ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2008 Математика и механика № 2(3)

УДК 519.684

Е.А. Данилкин

К ВОПРОСУ ОБ ЭФФЕКТИВНОСТИ 3D-ДЕКОМПОЗИЦИИ ПРИ ЧИСЛЕННОМ РЕШЕНИИ УРАВНЕНИЯ ПЕРЕНОСА С ИСПОЛЬЗОВАНИЕМ МВС С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ[1]

В статье сравниваются различные способы декомпозиции сеточной области при численном решении уравнения переноса методом конечного объема. Проведен предварительный теоретический анализ затрат на пересылку данных, а также выполнен вычислительный эксперимент на кластере ТГУ СКИФ Cyberia, показавший, что 3D-декомпозиция не дает выигрыша во времени по сравнению с 2D-декомпозицией, по крайней мере, для количества процессоров, не превышающего 250.

Ключевые слова: *геометрическая декомпозиция, математическое моделирование, параллельные вычисления.*

Математическое моделирование и численный эксперимент широко распространены в научных и прикладных исследованиях. В последнее время появились и широко используются разнообразные программные продукты для математического моделирования, например для изучения процессов гидродинамики и тепломассообмена (FLUENT, ANSYS CFX, OpenFOAM). В институтах и университетах для подобных целей разрабатываются свои математические модели и вычислительные приложения, но уже более узконаправленные. При этом основная идея математического моделирования – это описание исследуемого процесса с помощью системы дифференциальных уравнений. От модели к модели меняются лишь коэффициенты в уравнениях и способы их аппроксимации, при этом сами уравнения остаются неизменными.

Ни одна математическая модель гидро- или газодинамики не обходится без уравнения переноса, будь это перенос примеси или уравнение движения для одной из компонент скорости. Поэтому в данной статье сравнение различных способов декомпозиции будет осуществляться на примере уравнения переноса.

Задачи математического моделирования требуют большого количества компьютерного времени. Одним из способов преодоления этой проблемы является использование многопроцессорных вычислительных систем (МВС). В данной работе будут рассматриваться МВС с распределенной памятью, так как в настоящее время большинство супер-ЭВМ, которыми мы располагаем, построены именно по такому принципу. Наиболее общим подходом равномерного распределения вычислительной нагрузки между узлами МВС при решении сеточных уравнений является разделение вычислительной области на подобласти, количество которых совпадает с числом используемых процессов, т.е. использование принципа геометрической декомпозиции [1]. Причем самым простым способом решения этих разделенных на подобласти конечно-разностных задач является применение явных разностных схем, которые и будут рассматриваться здесь для численного ре-

[1] Работа выполнена при финансовой поддержке РФФИ, грант № 07-05-01126 и программы СКИФ-ГРИД.

шения поставленной задачи. Такие схемы отличаются относительно небольшим шагом интегрирования по времени для нестационарных задач, ограничение на который накладывает условие устойчивости.

Математическая постановка

Требуется решить трехмерное уравнение переноса в единичном кубе $(x, y, z) \in (0,1)\times(0,1)\times(0,1)$:

$$\frac{\partial \Phi}{\partial t}+\frac{\partial (u\Phi)}{\partial x}+\frac{\partial (v\Phi)}{\partial y}+\frac{\partial (w\Phi)}{\partial z}=\frac{\partial}{\partial x}\left[\Gamma\frac{\partial \Phi}{\partial x}\right]+\frac{\partial}{\partial y}\left[\Gamma\frac{\partial \Phi}{\partial y}\right]+\frac{\partial}{\partial z}\left[\Gamma\frac{\partial \Phi}{\partial z}\right]+S,\ t>0, \quad (1)$$

с граничными условиями второго рода и заданными начальными условиями

$$\begin{aligned} &\left.\frac{\partial \Phi}{\partial x}\right|_{x=0}=0; \quad \left.\frac{\partial \Phi}{\partial y}\right|_{y=0}=0; \quad \left.\frac{\partial \Phi}{\partial z}\right|_{z=0}=0; \quad \Phi|_{t=0}=0; \\ &\left.\frac{\partial \Phi}{\partial x}\right|_{x=1}=0; \quad \left.\frac{\partial \Phi}{\partial y}\right|_{y=1}=0; \quad \left.\frac{\partial \Phi}{\partial z}\right|_{z=1}=0. \end{aligned} \quad (2)$$

Функции *u, v, w*, $\Gamma > 0$ и *S* определены в рассматриваемой области.

Аппроксимация и численное решение

Аппроксимация дифференциальной задачи осуществляется на основе метода конечного объема [2]. Основная идея этого метода заключается в разбиении расчетной области на непересекающиеся, граничащие друг с другом конечные объемы так, чтобы один узел расчетной сетки содержался только в своем конечном объеме. Разбив таким образом расчетную область, интегрируем уравнение по каждому конечному объему. При вычислении интегралов используются кусочно-полиномиальные приближения для зависимых величин. Аппроксимация конвективных членов уравнений переноса выполняется с использованием противопотоковой схемы. Для расчета применялась явная схема, поэтому результатом приближенного интегрирования по одному конечному объему является готовая для вычислений формула:

$$\begin{aligned} ap^0\Phi^{n+1}_{i,j,k} = {} & ap_{i,j,k}\Phi^{n}_{i,j,k} + ae_{i,j,k}\Phi^{n}_{i+1,j,k} + an_{i,j,k}\Phi^{n}_{i,j+1,k} + at_{i,j,k}\Phi^{n}_{i,j,k+1} + \\ & + aw_{i,j,k}\Phi^{n}_{i-1,j,k} + as_{i,j,k}\Phi^{n}_{i,j-1,k} + as_{i,j,k}\Phi^{n}_{i,j,k-1} + b_{i,j,k}. \end{aligned} \quad (3)$$

Полученная разностная схема имеет первый порядок аппроксимации по времени и по пространству и является условно устойчивой и монотонной.

Распараллеливание

В рассматриваемом примере возможны три различных способа разделения значений сеточной функции по вычислительным узлам – одномерная, или ленточная, схема, двухмерное, или блочное, разбиение или трехмерное разбиение узлов вычислительной сетки.

Каждому процессорному элементу вместе с выделенной сеточной подобластью распределяются все значения сеточной функции из этой подобласти, в том числе и $\Phi^{n}_{i,j,k}$.

После этапа декомпозиции, когда производится разделение данных на блоки для построения параллельного алгоритма, переходим к этапу установления связей между блоками, вычисления в которых будут выполняться параллельно, – планированию коммуникаций. Из-за используемого шаблона явной разностной схемы для вычисления очередного приближения в приграничных узлах каждой подобласти необходимо знать значения сеточной функции с соседнего граничащего процессорного элемента. Для этого на каждом вычислительном узле создаются фиктивные ячейки для хранения данных с соседнего вычислительного узла и организуются пересылки этих граничных значений, необходимых для обеспечения однородности вычислений по явным формулам (3) [3] (рис. 1). Пересылка данных осуществляется с использованием процедур библиотеки MPI [4].

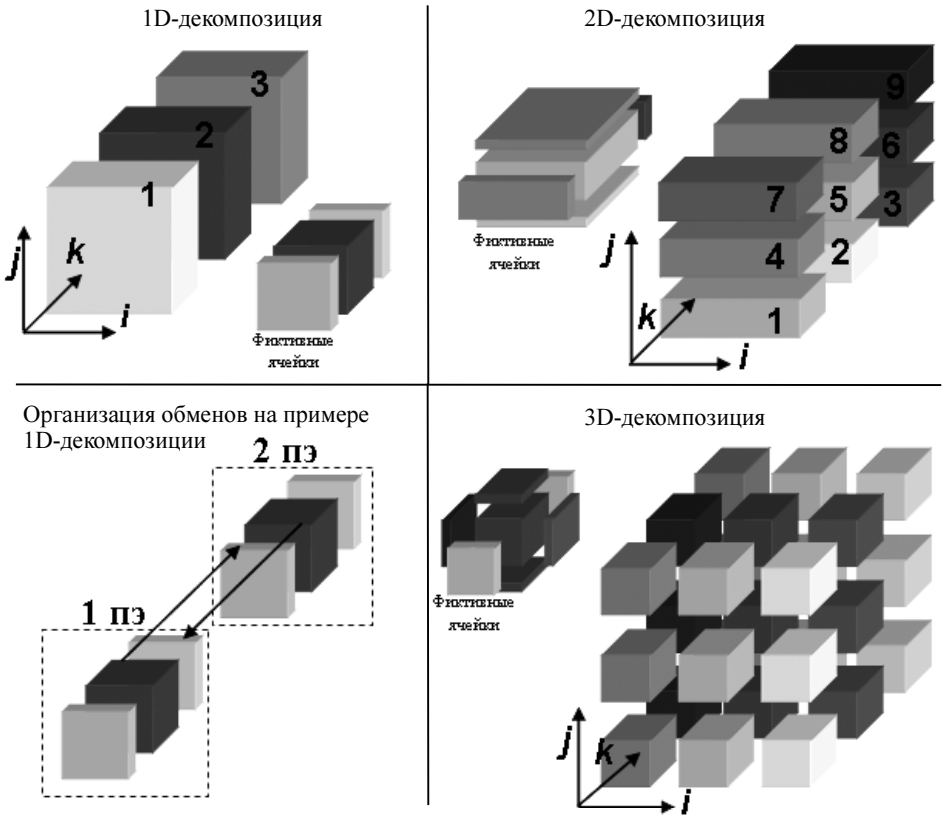

Рис. 1. Различные способы декомпозиции.
Схема организации обменов для 1D-декомпозиции

Обратимся к предварительному теоретическому анализу эффективности различных способов декомпозиции расчетной области для рассматриваемого случая. Будем оценивать время работы параллельной программы как время работы последовательной программы T_{calc}, деленное на количество используемых процессоров, плюс время пересылок: $T_p = T_{calc}/p + T_{com}$. Время пересылок для различных способов декомпозиции можно приблизительно выразить через количество пересылаемых данных:

$$T^{1D}{}_{com} = t_{send} \cdot N^2 \times 2; \quad T^{2D}{}_{com} = t_{send} \cdot N^2 \times 4/p^{1/2}; \quad T^{3D}{}_{com} = t_{send} \cdot N^2 \times 6/p^{2/3}; \qquad (4)$$

где N^3 – размерность задачи, p – количество вычислительных узлов, t_{send} – время пересылки одного числа.

Из формулы (4) видно, что для разных способов декомпозиции затраты на пересылку данных можно представить в виде $T_{com} = t_{send} \cdot N^2 \times k(p)$, где коэффициент пропорциональности $k(p)$ зависит от способа декомпозиции и числа используемых процессорных элементов.

В табл. 1 представлены числовые значения $k(p)$. Видно, что при $p > 5$ более эффективными будут 2D- и 3D-декомпозиции, а при $p > 11$ в случае 3D-декомпозиции потребуется пересылать между процессорными элементами меньшее количество сеточных значений функции Ф и можно ожидать, что в этом случае затрачиваемое на пересылку время будет минимально.

Таблица 1

Зависимость коэффициента пропорциональности $k(p)$ от числа вычислительных узлов и способа декомпозиции

Число процессов	3	4	5	6	10	11	12	16	60	120	250
1D-декомпозиция	2	2	2	2	2	2	2	2	2	2	2
2D-декомпозиция	2,31	2,00	1,79	1,63	1,26	1,20	1,15	1,00	0,51	0,36	0,25
3D-декомпозиция	2,88	2,38	2,05	1,82	1,29	1,21	1,14	0,94	0,39	0,24	0,15

Результаты

Расчеты производились на кластерной системе ТГУ СКИФ Cyberia на сетках 240 × 240 × 240 и 360 × 360 × 360 при использовании до 144 процессоров. Результаты вычислительного эксперимента показали наличие хорошего ускорения при решении задач данного класса. Основное внимание уделялось сравнению времени пересылок и времени вычислений при различных способах декомпозиции.

На первом этапе использовалась одна общая программа, размеры массивов от запуска к запуску не менялись, на каждом процессорном элементе нумерация элементов массивов начиналась с нуля. Несмотря на то, что в соответствии с теоретическим анализом 3D-декомпозиция является оптимальным вариантом распараллеливания, вычислительные эксперименты показали, что лучших результатов можно достичь, используя 2D-декомпозицию при числе используемых процессов от 25 до 144 (рис. 2).

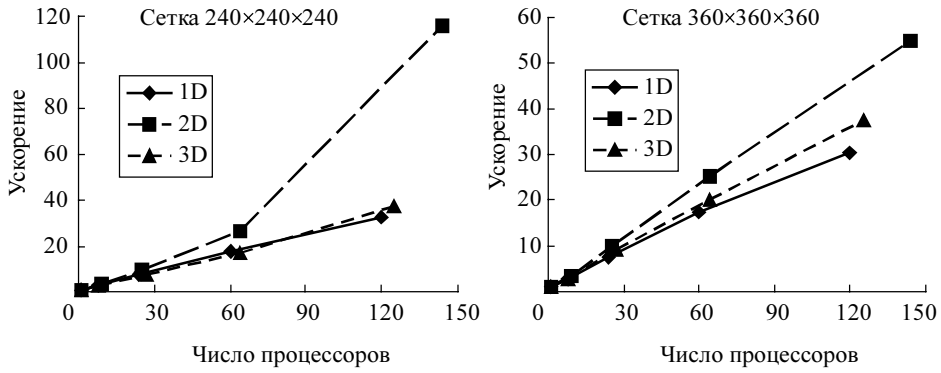

Рис. 2. Ускорение для различных способов декомпозиции расчетной области

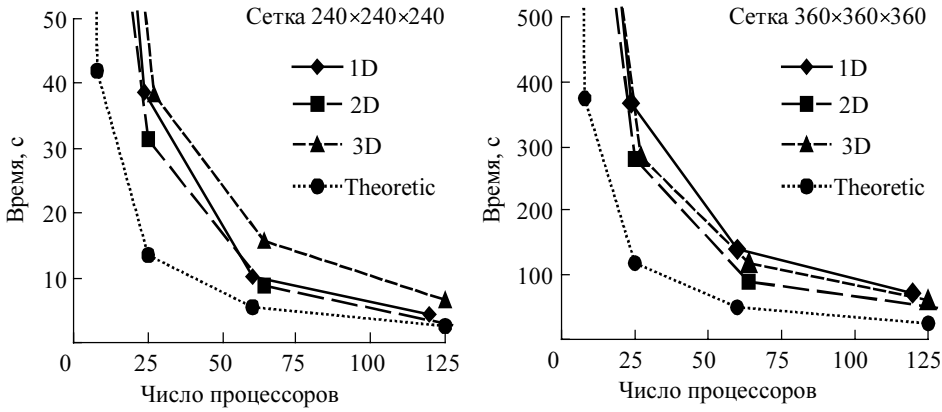

Рис. 3. Время вычислений для различных способов декомпозиции сеточной области

Исходя из предварительного теоретического анализа, на графиках должны присутствовать следующие закономерности. Время вычислений без учета межпроцессорных коммуникационных затрат при разных способах декомпозиции должно примерно совпадать для одинакового числа процессоров и уменьшаться как T_{calc}/p. В реальности же расчетные данные (рис. 3) указывают на то, что использование 2D-декомпозиции на различных сетках дает минимальные затраты на проведение вычислений и расчетные графики зависимости времени вычислений от числа используемых процессоров размещаются существенно выше, чем T_{calc}/p.

Для объяснения полученных результатов необходимо обратить внимание на допущения, которые были сделаны при предварительном теоретическом анализе поставленной задачи. Во-первых, предполагалось, что независимо от способа распределения данных на одном процессорном элементе выполняется одинаковый объем вычислительной работы, который должен приводить к идентичным временным затратам. Во-вторых, принималось, что время, затраченное на межпроцессорную пересылку любой последовательности одинакового количества данных, не зависит от способа их выборки из оперативной памяти.

Для того чтобы разобраться, что происходит в реальности, был проведен следующий набор тестовых расчетов. Для оценки состоятельности первого допущения был рассмотрен случай, когда программа запускалась в однопроцессорном варианте и при этом имитировались различные способы геометрической декомпозиции данных при одинаковом объеме вычислений, выполняемом каждым процессором. Было реализовано две версии программы: в первой использовались статические массивы размером N^3, во второй – «динамические», когда размеры массивов специально подбирались в соответствии со способом декомпозиции для того, чтобы минимизировать временные затраты, связанные с выборкой данных из оперативной памяти.

В табл. 2 представлено время в секундах, которое требуется процессору для проведения вычислений в своей сеточной подобласти при моделировании различных способов геометрической декомпозиции. От расчета к расчету объем вычислительной работы сохраняется, меняется лишь характер расположения обрабатываемых данных в оперативной памяти.

Таблица 2

Влияние способа декомпозиции сеточной области и размерности используемых массивов на время вычислений на одном вычислительном узле

Размерность задачи 360^3	
Без учета размерности массивов	С учетом размерности массивов
p 1D 2D 3D	*p* 1D 2D 3D
8 9,3 8,1 10,4	8 8,8 7,5 7,3
24 3,1 2,6 3,7	24 3,0 2,5 2,5
60 1,2 1,0 1,6	60 1,2 0,9 1,0

Проанализировав представленные результаты, можно сделать вывод, что при написании параллельной программы целесообразно использовать «динамические» массивы с подстраиваемыми под выделенное число процессоров размерами. В большей степени это существенно для 3D-декомпозиции. Преимущества использования «динамических» массивов, на наш взгляд, определяется меньшими временными затратами на выборку обрабатываемых данных из оперативной памяти и их передачу через КЭШ-память.

Для оценки реализуемости второго допущения ($T_{\text{com}} = t_{\text{send}} \cdot N^2 \times k(p)$) был рассмотрен тест, в котором измерялось время MPI-пересылки различных сечений массива с размерностью 3. Заметим, что решение этой задачи (пересылки, скажем, $i-j$-, $i-k$-, $j-k$-сечений массива, элементы которого имеют индексы i, j, k) в стандарте MPI может осуществляться различными способами в зависимости от того, какие сечения массивов рассматриваются.

Вычислительный эксперимент был организован следующим образом. В параллельной программе многократно выполнялась пересылка сечения массива одного типа между двумя процессорными элементами и фиксировалась продолжительность этого процесса. Использовались функции MPI_SEND и MPI_RECV и процедуры создания производных типов MPI_TYPE_VECTOR и MPI_TYPE_INDEXED [5]. В тестовых расчетах рассматривались массивы с количеством элементов 30^3, 60^3, 120^3. Полученные результаты показали, что и при организации межпроцессорной передачи данных также решительное преимущество имеет использование «динамических» массивов, поскольку в этом случае допускается использование производного типа MPI_TYPE_VECTOR, который для пересылки сечения $j-k$ в несколько раз по времени предпочтительней, чем более универсальный тип MPI_TYPE_INDEXED.

В табл. 3 представлено время, затрачиваемое на 100-кратную пересылку данных из разных сечений массива, осуществляемую с помощью производного типа MPI_TYPE_VECTOR. Анализируя данные, можно отметить, что время пересылки сечений массивов в MPI-FORTRAN-программе зависит от их типа. Поэтому наименьшие затраты на передачу данных дает 1D-декомпозиция, наибольшие – 3D-декомпозиция, причем это отличие становится более существенным при увеличении размера сечения от 30 × 30 до 120 × 120. Последнее связанно с увеличением времени на выборку передаваемых данных из оперативной памяти. Таким образом, в формулах (4) необходимо вводить дополнительные коэффициенты, которые бы учитывали тип пересылаемых сечений массивов сеточных функций.

Таблица 3

Время на пересылку различных сечений трехмерного массива (с)

Количество элементов в массиве	Сечение		
	$i-j$	$i-k$	$j-k$
30^3	0,00189	0,00192	0,00265
60^3	0,00402	0,00556	0,00884
120^3	0,01601	0,02151	0,03563

На рис. 4 представлен график ускорения параллельной программы, использующей «динамические» массивы, для различных способов декомпозиции. Здесь, также как и на рис. 2, наилучшие показатели демонстрирует 2D-декомпозиция, причем имеет место прирост ускорения. Незначительно ей уступает 3D-декомпозиция, проблемы которой в маштабируемости параллельной программы связаны, главным образом, с большими, чем в 1D- и 2D-случаях, затратами на выборку, обработку и пересылку данных из оперативной памяти.

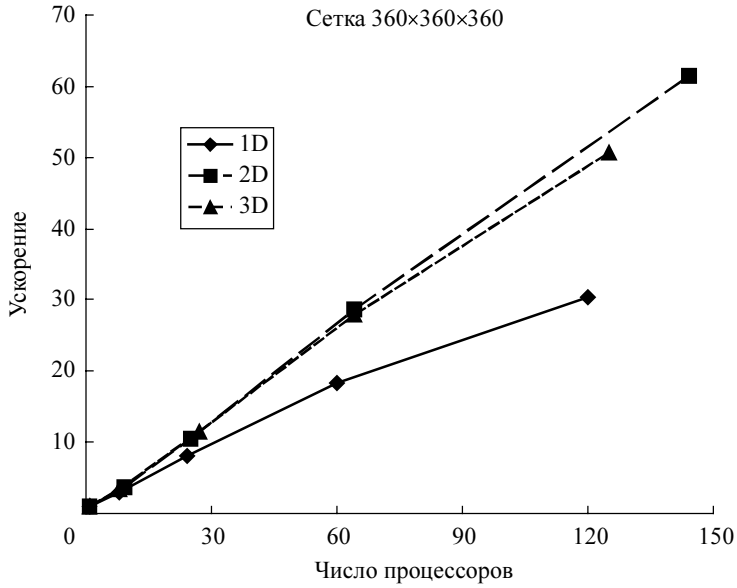

Рис. 4. График ускорения при использовании «динамических» массивов

Заключение

Для явных разностных методов решения уравнения переноса возможно применение одномерной, двумерной и трехмерной декомпозиции, однако результаты тестирования программ показали, что 3D-декомпозиция не дает выигрыша во времени по сравнению с 2D-декомпозицией, по крайней мере, для количества процессоров, не превышающего 250, при этом 3D-декомпозиция отличается более трудоемкой программной реализацией.

ЛИТЕРАТУРА

1. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
2. *Патанкар С.* Численные методы решения задач теплообмена и динамики жидкости: Пер. с англ. М.: Энергоатомиздат, 1984. 149 с.
3. *Данилкин Е.А.* К выбору способа декомпозиции при численном решении систем связанных дифференциальных уравнений на многопроцессорной технике с распределенной памятью // Третья Сибирская школа-семинар по параллельным вычислениям / Под ред. проф. А.В. Старченко. Томск: Изд-во Том. ун-та, 2006. С. 95 – 101.
4. *Старченко А.В., Есаулов А.О.* Параллельные вычисления на многопроцессорных вычислительных системах. Томск: Изд-во Том. ун-та, 2002. – 56 с.
5. *Букатов А.А., Дацюк В.Н., Жегуло А.И.* Программирование многопроцессорных вычислительных систем. Ростов н/Д: Изд-во ООО «ЦВВР», 2003. 208 с.

Статья принята в печать 19.07.2008 г.