may remain in the same subdomain or move to one of the 26 adjacent subdomains. To ensure conflict-free distribution of fragments across subdomains, it is enough to associate with each fragment a special index indicating the direction of the transition (from 1 to 27) and send it along with the coordinates. Conflict-freeness is guaranteed by the fact that the first time the fragments are distributed by single processor when preparing the initial configuration, and the control of their movement is performed by the head processors of domain groups, each in its own domain.

The computer codes (fortran77 + MPI 1.2) developed by author as well as instructions 'how to run' are available from the author on reasonable request.

3. Performance and Scalability of the Computational Algorithm

To verify the program and evaluate the performance and scalability of the underlying algorithm, two well-studied liquids were chosen: water and *n*-hexane. The calculation model of the first one is represented by molecules consisting of one fragment (H_2O), while the molecules of the second one are composed of six fragments connected via single covalent bonds (Fig. 1). For each substance, three variants of simulated systems were prepared and brought into thermodynamic equilibrium at a temperature of 298 K and atmospheric pressure, differing in the size and number of molecules in the main cell (see section 'Systems under Consideration'). Small-sized systems were divided into 8 (2×2×2), medium-sized systems into 64 (4×4×4), and large ones into 216 (6×6×6) domains. The edge length of the domain of aqueous systems was, on average, 50.15 Å, while that of *n*-hexane systems was 40.3 Å.

In the course of computational experiments, we measured the execution time for the task of calculating the density and heat of evaporation of the simulated liquid, which requires, on average, 5000 trial moves of the Metropolis et al. procedure [2, 18] per each fragment. For each system, a series of calculations was performed, gradually increasing the number of processor cores in domain groups in the following order: 1, 2, 4, 7, 14, 28, and 35. Due to limited computing resources (no more than 80 nodes equipped with 28 cores), no tests were performed on medium-sized systems for 35, and for large models for 28 and 35 cores in domain groups. Recall that $M{=}m{\cdot}N$ processor cores are required to run the test. Here N is the number of domains, and m is the number of processor cores in the domain group.

Figure 3 shows the dependence of the test execution time on the number of processor cores in the domain group on a logarithmic scale. Data for large systems (N=216 domains, m from 1 to 14) are not shown, as they are only a fraction of a percent higher than the corresponding

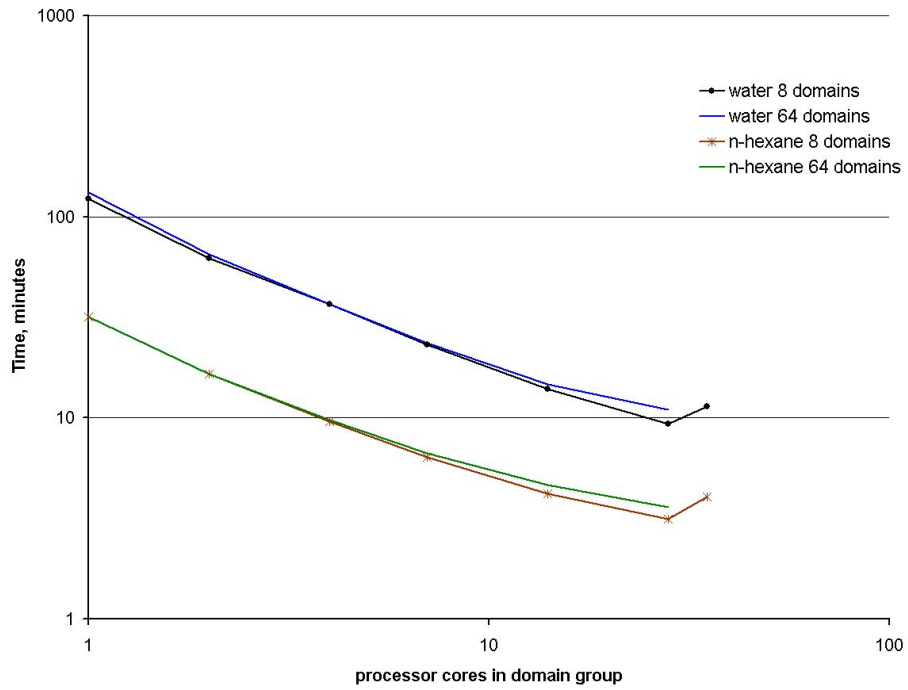

Figure 3. The test execution time for water and n-hexane models of small and medium size depending on the number of processor cores in the domain group. The symbols on the lines indicate the results for 1, 2, 4, 7, 14, 28 and 35 processor cores

medium sized systems. The difference in test execution time for small and medium systems is more noticeable and may be due to fewer data exchanges in the case of eight domains [14, 25, 26]. This figure clearly shows that after 28 a further increase in the number of processor cores in domain groups becomes impractical. Note that, obeying the Amdahls law, this effect is universal, since it practically does not depend on the kind of fragments and the presence of covalent bonds between them, as well as on the size of the domains (on average, the domain of the aqueous system contains 4208, and n-hexane – 1800 fragments) and the number of interaction partners for each fragment (for a water molecule, on average, about 1380, and for a fragment of an n-hexane molecule, 850). As it has been seen in Fig. 4, despite the fact that the Amdahls law limits the increase in performance due to the increase in the number of cores in domain groups, the tenfold acceleration of calculations is a very powerful incentive to use parallelization/splitting of the loop over interaction partners in addition to domain decomposition.

The most important result obtained in these computational experiments is the absence of a noticeable increase in the calculation time with an increase in the size of the simulated system, which is achieved by proportionally increasing the number of processor cores used. The curves in Fig. 3, which show the execution time of test calculations for systems of different sizes, begin to diverge noticeably only when large domain groups of processor cores are used.

As shown by additional calculations, in our model, the root-mean-square deviation of n-hexane atoms from the average position after performing 24000 trial moves (on average) for each fragment is 0.76±0.19 Å at T=306 K. In the calculations of the protein molecule by the Monte Carlo method, where parameters of the force fields relevant at that time, a significantly lower value of 0.082 Å was obtained for this characteristic [20]. In turn, it was noted that the rmsd of atoms of the same molecule reaches 0.75 Å after 10^5 time steps performed by the molecular dynamics method. At the same time, the rmsd of n-hexane atoms after the same

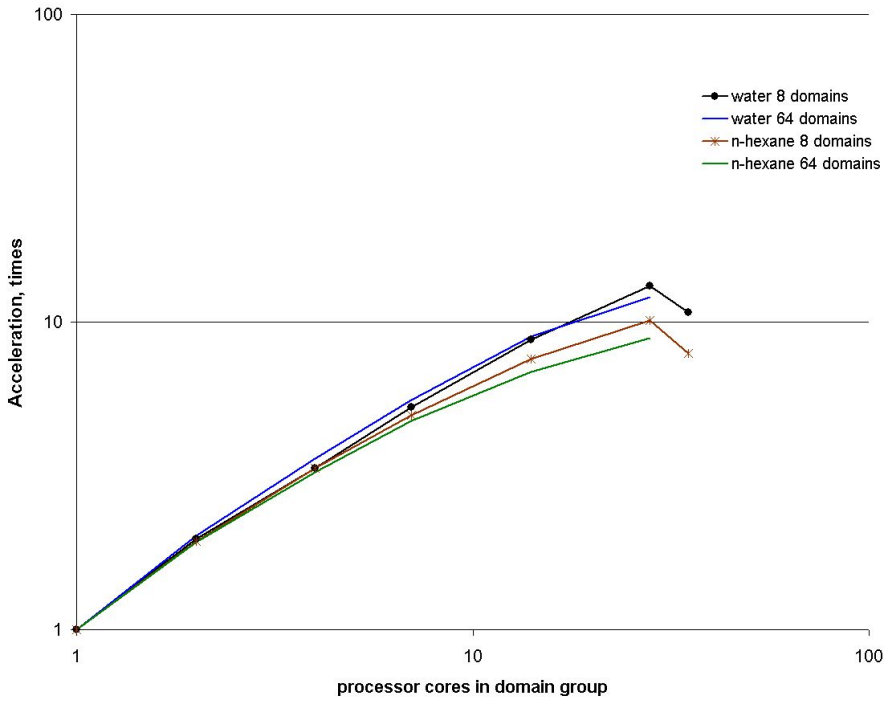

Figure 4. Acceleration of calculations of small and medium-sized water and n-hexane models depending on the number of processor cores in the domain group. The symbols on the lines indicate the results for 1, 2, 4, 7, 14, 28 and 35 processor cores

number of trial moves per each fragment is 1.36±0.37 Å in our model. It can be seen that the use of molecular models with moderate stiffness of covalent bonds between fragments makes it possible to surpass the molecular dynamics method in terms of the efficiency of scanning the configuration space of the model by increasing the displacement amplitude of atoms in one step of the computational procedure.

To date, several articles have been published that provide data on the performance of popular software packages. Thus, in the article [15] one can find data on the calculations of the F_1-ATPase system (327506 atoms), comparable in size to one of the systems we considered: the medium-sized model of liquid n-hexane (384000 atoms). As seen in Fig. 4 of the article [15], it takes 100, 65 and 35 ms to execute one time step by the NAMD2 package when using 256, 448 (interpolation) and 1000 processor cores on the IBM Blue Gene/L supercomputer. In our case, it takes 117.6, 80.3 and 55.7 ms to perform similar work for the medium-sized liquid n-hexane model using 256, 448 and 896 processor cores on the K60 supercomputer (https://ckp.kiam.ru). This comparison shows that even without deep, including low-level, code optimization, our approach ensures high performance of the program developed on its basis.

Concluding this section, we recall that the development of a highly efficient program for molecular Monte Carlo calculations requires solving two main problems: taking into account intramolecular degrees of freedom and parallelizing the calculations over the Markov chain underlying the computational algorithm. The first of them is proposed to be eliminated by dividing biopolymer molecules into rigid compact fragments connected by single covalent bonds of moderate stiffness. To solve the second one, domain decomposition in combination with splitting loops over lists of interaction partners is well suited. Note that the implementation of domain decomposition in the Monte Carlo method is fundamentally different from its implementation in the molecular dynamics method, where all atoms are displaced at one time step. We add that

the problem of applying domain decomposition in calculations of molecular systems using the procedure of Metropolis et al. [2, 18] is very poorly covered in the literature, so we believe that this article will provide new information useful for developers of supercomputer programs.

Conclusion

In this paper, we have considered the main factors affecting the performance of Monte Carlo calculations for large models of molecular aggregates, and also proposed effective methods for solving the identified problems. Thus, the use of domain decomposition and distributed computing on hundreds of processor cores not only eliminates the problem of lack of RAM, but also provides the possibility of simultaneous execution of the Metropolis et al. procedure [2, 18] in each domain. The minimum size of a domain is determined by the cutoff radius for interactions between atoms and the thickness of the buffer zone; therefore, each domain can contain several thousand atoms. To further speed up calculations in domains, it is necessary to split the loop over the list of interaction partners, assigning to each domain not one processor core, but a group of 7–14 cores, using data replication. This will further reduce the calculation time by an order of magnitude. The problem of slow diffusion in the configuration space of molecules with internal degrees of freedom [20] is proposed to be solved using a simplified treatment of the strain energy of covalent bonds and angles [27, 31, 32].

As shown by computational experiments with all-atom models of water and n-hexane, the strategy of parallelization of calculations proposed by us makes it possible to develop highly efficient programs for modeling biopolymers in an aqueous solution by the Monte Carlo method in the framework of classical physics.

References

1. Allen, F.H., Kennard, O., Watson, D.G., *et al.*: Tables of bond lengths determined by X-ray and neutron diffraction. Part 1. Bond lengths in organic compounds. J. Chem. Soc., Perkin Trans. 2 pp. S1–S19 (1987). https://doi.org/10.1039/P298700000S1

2. Allen, M.P., Tildesley, D.J.: Computer Simulation of Liquids. Cambridge University Press, New York (1987)

3. Anderson, J.A., Jankowski, E., Grubbb, T.L., *et al.*: Massively parallel Monte Carlo for many-particle simulations on GPUs. J. Comput. Phys. 254, 27–38 (2013). https://doi.org/10.1016/j.jcp.2013.07.023

4. Berendsen, H.J.C., Grigera, J.R., Straatsma, T.P.: The missing term in effective pair potentials. J. Phys. Chem. 91, 6269–6271 (1987). https://doi.org/10.1021/j100308a038

5. Cornell, W.D., Cieplak, P., Bayly, C.I., *et al.*: A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J. Am. Chem. Soc. 117, 5179–5197 (1995). https://doi.org/10.1021/ja00124a002

6. Dubbeldam, D., Torres-Knoop, A., Walton, K.S.: On the inner workings of Monte Carlo codes. Mol. Simul. 39, 1253–1292 (2013). https://doi.org/10.1080/08927022.2013.819102

7. Durstenfeld, R.: Algorithm 235: Random permutation. Commun. ACM 7, 420 (1964). https://doi.org/10.1145/364520.364540

8. Gō, N., Scheraga, H.A.: Analysis of the contribution of internal vibrations to the statistical weights of equilibrium conformations of macromolecules. J. Chem. Phys. 51, 4751–4767 (1969). https://doi.org/10.1063/1.1671863

9. Hammersley, J.M., Handscomb, D.C.: Monte Carlo Methods. Methuen, London (1964)

10. Heffelfinger, G.S., Lewitt, M.E.: A comparison between two massively parallel algorithms for Monte Carlo computer simulation: An investigation in the grand canonical ensemble. J. Comput. Chem. 17, 250–265 (1996). https://doi.org/10.1002/(SICI)1096-987X(19960130)17:2<250::AID-JCC11>3.0.CO;2-N

11. Jorgensen, W.L., Maxwell, D.S., Tirado-Rives, J.: Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. J. Am. Chem. Soc. 118, 11225–11236 (1996). https://doi.org/10.1021/ja9621760

12. Jorgensen, W.L., Tirado-Rives, J.: Monte Carlo vs molecular dynamics for conformational sampling. J. Phys. Chem. 100, 14508–14513 (1996). https://doi.org/10.1021/jp960880x

13. Kalos, M.H., Whitlock, P.A.: Monte Carlo Methods. Wiley-VCH, Weinheim (2008)

14. Kalugin, M.D., Teplukhin, A.V.: Study of caffeineDNA interaction in aqueous solution by parallel Monte Carlo simulation. J. Struct. Chem. 50, 841–852 (2009). https://doi.org/10.1007/s10947-009-0126-8

15. Kumar, S., Huang, C., Zheng, G., *et al.*: Scalable molecular dynamics with NAMD on the IBM Blue Gene/L system. IBM Journal of Research and Development 52(1-2), 177–188 (2008). https://doi.org/10.1147/rd.521.0177

16. Landau, D.P., Binder, K.: A Guide to Monte Carlo Simulations in Statistical Physics. Cambridge University Press, New York (2000)

17. Mavrantzas, V.G.: Using Monte Carlo to simulate complex polymer systems: Recent progress and outlook. Front. Phys. 9, 661367 (2021). https://doi.org/10.3389/fphy.2021.661367

18. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., *et al.*: Equation of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1092 (1953). https://doi.org/10.1063/1.1699114

19. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Statist. Assoc. 44, 335–341 (1949). https://doi.org/10.1080/01621459.1949.10483310

20. Northrup, S.A., McCammon, J.A.: Simulation methods for protein structure fluctuations. Biopolymers 19, 1001–1016 (1980). https://doi.org/10.1002/bip.1980.360190506

21. Pawley, G.S., Bowler, K.C., Kenway, R.D., Wallace, D.J.: Concurrency and parallelism in MC and MD simulations in physics. Comput. Phys. Comm. 37, 251–260 (1985). https://doi.org/10.1016/0010-4655(85)90160-2

22. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. 117, 1–19 (1995). https://doi.org/10.1006/jcph.1995.1039

23. Preis, T., Virnau, P., Paul, W., Schneider, J.J.: GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model. J. Comput. Phys. 228, 4468–4477 (2009). https://doi.org/10.1016/j.jcp.2009.03.018

24. Rader, A.J.: Coarse-grained models: Getting more with less. Curr. Opin. Pharmacol. 10, 753–759 (2010). https://doi.org/10.1016/j.coph.2010.09.003

25. Teplukhin, A.V.: Multiprocessor simulation of mesoscopic DNA fragments hydration. Matem. Model. 16(11), 15–24 (2004), http://www.mathnet.ru/links/037077e2238f8f9887f8fa8da9fae408/mm219.pdf

26. Teplukhin, A.V.: Parallel and distributed computing in problems of supercomputer simulation of molecular liquids by the Monte Carlo method. J. Struct. Chem. 54, 65–74 (2013). https://doi.org/10.1134/S0022476613010095

27. Teplukhin, A.V.: A simplified treatment of the deformations of covalent bonds and angles during the all-atom Monte Carlo simulation of polymers. Polym. Sci., ser. C 55, 103–111 (2013). https://doi.org/10.1134/S1811238213050044

28. Teplukhin, A.V.: Short-range potential functions in computer simulations of water and aqueous solutions. J. Struct. Chem. 57, 1627–1654 (2016). https://doi.org/10.1134/S0022476616080205

29. Teplukhin, A.V.: Monte Carlo simulation of the local ordering of water molecules. II. Spatial correlations and hydrogen bonds. J. Struct. Chem. 59, 1624–1630 (2018). https://doi.org/10.1134/S0022476618070144

30. Teplukhin, A.V.: Thermodynamic and structural characteristics of SPC/E water at 290 K and under high pressure. J. Struct. Chem. 60, 1590–1598 (2019). https://doi.org/10.1134/S0022476619100044

31. Teplukhin, A.V.: Monte Carlo calculation of thermodynamic and structural characteristics of liquid hydrocarbons. J. Struct. Chem. 62, 7082 (2021). https://doi.org/10.1134/S002247662101008X

32. Teplukhin, A.V.: Parametrization of the torsion potential in all-atom models of hydrocarbon molecules using a simplified expression for the deformation energy of valence bonds and angles. J. Struct. Chem. 62, 1653–1666 (2021). https://doi.org/10.1134/S0022476621110019

33. Uhlherr, A., Leak, S.J., Adam, N.E., *et al.*: Large scale atomistic polymer simulations using Monte Carlo methods for parallel vector processors. Comput. Phys. Comm. 144, 1–22 (2002). https://doi.org/10.1016/S0010-4655(01)00464-7

34. Vitalis, A., Pappu, R.V.: Methods for Monte Carlo simulations of biomacromolecules. Annu. Rep. Comput. Chem. 5, 49–76 (2009). https://doi.org/10.1016/S1574-1400(09)00503-9

35. Wichmann, B.A., Hill, I.D.: Generating good pseudo-random numbers. Comput. Statist. Data Anal. 51, 1614–1622 (2006). https://doi.org/10.1016/j.csda.2006.05.019