

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ ЭЛЕКТРОННОГО ПУЧКА С ПЛАЗМОЙ НА СОВРЕМЕННЫХ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСАХ

В.А. Вшивков, А.В. Снытников

*Институт Вычислительной Математики и Математической Геофизики СО РАН
Россия, 630090, Новосибирск, пр. Лаврентьева, 6*

E-mail: vsh@ssd.sccc.ru, snytav@ssd.sccc.ru

Проводится моделирование взаимодействия плазмы с релятивистским электронным пучком в трехмерной постановке. Модель построена на основе метода частиц в ячейках. Основную часть времени вычислений при этом занимает расчет движения частиц. В связи с этим проводится оптимизация вычислений, направленная на повышение эффективности использования кэш-памяти. А именно, выполняется частичное упорядочивание модельных частиц по положению в пространстве и все компоненты электрического и магнитного поля хранятся в одном 4-мерном массиве. Использование этих методик позволяет сократить время расчета движения частиц в 2 раза. Работа выполнена при поддержке Российского Фонда Фундаментальных Исследований, гранты 08-01-615 и 08-01-622, а также интеграционных проектов СО РАН № 103, № 113 и № 26.

PARALLEL IMPLEMENTATION OF BEAM-PLASMA INTERACTION MODEL FOR THE MODERN MULTIPROCESSOR COMPUTERS / A.V. Snytnikov, V.A. Vshivkov (Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Russia, 630090, Novosibirsk, Prospekt Lavrentieva, 6). The 3D simulation of the interaction of the relativistic electron beam with plasma is conducted. The model is built on the basis of the PIC method. The main part of computation time is taken by particle pushing procedure. Due to this reason the optimization of the code is performed in order to increase cache efficiency. In particular the partial ordering of particles is performed considering their position. Moreover, all the components of the electric and magnetic fields are stored in one 4-dimensional array. This technique allows to reduce particle pushing time by the factor of 2.

При моделировании плазмы методом частиц в ячейках для вычисления новых значений координаты и импульса модельной частицы используются значения электрического и магнитного поля. Каждая компонента поля хранится в отдельном трехмерном массиве. Таким образом на каждом временном шаге для каждой модельной частицы происходит обращение к шести трехмерным массивам. Модельные частицы расположены внутри расчетной области случайным образом. Если даже модельные частицы расположены рядом в массиве, где хранятся их координаты, то сами значения координат будут близкими только вначале. В дальнейшем модельные частицы перемешиваются. Это означает, что обращения к трехмерным массивам, содержащим электрическое и магнитное поля, являются неупорядоченными, и использование кэш-памяти в данном случае не позволяет сократить время счета.

Описание модели.

Численная модель, используемая для решения задачи о релаксации пучка, состоит из уравнений Власова для электронной и ионной компонент плазмы и системы уравнений Максвелла. В данной работе используется алгоритм решения этих уравнений, описанный в работе [3].

Уравнения Власова решаются методом частиц в ячейках (PIC). В рамках этого метода решаются уравнения движения модельных частиц, которые являются уравнениями характеристик для уравнения Власова. Для решения уравнений движения используется схема с перешагиванием. Для нахождения электрических и магнитных полей используется схема, в которой поля определяются из разностных аналогов законов Фарадея и Ампера. Эта схема имеет второй порядок аппроксимации по пространству и по времени.

Постановка задачи.

В начальный момент в трехмерной области решения, которая имеет форму прямоугольного параллелепипеда находится плазма, состоящая из электронов и ионов. Модельные частицы распределены по области равномерно. задается плотность плазмы и температура электронов, температура ионов считается нулевой. Дополнительно в области присутствуют электроны пучка, которые также распределены по области равномерно (предполагается, что пучок уже вошел в расчетную область). Электроны пучка отличаются от электронов плазмы тем, что они имеют кинетическую энергию направленного движения 1 МэВ, а их температура равна нулю. Модельные частицы, соответствующие электронам пучка, имеют меньшую массу, нежели модельные частицы, соответствующие электронам плазмы (отношение их масс равно отношению плотности плазмы и плотности пучка).

Итак, исходными параметрами задачи являются: плотность и температура электронов плазмы, отношение плотности электронов плазмы к плотности электронов пучка, энергия электронов пучка.

Параллельная реализация

Распараллеливание выполнено методом декомпозиции расчетной области по направлению, перпендикулярному направлению движения электронного пучка. Используется смешанная эйлерово-лагранжева декомпозиция. Сетка, на которой решаются уравнения Максвелла, разделена на одинаковые подобласти по одной из координат. С каждой подобластью связана группа процессоров (в том случае, когда вычисления производятся на многоядерных процессорах, процессором для единообразия будет именоваться отдельное ядро). Далее, модельные частицы каждой из подобластей разделяются между процессорами связанной с этой подобластью группы равномерно, вне зависимости от координаты.

Каждый из процессоров группы решает уравнения Максвелла во всей подобласти. Далее

решаются уравнения движения модельных частиц. После этого происходит суммирование значений тока по всей подобласти. Один из процессоров группы производит обмен граничными значениями тока и полей с соседними подобластями, и затем рассылает полученные граничные значения всем процессорам своей группы. В случае, если и уравнения Максвелла для подобласти, и уравнения движения всех частиц подобласти частиц решаются на одном процессоре, вычисления с частицами занимают в 10-20 раз больше времени.

Оптимизация.

Модельные частицы расположены внутри расчетной области случайным образом. Если даже модельные частицы расположены рядом в массиве, где хранятся их координаты, то сами значения координат будут близкими только вначале. В дальнейшем модельные частицы перемешиваются. Это означает, что обращения к трехмерным массивам, содержащим электрическое и магнитное поля, являются неупорядоченными, и использование кэш-памяти в данном случае не позволяет сократить время счета.

Использование кэш-памяти было бы более эффективным, если бы частицы были упорядочены. Тогда значения полей, загруженных в кэш при расчете движения некоторой частицы, могли бы быть использованы и для следующей частицы, если она расположена близко. Для этого достаточно упорядочить частицы по ячейкам сетки, то есть, хранить каким-то образом вместе все частицы, которые расположены внутри каждой ячейки. Это означает, что полная сортировка массива частиц не нужна, так как с точки зрения использования кэш-памяти не имеет значения, как частицы расположены внутри ячейки.

Модельные частицы, принадлежащие некоторой ячейке сетки, можно хранить в виде связанного списка или в виде массива. Преимущества списка очевидны: нет ограничения на число частиц в ячейке, простота добавления/удаления, но есть и недостатки, а именно большее по сравнению с массивом время доступа. Если же частицы каждой ячейки хранятся в виде массива (статического массива), то применительно к трехмерной задаче для двух сортов частиц это даст 5-мерный массив для одной только координаты X всех частиц (напомним, что модельная частица в данном случае характеризуется шестью признаками).

Но основная проблема в случае статического массива - это максимальное число частиц в ячейке. Это означает, что заранее неизвестно, какой длины массив потребуется для хранения всех частиц в каждой ячейке. Из проведенных расчетов известно, что максимальное значение плотности электронов было равно 5 (в единицах начальной плотности). Таким образом, можно было бы задать длину массива частиц в каждой ячейке $5N$, где N - число частиц в ячейке в начальный момент времени. Но в этом случае размер массива частиц увеличится в 5 раз, а его размер составляет 70 Гб, например, для сетки $512 \times 64 \times 64$ и $N = 150$.

Использование для этой цели динамических массивов решает проблему перерасхода памяти, но создает другую: необходимость иметь внутри программы свой эффективный менеджер динамической памяти, что, возможно, является решаемой задачей, но едва ли приведет к существенному уменьшению времени работы программы в целом.

Поэтому был реализован компромиссный вариант: для каждого сорта частиц (в данном случае 2 сорта: электроны и ионы) в каждой ячейке в целочисленном массиве длины $5N$ хранить номера частиц, находящихся в данный момент в данной ячейке. Номер задает позицию частицы в больших (порядка 100 млн. элементов) вещественных массивах, в которых хранятся координаты и импульсы модельных частиц. Таким образом, при перемещении частицы из одной ячейки в другую (обязательно в соседнюю - это определяется соображениями устойчивости метода частиц) перемещается только номер частицы: он удаляется из массива номеров, описывающих текущую ячейку, и добавляется в массив номеров одной из соседних ячеек. Внутри самих массивов координат и импульсов частицы никогда не перемещаются.

Также был реализован вариант с хранением частиц каждой ячейки в виде связанного списка. В этом случае имеется 4-мерный массив указателей, задающий первый элемент списка в

каждой ячейке, и отсутствуют большие массивы: вся информация по частицам хранится только в списках.

В обоих вариантах на вход процедуры интегрирования уравнений движения модельных частиц подаются шесть небольших (размером не более $5N$) массивов, хранящих координаты и импульсы частиц для каждой конкретной ячейки. Эти массивы формируются либо на основе списка частиц, либо на основе массива номеров частиц этой ячейки.

Далее рассмотрим результаты тестов, показывающих эффективность выполненной оптимизации. Тестовые расчеты проводились на рабочей станции с процессором AMD Phenom, и на кластере Новосибирского Государственного Университета, оснащенного процессорами Intel Nehalem. В обоих случаях была выбрана сетка такого размера, что даже один трехмерный массив, содержащий, например, одну из компонент поля, заведомо не помещается в кэш.

Размер сетки: (рабочая станция, процессор Phenom, последовательный счет) $64 \times 32 \times 32$ узла, 50 частиц в ячейке, (кластер, процессор Nehalem, в пересчете на один процессор), $512 \times 2 \times 64$ узла, 50 частиц в ячейке.

Время счета движения частиц, один временной шаг (рабочая станция, процессор Phenom):

- 1) Исходный неоптимизированный вариант: 13.25 сек.
- 2) Хранение значений поля в 4-мерном массиве: 8.8 сек.
- 3) Упорядочивание частиц с помощью массивов номеров: 12.51 сек.
- 4) Упорядочивание частиц с помощью связанного списка: 10.5 сек.
- 5) Сочетание вариантов 2 и 3: 10.92 сек.

Время счета движения частиц, один временной шаг (кластер, процессор Nehalem):

- 1) Исходный неоптимизированный вариант: 7.22 сек.
- 2) Хранение значений поля в 4-мерном массиве: 6.72 сек.
- 3) Упорядочивание частиц с помощью массивов номеров: 5.67 сек.
- 4) Упорядочивание частиц с помощью связанного списка: 10.3 сек.
- 5) Сочетание вариантов 2 и 3: 3.67 сек.

Отсюда можно сделать вывод, что сочетание упорядочивания модельных частиц и хранения значений поля в одном 4-мерном массиве приводит к значительному сокращению времени вычислений с частицами, в данном случае, почти в 2 раза. Хранение частиц в виде связанных списков оказалось неэффективным для процессора Nehalem, но, возможно, окажется полезным на других архитектурах (так позволяют думать измерения времени на процессоре Phenom) или в тех случаях, когда установить максимальное число частиц в ячейке невозможно.

Список литературы

1. Астрелин В.Т., Бурдаков А.В., Поступаев В.В. "Подавление теплопроводности и генерация ионно-звуковых волн при нагреве плазмы электронным пучком" //Физика плазмы, 1998, том 24, № 5, с.450-462
2. Кролл Н., Трайвелпис А. "Основы физики плазмы", М: "Мир", 1975.
3. Григорьев Ю.Н., Вшивков В.А. "Численные методы "частицы-в-ячейках" // - Новосибирск: "Наука", 2000.
4. Ч. Бедсел, Б.Лэнгдон "Физика плазмы и математическое моделирование", М:"Мир", 1989.