

УДК 004.021+519.6

**УСКОРЕНИЕ МОЛЕКУЛЯРНО-ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ  
НЕПОЛЯРНЫХ МОЛЕКУЛ ПРИ ПОМОЩИ GPU**

© 2014 г.

**В.Л. Мальшев,<sup>1,2</sup> Д.Ф. Марьин,<sup>1,2</sup> Е.Ф. Мусеева,<sup>1</sup>  
Н.А. Гумеров,<sup>1,3</sup> И.Ш. Ахатов<sup>1,4</sup>**<sup>1</sup>Центр микро- и наномасштабной динамики дисперсных систем  
Башкирского государственного университета, Уфа<sup>2</sup>Институт механики им. Р.Р. Мавлютова Уфимского научного центра РАН<sup>3</sup>Университет Мэриленда, Институт передовых компьютерных исследований,  
Колледж Парк, Мэриленд, США<sup>4</sup>Университет Северной Дакоты, Фарго, Северная Дакота, США

victor.L.malyshev@gmail.com

*Поступила в редакцию 20.02.2014*

Описывается применение структуры данных в молекулярно-динамическом моделировании неполярных молекул. Взаимодействие между атомами описывается потенциалом Леннарда–Джонса. Разработанная структура данных позволяет уменьшить вычислительную сложность алгоритма с квадратичной до линейной. Приведена схема реализации алгоритма для гетерогенной архитектуры, состоящей из CPU и GPU. Использование GPU позволяет достичь значительного ускорения вычислений. Показано, что описанная методика может быть использована для моделирования динамики неполярных молекул в области с характерными размерами в десятки нанометров на персональных суперкомпьютерах, оборудованных несколькими GPU.

*Ключевые слова:* молекулярная динамика, структура данных, GPU.

**Введение**

В настоящее время при проведении исследований микро- и наномасштабных явлений и объектов значительное внимание уделяется вопросам применения методов молекулярной динамики (МД), так как они позволяют изучать процессы в системах, находящихся как в состоянии термодинамического равновесия, так и в его отсутствие. Несмотря на то, что методы МД успешно применяются для подобного рода задач, исследование с их помощью реальных физических процессов было и остается весьма сложной задачей, так как требует рассмотрения большого количества молекул и, как следствие, проведения большого количества вычислительных операций в минуту. Ускорение расчетов возможно как за счет использования современных численных алгоритмов, так и за счет высокопроизводительного аппаратного обеспечения.

Эффективность МД-моделирования может быть увеличена за счет алгоритмических усовершенствований метода. При исследовании одноатомных жидкостей широко используется потенциал взаимодействия Леннарда–Джонса. Так как он является близкодьющим (на каждый атом существенно влияние лишь его ближайшего окружения), то нет необходимости в расчете сил взаимодействия каждой молекулы

с каждой. Упростить этот процесс позволяет использование списков соседей или структур данных.

Первые попытки в этом направлении были описаны М.Р. Allen и D.J. Tildesley [1] с построением списка соседей. Использование такого метода было основано на малом радиусе взаимодействия потенциала Леннарда–Джонса.

Позже были разработаны иерархические структуры данных. Такие структуры сложны в реализации, но имеют преимущество в применении к дальнедействующим потенциалам. Один из способов построения такой структуры можно найти в работах [2, 3].

К высокопроизводительному аппаратному обеспечению можно отнести реализацию МД-моделирования на системах, обладающих возможностью параллельных вычислений. На сегодняшний день существует ряд пакетов прикладных программ, направленных на моделирование методами МД, таких как LAMMPS [4], DLPOLY [5], NAMD [6], GROMACS [7] и ESPResSO [8], которые достаточно эффективно реализуют метод молекулярной динамики на вычислительных кластерах, однако лишь некоторые из них включают в себя возможность реализации вычислений на графических процессорах.

Многие авторы уделяли внимание моделированию методами МД на GPU [9–11], однако

впервые данный метод полностью был реализован на GPU в 2008 году Андерсеном и др. [12]. В своей работе они представили первую реализацию моделирования одноатомных веществ методами молекулярной динамики, в которой все шаги алгоритма выполнялись на GPU с использованием технологии NVIDIA CUDA. Для сокращения сложности алгоритма они использовали списки соседей. Одновременно с ними подобную реализацию алгоритма представил J.A. van Meel [13]. В 2011 году Trott et al. [14] представили реализацию алгоритма молекулярной динамики на кластере из GPU с использованием пакета LAMMPS. Для ускорения расчетов в алгоритме так же были использованы списки соседей. Вычисления были проведены для различных классов материалов (биомолекулы, полимеры, металлы, полупроводники), а скорость расчетов была сравнена с существующими реализациями МД-моделирования на GPU.

В представленной работе описан процесс построения новой структуры данных для неполярных молекул. Использование такой структуры данных, совместно с графическими процессорами, позволяет значительно ускорить вычисления близкодействующих потенциалов в методе МД.

### Математическая модель

Молекулярная динамика – метод, используемый для определения макроскопических свойств системы из  $N$  тел, в котором движение частиц подчиняется второму закону Ньютона. Система состоит из атомов, взаимодействующих друг с другом согласно некоторому потенциалу (или нескольким потенциалам) взаимодействия. В классической молекулярной динамике положение атомов вычисляется из начальных условий  $(r_0, v_0)$  посредством разрешения уравнений движения

$$\frac{dr_i}{dt} = v_i, \quad \frac{dv_i}{dt} = \frac{F(r_i)}{m_i}, \quad F(r_i) = -\frac{\partial}{\partial r_i} U_i(r^N),$$

где  $r^N = (r_1, r_2, \dots, r_N)$  – набор расстояний от  $i$ -й частицы до всех остальных,  $r_i$  – радиус-вектор  $i$ -й частицы,  $m_i$  – масса  $i$ -й частицы,  $v_i$  – ее скорость,  $U_i$  – суммарный потенциал взаимодействия  $i$ -й молекулы со всеми остальными. За исключением простейших случаев, эта система уравнений решается численно согласно выбранному алгоритму (метод скоростей Верле, leapfrog или др.).

В качестве потенциала взаимодействия рассматривается потенциал Леннарда–Джонса [15]

$$U(r_j) = 4\epsilon \left[ \left( \frac{\sigma}{r_j} \right)^{12} - \left( \frac{\sigma}{r_j} \right)^6 \right].$$

Ввиду его быстроубывающей природы он обрезается на расстоянии  $r_{cutoff}$  (радиус обрезки). В данной работе используется обезразмеривание переменных согласно [1].

Основная цель данной работы – разработка структуры данных, удобной при использовании гетерогенной архитектуры, состоящей из CPU и GPU. Поэтому результаты направлены на описание алгоритма построения структуры данных при моделировании динамики одноатомных жидкостей, взаимодействие между которыми описывается близкодействующим потенциалом Леннарда–Джонса. Для определения статистических свойств системы рассматривается NVT-ансамбль (Number Volume Temperature ensemble), в котором фиксирована кинетическая энергия молекул. Поддержание постоянной температуры осуществляется при помощи термостата Берендсена [16]. Периодические граничные условия применяются во всех направлениях.

### Алгоритм построения структуры данных

Для эффективного применения параллельных алгоритмов необходим удобный и быстрый доступ к данным. Для этого была разработана эффективная структура данных. Опишем основные моменты её построения. Алгоритм состоит из следующих этапов:

*Предварительные вычисления.* Построение сетки. Определение основных параметров: количество боксов, их линейные размеры, порядок нумерации, положение центра каждого бокса. Построение списка соседей для каждого бокса и централизованного вектора соседей.

Данные вычисления выполняются до цикла по времени. Такие части алгоритма достаточно реализовать на CPU, так как эти вычисления производятся лишь один раз при генерации начальных данных.

*Генерация структуры данных.* Так как предполагается разбиение области моделирования на боксы, то самым лучшим способом будет расположить координаты в такой последовательности, когда сначала записаны все частицы, находящиеся в 0-м боксе, далее в 1-м, и так до последнего бокса ( $N_{AllBox} - 1$ ). Для построения структуры данных используется bucket-сортировка атомов [17].

*Преобразование позиций в локальные координаты.* Данная операция необходима для более удобного вычисления расстояния между частицами – она позволяет избежать лишнего ветвления алгоритма, которое плохо сказывается на производительности GPU.

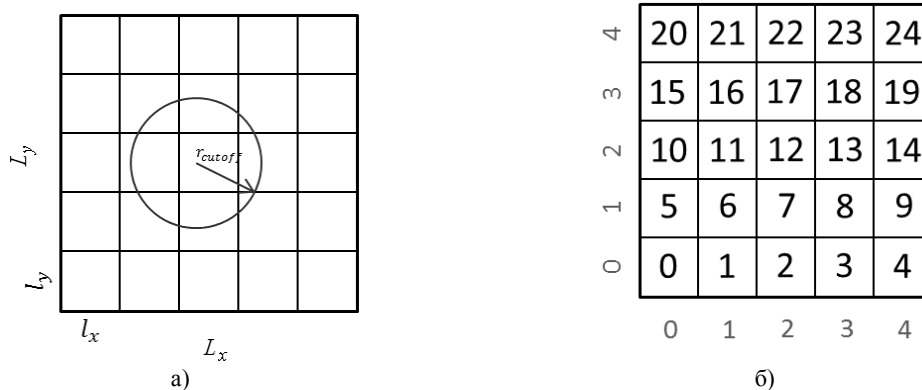


Рис. 1. (а) схема расположения боксов; (б) порядок нумерации боксов

Ниже приведем детальное описание каждого из представленных пунктов. Все формулы записаны для трёхмерного случая. Для простоты и наглядности изложения материала рисунки приведем для двумерной структуры данных.

### Число боксов и их нумерация

Предположим, что исследуемая область имеет размер  $L_x \times L_y \times L_z$ , радиус взаимодействия Леннарда–Джонса  $r_{cutoff}$ , тогда область можно разбить на боксы, где их число в каждом направлении соответственно равно

$$N_{xBox} = \left\lfloor \frac{L_x}{r_{cutoff}} \right\rfloor, \quad N_{yBox} = \left\lfloor \frac{L_y}{r_{cutoff}} \right\rfloor, \quad N_{zBox} = \left\lfloor \frac{L_z}{r_{cutoff}} \right\rfloor.$$

Тогда общее количество боксов  $N_{AllBox} = N_{xBox} \times N_{yBox} \times N_{zBox}$ . Линейные размеры каждого бокса

$$l_x = \frac{L_x}{N_{xBox}}, \quad l_y = \frac{L_y}{N_{yBox}}, \quad l_z = \frac{L_z}{N_{zBox}}.$$

Исходя из этих данных можно вычислить положение центра каждого бокса.

Введем два вида индексации боксов: сквозную и декартову. Сквозной индекс будем обозначать  $n$ , а декартов  $[i, j, k]$ . Нумерацию будем производить с нуля. Сквозной индекс представляет собой последовательную нумерацию всех боксов с «быстрым» индексом вдоль оси  $x$  и «медленным» вдоль оси  $z$ , т.е. сквозной индекс проходит по кривой, заполняющей область (space filling curve). Декартов способ определяет бокс согласно сдвигу вдоль каждой из осей относительно начала отсчёта (началом отсчёта выбирается левый нижний угол). Схема построения боксов и порядок их нумерации изображены на рис. 1. Переход от одной системы нумерации к другой осуществляется по простым формулам, которые используют только нумерацию бокса и количество боксов в каждом направлении.

### Вычисление соседей

Согласно построенному разбиению области каждый атом, расположенный в каком-либо боксе, может взаимодействовать лишь с атомами, находящимися в соседних боксах. Для нахождения соседних боксов удобно использовать декартову нумерацию. Таким образом, для любого бокса с координатами  $[i, j, k]$ , соседними будут 27 ячеек (включая сам бокс):  $[i-1, j-1, k-1]$ ,  $[i, j-1, k-1]$ , ...,  $[i+1, j+1, k+1]$ . Возникает вопрос в нахождении соседей для боксов, находящихся на границе области. Их можно найти путём использования периодических граничных условий. Таким образом, может быть составлен список, состоящий из  $27N_{AllBox}$  элементов, который каждому боксу ставит в соответствие всех его соседей.

Введем понятие центрированного вектора соседей. Построим 27 векторов (включая нулевой вектор), которые соединяют центр произвольного бокса с центрами боксов всех его соседей. В общем случае это векторы  $\{[il_x, jl_y, kl_z]\}$ , где  $i, j, k$  принимают значения  $\{-1, 0, 1\}$ . Схема построения периодизации системы и введения центрированных векторов соседей представлена на рис. 2.

### Построение структуры данных

Для реализации удобного доступа к данным необходимо пересортировать частицы в системе. Алгоритм сортировки состоит из следующих шагов:

1. Построение гистограммы размещения. Она показывает количество частиц в каждом боксе и локальный индекс для каждой частицы в данном боксе.

2. Выполнение параллельного сканирования с определением «маркеров», которые показывают, чему равны значения индексов первого и последнего атомов, находящихся в заданном

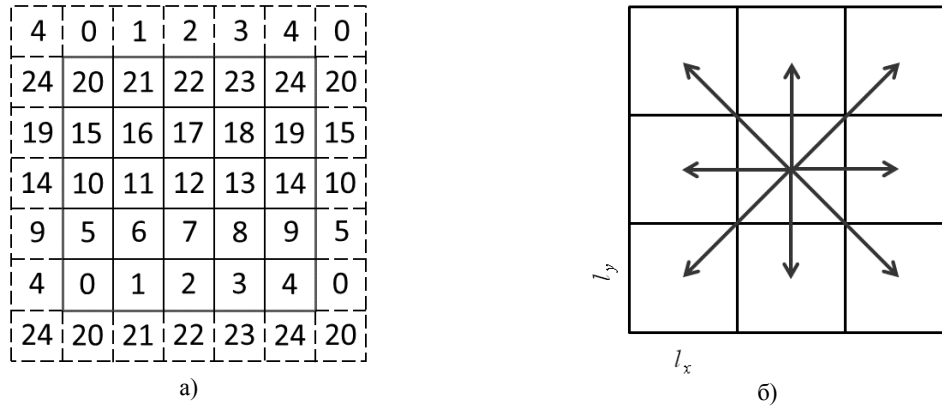


Рис. 2. (а) схема периодизации системы; (б) построение центрированных векторов соседей

боксе в новом, отсортированном массиве частиц, и определение глобального индекса частиц.

3. Упорядочивание частиц в соответствии с их глобальным индексом.

### Преобразование позиций в локальные координаты

При решении системы уравнений Ньютона необходимо вычислять силы взаимодействия между атомами, что требует нахождения расстояния между частицами. Для унифицирования процесса вычисления этого расстояния, а впоследствии и вычисления сил взаимодействия при использовании периодических граничных условий преобразуем все координаты в векторы  $\vec{r}_i^* = \vec{r}_i - \vec{r}_{\text{BoxCenter}(i)}$ , где  $\vec{r}_i$  – позиции частиц,  $\vec{r}_{\text{BoxCenter}(i)}$  – центр бокса, в котором находится  $i$ -я частица. Это позволяет избежать лишнего ветвления алгоритма при учете периодических граничных условий, которое плохо сказывается на производительности GPU.

В случае введения таких дополнительных векторов вычисление расстояния между двумя точками можно записать следующим образом (см. рис. 3). Предположим, необходимо вычислить расстояние  $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ . Так как все координаты были преобразованы в локальные, то вместо векторов  $\vec{r}_i, \vec{r}_j$  есть векторы  $\vec{r}_R, \vec{r}_S$ . Также существует один из векторов соседей (обозначен  $\vec{r}_N$ ). Согласно правилу суммирования векторов  $\vec{r}_{ij} = \vec{r}_R - \vec{r}_N - \vec{r}_S$ .

### Верификация кода

Для валидации расчетов методом МД рассмотрим свободную динамику паро-жидкостной среды аргона. Область моделирования представляет собой параллелепипед с размерами

$L_x = L_y = 24\sigma$ ,  $L_z = 48\sigma$ . Значения параметров потенциала взаимодействия Леннарда–Джонса выбираются следующими:  $\sigma = 0.34$  нм,  $\epsilon = 1.64 \cdot 10^{-21}$  Дж, что соответствует аргону, который помещается в центр области моделирования с параметрами  $\rho^* = 0.7$  ( $\rho = 1182$  кг/м<sup>3</sup>) и  $T^* = 1.0$  ( $T = 120$  К).

Для определения линии насыщения область моделирования разбивается на слои вдоль оси  $z$  параллельно плоскости  $L_x L_y$ . Количество таких плоскостей обозначим  $N_{bin}$  и в данной работе примем его равным 100. Плотность в каждом слое будем определять по формуле

$$\rho_i = \frac{N_i N_{bin}}{L_x L_y L_z},$$

где  $N_i$  – число частиц, находящихся в соответствующем  $i$ -м слое. Значение  $i$  изменяется от 1 до  $N_{bin}$ .

На рис. 4 изображены профили плотности при различных температурах системы. Для создания симметричной картины через каждые 500 шагов производится центрирование частиц относительно центра параллелепипеда с сохранением межмолекулярных расстояний.

Исходя из построенных распределений (рис. 4) можно определить плотность жидкого аргона по центральной части графика и газообразного по крайним участкам. Вычисляя таким способом плотности при различных температурах, можно построить кривую насыщения. Начальная плотность аргона, находящегося в центре параллелепипеда,  $\rho^* = 0.7$ . Температура  $T^*$  изменяется в диапазоне от 0.7 до 1.25 (от 85 К до 150 К). На рис. 5 отображены полученные результаты в размерных величинах. На графике приведены результаты расчетов методами молекулярной динамики и экспериментальные данные из [18]. Из графика видно, что результаты численных расчетов хорошо согласуются с экспериментом.

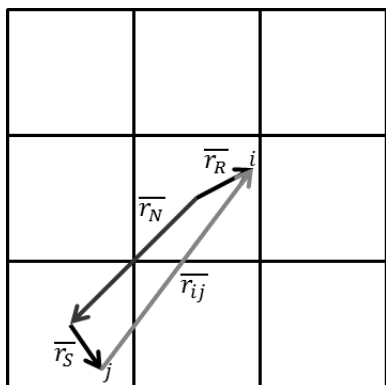


Рис. 3. Схема вычисления расстояния между частицами

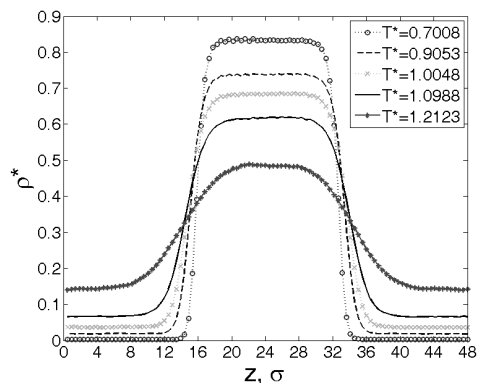


Рис. 4. Профиль плотности при различных температурах

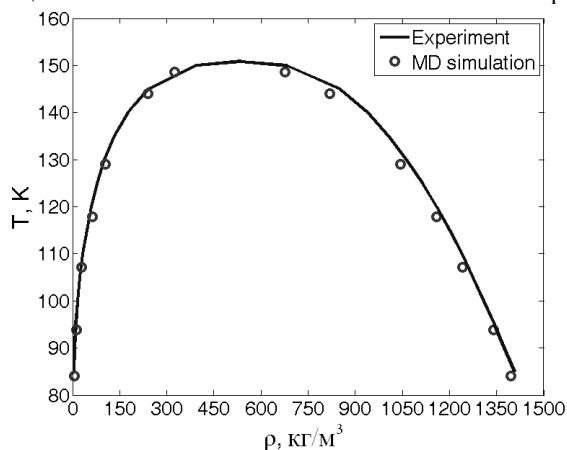


Рис. 5. Линия насыщения

### Применение GPU

В описанном алгоритме построения структуры данных есть операции, которые выполняются до цикла по времени. Такие части алгоритма могут быть реализованы на CPU, так как эти вычисления производятся лишь один раз при генерации начальных данных (определение основных параметров, построение общего списка соседей). Операции, которые выполняются на каждом временном шаге, реализуются на GPU: расчёт положений частиц, генерация структуры данных (СД), расчёт сил и потенциала, термостатирование. На рис. 6 представлена блок-схема рассматриваемой реализации структуры данных и метода молекулярной динамики для гетерогенного кластера, состоящего из двух нод, каждая из которых оснащена несколькими графическими процессорами.

Изначально структура данных генерируется на главной ноде, и она используется для распределения данных по нодам. На каждой из нод производятся операции только с частицами, принадлежащими этим нодам. На каждом временном шаге производится проверка на необходимость регенерации структуры данных на данной ноде: рассчитывается число частиц, перелетевших между боксами внутри ноды и ме-

жду нодами. В случае необходимости структура данных регенерируется на одной или нескольких нодах. Далее производится обмен граничными условиями (информацией по частицам в граничных для нод боксах). После производится расчёт межмолекулярного взаимодействия параллельно на нескольких GPU внутри каждой ноды.

### Результаты

В данной работе производится моделирование динамики жидкого аргона с использованием разработанной структуры данных. Рассматривается жидкость с плотностью  $1013 \text{ кг/м}^3$  при температуре  $130 \text{ K}$  и  $r_{cutoff} = 5.0\sigma$ . Вычисления производятся на гетерогенной рабочей станции с двумя 6-ядерными CPU Intel Xeon 5660 2.8GHz (всего 12 физических ядер и 12 виртуальных ядер, с использованием технологии Hyper-Threading), 12GB RAM, и четырьмя GPU NVIDIA Tesla C2075 с 6GB RAM. Код на GPU был написан с использованием технологии NVIDIA CUDA, параллельная работа нескольких GPU была реализована при помощи технологии OpenMP, для коммуникации между нодами использовалась технология MPI. Расчёты

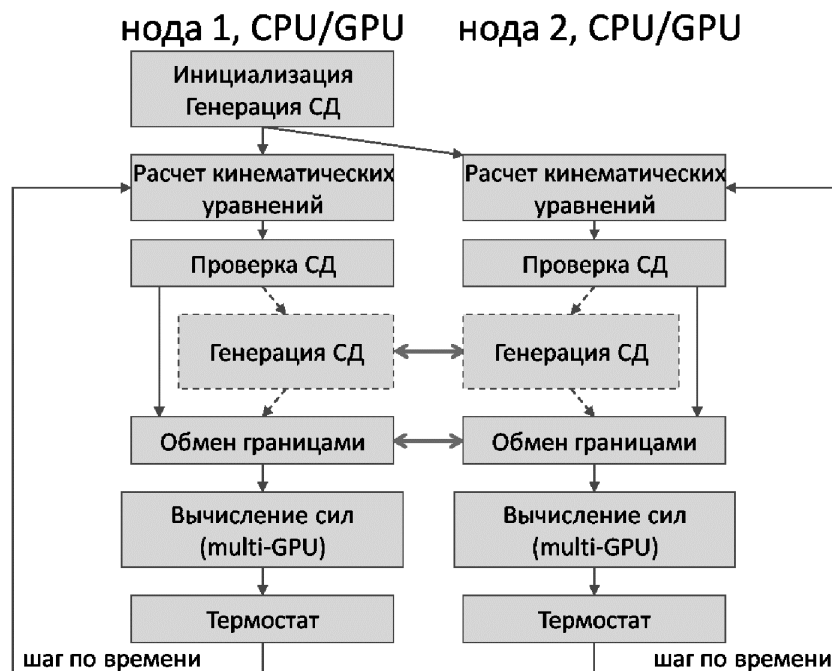


Рис. 6. Блок-схема для гетерогенного кластера на примере двух нод (СД-структура данных)

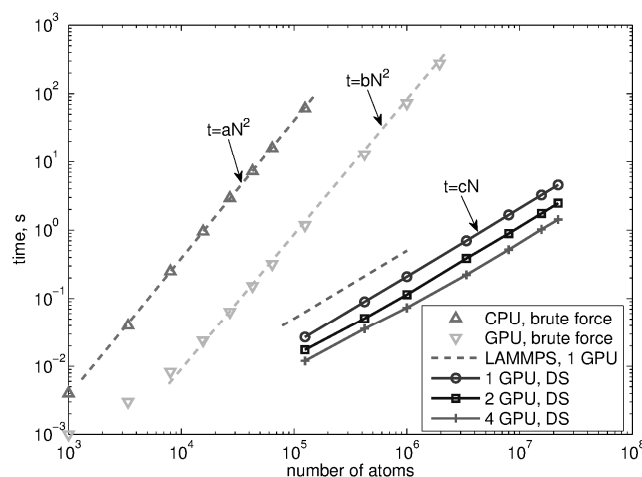


Рис. 7. Полное время расчета одного временного шага алгоритма

проводились с использованием чисел с плавающей точкой двойной точности.

На рис. 7 представлено полное время расчета одного шага алгоритма в зависимости от числа атомов в системе. На графике для примера приведены времена расчета методом прямого суммирования на CPU и на GPU (brute force). Из рис. 7 видно, что использование структур данных позволяет уменьшить вычислительную сложность с квадратичной до линейной.

Также было проведено сравнение по производительности с пакетом LAMMPS. Данный симулятор был выбран исходя из того, что в нём реализован метод, позволяющий снизить вычислительную сложность всего алгоритма, и реализована возможность использования GPU. Сравнение было проведено при одинаковых

физических параметрах и равномерном распределении молекул по области. Для расчетов на GPU в LAMMPS использовался пакет USER-CUDA. Сравнение проводилось на идентичных вычислительных станциях, описанных выше. Сравнение показало, что производительность настоящего метода до трёх раз выше, чем производительность LAMMPS на одном GPU.

В таблице показано ускорение расчета за счёт использования нескольких GPU на одной ноде по сравнению с одним GPU. Из таблицы видно, что использование нескольких GPU позволяет добиться достаточно хорошего ускорения с учётом закона Амдала для примерно 7% алгоритма, исполняемого только на одном GPU. Масштабируемость по нодам была исследована на примере двух нод, каждая с характеристика-

Таблица

Ускорение расчёта на нескольких GPU		
GPU	Ускорение	Закон Амдала для 7%
2	1.85	1.87
4	3.21	3.3

ми, описанными выше. Исследования показали, что масштабируемость близка к линейной.

Использование структуры данных, описанной в данной работе, значительно ускоряет процесс вычисления сил для близкодствующих потенциалов взаимодействия. Применение такого подхода позволяет провести вычисления на одном шаге алгоритма для системы, состоящей из 20 миллионов частиц, менее чем за 2 секунды.

*Работа выполнена при поддержке гранта Министерства образования и науки РФ (11.G34.31.0040), гранта РФФИ (код проекта №12-01-31083 «мол\_а»).*

#### Список литературы

1. Allen M.P., Tildesley D.J. Computer Simulation of Liquids. Oxford: Clarendon Press, 1987. 385 p.
2. Gumerov N.A., Ramani Duraiswami. Fast multipole methods on graphics processors // J. Computational Physics. 2008. V. 227. P. 8290–8313.
3. Марьин Д.Ф., Малышев В.Л., Моисеева Е.Ф. и др. Ускорение молекулярно-динамических расчетов с помощью Быстрого Метода Мультиполей и графических процессоров // Журнал «Вычислительные методы и программирование». М.: МГУ. 2013. Т. 14. С. 483–495.
4. Plimpton Steve. Fast parallel algorithms for short-range molecular dynamics // J. Computational Physics. 1995. V. 117. № 1. P. 1–19.
5. Forester T.R., Smith W. DL POLY 2.0: A general-purpose parallel molecular dynamics simulation package // J. Mol. Graph. 1996. V. 14. № 3. P. 136–141.
6. Phillips James C., Braun R., Wang Wei, et al. Scalable molecular dynamics with namd // J. Computational Chemistry. 2005. V. 26. № 16. P. 1781–1802.
7. Berendsen H.J.C., Van Der Spoel D., Van Duren R. Gromacs: A message-passing parallel molecular dynamics implementation // Comp. Phys. Comm. 1995. V. 91. P. 43–56.
8. Limbach H.J., Arnold A., Mann D.A., Holm C. Espresso – an extensible simulation package for research on soft matter systems // Comp. Phys. Comm. 2006. V. 174. № 9. P. 704–727.
9. Yang Juekuan, Wang Yujuan, Chen Yunfei. GPU accelerated molecular dynamics simulation of thermal conductivities // J. Computational Physics. 2007. V. 221. № 2. P. 799–804.
10. Che Shuai, Boyer Michael, Meng Jiayuan, et al. A performance study of general-purpose applications on graphics processors using CUDA // J. Parallel and Distributed Computing. 2008. V. 68. № 10. P. 1370–1380.
11. Liu Weiguo, Schmidt Bertil, Voss Gerrit, Muller-Wittig Wolfgang. Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA // Comp. Phys. Comm. 2008. V. 179. № 9. P. 634–641.
12. Anderson Joshua A., Lorenz Chris D., Travesser A. General purpose molecular dynamics simulations fully implemented on graphics processing units // J. Computational Physics. 2008. V. 227. P. 5342–5359.
13. van Meel J.A., Arnold A., Frenkel D., et al. Harvesting graphics power for md simulation // Molecular Simulation. 2008. V. 34. № 3. P. 259–266.
14. Trott Christian R., Winterfeld Lars, Crozier Paul S. General-purpose molecular dynamics simulations on GPU-based clusters. 2011. URL:~http://arxiv.org/abs/1009.4330v2.
15. Jones J.E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas // Royal Society of London. Proc. Series A. 1924. V. 106. P. 463–477.
16. Berendsen H.J.C., Postma J.P.M., van Gunsteren W.F., et al. Molecular-Dynamics with Coupling to an External Bath // J. Chemical Physics. 1984. V. 81. № 8. P. 3684–3690.
17. Кормен Томас Х., Лейзерсон Чарльз И., Ривест Рональд Л., Штайн Клиффорд. Алгоритмы. Построение и анализ. М.: Вильямс, 2012. 1296 с.
18. Варгафтик Н.Б. Справочник по теплофизическим свойствам газов и жидкостей. М.: Наука, 1972. 721 с.

## GPU ACCELERATION OF MOLECULAR DYNAMICS SIMULATIONS OF NONPOLAR MOLECULES

*V.L. Malyshev, D.F. Marin, E.F. Moiseeva, N.A. Gumerov, I.Sh. Akhatov*

The use of data structure in molecular dynamics simulations of nonpolar molecules is described. The interaction between atoms is described by the Lennard-Jones potential. The developed data structure allows one to reduce the computational complexity from  $O(n^2)$  to  $O(n)$ . A scheme of the algorithm realization for a heterogeneous architecture consisting of CPU and GPUs is presented. The use of GPUs significantly accelerates the computation speed. It is shown that the technique described can be used in molecular dynamics simulations of nonpolar molecules in domains with characteristic sizes of tens of nanometers on personal supercomputers equipped with GPUs.

*Keywords:* molecular dynamics, data structure, GPU.

## References

1. Allen M.P., Tildesley D.J. *Computer Simulation of Liquids*. Oxford: Clarendon Press, 1987. 385 p.
2. Gumerov N.A., Ramani Duraiswami. Fast multipole methods on graphics processors // *J. Computational Physics*. 2008. V. 227. P. 8290–8313.
3. Mar'in D.F., Malyshev V.L., Moiseeva E.F. i dr. Uskorenie molekulyarno-dinamicheskikh raschetov s pomoshch'yu Bystrogo Metoda Mul'tipolej i graficheskikh processorov // *Zhurnal «Vychislitel'nye metody i programmirovaniye»*. M.: MGU. 2013. T. 14. S. 483–495.
4. Plimpton Steve. Fast parallel algorithms for short-range molecular dynamics // *J. Computational Physics*. 1995. V. 117. № 1. P. 1–19.
5. Forester T.R., Smith W. DL POLY 2.0: A general-purpose parallel molecular dynamics simulation package // *J. Mol. Graph.* 1996. V. 14. № 3. P. 136–141.
6. Phillips James C., Braun R., Wang Wei, et al. Scalable molecular dynamics with namd // *J. Computational Chemistry*. 2005. V. 26. № 16. P. 1781–1802.
7. Berendsen H.J.C., Van Der Spoel D., Van Duren R. Gromacs: A message-passing parallel molecular dynamics implementation // *Comp. Phys. Comm.* 1995. V. 91. P. 43–56.
8. Limbach H.J., Arnold A., Mann D.A., Holm C. Espresso – an extensible simulation package for research on soft matter systems // *Comp. Phys. Comm.* 2006. V. 174. № 9. P. 704–727.
9. Yang Juekuan, Wang Yujuan, Chen Yunfei. GPU accelerated molecular dynamics simulation of thermal conductivities // *J. Computational Physics*. 2007. V. 221. № 2. P. 799–804.
10. Che Shuai, Boyer Michael, Meng Jiayuan, et al. A performance study of general-purpose applications on graphics processors using CUDA // *J. Parallel and Distributed Computing*. 2008. V. 68. № 10. P. 1370–1380.
11. Liu Weiguo, Schmidt Bertil, Voss Gerrit, Muller-Wittig Wolfgang. Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA // *Comp. Phys. Comm.* 2008. V. 179. № 9. P. 634–641.
12. Anderson Joshua A., Lorenz Chris D., Travesser A. General purpose molecular dynamics simulations fully implemented on graphics processing units // *J. Computational Physics*. 2008. V. 227. P. 5342–5359.
13. van Meel J.A., Arnold A., Frenkel D., et al. Harvesting graphics power for md simulation // *Molecular Simulation*. 2008. V. 34. № 3. P. 259–266.
14. Trott Christian R., Winterfeld Lars, Crozier Paul S. General-purpose molecular dynamics simulations on GPU-based clusters. 2011. URL: <http://arxiv.org/abs/1009.4330v2>.
15. Jones J.E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas // *Royal Society of London. Proc. Series A*. 1924. V. 106. P. 463–477.
16. Berendsen H.J.C., Postma J.P.M., van Gunsteren W.F., et al. Molecular-Dynamics with Coupling to an External Bath // *J. Chemical Physics*. 1984. V. 81. № 8. P. 3684–3690.
17. Kormen Tomas H., Lejzerson Charl'z I., Rivest Ronal'd L., Shtajn Klifford. *Algoritmy. Postroenie i analiz*. M.: Vil'yams, 2012. 1296 c.
18. Vargaftik N.B. *Spravochnik po teplofizicheskim svoystvam gazov i zhidkostej*. M.: Nauka, 1972. 721 c.