

Разработка численных схем решения кинетического уравнения в кластерных средах на основе технологии MPI.

Ю. Ю. Клосс*, Н. И. Хохлов*, Ф. Г. Черемисин**, Б. А. Шурыгин*

* *Московский Физико-Технический Институт, Долгопрудный, Россия*

** *Вычислительный центр им. А.А. Дородницына РАН, Москва, Россия*

Поступила в редколлегию 16.11.2007

Аннотация—В данной статье описан метод распараллеливания решения уравнения Больцмана применительно к задачам течений разреженного газа. Рассмотрены как общие аспекты библиотеки MPI, так и особенности ее применения в конкретно данном случае. Приводится анализ полученных результатов, путем проведения тестов на скорость расчета.

1. ВВЕДЕНИЕ.

Течение газа в микро каналах, диаметр которых сравним с длиной свободного пробега молекул, сопровождается значительным отклонением от термодинамического равновесия, что делает невозможным их моделирование на основе уравнений гидродинамики. Необходимо переходить на более подробный уровень описания на основе кинетического уравнения Больцмана[1], которое определяет функцию распределения скоростей молекул в физическом пространстве и в пространстве скоростей. Задача становится многомерной, в общем случае с 7 независимыми переменными (три пространственные координаты x, y, z , три компоненты молекулярной скорости u, v, w и время t). Ее решение требует больших вычислительных мощностей и затрат оперативной памяти. Характерное число узлов сетки в шестимерном фазовом пространстве (координаты и скорости) меняется от 10 млн. для грубых и тестовых расчетов до 300 млн. для более точных и сложных расчетов. В каждом из этих узлов для одноатомного и однокомпонентного газа требуется вычислять пятимерный нелинейный интегральный оператор столкновений Больцмана. Нелинейность задачи приводит к тому, что даже для моделирования стационарных течений, требуется применять итерационный метод нахождения решения, наиболее естественным из которых является метод установления, т.е. решение нестационарного уравнения Больцмана до выхода на независящее от времени решение. Использование свойств симметрии течения и эффективного метода решения уравнения позволяют значительно снизить требуемые вычислительные ресурсы, но даже на сетках с небольшим числом ячеек время расчета на одном процессоре составляет десятки часов. Необходим перенос проблемно моделирующей среды на кластерную архитектуру. Существует несколько библиотек, позволяющих использовать в своих программах распределенные вычисления. Наиболее распространенные среди них – это MPI и PVM, но последняя из них уже устарела, а MPI сейчас широко используется и развивается, именно поэтому она и была выбрана.

2. ПРИМЕНЕНИЕ MPI ДЛЯ РЕШЕНИЯ РАЗНОСТНЫХ СХЕМ ПЕРВОГО ПОРЯДКА ТОЧНОСТИ НА ПРЯМОУГОЛЬНЫХ СЕТКАХ.

2.1. Алгоритм распределения расчета на узлы кластера.

Уравнение Больцмана

$$\frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial x} = \int_{-\infty}^{\infty} \int_0^{2\pi} \int_0^{b_m} (f' f_1' - f f_1) g b d b d \varepsilon d \xi_1 \quad (1)$$

решается на фиксированной пространственно-скоростной сетке конечно-разностным методом, разработанным в [2]. Для аппроксимации левой части уравнения используется явная консервативная конечно-разностная схема первого порядка точности. Решение уравнения находится методом расщепления по физическим процессам, при котором на шаге по времени dt последовательно решаются уравнения

$$\begin{aligned} \frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial x} &= 0, \\ \frac{\partial f}{\partial t} &= I \end{aligned} \quad (2)$$

Нами была разработана программа со структурой удобной для распараллеливания и разделения моделирования на несколько узлов, образующих кластер. Структура сеток, на которых велось моделирование была следующей: все моделирование происходило в шестимерном фазовом пространстве, при этом в пространстве координат расчет проводился на прямоугольной сетке, а в каждой ячейке координатной сетки находилась сетка пространства скоростей, содержащая значения функции распределения газа.

Как было отмечено ранее, для распараллеливания использовалась библиотека MPI[3,4] в реализации LAM-MPI[5]. При распараллеливании пространственная сетка делится между процессами. Деление сетки происходит по оси X, поскольку вдоль этой оси находится наибольшее число ячеек. На каждом шаге соседние процессы обмениваются граничными условиями. Пусть например у нас число ячеек по оси X равно X_n , тогда для сетки $100 \times 30 \times 30$ оно равно 100. При делении сетки между процессами каждому достанется некоторый отрезок сетки от X_{n_i} до $X_{n_{i+1}}$, причем X_{n_i} и $X_{n_{i+1}}$ будут находиться в пределах от 0 до X_n . В самом простейшем случае каждый процесс будет обрабатывать одинаковые отрезки сетки, то есть если существует N процессов, то каждый из них будет вести расчет на сетке размером по оси X X_n/N . Для двумерного случая это представлено на Рис.1.

После каждого шага в решении разностной схемы процессы должны обмениваться между собой граничными условиями, в данном случае это значения функций распределения на границах. Для этого каждый процесс содержит информацию не только о той сетке, на которой он ведет счет, но и информацию о двух граничных слоях слева и справа, для записи туда значений функции распределения от своих соседей. На каждом шаге происходит обмен граничными условиями между соседними процессами. Рассмотрим этот процесс подробнее. Пусть у i -й процесс считает на сетке начиная с X_{n_i} до $X_{n_{i+1}}$ ячеек. Для хранения граничных условий соседей ему также необходимо иметь X_{n_i-1} и $X_{n_{i+1}+1}$ слои пространственной сетки. Тогда на каждом шаге он будет получать от $i-1$, левого соседа X_{n_i-1} слой и отправлять ему X_{n_i} , а также отправлять $X_{n_{i+1}}$ слой $i+1$, правому соседу и получать от него $X_{n_{i+1}+1}$ слой. В двумерном случае процесс обмена граничными условиями показан на Рис. 2.

Если размер сетки $X_n \times Y_n \times Z_n$, а процессов N , то у каждого процесса будет кусок сетки размером $(X_n/N) \times Y_n \times Z_n$. А размер теневых граней будет $1 \times Y_n \times Z_n$, причем он не зависит от числа процессов. Тогда объем пересылок на каждом шаге будет равен $S \times Y_n \times Z_n$, где S - объем данных в одной ячейки. Но поскольку у процесса два соседа, а ему надо не только переслать, но и принять граничные условия, то это число нам надо еще умножить на 4, то есть получаем: $4 \times S \times Y_n \times Z_n$. Поскольку размер пересылок не зависит от числа процессов, то он должен хорошо масштабироваться.

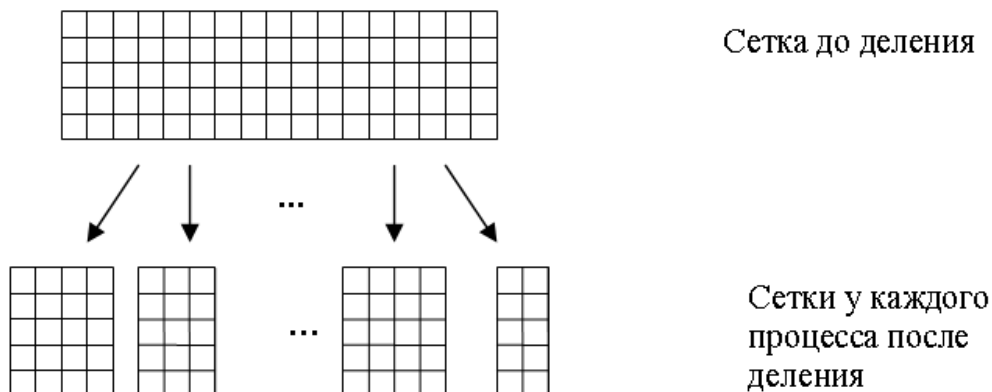


Рис.1. Пример деления сетки в двумерном случае.

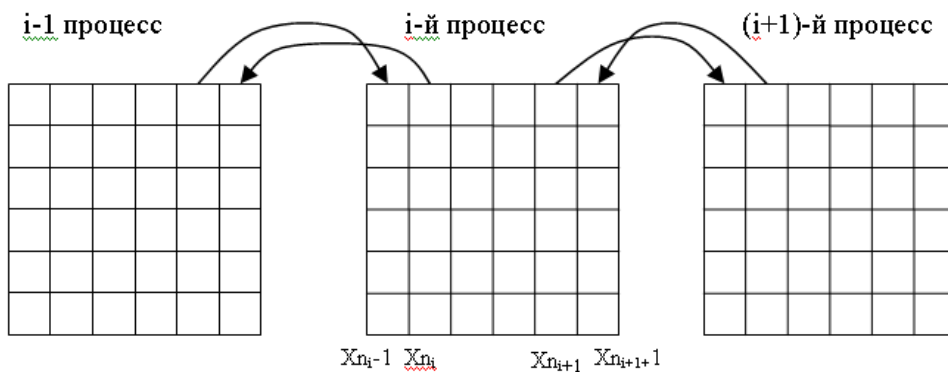


Рис. 2. Схема обмена граничными условиями между процессами.

2.2. Применение в реальных расчетах.

Данный алгоритм применялся в задаче расчета течения газов через различные микро структуры, как то системы отверстий, щели и др. Как пример одна из задач - система нанотрубок, рис 3. Из-за симметрии задачи расчет можно вести на геометрии, представленной на рис. 4. Которая и использовалась при получении результатов.

Для упрощения модели форма трубок берется прямоугольной. Такие системы используются в различных микро фильтрах и микро насосах.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РАСПАРАЛЛЕЛИВАНИЯ ЧИСЛЕННОЙ СХЕМЫ.

Описанный выше алгоритм был реализован в программе. Схема взаимодействия процессов в данном алгоритме представлена на Рис. 5. Сетка делится между процессами поровну, или когда деление на равные части невозможно, то остаток от деления разделяется между процессами. Каждому процессу по очереди добавляется еще один слой, пока остаток не закончится.

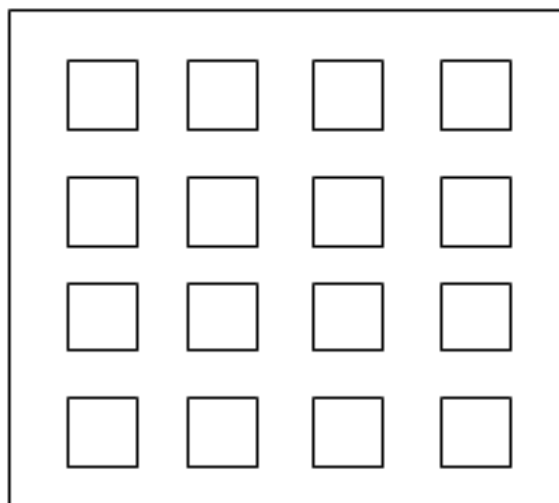


Рис. 3. Система нанотрубок.

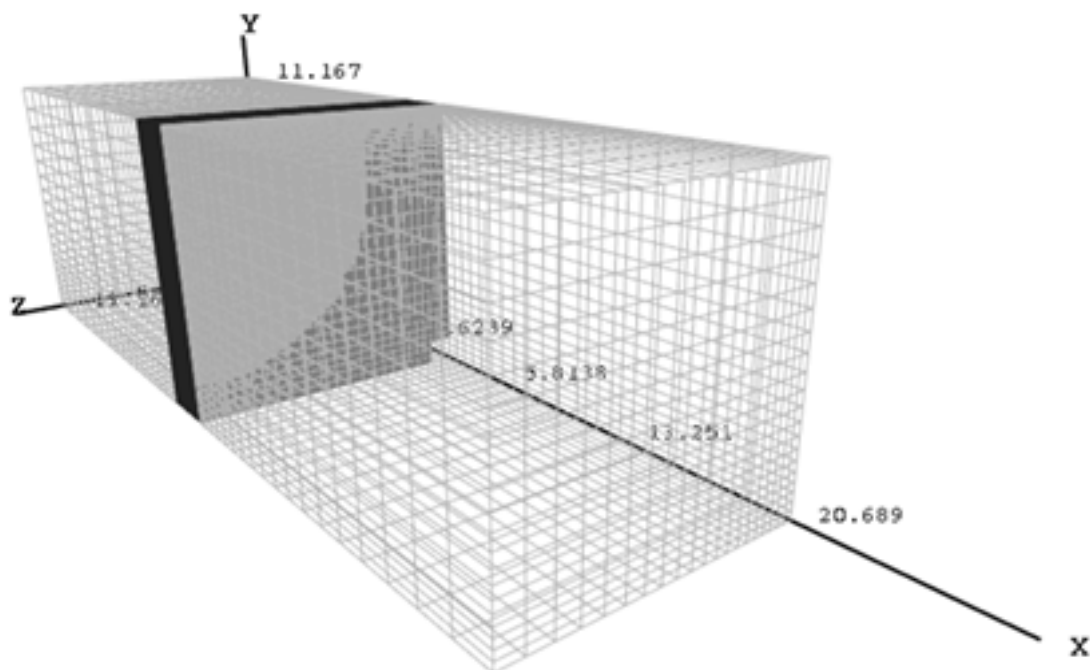


Рис. 4. Геометрия расчетной задачи.

Например у нас 100 клеток по оси X, а число процессов 16. Тогда каждому процессу достанется по 6 клеток, а 4-м их них по 7. Такое деление позволяет получить оптимальное распределение сетки между процессами. Как уже писалось выше все процессы равнозначны, чтобы расчет на всех узлах проводился с как можно более равными временами и узлы были загружены равномерно. Поскольку для пересылки данных используется функция `MPI_Sendrecv`, происходит синхронизация между процессами. Для того чтобы не возникло тупиковых ситуаций при пе-

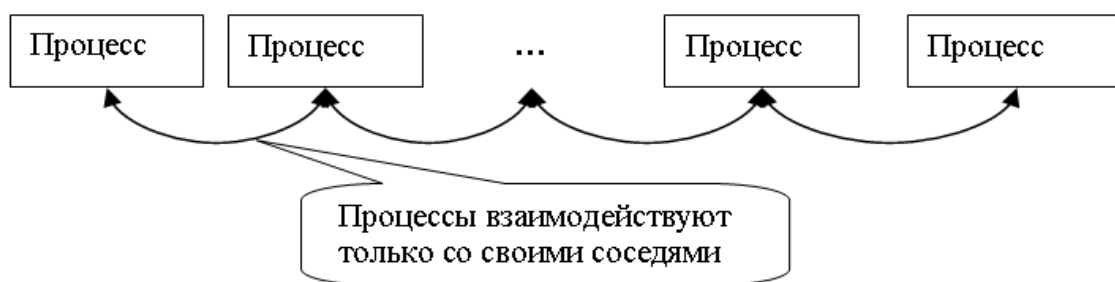


Рис. 5. Схема взаимодействия процессов.

ресылке, процессы разделены на четные и нечетные. И производят пересылку левых/правых граничных условий в разной последовательности.

Применение технологии распараллеливания MPI позволило значительно уменьшить временные затраты на проведение расчетов, а также снизить ограничения на минимально необходимый объем оперативной памяти компьютера, за счет разнесения сетки на различные узлы.

4. ОЦЕНКА УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ.

Тест проводился на кластере из 16 узлов. На каждом узле было по два двух ядерных процессора Xeon частотой 1.5 ГГц и с 2 Гб оперативной памяти. Итого 64 точки исполнения. Размер сетки был $488 \times 25 \times 25$. Ниже представлен график показывающий зависимость скорости расчета одного шага разностной схемы в зависимости от числа запущенных экземпляров рабочих процессов. Скорость отложена по оси Y и нормирована по скорости для одного процесса, по оси X отложено число рабочих процессов.

Суммарное увеличение скорости, при увеличении числа процессов в 64 раза - 55.4 раза. Сетка бралась довольно большой, порядка 1.7 Гб поэтому влияние кеша процессоров на скорость невелика. При маленьких сетках, порядка 200 Мб, при большом числе процессов сильно влиял на скорость. На графике хорошо видно, что при малых числах процессов он практически точно аппроксимируется прямой, а далее видно что в некоторых местах наблюдаются провалы уменьшения скорости. Это связано с делением сетки. Размер сетки по оси X, а именно по этой оси проходило деление, был равен 488. На графике в точке 32 виден провал. Максимальный размер части сетки, который был при таком делении, был равен $[488 / 32] + 1 = 16$. Но точно такой же размер был и при числе процессов 30. При малых числах процессов график практически линейен, это потому что каждый процесс получает достаточно большие части сетки, и разницей между соседними процессами можно пренебречь. А при большом числе процессов это различие уже становится существенно и влияет на скорость. Размер сетки по другим осям был 25×25 . Именно столько ячеек пересылалось на каждом шаге. Объем данных в одной ячейке порядка 4кб, таким образом, объем данных пересылаемый на каждом шаге между процессами был порядка 10мб.

В тестовом расчете размер сетки был взят такой чтобы не было влияния кеша процессоров на скорость вычислений. В реальных расчетах сетки чаще всего берутся меньше, поэтому при расчетах на большом числе процессоров, когда объем сетки на каждый процессор был небольшой и помещался в кеш, получалось еще большее ускорение. Так при расчете на сетке размером $1600 \times 15 \times 1$, при увеличении числа процессов в 160 раз было получено увеличение скорости в 270 раз. Расчет такой задачи на кластере длился 5 часов, а при расчете на одном узле это могло потребовать 50 суток.

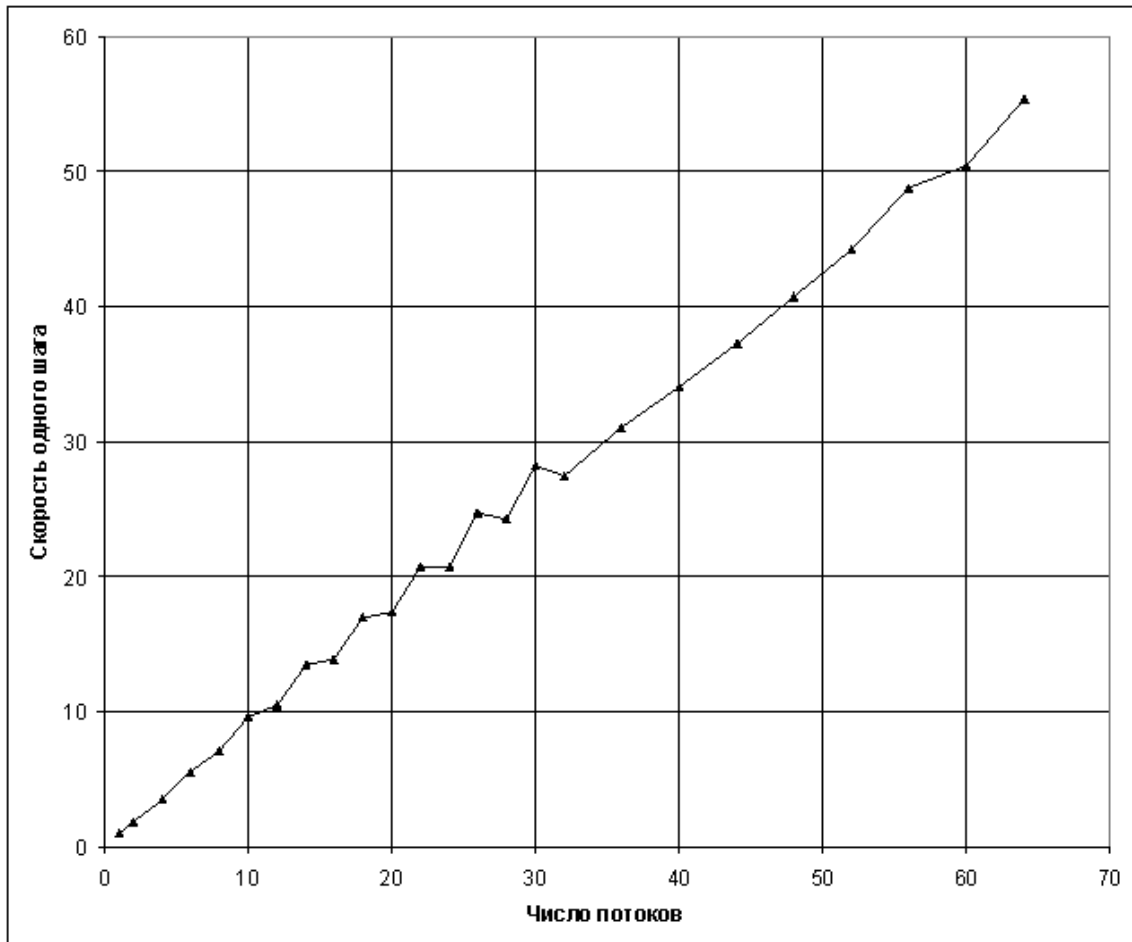


Рис. 6. Результаты тестирования алгоритма.

5. ЗАКЛЮЧЕНИЕ.

В результате работы по распараллеливанию алгоритма решения численной схемы на прямоугольных сетках, удалось добиться существенного увеличения скорости расчета при вычислении на кластерных системах. Были проведены серии тестов на увеличение производительности, которые дали положительные результаты. Из полученных результатов можно сделать вывод об эффективности примененных алгоритмов распараллеливания к вычислительным задачам такого типа. Полученное увеличение скорости практически линейно с увеличением числа точек исполнения программы, что говорит о хорошей масштабируемости алгоритма.

В заключении статьи авторы хотели выразить искреннюю благодарность сотрудникам ИСП РАН Аветисяну А.И., Бобковой В.В., Гайсаряну С.С., Калугину М.Д. за конструктивные замечания и предложения, которые были учтены в разработке прикладной программной системы и нашли отражение в настоящей статье.

СПИСОК ЛИТЕРАТУРЫ

1. Коган М.Н. Динамика разреженного газа. М.: Наука. 1967
2. Черемисин Ф.Г. Консервативный метод вычисления интеграла столкновений Больцмана. Доклады РАН. Т. 357. N. 1. с.1-4. 1997
3. <http://www.parallel.ru>

4. Воеводин В. В., Воеводин Вл. В.. Параллельные вычисления. "БХВ - Петербург", 2002
5. <http://www.lam-mpi.org>