

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ МОЛЕКУЛЯРНОЙ ДИНАМИКИ

А.В. Комолкин, В.С. Неверов

1. Введение

Прогресс компьютерных методов моделирования молекулярной динамики тесно связан с переходом к моделированию наноразмерных систем на наносекундных масштабах времени. Этот переход, с одной стороны, вызван необходимостью моделировать медленные процессы, такие как конформационные переходы в больших белковых молекулах, моделирование взаимодействия макромолекул с относительно небольшими молекулами в растворах (молекулярный докинг), взаимное упорядочение молекул в жидких кристаллах и другие. С другой стороны, большинство приложений компьютерного моделирования (в том числе и указанные выше) требуют использования атомарных моделей молекул, то есть требуют знания расположения всех атомов в молекулах, включая атомы водорода, и вычисления энергии атом-атомных взаимодействий в процессе моделирования эволюции молекулярных систем.

Современные исследователи ожидают, что программы компьютерного моделирования будут оперировать с системами с размерами до нескольких десятков нанометров и выполнять моделирование на протяжении десятков наносекунд. Например, кубическая ячейка с размером ребра 30 нм будет содержать около 1 миллиона молекул воды, то есть около 3 миллионов атомов. В то же время фрагмент ДНК, длиной 30 нм, является «коротким».

Моделирование молекулярной динамики заключается в решении уравнений движения атомов, рассматриваемых как материальные точки и взаимодействующих на основе классических потенциалов (не квантово-механических). Алгоритм моделирования можно кратко представить так:

1. в текущий момент времени t по координатам всех атомов вычислить энергию взаимодействий и силы, направленные на каждый атом;
2. в следующий момент времени $t + \Delta t$ по силам вычислить скорости и координаты атомов, и этот момент времени сделать текущим;
3. в случае моделирования ансамблей NVT или NpT (с постоянной температурой и постоянными температурой и давлением соответственно) корректировать температуру (скорости атомов) и давление (объем системы);
4. проанализировать состояние молекулярной системы в текущий момент времени;
5. выполнить следующий шаг моделирования, перейдя к пункту 1.

Очевидно, что параллелизм программы моделирования может быть заключен только внутри одного шага моделирования, так как исходные данные для вычислений в пункте 1 изменяются в пунктах 2 и 3. Пункт 4 (анализ состояния) обычно заключается в записи текущей конфигурации системы на диск для последующего анализа различными программами, поэтому мы не будем здесь рассматривать параллелизм в процедурах анализа. Это связано дополнительно и с тем, что анализ выполняется значительно быстрее собственно моделирования (пункты 1-3) и часто выполняется в последовательных программах.

Переформулируя закон Амдала с учетом накладных расходов на передачу данных, можно полное время работы программы на P процессорах записать как:

$$T_{\text{полное}}(P) = T_{\text{послед}} + \frac{T_{\text{нап}}}{P} + T_{\text{обм}}(P) \quad (1)$$

где в правой части указаны соответственно времена последовательной части программы, параллельной части и время обмена данными между процессорами, которое может возрастать с увеличением их числа. Причиной плохой масштабируемости программы может стать как большое время последовательной её части, так и существенное возрастание времени обмена данными при неправильном выборе методов распараллеливания.

Рассмотрим подробнее методы распараллеливания программы.

2. Параллелизм в алгоритмах вычисления энергий и сил

В данном параграфе рассматриваются возможные варианты параллельного вычисления энергии атом-атомных взаимодействий.

В общем случае потенциальная энергия взаимодействий зависит от координат всех N атомов $\{\vec{R}_i\}, i = 1 \dots N$ системы:

$$E_{\text{пот}} = E(\{\vec{R}_i\}) \quad (2)$$

Сила, действующая на атом, вычисляется с помощью оператора Гамильтона по координатам данного атома, то есть выражается в виде:

$$\vec{F}_i = -\vec{\nabla}_i E_{nom} = -\left(\vec{e}_x \frac{\partial}{\partial x_i} + \vec{e}_y \frac{\partial}{\partial y_i} + \vec{e}_z \frac{\partial}{\partial z_i} \right) E_{nom} \quad (3)$$

Поскольку атомы объединены в молекулы (а мы будем в данной работе рассматривать моделирование молекул в растворах), то полную энергию системы можно будет подразделить на две части — внутримолекулярных и межмолекулярных взаимодействий.

Не совсем очевидно, но энергия внутримолекулярных взаимодействий может зависеть от координат атомов в других молекулах — это справедливо для поляризуемых моделей. В таких моделях частичные электрические заряды на атомах могут меняться в зависимости от расположения внешних (по отношению к данной молекуле) зарядов, то есть некоторого внешнего электрического поля. Это поле должно быть самосогласованным, то есть изменение частичных зарядов на атомах данной молекулы влияет на распределение зарядов на соседних молекулах. После расчета электрического поля корректируются заряды на атомах, исходя из их связей только в каждой конкретной молекуле. Процедура самосогласования электрического поля и частичных зарядов носит итерационный характер и затрагивает все атомы (все молекулы) в моделируемой системе. Обращаем внимание, что при самосогласовании каждый процессор должен знать заряды на всех атомах системы, то есть, после каждой итерации придется выполнять передачу новых значений электрических зарядов всем процессорам.

После такого самосогласования можно считать, что заряды на атомах на данном шаге моделирования фиксированы. Это позволяет в дальнейшем полагать, что энергия внутримолекулярных взаимодействий должна зависеть только от координат атомов данной молекулы. В таком предположении параллелизм вычисления внутримолекулярных взаимодействий должен основываться на парадигме *параллелизма задач*, поскольку в системе могут присутствовать молекулы разного вида (с разным числом атомов), а распараллеливание алгоритма должно выполняться по молекулам.

Если модель не учитывает поляризацию, то этап самосогласования электрического поля отсутствует, что резко упрощает и ускоряет вычисления. Подавляющее большинство современных моделей не учитывает поляризацию молекул.

Межмолекулярная энергия складывается обычно из энергии невалентных (Ван-дер-Ваальсовских) взаимодействий, кулоновских (электростатических) взаимодействий и энергии водородной связи. Все эти вклады являются парными, то есть полная энергия зависит аддитивно от энергии взаимодействия каждой пары атомов i и j , которые должны принадлежать разным молекулам:

$$E_{nom} = \sum E_{ij} ; E_{ij} = E(\vec{R}_i - \vec{R}_j) ; \vec{F}_{ij} = -\vec{F}_{ji} \quad (4)$$

В случае использования поляризуемых моделей заряды можно считать фиксированными после процедуры согласования, то есть можно использовать приближение парных взаимодействий для вычисления всех межмолекулярных сил. В этой части алгоритма удобно также пользоваться парадигмой *параллелизма задач* для вычисления половины (точнее, меньше, чем половины) матрицы сил $[\vec{F}_{ij}]$ в двойном цикле по $i = 1 \dots N - k$ и $j = i + k'(i) \dots N$ (здесь k и $k'(i)$ должны быть такими, чтобы i и j принадлежали разным молекулам).

Электростатический потенциал, в отличие от других перечисленных выше, является дальнедействующим и вычисляется дополнительно методом Эвальда. Не вдаваясь в математические подробности, которые можно найти, например, в классической монографии [1], укажем, что алгоритм метода Эвальда состоит из тройного цикла по копиям (образам) центральной ячейки моделирования, находящимся внутри шара (это накладывает ограничения на параметры циклов: $l^N + m^N + n^N \leq const$), тело которого содержит четвертый вложенный цикл по всем атомам. Существенным фактором, ограничивающим быстродействие четверного цикла, является необходимость вычисления функций \sin и \cos от аргументов

$lx_i + my_i + nz_i$. Для ускорения вычислений обычно применяют табулирование значений $sx_{l,i} = \sin(lx_i)$, $cx_{l,i} = \cos(lx_i)$ и всех аналогичных для использования их в формулах для тригонометрических функций суммы трёх углов. Дополнительное ускорение, связанное с вычислением $\sin(k\alpha)$ через $\sin(\alpha)$, приводит к недопустимому падению точности вычислений.

Внутренний цикл по атомам может быть распараллелен на основе парадигмы *параллелизма данных*, но более эффективным является подход к распараллеливанию наиболее внешнего цикла на основе парадигмы *параллелизма задач*.

Параллельная организация тройного цикла по шару с учетом оптимизации табулирования тригонометрических функций и равномерной загрузки процессоров являлась одной из целей работы. Реализован подход на основании поиска наиболее длинных непрерывных цепочек (столбиков) ячеек, для

которых меняется только один параметр цикла, а два других имеют фиксированные значения. Очевидно, что самая длинная цепочка проходит через центр шара. Все найденные цепочки сортируются по убыванию длины и распределяются между параллельными процессорами так, чтобы суммарное число ячеек на всех процессорах было приблизительно одинаково. Предполагается, что время вычисления энергии и сил в каждой ячейке одинаковы. Таким образом, тройной цикл заменяется на на каждом процессоре на двойной цикл по цепочкам ячеек, приписанным данному процессору, и собственно по ячейкам. При этом в каждой цепочке ячеек изменяется только один из индексов тройного цикла, что позволяет оптимизировать вычисление тригонометрических функций через сумму двух углов — фиксированного и изменяющегося. Показано, что данный алгоритм является хорошо масштабируемым на системе до 40 процессоров, на которой выполнялось тестирование и проводились научные вычисления. Основное преимущество в масштабируемости достигнуто за счет оптимизации вычислений тригонометрических функций в цепочках ячеек.

Например, в шаре радиуса 7 содержатся 73 цепочки длиной от 13 (максимально) до 5 (минимально) ячеек и только 2 «цепочки» по одной ячейке. При распределении нагрузки на 40 процессоров 33 из них получают от 18 до 20 ячеек в двух цепочках, и оставшиеся 7 процессоров — от 13 до 14 ячеек. Время выполнения параллельного алгоритма соответствует времени подготовки (табуляции функций) и времени вычисления сил в 20 ячейках.

Другой хорошо масштабируемый параллельный метод прямого вычисления дальнедействующих сил (по формуле Кулона) не использует тригонометрические функции, но использует значительно большее число копий центральной ячейки (радиус шара должен выбираться до 100). Несмотря на то, что этот алгоритм линейно масштабируется с увеличением числа процессоров и требует меньше времени для вычислений в каждой ячейке, суммарное время работы алгоритма больше, чем реализованный в данной работе параллельный метод Эвальда. Мы оцениваем, что на системах до 128 процессоров предложенный нами метод будет работать быстрее.

В результате параллельного вычисления энергии взаимодействий и сил с использованием парадигмы *параллелизма задач*, каждый из процессоров вычисляет частичную силу, действующую на каждый атом. Для вычисления полной силы необходимо вычислить сумму частичных сил, что в терминах парадигмы *параллелизма задач* (или *передачи сообщений*, *message passing*) называется операцией уменьшения размерности (*reduce*). Такая операция связана с передачей всех массивов одному главному процессору и вычислению суммы. При оптимизации обмена данными по графу в виде бинарного дерева со сложением промежуточных массивов, время обмена растет пропорционально $\log_2(P)$.

Дальнейшая оптимизация вычисления сил в программе связана с балансировкой нагрузки на процессоры и с уменьшением числа процессоров, выполняющих вычисления по методу Эвальда. Такая оптимизация должна затронуть основной алгоритм программы, в котором необходимо будет выполнять параллельно на разных наборах процессоров вычисление Ван-дер-Ваальсовых сил и электростатических методом Эвальда с последующей балансировкой и перераспределением нагрузки, если одна из процедур будет выполнена быстрее другой.

3. Параллелизм в алгоритмах перемещения атомов

Перемещение атомов зависит только от скоростей атомов и сил, действующих на эти атомы. Этот алгоритм может быть реализован в парадигме *параллелизма данных* при обработке массивов координат атомов. При распараллеливании этой части алгоритма молекулярной динамики надо учитывать и фактор времени передачи данных в формуле (1).

Если включить в алгоритм молекулярной динамики, приведенный в 1-ом параграфе, передачу данных, то полный алгоритм будет выглядеть так:

1. разослать всем процессорам из главного координаты всех атомов;
2. вычислить параллельно частичные силы, действующие на атомы;
3. собрать все частичные силы на главном процессоре и вычислить полные силы;
4. разослать всем процессорам из главного полные силы;
5. выполнить перемещение атомов с учетом различных ограничений — на длины химических связей, на моделирование условий постоянной температуры и давления;
6. собрать координаты на главном процессоре;
7. сохранить координаты на данном шаге моделирования на диске;
8. перейти к пункту 1.

Пункты 2 и 5, как сказано выше, могут быть реализованы в различных парадигмах. Основное время вычислений занимает расчет сил. Но при увеличении числа процессоров возрастает суммарное время передачи данных (пункты 1, 3, 4 и 6).

В таблицу 1 сведены данные о времени моделирования 1000 молекул воды на кластере двухпроцессорных ЭВМ с использованием системы передачи данных MPICH с механизмом обмена внутри узла *shared-memory*. Данные системы (как молекулярная система, так и компьютерный кластер) не являются показательными или идеальными для выполнения моделирования молекулярной динамики. При увеличении

числа молекул время вычисления сил растет практически пропорционально N^2 , а время обмена — N . Поэтому при увеличении модельной системы в 10-30 раз показатель масштабируемости увеличится.

Проведенный анализ быстродействия программы показывает, что использование всех имеющихся процессоров для перемещения атомов может оказаться нецелесообразным из-за увеличения времени обмена силами и координатами в пунктах 4 и 6 алгоритма. Другой крайний случай — использование только одного процессора для перемещения атомов может оказаться до некоторой степени эффективным, поскольку полностью исключаются пункты 4 и 6.

Необходимо отметить, что собственно перемещение атомов при использовании ансамблей NVT и NpT скрыто содержит в себе обмен данными. Для поддержания температуры и давления требуется знать средние температуру всей системы и давление, которые вычисляются по всем атомам. Если атомы будут распределены по процессорам, то объем передаваемых данных будет составлять всего один или два вещественных числа, но обмен будет затрагивать все процессоры. Время перемещения атомов с поддержанием постоянной температуры в ансамбле NVT уменьшается с ростом числа процессоров от 1 до 8 с 4,5 мс/шаг до 2,5 мс/шаг.

Анализ быстродействия показывает, что ускорение в вычислении сил составляет на 8 процессорах $273 \text{ мс} : 37 \text{ мс} \approx 7,4$, что можно рассматривать как хороший показатель. При этом ускорение вычисления короткодействующих межмолекулярных сил составило $233 \text{ мс} : 29,5 \text{ мс} \approx 7,9$, а ускорение метода Эвальда — $38,9 \text{ мс} : 7,25 \text{ мс} \approx 5,4$ (эти данные не вошли в таблицу).

Видно, что основной вклад в уменьшение масштабируемости программы вносит обмен данными. Сокращение обмена путем уменьшения числа параллельных ветвей на этапе перемещения атомов может увеличить эффективность программы. Количество процессоров необходимо подбирать (оптимизировать) для каждого типа молекулярных систем и для каждой вычислительной системы. Например, на параллельных многопроцессорных системах, возможно, будет эффективным использование всех процессоров, в то время как на кластерах типа Beowulf может быть оптимальным использование только одного узла или, даже, одного процессора.

4. Заключение

В работе предложен и реализован новый параллельный алгоритм метода Эвальда. Проанализированы причины возможной плохой масштабируемости программ молекулярной динамики. Предложены пути оптимизации быстродействия программы как с точки зрения параллельности частей (задач) алгоритма на основе усложнения парадигмы *параллелизма задач*, так и с точки зрения выполнения программы на конкретных вычислительных системах.

Данная работа частично поддержана грантами РФФИ 09-02-00010 и 09-03-01105.

Таблица 1. Время и ускорение вычислений в моделировании тестовой молекулярной системы.

Число процессоров P	Время работы (мс/шаг)		Относительное время (%)		Общее время (мс/шаг)	Ускорение
	Вычисление сил	Перемещение атомов + обмен	Вычисление сил	Перемещение атомов + обмен		
1	273	6	98	2	279	1
2	139	6	96	4	144,5	1,93
4	71	10,5	87	13	81,5	3,42
6	48,3	12,8	79	21	61,1	4,56
8	37	16,1	69,5	40,5	53,1	5,25

ЛИТЕРАТУРА:

1. M.P. Allen, D. Tildesley. Computer Simulation of Liquids. — Oxford: Clarendon Press, 1987