Доклады Международной конференции "Математическая биология и биоинформатика". *Под ред. В.Д. Лахно. Том 7. Пущино: ИМПБ РАН, 2018. Статья № e44. doi: 10.17537/icmbb18.53*

Использование параллелизма различных уровней в программе моделирования молекулярной динамики PUMA-CUDA

Лихачев И.В.[1, 2], Балабаев Н.К.[1]

[1]*ИМПБ РАН – филиал ИПМ им. М.В. Келдыша РАН*
[2]*Тульский государственный университет*

ilya_lihachev@mail.ru

Описывается создание программно-алгоритмического комплекса PUMA-CUDA с поддержкой параллельной обработки данных разных уровней: внутри-процессорного OpenMP, межпроцессного MPI, массивно-параллельного NVIDIA CUDA с поддержкой произвольного количества графических ускорителей.

Ключевые слова: *молекулярная динамика, высокопроизводительные вычисления, параллельные вычисления, GPGPU, CUDA.*

Parallelism of different levels in the program of molecular dynamics simulation PUMA-CUDA

Likhachev I.V.[1, 2], Balabaev N.K.[1]

[1]*IMPB RAS – Branch of KIAM RAS*
[2]*Tula State University*

The developing of the PUMA-CUDA software complex with the support of parallel data processing of different levels is described: inside the processor OpenMP, interprocess MPI, massively parallel NVIDIA CUDA with the support of an arbitrary number of graphics accelerators.

Key words: molecular dynamics simulation, high performance calculations, parallel computing, GPGPU, CUDA.

1. Введение

Метод моделирования молекулярной динамики используется во многих научно-исследовательский лабораториях. Размеры систем за последние годы существенно выросли и составляют порядка 10^5 атомов. Выполнение программ по молекулярной динамике, как правило, производят на специальных вычислительных кластерах, называемых нередко суперкомпьютерами.

Большинство вычислительных кластеров представляет собой массив достаточно мощных многоядерных ЭВМ, соединенных по определенной топологии. При этом каждому пользователю кластера доступно в порядке очереди некоторое количество вычислительных ядер. В последние годы получают популярность гетерогенные вычислительные среды – узлы с современным многоядерным центральным процессором и одним или несколькими графическими ускорителями, предполагающими их использование в качестве математического сопроцессора.

2. Задача моделирования молекулярной динамики

Задача моделирования молекулярной динамики состоит в интегрировании уравнений движения системы N взаимодействующих материальных точек (атомов), движение которых описывается классическими уравнениями Ньютона:

$$m_i \cdot \ddot{\mathbf{r}}_i = F_i \equiv -\frac{\partial U}{\partial \mathbf{r}_i}$$

где $I = 1,..., N$,

$$U \equiv U(\mathbf{r}_1,...,\mathbf{r}_N) = U_{\substack{\text{валентных}\\\text{связей}}} + U_{\substack{\text{валентных}\\\text{углов}}} + U_{\substack{\text{торсионных}\\\text{углов}}} +$$

$$+ U_{\substack{\text{плоских}\\\text{групп}}} + U_{VdW} + U_{qq}$$

Вычислительно задача сводится к нахождению энергии, сил, интегрирования уравнений движения, а также учета внешних воздействий и граничных условий. Для моделирования системы при заданной

Proceedings of the International Conference "Mathematical Biology and Bioinformatics". *Ed. V.D. Lakhno. Vol. 7. Pushchino: IMPB RAS, 2018. Paper No. e44. doi: 10.17537/icmbb18.53*

температуре используется столкновительный термостат [1, 2].

Решение задачи моделирования молекулярной динамики – сложный и разнородный вычислительный процесс, состоящий из нескольких этапов. Каждый этап имеет свою вычислительную сложность, свои особенности. Приведем список этапов задачи.

Таблица 1. Список основных этапов моделирования

Этапы моделирования	Сложность	max число потоков
Валентные связи, валентные углы, торсионные углы, плоские группы	N	N
Невалентные взаимодействия: Ван-дер-Ваальс и Кулон	N^2	N
Термостат	N	1
Интегрирование уравнений движения	N	N
Сбор статистической информации, консольный и файловый ввод / вывод	N (не на каждом шаге)	1

Примечание: N – число атомов в системе.

Основную часть времени программа моделирования молекулярной динамики вычисляет силы, действующие на атомы со стороны других атомов. Следует заметить, что в системе взаимодействуют друг с другом не все пары атомов, а только расположенные на расстоянии друг от друга, не превышающем радиуса взаимодействия атомов (порядка 10–15 Å, в зависимости от условий моделирования). Можно ускорить выполнение программы расчета сил, убрав из расчета невзаимодействующие пары атомов.

Проблема ускорения МД-расчетов за счет введения вспомогательных структур, описывающих взаимодействующие атомы, в литературе решается двумя способами [3]. Используют метод сканирования по пространству (≡ метод присоединенных списков [4]) или метод составления списков пар взаимодействующих атомов (≡ метод списков Верле [5]).

Алгоритм сканирования по пространству заключается в разбивке трехмерного пространства на ячейки. Зная ячейку, мы знаем соседние ячейки, знаем атомы, взаимодействующие с данным. Процесс помещения атома в ячейку имеет линейную сложность и эффективно выполняется на центральном процессоре. Метод является общепринятым при составлении программы моделирования молекулярной динамики на традиционных (центральных) процессорах.

Алгоритм составления списков взаимодействующих атомов заключается в составлении списка всех пар взаимодействующих атомов. Если нет сведений о расположении атомов системы, то при работе на центральном процессоре он имеет квадратичную сложность ($N^2/2$) от количества частиц, из-за чего проигрывает алгоритму сканирования по пространству при составлении традиционных программ. Также он использует больше памяти.

Невалентные взаимодействия занимают большую часть машинного времени процесса вычислений. Именно их нужно ускорить, используя графические процессоры в качестве математических сопроцессоров общего назначения.

В процессе анализа задачи был сделан вывод, что следующие этапы нуждаются в оптимизации машинного времени:

- вычисление невалентных взаимодействий;
- составление вспомогательных структур для ускорения расчетов невалентных взаимодействий.

3. Методы ускория расчетов

3.1. Массивно-параллельный алгоритм составления списков взаимодействующих атомов

Ускорение работы программы моделирования молекулярной динамики достигается за счет реализации алгоритма составления списков на CUDA.

Список представляется в виде зубчатого (рваного) двумерного массива. Число столбцов в массиве соответствует количеству атомов в системе. А в каждой строке записываются все соседи данного атома в области, соответствующей радиусу обрезания.

Каждый столбец массива может быть заполнен независимо от других. Задача параллелится на N потоков. При больших числах N использование графических ускорителей оправдано. Тем более, что полученный вспомогательный массив остается в памяти графического ускорителя, где и будет использоваться другими алгоритмами.

Составление списка взаимодействующих атомов достаточно производить не на каждом шаге, а один раз за несколько шагов или даже десятков шагов, когда координаты атомов изменятся существенно. Для этого, помимо радиуса взаимодействия, вводится еще один параметр – радиус запоминания.

3.2. Массивно-параллельный расчет невалентных взаимодействий

Вычисление невалентных взаимодействий занимает большую часть машинного времени. От эффективности реализации данной задачи зависит быстродействие задачи в целом.

Силы Ван-дер-Ваальса и Кулона, действующие на каждый атом со стороны его соседей, могут быть вычислены независимо. Таким образом, можно

организовать N параллельных потоков, где N – число частиц в системе.

Если N достаточно велико – порядка количества процессоров в графических ускорителях, – то целесообразность использования массивного параллелизма оправдана. Вычисления проводятся с системами от нескольких сотен до нескольких сотен тысяч атомов. Чем больше атомов, тем преимущество графических процессоров более заметно. Роль накладных расходов уменьшается, и мы уходим от квадратичной сложности задачи, присущей центральному процессору, к линейной.

Общий алгоритм решения подзадачи с использование графический ускорителей выглядит следующим образом:

1. исходные данные находятся в памяти хоста (координаты, параметры работы системы и параметры силового поля);
2. копирование данных в память устройства;
3. параллельный запуск для всех атомов функции ядра – функции вычисления силы и энергии, действующей на один атом;
4. копирование результатов в память хоста.

В GPGPU[1]-вычислениях под хостом (host) подразумевается центральный процессор и его оперативная память, а под устройством (device) – графический ускоритель со своей собственной памятью. И центральный, и графический процессоры работают только со своей памятью. Поэтому нужно производить операции обмена информацией по шине PCI-Express.

3.3. Применение двух графических ускорителей

Расчет Кулоновских и Ван-дер-Ваальсовых взаимодействий – две структурно похожие функции. Они могут выполняться независимо друг от друга. Многие современные вычислительные кластеры в случае предоставления графических ускорителей в качестве вычислительной среды имеют по два ускорителя на один хост.

Эти факты позволяют параллельно выполнять расчет Ван-дер-Ваальсовых и Кулоновских взаимодействий на двух ускорителях при помощи асинхронных средств CUDA, т.е. не используя межпотоковое и межпроцессное взаимодействие на уровне операционной системы.

Алгоритм применения двух графических карт одного хоста выглядит следующим образом.

1. Начальные данные в памяти хоста (координаты, заряды).
2. Копирование данных в память первого устройства.
3. Параллельный запуск функции ядра вычисления Кулоновских взаимодействий на первом устройстве.
4. Копирование данных в память второго устройства.
5. Параллельный запуск функции ядра вычисления Ван-дер-Ваальсовых взаимодействий на втором устройстве.
6. Копирование результатов в память хоста с первого ускорителя.
7. Копирование результатов в память хоста со второго ускорителя.

После параллельного запуска функции ядра на каком-либо ускорителе управление передается основной программе на центральном процессоре, не дожидаясь окончания работы ускорителя. Эта особенность позволяет подготовить данные для второго ускорителя и запустить на нем расчет. Ожидание готовности ускорителя происходит непосредственно при вызове функции копирования результатов из памяти ускорителя в память хоста.

Наибольший теоретический выигрыш от использования двух видеокарт – 2 раза.

3.4. Масштабируемость задачи на произвольное количество хостов

Как уже упоминалось ранее, основное количество машинного времени уходит на вычисление невалентных взаимодействий. Силы, действующие на каждый атом, можно найти независимо, параллельно на N процессорах, где N – количество атомов в системе.

Если количество атомов существенно превосходит количество CUDA-процессоров, целесообразно разделить расчёты между несколькими хостами (в данном контексте под хостом имеется в виду системный блок с парой видеокарт, а не центральный процессор с оперативной памятью).

Каждому хосту необходимо рассчитать часть системы, состоящую из N/m атомов, где N – количество атомов в системе, m – количество хостов.

Приведем алгоритм проведения расчетов молекулярной динамики на произвольном количестве хостов благодаря совместному применению технологий MPI и NVIDIA CUDA.

1. Файловый ввод / вывод и большинство вспомогательных операций осуществляет первый процесс.
2. Каждый процесс считывает параметры силового поля и параметры моделирования. У каждого процесса они одинаковы.
3. Первый процесс передает всем процессам координаты всех атомов, используя коллективную операцию MPI Broadcast.
4. Каждый хост использует два графических ускорителя для расчетов невалентных взаимодействий своей части задачи.
5. Полученные части массивов сил и энергий со всех хостов собираются на первом узле, использую коллективную операцию MPI Gather.
6. Первый узел рассчитывает валентные взаимодействия, производит термостатирование и интегрирование уравнений движения.

[1] *General-Purpose Computing on Graphics Processing Units* или Неспециализированные вычисления на графических процессорах

Таким образом, МД-расчеты могут быть масштабированы на произвольное количество хостов.

3.5. Сравнение производительности с классическими алгоритмами

Сравнение производительности описываемой гетерогенной программы моделирования молекулярной динамики PUMA-CUDA с классической программой PUMA [6–8] проводилось на одних и тех же системах (см. рис. 1). Из Protein Data Bank был выбран иммуноглобулин, связывающий белок с кодовым названием 1PGB, содержащий 853 атома. Белок был окружен водной оболочкой из заданного числа молекул воды таким образом, чтобы получить последовательно системы из 1 000, 2 000, 4 000, …, 256 000 атомов.

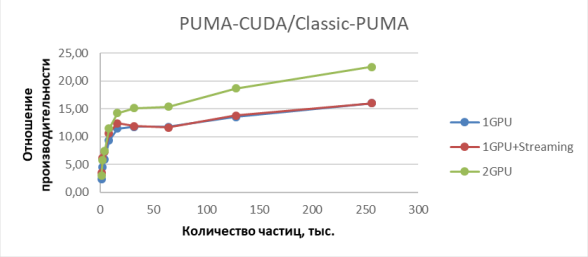

Рис. 1. Сравнение производительности гетерогенной программы моделирования молекулярной динамики PUMA-CUDA с классической программой PUMA от числа частиц.

Результаты сравнения производительности показали, что гетерогенной программой моделирования молекулярной динамики целесообразно пользоваться, начиная с нескольких тысяч атомов. Производительность растёт быстро примерно до 16 000 атомов, после чего выходит на линейную зависимость.

Особо хотелось бы отметить, что при работе с крупными системами (256 000 атомов) спад производительности не наблюдается.

Результаты сравнения производительности на разных вычислительных установках могут отличаться.

4. Благодарности

Работа выполнена с использованием оборудования Института прикладной математики им. М.В. Келдыша и оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова [9].

5. Список литературы

1. Lemak A.S., Balabaev N.K. A comparison between collisional dynamics and Brownian dynamics. *Mol. Simul.* 1995. V. 15. P. 223–231.
2. Lemak A.S., Balabaev N.K. Molecular dynamics simulation of a polymer chain in solution by collisional dynamics method. *J. Comp. Chem.* 1996.V. 17. P. 1685–1695.
3. Балабаев Н.К., Шайтан К.В. Компьютерное моделирование молекулярной динамики. В: *Методы компьютерного моделирования для исследования полимеров и биополимеров*. М.: Издательство ЛКИ, 2009. С. 35–62.
4. Allen M.P. *Computer simulation of liquids*. Oxford: Claredon Press, 1987. P. 385.
5. Verlet L. *Phys. Rev. A*. 1964. V. 139. P. 405–411.
6. Glyakina A.V., Likhachev I.V., Balabaev N.K., Galzitskaya O.V. Comparative mechanical unfolding studies of spectrin domains R15, R16 and R17. *J. Struct. Biol.* 2018. V. 201. № 2. P. 162–170.
7. Glyakina A.V., Likhachev I.V., Balabaev N.K., Galzitskaya O.V. Mechanical stability analysis of the protein L immunoglobulin-binding domain by full alanine screening using molecular dynamics simulations. *Biotechnol. J.* 2015. V. 10. № 3. P. 386–394.
8. Glyakina A.V., Likhachev I.V., Balabaev N.K., Galzitskaya O.V. Right- and left-handed three-helix proteins. II. Similarity and differences in mechanical unfolding of proteins. *Proteins*. 2014. V. 82. № 1. P. 90–102.
9. Sadovnichy V., Tikhonravov A., Voevodin V., Opanasenko V. "Lomonosov": Supercomputing at Moscow State University. In: *Contemporary High Performance Computing: From Petascale toward Exascale*. Boca Raton: CRC Press, 2013. P. 283–307.