

2003, том 39, № 3

**РЕШЕНИЕ БОЛЬШИХ ЗАДАЧ
НА ПАРАЛЛЕЛЬНЫХ СУПЕРКОМПЬЮТЕРАХ**

УДК 681.3 : 518.5

Г. А. Тарнавский, В. Д. Корнеев

(Новосибирск)

**ДЕКОМПОЗИЦИЯ ПРОГРАММНОГО КОМПЛЕКСА
МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ
ВЫСОКОСКОРОСТНЫХ ТЕЧЕНИЙ РЕАЛЬНОГО ГАЗА***

Данная работа ориентирована на применение современных вычислительных технологий и методов параллельного программирования, направленных на повышение эффективности решения фундаментальных научных и прикладных проблем. Рассмотрены технологические аспекты параллелизации вычислительной системы «Поток-3», предназначенной для моделирования задач аэродинамики и физической газовой динамики. Предложены и исследованы способы и проблемы глобального распараллеливания программного комплекса по главным входным параметрам, а также *C*-, *L*- и *V*-типы параллелизаций процедур основного итерационного ядра системы для локального ускорения операций.

Введение. Быстрый рост возможностей вычислительной техники, поддерживающих программные комплексы, и связанное с этим появление новых компьютерных технологий (параллелизация расчета на многопроцессорных вычислительных системах) обеспечили существенное продвижение математического моделирования задач газовой динамики в область численного исследования сложных высокоскоростных течений газа с учетом его реальных свойств, в том числе аномальных, с отклонением уравнения состояния от идеального. Однако это, делая практически реализуемым проведение многопараметрических циклов расчетов с подробной мелкошаговой вариацией определяющих параметров, а также практику сохранения результатов в банках данных для их последующего комплексного анализа, ставит на качественно новый уровень ряд вопросов компьютерного моделирования в данной области знания.

В [1, 2] была разработана общая стратегия параллелизации большого вычислительного комплекса, предназначенного для компьютерного моделирования задач аэродинамики и физической газовой динамики. В работах [3, 4] рассмотрены различные схемы распараллеливания операций многомерной

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (проекты № 01-01-00781, № 02-01-00097).

скалярной прогонки, составляющих основной алгоритм численного решения системы дифференциальных уравнений Навье – Стокса динамики вязкого теплопроводного газа, и проведены многоцелевые вычислительные эксперименты по определению эффективных способов распараллеливания и их реализации на различных мультипроцессорных системах. В [5] изучались некоторые аспекты организации проведения расчетов задач в предметной области, связанных с обеспечением численного метода начальными данными, и частично обсуждались проблемы вывода полученной информации и размещения ее на долгосрочное хранение в базах данных. Эта работа представляет идеологию и опыт параллелизации вычислительной системы и многосторонний анализ примененных различных типов распараллеливания.

Глобальная параллелизация. Из рассмотренных в [1, 2] общих видов параллелизации, связанных с физико-математической, геометрической и технологической декомпозициями полной задачи на ряд исполняемых параллельно подзадач, на первом этапе реорганизации однопроцессорного программного комплекса «Поток-2» в параллельную вычислительную систему «Поток-3», ориентированную на многопроцессорный счет с возможностью ее использования широким кругом пользователей, было применено несколько способов распараллеливания. Одним из них является глобальная параллелизация системы по главным входным параметрам.

Рассмотрим физическую сущность этого способа. Вычислительный комплекс обеспечивает решение задачи обтекания летательного аппарата потоком газа в широком диапазоне определяющих параметров. Главными из них (имеются в виду только физические параметры, определяющие основные режимы течения; алгоритмические параметры, хотя на них также может быть распространена излагаемая далее идеология, здесь не рассматриваются) являются: форма поверхности тела, тепловое условие на нем, высота и скорость полета, термодинамические свойства газа. Для ясности изложения выберем два из них, например высоту H и скорость V полета. Как правило, для изучения процессов обтекания необходимо провести значительное количество расчетов с вариацией как H , так и V . Такой цикл расчетов может быть организован в едином запуске вычислительного комплекса с параллелизацией «по глобальному параметру», схема которой приведена на рис. 1. Показан ряд уровней распараллеливания, образующий некоторый односвязный граф: древовидную ветвящуюся непересекающуюся систему связей, выходящих из единого корня – входа в программный комплекс. (ЛБД – локальная база данных.)

Каждый уровень представлен символом

$$G_{g_1, \dots, g_m}^m : m \in [1, M]; g_1 \in [1, g_{1F}]; \dots, g_m \in [1, g_{mF}], \quad (1)$$

где m – номер горизонтального уровня графа, отсчитываемого от корня, или номер вида глобального параметра, по которому производится параллелизация, например, $G^1 \equiv H$ (высота полета), $G^2 \equiv V$ (скорость полета) и т. д.; M – глубина распараллеливания (число горизонтальных уровней графа), т. е. длина списка физико-математических параметров задачи, по которым производится параллелизация; g_{1F} – длина списка (число вариантов) значений параметра первого уровня; g_1 – текущий номер в этом списке; g_{mF} – длина списка (число вариантов) значений параметра в одной подгруппе m -уровня; g_m – текущий номер в этом списке. (Все индексы целочисленные.)

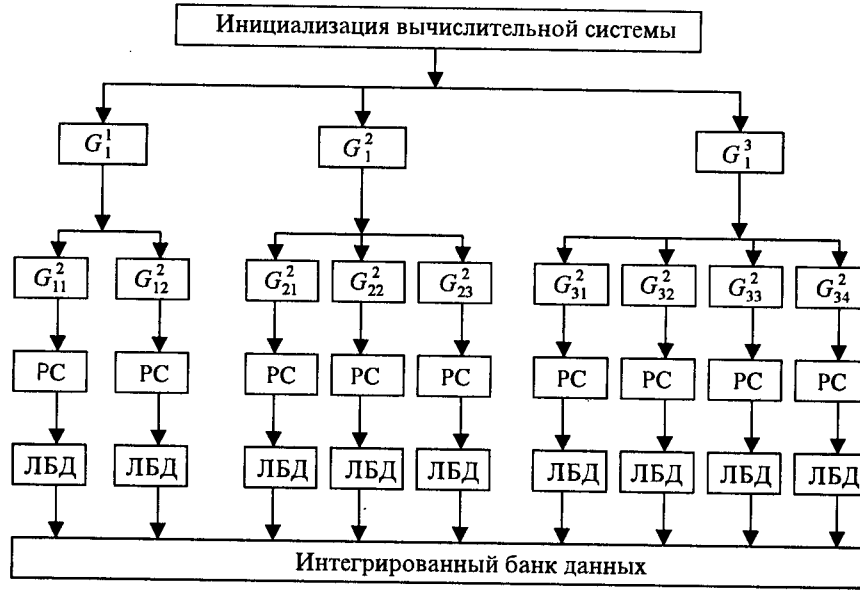


Рис. 1. Глобальная параллелизация вычислительной системы «Поток-3»

Заметим, что в общем случае длины списков во всех подгруппах m -уровня могут быть различными и определяться принадлежностью к той или иной подгруппе предыдущего уровня, что формализованно может быть представлено рекуррентной формулой

$$g_{mF} = g_{mF}(g_{m-1}), \quad m = \overline{2, M}. \quad (2)$$

Такая ситуация показана на рис. 1, иллюстрирующем двухуровневое глобальное распараллеливание. На первом G^1 уровне длина списка параметров $g_{1F} = 3$, т. е. организуется решение задачи с тремя значениями высоты полета H_1, H_2, H_3 . На втором G^2 уровне длины списков параметров определяются принадлежностью к той или иной подгруппе (ветви графа): для первой длина списка $g_{1,2F} = 2$, для второй $g_{2,2F} = 3$, для третьей $g_{3,2F} = 4$. В поясняющих рис. 1 терминах это означает заказ на организацию расчета: в варианте с высотой полета $H = H_1$ два подварианта со значениями скорости полета $V = V_{11}, V_{12}$; в варианте с $H = H_2$ три подварианта со значениями $V = V_{21}, V_{22}, V_{23}$; в варианте с $H = H_3$ четыре подварианта со значениями $V = V_{31}, V_{32}, V_{33}, V_{34}$. После организации файлов вариантов входных параметров с «индивидуальными» значениями $G_{g_1}^1, G_{g_1, g_2}^2$ каждый файл (окончательная наиболее удаленная от корня графа структура) передает данные в программный комплекс (РС) и инициирует его процессорную систему. Подчеркнем еще раз, что обмен данными в процессе исполнения всего задания в целом не происходит между процессорными подгруппами G^2 , ни тем более между процессорными блоками G^1 . В этом состоит сущность данного типа параллелизации – распараллеливания по глобальным параметрам программного комплекса (основным входным параметрам, как правило, скалярам, определяющим физико-математический смысл задачи).

Локальная параллелизация. Локальное распараллеливание, относящееся по классификации [1] к технологическому типу параллелизации «низшего уровня», предназначено для ускорения процесса вычисления в подпрограммах, составляющих основное алгоритмическое ядро РС (см. рис. 1) вычислительного комплекса «Поток-3» [3]:

$$\{PC\} = \sum_{n=1}^N \sum_{k=1}^K \sum_{i=1}^I \sum_{j=1}^J k - \text{OPERATIONS}(i, j; n), \quad (3)$$

$$k - \text{OPERATIONS}(i, j; n) \Rightarrow F_{ij}^{kn}. \quad (4)$$

Символическая запись (3), (4) означает следующее. Результатом функционирования k -й процедуры алгоритма, реализованной в виде отдельной подпрограммы

$$\text{SUBROUTINE}(k, \text{input parameters}, \text{output parameters}), \quad k \in [1, K], \quad (5)$$

на n -м итерационном слое (шаге по времени n)

$$t_n = \varphi_n(n), \quad n \in [1, N], \quad (6)$$

является двумерный массив F_{ij}^{kn} – прямоугольная таблица чисел (для краткости изложения рассматривается двумерная задача предметной области – плоская и осесимметричная), где i и j нумеруют соответственно по каждому (y, x) -координатному направлению узлы расчетной сетки, дискретизирующей непрерывную дифференциальную задачу:

$$y_i = \varphi_y(i), \quad i \in [1, I], \quad (7)$$

$$x_j = \varphi_x(j), \quad j \in [1, J]. \quad (8)$$

При использовании равномерного пространственно-временного шаблона (6)–(8) имеют вид $t_n = \tau n$; $y_i = i\Delta y$; $x_j = j\Delta x$, где τ , Δy и Δx – шаги по времени, y - и x -координатам соответственно.

Подпрограммы $k - \text{OPERATIONS}(i, j; n)$ многократно исполняются в главном итерационном процессе решения сложной системы дифференциальных уравнений и содержат внутри себя циклические операторы перебора индексов; внутри некоторых циклов содержится значительное число арифметических операций.

Для ускорения исполнения всех k -процедур главного итерационного блока программного комплекса были разработаны и использованы C -, L -, V -типы параллелизации, соответствующие распараллеливанию операций по столбцам, по строкам, по вектору (Column, Line, Vector). Одной из первых работ по распараллеливанию, и в частности рассмотрению типов параллелизации, отмеченных выше, является работа [6].

Условная схема параллелизации основного вычислительного ядра программного комплекса «Поток-3» (на рис. 1 обозначен символом РС) представлена на рис. 2. Каждый из типов параллелизации ориентирован на индивидуальные алгоритмические и реализующие их программные особенности

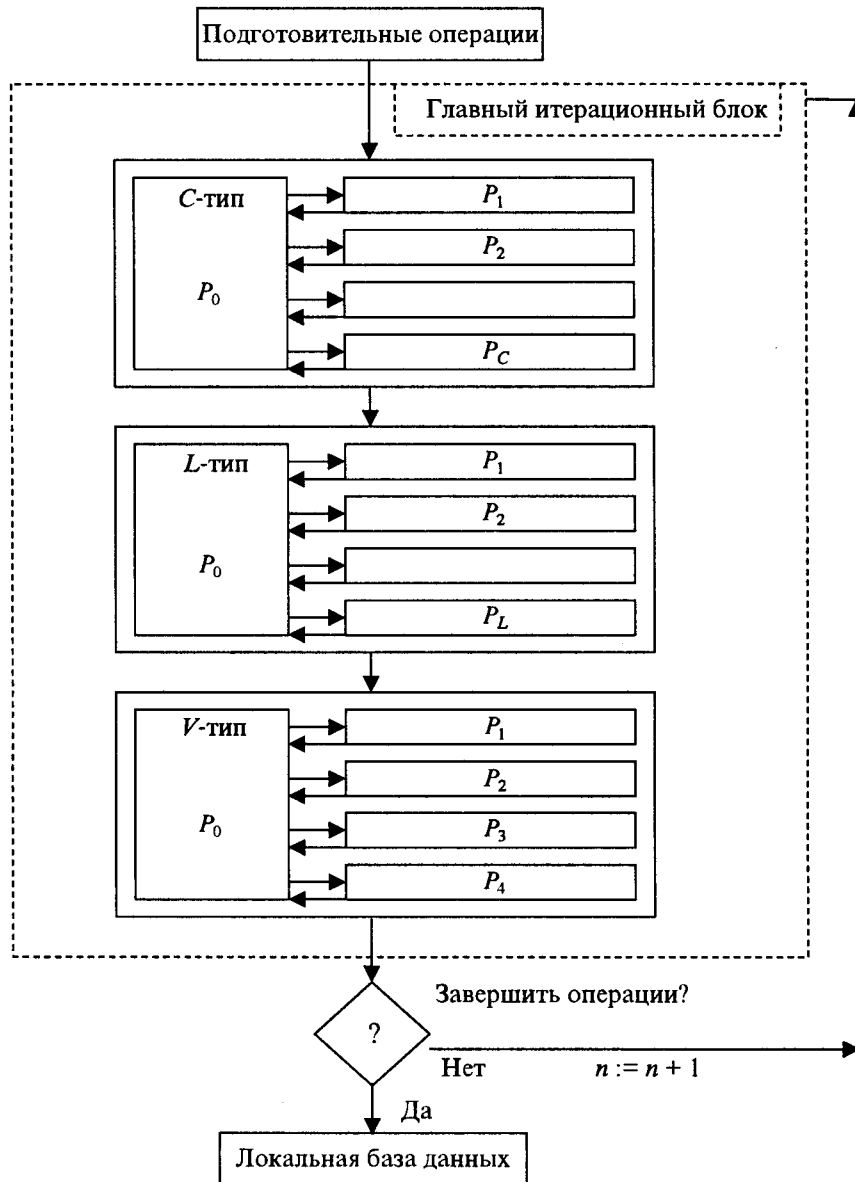


Рис. 2. Локальная параллелизация основного алгоритмического ядра вычислительной системы «Поток-3»

каждой k -процедуры с целью получить максимальный выигрыш в ускорении вычислительного процесса с минимизацией (насколько возможно) времени информационного интерфейса: подготовка к распараллеливанию процессором-диспетчером P_0 , рассылка заданий и данных по процессорам-исполнителям P_1, P_2, P_3 и т. д., сбор данных после исполнения заданий, их реорганизация в единый массив данных и передача управления в следующую $(k + 1)$ -процедуру.

Заметим, что количество и порядок следования C -, L - и V -типов параллельных процедур определяется алгоритмом и конкретным видом реализующих его подпрограмм.

Рассмотрим подробнее сущность каждого типа параллелизации (далее, если это не оговаривается особо, имеется в виду параллельная реализация подпрограмм комплекса, ориентированного на многопроцессорные системы с распределенной памятью).

Неявный конечно-разностный алгоритм главного итерационного ядра вычислительного комплекса «Поток-3» основан на расщеплении системы дифференциальных уравнений Навье – Стокса, применяющихся для моделирования нестационарных задач аэродинамики и физической газовой динамики,

$$\frac{\partial F}{\partial t} = W \left(F, \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial^2 F}{\partial x^2}, \frac{\partial^2 F}{\partial y^2}, \frac{\partial^2 F}{\partial x \partial y} \right), \quad (9)$$

$$F = F(t, x, y) \quad (10)$$

по координатным направлениям x и y :

$$W = W_1 + W_2 + W_3. \quad (11)$$

Расщепление (11) производится таким образом, чтобы разделить по возможности члены, содержащие координатные производные

$$W_1 = W_1 \left(F, \frac{\partial F}{\partial x}, \frac{\partial^2 F}{\partial x^2} \right), \quad (12)$$

$$W_2 = W_2 \left(F, \frac{\partial F}{\partial y}, \frac{\partial^2 F}{\partial y^2} \right), \quad (13)$$

$$W_3 = W_3 \left(F, \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial^2 F}{\partial x^2}, \frac{\partial^2 F}{\partial y^2}, \frac{\partial^2 F}{\partial x \partial y} \right), \quad (14)$$

где W_1 содержит производные только по x , W_2 – только по y , а W_3 является некоторой координатно-нерасщепляющейся частью.

Блоки системы уравнений W_1 и W_2 разделяются далее по физическим процессам (динамике и диссипации):

$$W_1 = W_{11} + W_{12}, \quad (15)$$

$$W_2 = W_{21} + W_{22}. \quad (16)$$

При этом зависимости W_{11} , W_{12} , W_{21} , W_{22} от вектора искомых функций F , записанные в матричной форме

$$W_{11} = A_{11} F, \quad (17)$$

$$W_{12} = A_{12}F, \quad (18)$$

$$W_{21} = A_{21}F, \quad (19)$$

$$W_{22} = A_{22}F, \quad (20)$$

характеризуются следующими свойствами: матрицы A_{11} и A_{21} являются диагональными, а матрицы A_{12} и A_{22} имеют существенную заполненность ненулевыми элементами (даже без диагонального преобладания).

Конкретный вид (9)–(20) для дальнейшего изложения значения не имеет и здесь не приводится (см. [7, 8]). Данная структура метода расщепления (9)–(20) детерминирует в общей стратегии параллелизации программного комплекса тактику локального (технологического по терминологии [1]) распараллеливания подпрограмм, реализующих этапы численного алгоритма (9)–(16) с учетом их особенностей (17)–(20).

С-тип параллелизации. Этот тип распараллеливания части вычислительного алгоритма ориентирован на ускорение вычислений в подпрограммах, реализующих расчет членов уравнений типа (13), т. е. содержащих производные (точнее, конечно-разностные операторы) только по y -координатному направлению. В сеточном (дискретном) пространстве операций (4) данный тип операндов может быть записан (индекс k для простоты записи опущен) как

$$F_{ij}^{n+v} = \text{OPERATIONS}(F_{ij}^n, F_{i\pm 1, j}^n, F_{i\pm 2, j}^n). \quad (21)$$

Запись (21) означает, что по известным значениям F на n -слое вычисляются значения на $(n+v)$ -слое. Символ v означает некоторый «дробный» (например, $v=1/6$, $v=2/6$ и т. д.) этап перехода на $(n+1)$ -й итерационный слой, однако в представляемой работе с неизбежным беглым изложением конкретизация и уточнения алгоритма нецелесообразны (см. [7]).

В операциях типа (21) имеет особое, даже ключевое, значение то обстоятельство, что при вычислении массива F_{ij}^{n+v} используются элементы массивов F_{ij}^n только с вариацией первого индекса ($i, i\pm 1, i\pm 2$), т. е. имеет место сцепление операций по первому (i) индексу и их независимость по второму индексу (j). Это означает возможность проведения операций одновременно (параллельно) по всем колонкам (столбцам) расчетной сетки, что показано на рис. 3, иллюстрирующем сущность C -типа распараллеливания. Общее сеточное пространство R :

$$\{R_{ij}\} = i \otimes j, \quad (22)$$

в области (7), (8) может быть разделено на P_C подпространств (вертикальных колонок), в каждой из которых операции (21) являются независимыми процессами и могут быть проведены параллельно:

$$F_{ij}^{n+v} = \sum_{p=1}^{P_C} p F_{ij}^{n+v}. \quad (23)$$

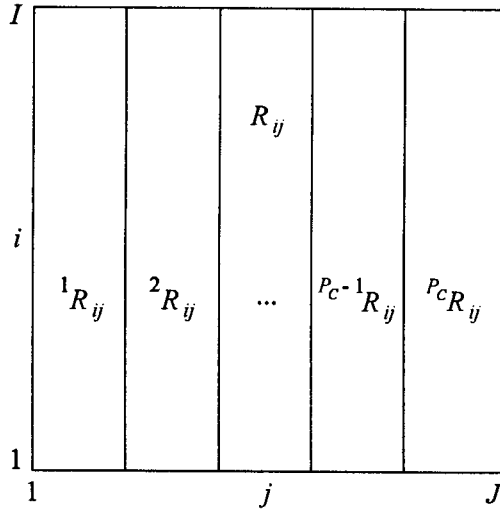


Рис. 3. С-тип параллелизации вычислений

Число независимых процессов P_C , естественно, должно удовлетворять условию

$$P_C \leq J. \quad (24)$$

Таким образом, для ускорения вычислений в (21) возможно использование P_C машин (процессоров).

Размеры колонок (см. рис. 3) могут быть разными, поскольку в общем случае при $P_C < J$ отношение J/P_C не является целым числом, но, очевидно, разбиение R на почти одинаковые колонки оптимально, поскольку при этом число операций в колонках почти одинаково, и, следовательно, время ожидания общего завершения всех процессов, т. е. операций (21) и (23), в R минимизируется.

Реализация этого типа распараллеливания в программном комплексе «Поток-3» была осуществлена следующим образом. По заказу пользователя числа процессоров P_C особый процессор-диспетчер P_0 производит сегментацию R -пространства на P_C подпространств, т. е. разделяет все задействованные в (21) массивы F_{ij} на P_C подмассивов ${}^P F_{ij}$ и рассылает их в P_C машин, которые осуществляют с ними операции (21) и затем возвращают их процессору-диспетчеру, составляющему по (23) полный массив $F_{ij}^{n+\nu}$.

L-тип параллелизации. Этот вид распараллеливания части вычислительного алгоритма ориентирован на ускорение вычислений в подпрограммах, реализующих расчет членов типа (12), т. е. содержащих производные (точнее, конечно-разностные операторы) только по x -координатному направлению. В сеточном (дискретном) пространстве операций (4) данный тип может быть записан (индекс k для простоты опущен) как

$$F_{ij}^{n+\mu} = \text{OPERATIONS}(F_{ij}^n, F_{i, j \pm 1}^n, F_{i, j \pm 2}^n). \quad (25)$$

Запись (25) также означает, что по известным значениям F на n -слое вычисляются значения на $(n + \mu)$ -слое. Словом, μ есть некоторый промежуточ-

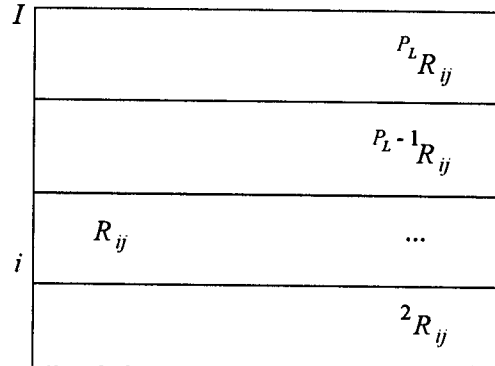


рис. 4. L-тип параллелизации вычислений

ный этап перехода на $(n+1)$ -й итерационный слой. В операциях (25) в отличие от операций (21) имеют место сцепление вычислений по второму (j) и их независимость по первому (i) индексам. Это означает возможность проведения операций параллельно по всем строкам расчетной сетки, как показано на рис. 4. Общее сеточное R -пространство (22) в области (7), (8) может быть разделено на P_L подпространств (горизонтальных полос), в каждом из которых операции (25) являются независимыми и могут быть выполнены параллельно:

$$F_{ij}^{n+\mu} = \sum_{\rho=1}^{P_L} {}^{\rho}F_{ij}^{n+\mu}. \quad (26)$$

Число независимых процессов P_L очевидно и должно удовлетворять условию

$$P_L \leq I. \quad (27)$$

Таким образом, для ускорения вычислений в (25) возможно использование P_L процессоров.

Размеры полос (см. рис. 4) могут быть разными, поскольку в общем случае при $P_L < I$ отношение I/P_L не является целым числом, но, естественно, разбиение R на почти одинаковые полосы оптимально.

Реализация этого типа распараллеливания в программном комплексе «Поток-3» аналогична реализации S -типа. По заказу пользователя числа процессоров P_L особый процессор-диспетчер P_0 осуществляет распределение R -пространства на P_L подпространств, т. е. разделяет все задействованные в (25) массивы F_{ij} на P_L подмассивов ${}^{\rho}F_{ij}$ и направляет их в P_L машин, которые производят с ними операции (25) и затем возвращают их процессору-диспетчеру, составляющему по (26) полный массив $F_{ij}^{n+\mu}$.

Несмотря на идеологическое сходство C - и L -типы параллелизации весьма существенно отличаются при их компьютерной реализации на много-процессорных системах. Рассмотрим конкретно функционирование процессора-диспетчера (рис. 5), распределяющего элементы полного массива a_{ij} по процессорам-исполнителям ($P_C = P_L = 3$).

На рис. 5, *a* показано традиционное математическое представление массива a_{ij} с распределением его элементов вдоль строк (вариация j) и вдоль столбцов (вариация i). В компьютерном представлении двумерный массив a_{ij} реконфигурируется в некоторое адресное пространство, которое можно для иллюстрации трактовать в геометрическом смысле как одномерную прямую линию (рис. 5, *b*) с последовательным расположением элементов массива a_{ij} .

На рис. 5, *c* над узлами, изображающими элементы a_{ij} , поставлены цифры, соответствующие номерам процессоров-получателей этих элементов, выполняющих над ними арифметические операции (21) и по их окончании возвращающих эти подмассивы диспетчеру для «сборки» в общий массив (23).

При L -типе параллелизации процессор-диспетчер при разделении массива a_{ij} на P_L подмассивов (рис. 5, *d*, $P_L = 3$) определяет в адресном пространстве (см. рис. 5, *b*) расположение соответствующих элементов a_{ij} . На рис. 5, *d* над узлами, изображающими элементы a_{ij} , поставлены цифры, соответствующие номерам процессоров-получателей этих элементов, выполняющих над ними арифметические операции (25) и по их окончании возвращающих эти подмассивы диспетчеру для «сборки» в общий массив (26).

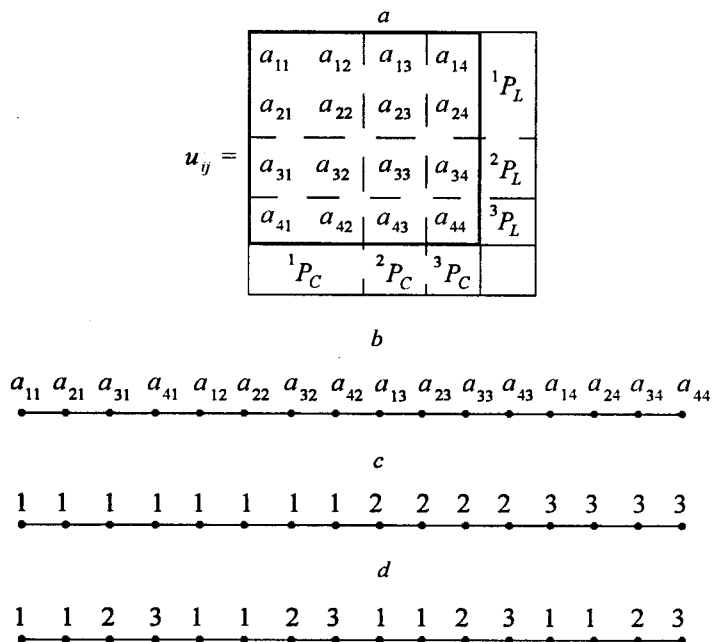


Рис. 5. Реконфигурация процессорного пространства: *a* – двумерное представление массива a_{ij} , $i \in [1, 4]$, $j \in [1, 4]$; *b* – одномерная Fortran-развертка двумерного массива a_{ij} ; *c* – распределение элементов массива a_{ij} при C -типе параллелизации на трех процессорах; *d* – распределение элементов массива a_{ij} при L -типе параллелизации на трех процессорах

На рис. 4 ширина полосы или 2, или 1, так как среднее значение ширины

$$S = I/P_L = 4/3. \quad (28)$$

Таким образом, элементы 1, 2 и 3 подмассивов будут располагаться группами по 2 элемента для первой полосы и одиночно для второй и третьей полос, а интервалы между элементами одной и той же группы, например первой, будут определены как

$$\text{Int} = \frac{J(P_L - 1)}{P_L}. \quad (29)$$

Отметим, что в случаях, когда I не только не делится нацело на P_L , но и используются полосы разной ширины, формула (29) существенно усложняется.

В связи с вышеизложенным можно констатировать, что L -тип параллелизации в практической реализации значительно более сложен, чем C -тип, требует существенного внимания от программиста и более затратен по компьютерным ресурсам (времени реконфигурации вычислительного пространства).

V -тип параллелизации. Этот вид распараллеливания части вычислительного алгоритма ориентирован на ускорение вычислений в подпрограммах, реализующих расчет членов уравнений типа (17) и (19), и может дополнять в них C - и L -типы параллелизации. V -тип параллелизации детерминруется диагональностью матриц A_{11} и A_{21} в (17) и (19), что приводит к независимости в некоторых подпрограммах комплекса «Поток-3» вычисления (10) компонент вектора $F = (F_1, F_2, F_3, F_4)$. Для трехмерной задачи вектор F имеет пять компонент (см. [7]).

Этот тип распараллеливания достаточно прост в реализации: процессор-диспетчер рассылает необходимые данные четырем процессорам-исполнителям, которые после проведения необходимых операций возвращают новые значения массивов $F_{1_u}^{n+u}$, $F_{2_u}^{n+u}$, $F_{3_u}^{n+u}$, $F_{4_u}^{n+u}$ (здесь, как и выше, символом u обозначен некоторый «дробный» шаг продвижения решения в общем алгоритме от F^n к F^{n+1}).

Обозначим для общности число процессоров, необходимых для этого типа параллелизации, как P_V (плюс процессор-диспетчер, хотя в этом случае процесс организации данных, их рассылки и последующего приема весьма незатратен по компьютерным ресурсам, и диспетчеру можно «поручить» работу над одной из компонент вектора F , уменьшая на единицу число процессоров-исполнителей). Рассмотрим кратко некоторые технические вопросы, связанные с соотношением значений P_C , P_L и P_V . Если первые два могут принимать любые положительные значения, то значение P_V жестко детерминировано: $P_V = 4$ для двумерной и $P_V = 5$ для трехмерной задач.

Если выбрать P_C или P_L значительным (этот выбор определяется только возможностями доступа вычислителя к той или иной многопроцессорной системе с различными техническими данными), то в момент исполнения подпрограмм, параллелизованных по V -типу, $\max(P_C, P_L) - P_V$ процессоров будут находиться в режиме ожидания (т. е. простаивать), что не вполне оптимально.

Заметим, что подобная проблема существует и для соотношения значений P_C и P_L , которая может быть решена простым выбором $P_C = P_L$. Вообще говоря, этот вопрос более сложен, чем кажется на первый взгляд, и весьма дискуссионен, поскольку требования к эффективности решения научной задачи и оптимальности использования ресурсов ЭВМ могут находиться в сильном противоречии. Разумеется, никто не станет возражать против того, чтобы научная задача была решена эффективно с минимальными затратами компьютерных ресурсов, но если нет оптимума между этими полюсами, то следует, на наш взгляд, предпочесть эффективное решение с существенно «непродуктивным» использованием МВС, чем «экономное» использование МВС для получения неэффективного решения.

Авторы выражают благодарность В. Э. Малышкину, Н. В. Кучину, В. А. Вшивкову и Г. С. Хакимзянову за полезные обсуждения и внимание к работе.

СПИСОК ЛИТЕРАТУРЫ

1. Тарнавский Г. А., Шпак С. И. Декомпозиция методов и распараллеливание алгоритмов решения задач аэродинамики и физической газовой динамики: вычислительная система «Поток-3» // Программирование. 2000. № 6. С. 45.
2. Тарнавский Г. А., Вшивков В. А., Тарнавский А. Г. Параллелизация алгоритмов и кодов вычислительной системы «Поток-3» // Программирование. 2003. № 1. С. 1.
3. Тарнавский Г. А., Шпак С. И. Схемы распараллеливания операций решения систем алгебраических уравнений методом многомерной скалярной прогонки // Вычислительные методы и программирование. 2000. 1. С. 21: <http://www.srcc.msu.su/num-meth>
4. Вшивков В. А., Тарнавский Г. А., Неупокоев Е. В. Распараллеливание алгоритмов прогонки: многоцелевые вычислительные эксперименты // Автометрия. 2002. 38, № 4. С. 74.
5. Тарнавский Г. А., Тарнавский А. Г. Современные компьютерные технологии и неединственность решений задач газовой динамики // Симметрия и дифференциальные уравнения / Под ред. В. К. Андреева. Красноярск: Изд-во ИВТ СО РАН, 2002. С. 209.
6. Евреинов Э. В., Косарев Ю. Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск: Наука, 1966.
7. Ковеня В. М., Тарнавский Г. А., Черный С. Г. Применение метода расщепления в задачах аэродинамики. Новосибирск: Наука, 1990.
8. Тарнавский Г. А., Шпак С. И. Проблемы численного моделирования сверхзвукового ламинарно-турбулентного обтекания тел конечного размера // Математическое моделирование. 1998. 10, № 6. С. 53.

*Институт теоретической и прикладной механики СО РАН,
Институт вычислительной математики
и математической геофизики СО РАН,
E-mail: tarnav@itam.nsc.ru,
korneev@ssd.sccc.ru*

*Поступила в редакцию
25 марта 2003 г.*