УДК 519.6

А. В. Уткин, М. С. Ожгибесов

Применение технологий CUDA и MPI к решению задач молекулярной динамики

A. V. Utkin, M. S. Ozhgibesov

CUDA and MPI Programming Models for Problems of Molecular Dynamics

Одним из наиболее сложных моментов при выполнении моделирования в рамках метода молекулярной динамики является значительное расчетное время задачи даже для сравнительно небольших систем атомов. Первый способ решения данной проблемы состоит в применении высокоэффективных параллельных масштабируемых алгоритмов с использованием программного интерфейса MPI. Второе перспективное направление — создание параллельных программ, позволяющих проводить вычисления с использованием графических процессоров, поддерживающих технологию CUDA. Появление гетерогенных вычислительных кластеров обусловило необходимость разработки новых алгоритмов и кодов, которые используют технологию CUDA для проведения ресурсоемких расчетов, а для связи между различными GPU, физически принадлежащих разным узлам кластера, применяется технология MPI. В ходе работы был создан гибридный алгоритм (третий подход), который позволил объединить технологии CUDA и MPI в одной программе. Был проведен подробный анализ разработанных программных комплексов (MPI, CUDA и CUDA-MPI) и определены оптимальные условия использования каждого подхода.

Ключевые слова: молекулярная динамика, параллельное программирование.

One of the most difficult issues in molecular dynamic modeling is known to be a large computational time required for simulation even relatively small systems of atoms. Implementation of highly efficient parallel scalable algorithms is a key for effective solution of the problem mentioned above. In the paper, three implementations of parallel MD algorithms are tested. The first approach is based on a highly parallel MPI-based program design for a computer cluster with distributed memory. The second parallel algorithm is implemented on CUDA based GPUs by NVIDIA. It should be noted that modern high performance computing systems are combinations of MPI clusters equipped with GPGPUs, which turns them into so-called heterogeneous computing clusters. In this case, MPI technology is used for internode communications, while all computations are carried out by GPUs. Thus, the third approach of parallelization discussed in this study is based on a design of CUDA-MPI algorithm. The detailed studies and comparison of all three approaches (MPI, CUDA and CUDA-MPI) are performed to define optimal parameters and conditions of each algorithm applicability.

Key words: molecular dynamics, parallel computing.

DOI 10.14258/izvasu(2014)1.1-28

Физическая система и алгоритм расчета. Для анализа эффективности созданных алгоритмов, в качестве физической системы был выбран резервуар, наполненный газом (аргоном). Размеры резервуара составляли 2500x2500x2500Å (503000 атомов). Взаимодействие атомов со стенкой резервуара описывалось функцией зеркального отражения, а взаимодействие атомов между собой моделировалось при помощи парного потенциала Леннарда-Джонса [1]. Интегрирование уравнения движений поводилось с использованием схемы Верле второго порядка точности по временному шагу. Расчет сил межатомного взаимодействия является наиболее трудоемкой с точки зрения вычислительных ресурсов частью молекулярно-динамического моделирования, поскольку при расчете сил, действующих на атом i, необходимо учитывать вклад со стороны всех соседних атомов. Существует ряд общепринятых методик оптимизации расчетов в методе молекулярной динамики, которые позволяют значительно уменьшить время расчета [2].

1. Список Верле (список соседей). При моделировании большой системы атомов и использовании радиуса обрезания r_c, меньшего, чем моделируемая область, определенная часть атомов не будет вносить никакого вклада в силу, действующую на атом i.

В рассматриваемом методе вводится дополнительный радиус обрезания r_v (больший, чем r_c).

Перед вычислением взаимодействий для каждого атома i, создается свой список соседних атомов, попавших в сферу с радиусом r_v. Но при расчете сил, действующих на атом i, учитываются только соседние атомы из этой сферы. Интервал между созданием нового списка соседей обычно составляет 10–20 временных шагов или определяется автоматически.

2. Метод связанных списков (Cell linked list method-CLLM). Альтернативным методом для эффективного определения атомных соседей в достаточно больших системах (по крайней мере, содержащих больше 1000 атомов) является метод связанных списков. Трехмерная расчетная область делится на пронумерованные ячейки, чей размер немного больше или равен радиусу обрезания потенциала межатомного взаимодействия r_c. В начале вычислений создается массив, содержащий список номеров соседей каждой ячейки. Каждый атом взаимодействует только с атомами своей ячейки или с атомами соседних ячеек (26 ячеек). Одним из дополнительных преимуществ, которые дает метод связанных списков, является возможность использования третьего закона Ньютона при расчете сил $F_{ij} = -F_{ji}$. Это позволяет избежать двойного расчета силы в паре атомов i–j и уменьшает число соседних ячеек с 26 до 13, что приводит к существенному уменьшению времени счета задачи.

3. Различные гибридные комбинации метода связанных списков и списка Верле (гибридный список Верле). Самой очевидной гибридной комбинацией является метод, где список Верле для атома i создается не на основе перебора всех атомов системы, а на основе анализа расстояния до атомов в ячейке, которой принадлежит атом i и расстояния до атомов в соседних 26 ячейках. Тем самым исключается одна из основных проблем, связанных с необходимостью перебора всего набора атомов системы, при построении списка Верле.

Программная реализация с использованием технологии MPI, основанная на одномерной параллелизации. Важным элементом программирования для кластерных систем с разделенной памятью является правильная организация обменов данными между узлами и их синхронизация. В настоящем исследовании был использован метод разбиения расчетной области на подобласти с последующей динамической корректировкой их размеров. Динамическая балансировка размеров областей необходима для обеспечения равномерной загрузки всех вычислительных узлов. При определении сил, действующих на атом со стороны остальных частиц, внутри такой подобласти использовался метод связанных списков.

Программная реализация параллельного алгоритма для графических процессоров (GPU), основанная на технологии CUDA NVIDIA. Особенность реализации программы на GPU заключается в выполнении одинакового набора инструкций каждым вычислительным процессором (один атом — одна вычислительная нить). Необходимо отметить, что узким местом при использовании GPU является низкая скорость передачи данных из главной памяти компьютера в память GPU, таким образом, основная задача состоит в минимизации обмена данными между оперативной памятью компьютера и памятью GPU. Было создано два варианта GPU кода. Первый вариант основан на методе связанных списков, во втором варианте расчетного кода использовалась гибридная комбинация метода связанных списков и списка Верле.

Программная реализация с использованием гибридной технологии MPI и CUDA NVIDIA. Появление гетерогенных вычислительных кластеров обусловило необходимость создания новых алгоритмов и кодов, которые используют технологию CUDA для проведения трудоемких расчетов, а для связи между различными GPU, физически принадлежащими разным узлам кластера, используется технология MPI. При создании расчетного кода за основу был взят созданный вариант MPI программы с одномерной параллелизацией. Поскольку расчет силовых взаимодействий является самой трудоемкой частью вычислений, именно эта часть переносилась с CPU на GPU. Как отмечено выше, операция копирования данных с CPU на GPU отнимает много времени, поэтому принято решение создавать метод связных списков непосредственно на GPU, а не копировать уже готовые массивы данных из основной памяти компьютера. Соответственно, расчет сил проводился на GPU при помощи метода связных списков (26 соседних ячеек).

Очевидным недостатком данной схемы является необходимость копирования большого количества информации между CPU и GPU на каждом расчетном шаге, с целью организации обмена данными между GPU о перемещении атомов внутри расчетной области. Меняющееся на каждом шаге число атомов N_{CPU} на каждом вычислительном ядре CPU делает невозможным использование списков Верле в силу их определения, данного выше.

Сравнительный анализ алгоритмов и программ. Было выполнено сравнительное тестирование рассмотренных программ. Общее время расчета составляло 0.05 нс. — 100000 шагов $5\,10^{-16}$ с.

Расчеты проводились на гибридном кластере, установленном в ССКЦ СОРАН — НКС-30Т+GPU: 80 (480 ядер) процессоров CPU (X5670) и 120 (61440 ядер) процессоров GPU (Tesla M 2090). Для расчетов также использовался ПК с установленной картой Tesla C1060 и процессором CPU i7–920.

Время расчета задачи (секунды)

	MPI (9CPU[1])	MPI+CUDA (9CPU+9GPU[1])	CUDA (1GPU[1])	CUDA (1 GPU[1])	CUDA (1 GPU[2])
Общее время счета задачи	11600	6840	4379*/4853**	839*/5096**	1602*/11477**
Общее время расчета сил	2100	548	3996*/3996**	137*/138**	199*/202**
Обмены между CPU и GPU		507			
Метод сортировки	CLLM	CLLM (26 ячеек)	CLLM	CLLM+Список Верле	

1 — гибридный многопроцессорный кластер; 2 — ПК с GPU Tesla C1060.

* — обновление списков соседей каждые 20 шагов; ** — обновление списков соседей на каждом шаге.

Из представленных в таблице временных зависимостей следует, что программа, основанная только на технологии CUDA и гибридных списках Верле, использует меньше всего временных ресурсов (запуск проводился на 1 GPU). Использование только метода связанных списков приводит к существенному увеличению времени расчета из-за необходимости обработки лишней информации из соседних ячеек (замедление почти в 5 раз). Код, основанный на гибридной схеме MPI+CUDA, считает быстрее, чем MPI программа (запуск проводился на девяти узлах CPU и 9 GPU). Время расчета сил уменьшилось почти в 4 раза, но добавилось время обмена между CPU и GPU, которое практически идентично времени, затраченному на расчет сил на GPU. Программа, основанная только на технологии CUDA и гибридных списках Верле, считает быстрее, чем программа MPI+CUDA, почти в 8 раз. Следует отметить, что алгоритм MPI+CUDA несет в себе все недостатки, присущие алгоритму MPI, которые связаны в первую очередь с необходимостью обменов между узлами CPU, а также с их неравномерной загрузкой. Как уже упоминалось выше, из-за постоянно меняющегося числа атомов на каждом узле невозможно использовать списки Верле, что также усугубляет ситуацию.

Однако, несмотря на все эти недостатки, полученный код будет эффективен при моделировании физических систем, состоящих из большого числа атомов. Это связано с тем, что рост числа атомов приведет к неизбежному росту требований к памяти для хранения информации. Возможна ситуация, когда программа, основанная только на технологии CUDA и гибридных списках Верле, затребует больше ресурсов, чем может обеспечить GPU. В случае же использования гибрида MPI+CUDA подобная ситуация маловероятна, поскольку размер запрашиваемых ресурсов будет обратно пропорционален числу используемых GPU и CPU.

Библиографический список

1. Allen M.P., Tildesley D.J. Computer Simulation of Liquids. — Oxford Science Publications, 2000.

2. Frenkel D., Smit B. Understanding Molecular Simulation, Second Edition // From Algorithms to Applications. — 2001.