

УДК 532.74

ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ В ЗАДАЧАХ СУПЕРКОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ МОЛЕКУЛЯРНЫХ ЖИДКОСТЕЙ МЕТОДОМ МОНТЕ-КАРЛО**А.В. Теплухин***Институт математических проблем биологии РАН, Пущино*
E-mail: tepl@impb.psn.ru*Статья поступила 3 апреля 2012 г.*

Предложена эффективная стратегия молекулярного моделирования методом Монте-Карло, основанная на объединении двух ключевых подходов к распараллеливанию вычислений. Пространственная (доменная) декомпозиция обеспечивает высокую масштабируемость вычислительных алгоритмов посредством разбиения "большой задачи" на несколько одновременно разрешимых маленьких подзадач. Однако в рамках этого метода можно уменьшать размеры доменов лишь до определенного значения. Декомпозиция по частицам (разбиение программных циклов на порции), наоборот, очень эффективна для изучения объектов малого и среднего размера, но плохо масштабируется и с ростом размера модели быстро исчерпывает ресурсы оперативной памяти компьютера. Сочетание этих подходов позволяет "нейтрализовать" их ограничения и создать эффективные суперкомпьютерные программы для изучения молекулярных моделей, состоящих из сотен миллионов атомов.

Ключевые слова: жидкость, вода, суперкомпьютер, моделирование, Монте-Карло, алгоритм, параллельные вычисления, MPI.

ВВЕДЕНИЕ

Компьютерные методы молекулярного моделирования являются мощным инструментом, позволяющим изучать строение и макроскопические характеристики различных веществ путем проведения численных экспериментов с "микросистемами", допускающими явное описание в конфигурационном (фазовом) пространстве. Однако компьютерные программы, используемые для моделирования, весьма требовательны к вычислительным ресурсам, поскольку приходится учитывать очень сложные взаимодействия компонентов исследуемого объекта. Широко применяемая в настоящее время модель молекулярной жидкости отражает приемлемый по современным нормам уровень соотношения точности и трудоемкости и представляет собой совокупность "частиц" с попарно аддитивными взаимодействиями. Каждая частица есть непустое множество силовых центров (обычно это центры атомов). Взаимодействия силовых центров выражаются в виде суммы кулоновского и ван-дер-ваальсового вклада. При моделировании полимеров и молекул с внутренними степенями свободы к некоторым силовым центрам добавляются дополнительные потенциальные функции, учитывающие энергию деформации химических связей и углов между ними.

Далее будем рассматривать модели, составленные из "жестких" неполяризуемых частиц. В этом случае положение отдельной частицы определяется, как у твердого тела, набором из 6 чисел (координаты "центра" и три угла эйлеровского типа), а совокупность значений координат всех частиц в данный момент времени представляет собой мгновенную конфигурацию ("фото-

графию") моделируемой системы. Искомые характеристики изучаемых систем рассчитываются путем усреднения по наборам мгновенных конфигураций, получаемым в ходе моделирования. В основе применяемых для этого компьютерных программ лежат либо детерминистские (решение уравнений молекулярной динамики), либо стохастические (реализации цепей Маркова в конфигурационном пространстве) методы получения серий мгновенных конфигураций [1].

В связи с этим особую остроту принимает общая для вышеупомянутых методов моделирования проблема представительности выборок мгновенных конфигураций. Для небольших моделей (десятки атомов) эта проблема снимается путем сравнения результатов расчетов, стартовавших из разных начальных конфигураций. Однако время, требуемое для получения одной "фотографии", катастрофически возрастает с увеличением размера исследуемого объекта. В свою очередь, усложняющаяся топология многомерного конфигурационного пространства (многочисленные потенциальные барьеры, явления типа "бутылочных горловин" и др.) влечет за собой потребность в генерации значительно большего числа мгновенных конфигураций для перехода в новую представительную область.

Цель данной работы — продемонстрировать высокую эффективность моделирования молекулярных жидкостей методом Монте-Карло путем глубокого распараллеливания вычислений. Обсуждаемые далее алгоритмы и соответствующие суперкомпьютерные программы были разработаны автором данной статьи на алгоритмическом языке Fortran77 с использованием коммуникационных подпрограмм MPI1.2 [2]. Для генерации равномерно распределенных случайных чисел использовался алгоритм [3], позволяющий создавать сотни независимых числовых последовательностей. В ходе исследований эффективности алгоритмов проводились расчеты водных систем, построенных на основе модели молекулы воды TIP4P2005 [4] с постоянными парциальными зарядами и фиксированной геометрией расположения ее четырех силовых центров.

СТАНДАРТНАЯ ПРОЦЕДУРА ВЫБОРКИ ПО МЕТРОПОЛИСУ И ПУТИ ЕЕ РАСПАРАЛЛЕЛИВАНИЯ

Для расчета структурных и термодинамических характеристик различных веществ методом Монте-Карло обычно применяют алгоритм Метрополиса и др. [5], реализованный по классической схеме [1]. В качестве модельной системы чаще всего рассматривают элементарную ячейку кубической формы с периодическими граничными условиями, содержащую десятки или сотни частиц. Для расчета энергии взаимодействий частиц между собой используют попарно аддитивные атом-атомные потенциальные функции. Размер ячейки (объем V), температура T и число частиц N в ходе расчетов не изменяются (NVT -ансамбль [1]). Начальное положение частиц выбирается случайным образом в пределах ячейки.

В нескольких фразах схема стандартной процедуры Метрополиса для NVT -ансамбля может быть изложена следующим образом: конкретная частица (далее — "испытуемая частица") поворачивается на случайный угол вокруг случайно выбранной оси, проходящей через ее центр, и смещается на случайный вектор (с учетом периодических граничных условий). Рассчитывается изменение энергии системы ΔE . Если энергия снижается, новое положение частицы принимается. В противном случае новое положение может быть принято с вероятностью $\exp(-\Delta E/(k_B T))$, здесь k_B — постоянная Больцмана. Если перемещение, ведущее к росту энергии, отвергнуто, частица остается в исходном положении, а молекулярная конфигурация, не изменившаяся в ходе этого испытания, принимается в качестве очередного состояния цепи Маркова. Затем эта последовательность действий применяется к следующей частице (циклически, по номерам частиц) и т.д. В нормальных условиях максимальное смещение за одно испытание составляет $0,2 \text{ \AA}$, а максимальный угол поворота — $0,15$ рад. Более подробное изложение можно найти в книге [1].

Многочисленное (сотни миллионов раз) применение этой процедуры ко всем частицам модели позволяет получить для расчета термодинамических средних представительную выборку мгновенных конфигураций, формирующую в бесконечном (по числу конфигураций) пределе канонический ансамбль Гиббса.

Таким образом, типичный последовательный (без распараллеливания) алгоритм программы молекулярного моделирования методом Монте-Карло содержит, как минимум, четыре вложенных цикла:

1. Цикл циклов по частицам;
2. Цикл по частицам (процедура Метрополиса);
3. Цикл по частицам, участвующим во взаимодействиях с испытываемой частицей (расчет приращения энергии системы ΔE);
4. Цикл по парам атомов (силовых центров) для расчета приращения энергии взаимодействия испытываемой частицы с ее партнером.

Конец циклов 1, 2, 3, 4.

При соблюдении определенных условий (см. ниже) каждый цикл допускает параллельную реализацию. Более того, возможны варианты, сочетающие различные комбинации "уровней распараллеливания". Например, можно распараллелить в одной программе только циклы 3 и 4 или 2 и 3, а можно осуществить распараллеливание на всех уровнях. Следует отметить, что если работы по распараллеливанию на нижних уровнях (4 и 3) не выходят за пределы сферы компетенции квалифицированного программиста, то организация параллельных вычислений для двух внешних циклов невозможна без предварительного анализа корректности разрабатываемых алгоритмов с позиций общей и статистической физики, а также теории марковских цепей.

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ В ЗАДАЧАХ МОЛЕКУЛЯРНОГО МОДЕЛИРОВАНИЯ МЕТОДОМ МОНТЕ-КАРЛО

Современные технологии позволяют размещать в корпусе одного процессора несколько вычислительных ядер, способных работать независимо друг от друга. Во избежание терминологической путаницы условимся далее под словом "процессор" подразумевать отдельное вычислительное ядро, на котором выполняется только одна копия программы (вычислительный процесс).

Время выполнения теста измеряли по завершению заданного количества итераций цикла 1 в "нулевом" процессоре (согласно правилам MPI [2], нумерация процессоров начинается с нуля). Каждый тест повторяли по 10 раз. Как правило, получаемые результаты отклоняются не более чем на 15 % от минимального значения по группе. Далее в таблицах и на графиках приводится наилучшее (минимальное) значение времени по конкретной позиции.

Цикл 4. В среднем на одну частицу приходится 3—10 силовых центров, поэтому типичный цикл этого типа "развертывается" в несколько десятков актов расчета энергии по соответствующей формуле потенциальной функции, ограничивая тем самым возможности по его распараллеливанию. Ускорение вычислений (в 1,5—2 раза) на этом уровне может быть получено только в результате оптимизации ассемблерного кода программы с использованием потоковых инструкций (SSE, SSE2 и т.п.) современных процессоров или директив OpenMP. Оптимальным средством для решения этой задачи является применение компиляторов, автоматически генерирующих исполняемый код с учетом таких возможностей.

Цикл 3. В зависимости от плотности моделируемой системы и радиуса действия атом-атомных потенциалов каждая частица может иметь до нескольких сотен партнеров по взаимодействиям. Благодаря попарной аддитивности потенциальных функций расчет приращения энергии системы ΔE можно серьезно ускорить, распределив список партнеров испытываемой частицы одинаковыми (по возможности) порциями среди доступных задач процессоров. Этот подход к распараллеливанию расчетов известен как декомпозиция по частицам (атомам или силовым центрам). Так, взяв M процессоров, можно было бы M -кратно уменьшить время вычислений, закрепив за каждым процессором свою группу частиц. Однако изменение координат частиц в процессе моделирования требует своевременного обмена информацией между процессорами, поэтому на реальном оборудовании эффект ускорения вычислений с ростом числа процессоров быстро ослабевает.

Логическая схема алгоритма для выяснения возможностей по ускорению вычислений за счет распараллеливания цикла 3 — следующая. При старте программы всем процессорам рас-

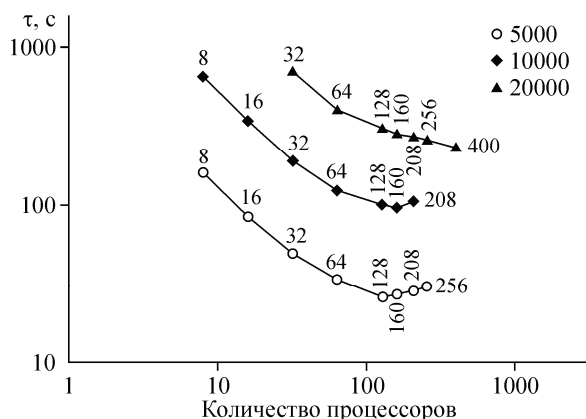


Рис. 1. Затраты времени на выполнение 100 итераций цикла 1 при расчете водных кластеров из 5000, 10 000 и 20 000 молекул воды. Распараллеливание цикла 3 по списку молекул. Вдоль линий данных указаны соответствующие количества процессоров

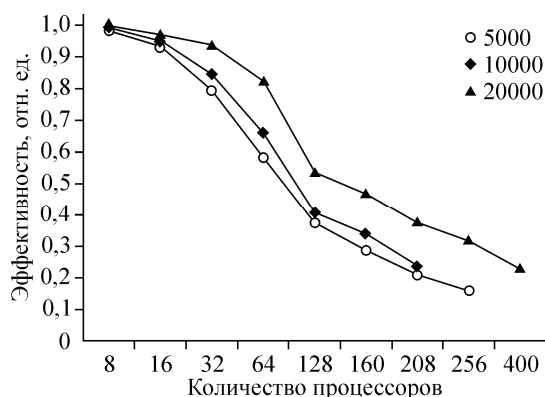


Рис. 2. Эффективность (см. текст статьи) распараллеливания вычислений в цикле 3 при расчете водных кластеров различного размера (см. подпись к рис. 1)

сылаются (только один раз, MPI_Bcast [2]) координаты всех частиц в начальной конфигурации. Цикл 4 здесь и далее распараллеливается компилятором. Распараллеливание выполняется только для расчета приращения энергии системы на этапе очередного испытания в рамках процедуры Метрополиса (цикл 3). "Новые" координаты испытываемой частицы генерируются процессором с номером 0 и рассылаются всем процессорам (MPI_Bcast). Каждый процессор считает приращение энергии взаимодействия испытываемой частицы со своей (определяемой в соответствии с номером данного процессора) порцией ее соседей. Сосчитанные приращения суммируются (MPI_reduce&MPI_sum [2]) у 0-го процессора, где и определяется: принять новую конфигурацию или сохранить старую. Результат (0 или 1) рассылается (MPI_Bcast) всем процессорам и, если получена цифра 1, все процессоры обновляют координаты испытанной частицы. Цикл 3 завершается переходом к очередной частице. Рассылку результата испытания можно совместить с рассылкой пробных координат для очередной частицы.

На суперкомпьютере МВС100К в МСЦ РАН (www.jssc.ru) было выполнено несколько серий расчетов для моделей кластеров, содержащих от 5000 до 20 000 молекул воды при температуре 300 К (радиус взаимодействий не ограничивали). В расчетах задействовали от 4 до 400 процессорных ядер. Благодаря использованию большого количества процессоров можно многократно снизить затраты календарного времени, требуемого для проведения вычислительных экспериментов (рис. 1). С практической точки зрения, эффективность алгоритма удобнее характеризовать величиной ускорения вычислений, достигаемого при заданном количестве процессоров. В этом случае "ускорение" вычисляется как отношение времени счета теста в режиме без какого-либо распараллеливания к соответствующему времени выполнения теста на заданном числе процессоров. В идеальном случае величина ускорения должна быть равной количеству задействованных процессоров. Эффективность реального алгоритма наиболее точно характеризуется отношением полученного ускорения к соответствующему количеству процессоров. Видно (рис. 2), что, начиная со 128 процессоров, дальнейшее распараллеливание с использованием декомпозиции по частицам становится неэффективным вне зависимости от размера моделируемой системы.

Цикл 2. На первый взгляд задача распараллеливания этого цикла кажется неразрешимой. Действительно, согласно теории цепей Маркова, лежащей в основе применения метода Монте-Карло для моделирования жидкостей [1], новое состояние моделируемой системы определяется только ее состоянием перед началом испытания. В контексте процедуры Метрополиса (см. предыдущий раздел) это означает, что результат испытания i -й частицы при выполнении цикла 2 не может стать известным до завершения испытания для частицы с номером $i - 1$. На

практике при использовании, к примеру, четырех процессоров для распараллеливания цикла 2, это приведет к тому, что процессор с наибольшим номером будет вынужден простаивать в ожидании результатов от предыдущих трех (а перед этим — третий процессор и т.д.). Вывод: простым разбиением цикла 2 на порции (как было сделано для цикла 3) ускорения вычислений добиться нельзя.

Тело цикла 2 представляет собой последовательность из трех групп операторов. Первая группа генерирует пробный набор координат для испытываемой частицы и может выполняться одновременно (параллельно) для нескольких испытываемых частиц. Вторая группа содержит вложенную пару циклов 3, 4 и отвечает за расчет приращения энергии системы ΔE . Эту группу операторов также можно выполнять одновременно для нескольких испытываемых частиц. Практически все время, затрачиваемое на одну итерацию цикла 2, уходит на выполнение операторов второй группы. Третья группа определяет: принять пробную конфигурацию или оставить ту, что была перед испытанием. Именно этот участок кода программы препятствует распараллеливанию цикла 2. Следовательно, можно добиться существенного ускорения вычислений, организовав код программы таким образом, чтобы операторы первой и второй групп выполнялись параллельно на каждом процессоре, а операторы третьей группы — только на одном из них.

Логическая схема алгоритма для выяснения возможностей по ускорению вычислений за счет распараллеливания цикла 2 следующая.

При запуске программы на np процессорах всем процессорам рассылаются (только один раз, MPI_Bcast) координаты начальной конфигурации всех частиц (хранятся в заранее подготовленном файле, считываемом 0-м процессором). Длина цикла 2 теперь не превосходит $N/np + 1$ (в нераспараллеленном варианте она была равна N — числу частиц в системе). В начале этого цикла каждый из np процессоров (независимо друг от друга) генерирует пробные координаты для нового положения одной из np , испытываемых на данной итерации цикла 2, частиц (каждый процессор работает со своей частицей). Каждый процессор получает пробные координаты всех np испытываемых частиц (межпроцессорный обмен MPI_AllGather [2]).

Далее каждый процессор дважды рассчитывает энергию взаимодействия "своей" испытываемой частицы с ее окружением (вложенные циклы 3 и 4). Если партнер по взаимодействию не входит в число молекул, испытываемых на данной итерации цикла 2, или если он из числа испытываемых частиц, но его порядковый номер в np группе больше, чем у "своей" испытываемой, то вычисляются значения энергии для пробного и для исходного положения "своей" испытываемой частицы. Для "чужой" испытываемой частицы оба раза используется ее исходное положение. Если партнер тоже испытываемая частица и ее порядковый номер в np группе меньше, чем у испытываемой в данном процессоре, то сначала рассчитывается энергия для случая, когда обе частицы в пробном положении, а затем для случая, когда "своя" испытываемая частица в исходном, а партнер в пробном положении.

Рассчитанные значения энергий каждый процессор сохраняет в специальном массиве из np пар элементов. Номер пары элементов массива соответствует номеру испытываемой частицы. Пара с номером "своего" процессора содержит суммарные значения энергий взаимодействия "своей" частицы со всеми партнерами, не являющимися испытываемыми частицами. После завершения цикла 3 все эти массивы пересылаются в 0-й процессор (MPI_Gather [2]). Общий массив, сформированный таким способом в 0-м процессоре, содержит компоненты приращений энергии системы для всех вариантов исходов испытаний np частиц, выполняемых одновременно на данной итерации цикла 2.

"Нулевой" процесс выполняет нераспараллеливаемую "третью группу" операторов в двойном цикле по np частицам. Во внутреннем цикле суммируются приращения (разности соответствующих пар элементов объединенного в 0-м процессе массива) энергии взаимодействия конкретной испытываемой частицы с партнерами, не принимавшими участия в испытаниях, а также с остальными испытываемыми частицами с учетом исходов испытаний, установленных на предыдущих итерациях внешнего цикла этой группы. Значение ΔE , полученное по завершении внутреннего цикла, используется по стандартной процедуре Метрополиса и др. [5] для принятия

решения об исходе испытания данной частицы. Результат испытания (1 — принять пробную конфигурацию, 0 — оставить исходную) заносится в индикаторный массив из np элементов. После завершения внешнего цикла "третьей группы" операторов индикаторный массив рассылается всем процессорам (MPI_Bcast). Рассылку результата испытания можно совместить с рассылкой пробных координат испытуемых частиц на очередной итерации цикла 2. Получив копию индикаторного массива, все процессоры обновляют координаты частиц, изменивших свое положение. Цикл 2 начинает очередную итерацию со следующей порцией из np частиц.

При изобарно-изотермическом моделировании (NpT -ансамбль [1]) объем системы становится еще одной переменной, определяющей (в совокупности с координатами частиц) мгновенную конфигурацию системы. Новые значения объема "разыгрываются" на основе метрополисовской выборки каждую 20-ю итерацию цикла 1. Величину максимального изменения длины ребра элементарной ячейки моделируемой системы при "испытании" пробного значения объема можно оценить по формуле: $\Delta L = 0,1 \cdot \sqrt{\frac{T}{L}} + 0,0001 \text{ \AA}$. Здесь T — абсолютная температура;

L — средняя длина ребра элементарной ячейки модели (в \AA), соответствующая равновесному значению плотности моделируемой жидкости при данной температуре. Расчет приращения энтальпии системы выполняется всеми процессорами (при этом каждый процессор "обрабатывает" свою порцию из N/np частиц, определяемую по его номеру) и совмещен с составлением списков ближайших соседей для всех частиц (партнеры по взаимодействиям, для цикла 3, межпроцессорный обмен MPI_AllGather).

Две частицы считаются ближайшими соседями, если расстояние между их центрами не превосходит R_b на момент составления списка. $R_b = R_c + R_1 + R_2 + 2r$. Здесь R_c — расстояние (в данном случае оно равно 14 \AA), свыше которого взаимодействие между двумя силовыми центрами обнуляется; R_1 и R_2 — размеры частиц (радиусы их минимальных охватывающих сфер; для модели TIP4P2005 [4] молекулы воды — $R_1 = R_2 = 0,757 \text{ \AA}$); r — максимальное расстояние, на которое может сместиться центр частицы за 20 итераций цикла 1. Длина ребра куба, вмещающего моделируемую систему, не должна быть меньше удвоенного значения R_b . На практике, благодаря возможности контролировать перемещения частиц по координатам, в качестве ближайших соседей частицы достаточно учитывать лишь частицы, попавшие в описанную вокруг нее сферу радиуса R_b , симметрично усеченную кубом с длиной ребра $D_b = 2(R_c + R_1 + R_2 + 2r_x)$. Здесь r_x — максимальная величина, на которую может измениться каждая из трех декартовых координат центра частицы за 20 итераций цикла 1. В данной работе $r_x = 1 \text{ \AA}$. Такой подход к определению партнеров частиц по взаимодействиям позволяет не только сократить их количество (а тем самым и длину цикла 3), но и заметно уменьшить минимальный размер моделируемой системы (длина цикла 2), не снижая величину R_c .

Если расстояние между двумя силовыми центрами (i и j), принадлежащими разным частицам, не превосходит R_c , то энергия их ван-дер-ваальсовых взаимодействий вычисляется по формуле $E_{vdw}(r) = 4\varepsilon[(\sigma/r)^{12} - (\sigma/r)^6] - 4\varepsilon[(\sigma/R_c)^{12} - (\sigma/R_c)^6]$, где r — расстояние между силовыми центрами. В свою очередь, энергия электростатических взаимодействий этих силовых центров рассчитывается согласно методу обобщенного реактивного поля [6] по формуле: $E_e(r) = (q_i q_j)[1/r - 12/(5R_c) + 4r^2/R_c^3 - 3r^3/R_c^4 + 2r^5/(5R_c^6)]$. Параметры ε , σ и парциальные заряды на силовых центрах (q_i и q_j) взяты из работы [4].

Тестирование быстродействия программы, сочетающей распараллеливание циклов 2 и 3, проводили на суперкомпьютере "Ломоносов" в МГУ, г. Москва (<http://parallel.ru/cluster>). Алгоритм этой программы организован таким образом, что каждому из np_2 процессоров, задействованных для "одновременного" выполнения процедуры Метрополиса с очередной группой из np_2 частиц (цикл 2), в расчетах приращения энергии системы (цикл 3) помогают еще по np_3-1 процессору. Таким образом, задача может быть запущена на $np = np_2 \cdot np_3$ процессорах. Соответствующие группы процессоров формируются вызовами подпрограмм MPI_Comm_Split [2] в начале программы. На стадии испытания пробного значения объема системы и при составле-

нии списков ближайших соседей частиц также используются все np процессоры. Моделируемая система кубической формы с периодическими граничными условиями содержала 14203 молекулы воды (TIP4P2005 [4]) при температуре 290 К и атмосферном давлении.

В табл. 1 указано время выполнения 2000 итераций цикла 1 при использовании различных сочетаний числа процессоров, участвующих в распараллеливании циклов 2 и 3 (np_2 и np_3 соответственно). Видно, что распараллеливание только цикла 2 (при $np_3 = 1$) позволяет добиться более высоких результатов, чем алгоритм с распараллеливанием только цикла 3 (при $np_2 = 1$).

Причина этого в том, что из-за ограничения радиуса взаимодействия между частицами количество ближайших соседей становится во много раз меньше числа частиц в модели. Так, в рассматриваемом примере до распараллеливания длина цикла 2 более чем в 15 раз превосходит длину цикла 3.

При увеличении числа процессоров порции вычислительной работы, приходящиеся на отдельный процессор, неуклонно уменьшаются, а время, затрачиваемое на пересылку данных, наоборот, растет. Резкое падение эффективности параллельных вычислений происходит, когда время вынужденного простоя процессоров (ожидание данных от других процессоров или момента завершения выполнения нераспараллеливаемого участка программы) начинает отнимать заметную долю. Большая длина цикла 2 дает возможность разбить его на большее количество порций, чем у цикла 3, без существенного снижения эффективности использования процессоров.

Программа, в которой распараллелены оба цикла (2 и 3), позволяет добиться еще более высоких результатов за счет выбора оптимального соотношения количества процессоров, участвующих в вычислениях в цикле 2 или 3. Например (см. табл. 1), при расчетах с использованием 512 процессоров можно организовать разбиение цикла 2 на 64 порции, а цикла 3 — на 8. С практической точки зрения для рассматриваемой в данном разделе модельной системы из 14203 молекул воды TIP4P2005 [4] самым оптимальным является вариант расчета на 64 процессорах по схеме $np_2 = 32$, $np_3 = 2$ (ускорение в 45 раз по сравнению с нераспараллеленной программой, где $np_2 = np_3 = 1$). Здесь np_2/np_3 почти равно отношению длин соответствующих циклов.

Цикл 1. В процессе выполнения цикла 1 происходит формирование представительной выборки мгновенных конфигураций изучаемой системы (генерируются в цикле 2), а также расчет усредняемых по этой выборке значений искомым характеристик моделируемой жидкости. В отличие от внутренних циклов (2—4), длина цикла 1 изначально неизвестна. Причина этого — медленная сходимости средних значений, рассчитываемых в ходе компьютерного моделирования с использованием цепей Маркова, обусловленная сильной корреляцией в последовательности мгновенных конфигураций, различающихся положением не более чем одной частицы. Некоторого ускорения сходимости можно добиться [7], включая в выборку только конфигурации, получаемые в конце цикла 2 на каждой 100-й или 200-й итерации цикла 1. Дополнительное преимущество такого подхода — экономия процессорного времени за счет соответствующего уменьшения частоты выполнения расчетов значений усредняемых величин.

На практике приходится выполнять работу последовательными порциями фиксированной длины. Обычно расчет завершается, когда частные средние, рассчитанные по таким порциям, перестают проявлять какие-либо систематические отклонения в большую или меньшую сторону.

Т а б л и ц а 1

Время (с), затрачиваемое на выполнение 2000 итераций цикла 1 для системы, содержащей 14203 молекулы воды. Приведены значения для различных сочетаний количества процессоров, задействованных в распараллеливании цикла 2 (np_2) и цикла 3 (np_3)

np_3	np_2							
	1	2	4	8	16	32	64	128
1	12793	5932	3317	1731	959	513	299	291
2	5921	3316	1585	906	478	286	183	234
4	3347	1600	924	487	295	187	131	208
8	1661	927	540	311	211	142	112	191
16	1091	637	378	236	178	127	104	185
32	799	526	317	208	176	121	102	185
64	721	510	317	218	180	123	105	—

К этому моменту суммарная длина цикла l может превысить сто миллионов итераций. Однако даже столь долгие расчеты не могут гарантировать надежность результатов, поскольку моделируемая система может оказаться "запертой" в непредставительной области конфигурационного пространства на все время вычислительного эксперимента.

Повышению результативности "сканирования" конфигурационного пространства служит группа методов, развиваемых в рамках идеологии моделирования "обобщенных ансамблей" [8]. Этот подход позволяет быстрее перемещаться в многомерном конфигурационном пространстве исследуемой системы, преодолевая энергетические барьеры между метастабильными состояниями, что существенно ускоряет формирование представительной выборки для расчетов искомых характеристик. В этом случае параллельные вычисления могут быть применены с очень высокой эффективностью, так как обобщенный ансамбль можно реализовать путем одновременного моделирования нескольких экземпляров изучаемой системы при разных температурах [9]. В настоящее время автор выполняет расчеты свойств воды в широком диапазоне температур с использованием разработанной им суперкомпьютерной программы для моделирования "многотемпературного" NpT -ансамбля с распараллеливанием циклов 2—4.

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ ДЛЯ "БОЛЬШИХ МОДЕЛЕЙ"

Принцип репликации данных, лежащий в основе упомянутых выше алгоритмов, является их ахиллесовой пятой. Каждый процессор хранит (и своевременно обновляет) координаты, а также списки партнеров по взаимодействиям для *всех* частиц моделируемого объекта. С ростом размера модели эффективность вычислений неуклонно снижается, несмотря на адекватное увеличение количества процессоров. Даже в таком режиме (задача хоть и медленно, но считается) увеличение размера модели возможно лишь до пределов, определяемых объемом оперативной памяти, доступной отдельному процессору. Для современного оборудования максимальной является модель, содержащая $\sim 10^6$ силовых центров, взаимодействующих в среднем с 500 партнерами.

В данной ситуации переход к распределенным вычислениям является единственным решением "проблемы роста", осуществляя разбиение одной большой задачи на несколько маленьких. Следует поддерживать "распределенность" на всех стадиях работы с изучаемым объектом, включая генерацию начальной конфигурации, выполнение процедуры Метрополиса, вывод контрольной информации в процессе расчета, запись/чтение данных для рестарта в "пакетном" режиме, построение изображений и т.п. При таком подходе размер модели будет ограничиваться только количеством доступных пользователю вычислительных узлов суперкомпьютера, а не ресурсами отдельного процессора. Следует отметить, что алгоритмы распределенных вычислений методом молекулярной динамики [10] имеют очень мало общего с их реализацией методом Монте-Карло [11], поскольку за одну итерацию конечно-разностных уравнений Ньютона изменяются положения всех частиц, а за один шаг стандартной процедуры Метрополиса — только одной из них. Тем не менее название "доменная декомпозиция" используется в обоих случаях, указывая на универсальность концепции.

При соблюдении некоторых условий (короткодействующие потенциалы) можно организовать параллельное выполнение процедуры Метрополиса одновременно для нескольких частиц [12]. Для этого элементарная ячейка кубической формы, содержащая моделируемую систему, представляется в виде совокупности кубов меньшего объема (P -кубы), образующих регулярную пространственную решетку из процессорных "доменов". Каждый процессор занят расчетами движения молекул "своего" P -куба, периодически обмениваясь необходимыми данными с соседними (в пространстве моделируемой ячейки) процессорами о перемещениях молекул как внутри P -кубов, так и между ними.

Статистическая независимость одновременных испытаний обеспечивается дополнительным разбиением пространства P -куба на 8 L -кубов меньшего размера. Длина ребра L -куба не должна быть меньше величины $1/2D_b$ (см. выше). В ходе расчетов каждый процессор "работает"

с частицами только одного L -куба, затем переходит к следующему и т.д., циклически по всем L -кубам. Последовательность выбора L -кубов для всех процессоров одинакова. При такой организации вычислений любая пара силовых центров испытываемых частиц всегда разделена расстоянием большим, чем радиус обнуления их взаимодействий R_c . Более подробное описание этого алгоритма представлено в работах [12, 13].

Рассматриваемый в этом разделе подход к организации распределенных вычислений практически полностью устраняет зависимость объема вычислений (в том числе и при межпроцессорных обменах информацией), приходящегося на отдельный процессор, от размера задачи. В то же время общая доля нераспараллеливаемого кода при таких расчетах очень мала (участок, где 0-й процессор "выносит решение": принять или отвергнуть новую величину объема всей модели). В таких условиях эффективность алгоритмов следует оценивать с позиций, изложенных в работе [14], зафиксировав отношение объема модели к количеству используемых для ее расчета процессоров, а не размер задачи (как в тестах программ из предыдущей главы, см. рис. 1 и табл. 1).

Изучение производительности программы, использующей доменную декомпозицию для распределенных вычислений методом Монте-Карло, проводили на суперкомпьютере "Ломоносов" в МГУ, г. Москва (<http://parallel.ru/cluster>). Была выполнена серия расчетов 7 систем различного объема, содержащих в элементарной ячейке кубической формы (периодические граничные условия, $R_c = 10 \text{ \AA}$) только молекулы воды. В этом случае каждая система может быть разбита пространственной решеткой на k^3 P -кубов (доменов) *одинакового* размера (здесь $k = 2, 4, 6, 8, 10, 16$ и 20). Каждый P -куб содержит в среднем 1156 молекул воды четырехцентровой модели TIP4P2005 [4]. Для расчета конкретной системы используется столько же процессоров, сколько в ней P -кубов. Так, систему минимального размера ($2 \times 2 \times 2$ P -кубов, 36 980 силовых центров, длина ребра ячейки $65,1 \text{ \AA}$) рассчитывали на 8 процессорах, а самую большую ($20 \times 20 \times 20$ P -кубов, 36 982 832 силовых центра, 651 \AA) — на 8000 (т.е. 20^3).

На рис. 3 приведено время выполнения 2000 итераций цикла 1 для "водных кубов" различного объема в NpT -ансамбле при температуре 290 К и атмосферном давлении. Модель минимального размера, составленная из 8 доменов, представляет собой особый случай с уменьшенным вдвое количеством межпроцессорных коммуникаций (в двухэлементной по каждой координатной оси решетке левый сосед одновременно является и соседом справа из-за периодических граничных условий). Оказалось, что при увеличении объема модели в 125 раз (с 64 P -кубов до 8000) соответствующие затраты времени расчета возрастают всего лишь на 7%, если в той же пропорции увеличить и количество процессоров. Таким образом, использование алгоритмов на основе доменной декомпозиции позволит исследователям изучать молекулярные объекты очень большого размера, оставаясь в привычных рамках затрат календарного времени.

Корректный учет электростатических взаимодействий обеспечивается применением метода обобщенного реактивного поля [6], моделирующего влияние окружающей среды, находящейся за пределами сферы действия "укороченных" потенциальных функций. При этом минимальный размер ребра P -куба не должен быть меньше диаметра сферы действия укороченного электростатического потенциала. Иными словами, именно выбор значения R_c устанавливает нижнюю границу "стоимости" расчета (время выполнения одной итерации цикла 1), запрещая дальнейшее дробление доменов.

Эффективность вычислений можно существенно повысить, используя декомпозицию циклов 2—4 (см. предыдущую главу о распараллеливании) при расче-

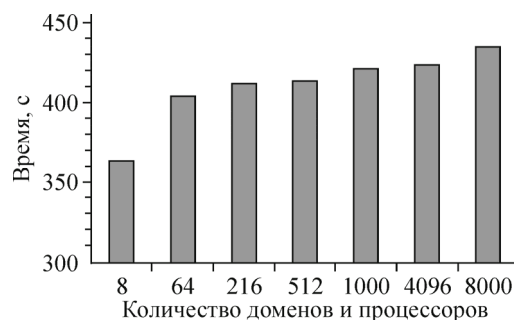


Рис. 3. Время выполнения 2000 итераций цикла 1 для ряда "водных кубов" различного объема (NpT -ансамбль, периодические граничные условия). Распределенные вычисления с использованием доменной декомпозиции (см. пояснения в тексте).

Т а б л и ц а 2

Время (с), затрачиваемое на выполнение 2000 итераций цикла 1 для системы, содержащей 14203 молекулы воды. Использовалась доменная декомпозиция типа $2 \times 2 \times 2$. Приведены значения для различных сочетаний количества процессоров, задействованных в распараллеливании цикла 2 (np_2) и цикла 3 (np_3)

np_3	np_2							
	1	2	4	8	16	32	64	128
1	1388	720	360	190	110	60	38	49
2	727	381	196	114	65	41	29	40
4	455	233	139	72	46	31	23	—
8	270	154	80	50	34	25	—	—
16	161	97	56	37	27	—	—	—
32	110	72	46	31	—	—	—	—
64	87	69	43	—	—	—	—	—

смаатривали в предыдущей главе при тестировании программы без доменной декомпозиции (см. табл. 1).

В табл. 2 указано время выполнения 2000 итераций цикла 1 при использовании в "доменной команде" различных сочетаний числа процессоров, участвующих в распараллеливании циклов 2 и 3 (np_2 и np_3 соответственно). Каждой клетке табл. 2 соответствует расчет на $8 \cdot np_2 \cdot np_3$ процессорах.

Прежде всего следует отметить, что введение доменной декомпозиции в программы с распараллеливанием циклов 2—4 дает ощутимый прирост скорости расчетов уже в своем самом простом варианте (ср. клетку $np_2 = np_3 = 1$ в табл. 2 с клеткой $np_2 = 4$, $np_3 = 2$ в табл. 1). В "экстремальной" для рассматриваемой модели комбинации из 2048 процессоров доменная декомпозиция дает более чем четырехкратное превосходство в быстродействии по сравнению с распараллеливанием без доменов (ср. клетку $np_2 = 64$, $np_3 = 4$ в табл. 2 с клеткой $np_2 = 64$, $np_3 = 32$ в табл. 1; жирный шрифт). Без использования параллельных и распределенных вычислений этот же тест будет длиться в 556 раз дольше (12 793 с вместо 23).

В настоящий момент автором ведется разработка программы, использующей доменную декомпозицию с распараллеливанием циклов по частицам, для моделирования обобщенных ансамблей [8, 9]. Аналогичный подход уже продемонстрировал высокую эффективность моделирования в рамках метода молекулярной динамики [15].

ЗАКЛЮЧЕНИЕ

Выше были рассмотрены способы многократного ускорения расчетов методом Монте-Карло. Использование параллельных и распределенных вычислений позволяет в течение часа выполнять на достаточно мощном суперкомпьютере сотни тысяч циклов испытаний по всем частицам даже для моделей, составленных из многих миллионов атомов. Однако само по себе быстродействие компьютерных программ еще не гарантирует представительности выборки конфигураций, необходимой для получения достоверных результатов. Более того, несмотря на отдельные успехи в развитии теории сходимости вычислительных алгоритмов молекулярного моделирования, соответствующие критерии до сих пор не выработаны [16]. В этих условиях модификация компьютерных программ автора данной статьи в духе идеологии обобщенных ансамблей [8] позволит поднять молекулярное моделирование методом Монте-Карло на качественно новый уровень.

тах энергии взаимодействия частиц P -куба между собой и с частицами соседних кубов. В этом случае каждому домену будет сопоставляться уже не один процессор, а целая "команда" из $np = np_2 \cdot np_3$ процессоров. Таким образом, k -доменная задача будет выполняться на $k \cdot np$ процессорах суперкомпьютера.

Изучение производительности программы, сочетающей доменную декомпозицию с распараллеливанием циклов 2—4, проводили на суперкомпьютере "Ломоносов" в МГУ, г. Москва (<http://parallel.ru/cluster>). Моделируемая система кубической формы с периодическими граничными условиями содержала 14 203 молекулы воды (TP4P2005 [4]) при температуре 290 К и атмосферном давлении. Элементарную ячейку модели разбивали на 8 P -кубов (по два домена вдоль каждой координатной оси). Точно такую же систему рас-

СПИСОК ЛИТЕРАТУРЫ

1. *Allen M.P., Tildesley D.J.* Computer simulation of liquids. – N.Y.: Oxford University Press, 1987.
2. *Snir M., Otto S., Huss-Lederman S., Walker D., Dongarra J.* MPI: The Complete Reference. – Boston: MIT Press, 1996.
3. *Wichmann B.A., Hill I.D.* // *Comput. Statist. Data Anal.* – 2006. – **51**. – P. 1614 – 1622.
4. *Abascal J.L.F., Vega C.* // *J. Chem. Phys.* – 2005. – **123**. – P. 234505 (1 – 12).
5. *Metropolis N.A., Rosenbluth A.W., Rosenbluth M.N. et al.* // *J. Chem. Phys.* – 1953. – **21**. – P. 1087 – 1092.
6. *Hummer G., Soumpasis D.M., Neumann M.* // *J. Phys.: Condens. Matter.* – 1994. – **6**. – P. A141 – A144.
7. Вуд В. Физика простых жидкостей. Экспериментальные исследования / Ред. Г. Темперли, Д. Роулинсон, Д. Рашбрук. – М.: Мир, 1973. – P. 275 – 394.
8. *Okamoto Y.* // *J. Mol. Graphics Modell.* – 2004. – **22**. – P. 425 – 439.
9. *Hukushima K., Nemoto K.* // *J. Phys. Soc. Jpn.* – 1996. – **65**. – P. 1604 – 1608.
10. *Plimpton S.* // *J. Comput. Phys.* – 1995. – **117**. – P. 1 – 19.
11. *Heffelfinger G.S., Lewitt M.E.* // *J. Comput. Chem.* – 1996. – **17**. – P. 250 – 265.
12. *Теплухин А.В.* // Математическое моделирование. – 2004. – **16**, № 11. – С. 15 – 24.
13. *Калугин М.Д., Теплухин А.В.* // Журн. структур. химии. – 2009. – **50**. – С. 878 – 889.
14. *Gustafson J.L.* // *Commun. ACM.* – 1988. – **31**. – P. 532 – 533.
15. *Slim H.A., Wilson M.R.* // *J. Chem. Theory Comput.* – 2008. – **4**. – P. 1570 – 1575.
16. *Zuckerman D.M.* // *Annu. Rev. Biophys.* – 2011. – **40**. – P. 41 – 62.