РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
АВТОМЕТРИЯ
№ 3 1995

УДК 519.68 : 681.32

В. А. Леус, А. И. Мишин
(Новосибирск)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ГЛОБАЛЬНОЙ И ЛОКАЛЬНОЙ РЕАЛИЗАЦИЙ АЛГОРИТМОВ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Проводится сравнительный анализ локальной и глобальной параллельных реализаций алгоритмов решения задач математической физики на расширенной N-процессорной решетке, в которой, кроме локальных межпроцессорных связей, имеются также и глобальные связи по строкам и столбцам. Показано, что временная сложность локальной параллельной реализации алгоритмов решения задач механики сплошной среды методом дробных шагов при числе процессоров $N = w^{r-1}$ (здесь r — число пространственных переменных, w — целочисленный размер сетки дискретизации по пространственной координате) такая же, что и при традиционном использовании этого метода, требующем в w раз больше процессоров. Лишь при решении задач малого размера, когда временем доставки данных к процессорам можно пренебречь, глобальная реализация не уступает локальной, а при распараллеливании алгоритмов с частыми сменами направлений глобальных пересылок информации (например, четно-нечетная редукция для ленточных матриц размером N) глобальная реализация оказывается существенно эффективнее локальной и позволяет получить ускорение в $\sqrt{N}$ раз на матричной системе со строчными и столбцевыми шинами. В зависимости от условий реализации (разовое или многократное использование, размер задачи, схема межпроцессорных связей) понятия «быстрый» и «медленный» применительно к алгоритму могут изменяться на противоположные. Так, в режиме потоковых вычислений классический алгоритм прогонки превосходит алгоритмы, основанные на коммуникационной схеме сдваивания и реализуемые специализированной вычислительной структурой «гиперкуб».

Введение. В данной работе проводится сравнительный анализ глобальной и локальной реализаций наиболее важных алгоритмов математической физики на вычислительной системе (ВС) с глобальными и локальными межпроцессорными связями. Даются оценки временной сложности с учетом затрат времени на операции суперпозиции и подстановки, соответствующие доставкам данных к процессорам. В зависимости от условий применения (разовое или многократное использование алгоритма, размер задачи, схема связи процессоров в ВС) понятие «быстрый» («медленный») алгоритм может изменяться на противоположное. В качестве вычислительной модели параллельной ВС используется двумерная процессорная решетка, в которой, помимо локальных межпроцессорных связей, могут устанавливаться (программно или технологически) также и необходимые глобальные. Возможность физической реализации такой модели рассмотрена в [1], где сравниваются также локальная и глобальная реализации преобразования Фурье.

В [2] даны оценки вычислительных затрат на решение многомерных задач механики сплошной среды. Исследовано распараллеливание на глобальной вычислительной системе с числом процессоров $N = w^r$, имитирующей пространственную сетку дискретизации с w^r узлами, где w — характерный размер сетки по координате, r — число пространственных переменных ($r = 1, 2, 3$). Полученные результаты справедливы при решении задач небольших размеров на параллельных ВС, в которых временем межпроцессорных обменов можно пренебречь в сравнении с временем вычислений. Для больших

ВС, использующих быстродействующие процессоры, эти оценки следует умножать на весовой коэффициент, полиномиально растущий с увеличением размера задачи (количества узлов сетки), чтобы учесть временные затраты на доставку данных к процессорам.

В работе рассматривается распараллеливание этих задач на параллельной ВС, где операции подразделяются на глобальные, у которых длина доставки операндов полиномиально растет с увеличением размера задачи, и локальные, для которых операнды берутся от соседних элементов. Решение задач механики сплошной среды методом дробных шагов [3] на локальной ВС, содержащей лишь w^{r-1} процессоров, позволяет достичь минимальной временной сложности.

Метод дробных шагов заменяет многомерную задачу на чередующиеся по пространственным переменным двумерные (координата и время) задачи, что при пятиточечном шаблоне приводит к необходимости решения систем линейных алгебраических уравнений с трехдиагональными матрицами. Широко применяемым для этого алгоритмом является прогонка (sweeping). Распараллеливанию прогонки, т. е. замене ее параллельным алгоритмом, решающим ту же задачу, посвящено большое число работ (см., например, [4]). Следует подчеркнуть, что эти работы не выходят за рамки традиционных представлений, игнорирующих временные затраты на операции суперпозиции. В [5] предложен алгоритм, разбивающий ленту матрицы на K частей, в каждой из которых прогонки ведутся независимо с участием так называемых параметрических переменных. Этот алгоритм, обобщающий известную встречную прогонку, эффективен при постановке на K-процессорной ВС для крупноблочного распараллеливания, когда K много меньше размера задачи $N = w^r$, поскольку время межпроцессорных обменов в этом случае значительно меньше вычислительного. В случае предельного (массового) параллелизма, когда $K \sim N$, вне конкуренции оказываются алгоритмы, использующие так или иначе схему сдваивания (binary tree layout).

Согласно вычислительной схеме сдваивания, строки ленточной матрицы обрабатываются попарно и независимо за $\sim\log_2 N$ алгоритмических этапов. На первом этапе операция производится с соседними строками, на втором этапе операндами являются строки, отстоящие на две позиции, на третьем этапе — строки, отстоящие на четыре позиции и т. д., на p-м этапе «взаимодействуют» строки, отстоящие на 2^{p-1} позиций. Таким образом, при числе процессоров $N = 2^p$ можно решить задачу за $\sim\log_2 N$ алгоритмических шагов (этапов). Отметим, что при решении задачи методом сдваивания на традиционной векторной машине (например, CRAY-1) ускорение вычислений не превышает семикратного и подходит к насыщению уже при $N = 1000$ [6]. Подобные эксперименты лишь подтвердили то, что теоретически стало ясным еще до их постановки: в эффективности параллельных вычислений решающую роль играет не только количество арифметических операций, но и сеть информационных коммуникаций в ВС и ее адекватность схеме взаимодействий между ветвями алгоритма.

Параллельное решение задач математической физики. Нестационарные многомерные задачи механики сплошной среды описываются системой дифференциальных уравнений в частных производных, которую можно представить в следующей векторной форме:

$$A\frac{\partial}{\partial t}U + B\frac{\partial}{\partial x}U + C\frac{\partial}{\partial y}U + EU = F, \qquad (1)$$

где U — упорядоченная совокупность из v неизвестных функций; F — вектор заданных функций (аналог свободных членов); A, B, C, E — квадратные матрицы размером v [2]. Считается, что вся зависимость коэффициентов, образующих матрицы, от пространственных координат и времени сосредоточена в их зависимости от компонент вектора неизвестных функций U, т. е. коэффициенты рассчитываются по найденным в точке значениям компонент U конечным числом арифметических действий. Краевая задача для системы

(1) состоит в отыскании на единичном квадрате такого решения U, которое удовлетворяет заданным начальным условиям $U(0, x, y) = U^0$ в квадрате и граничным условиям $U(t, r) = U^r$ в произвольной точке границы квадрата.

При дискретизации этой задачи вводится равномерная сетка в единичном квадрате с шагом $h = 1/n$, шаг по времени ε, а значение переменной U, вычисляемое в узле $x_m = mh$, $y_n = nh$ $(m, n = 0, ..., w)$ на i-м временном срезе $t^i = i$, обозначается $U_{mn} = U(t^i, x_m, y_n)$. Производные заменяются конечно-разностными выражениями

$$\frac{\partial U}{\partial t} \approx (U^{i+1}_{mn} - U^i_{mn})/\varepsilon, \qquad \frac{\partial U}{\partial x} \approx \frac{U^{i+1}_{m+1,n} - U^{i+1}_{m-1,n}}{2h}, \qquad \frac{\partial U}{\partial y} \approx \frac{U^i_{m,n+1} - U^i_{m,n-1}}{2h}, \tag{2}$$

причем для v-мерного вектора U эти представления имеют покомпонентный смысл.

Относя уравнение (1) к узлу (m, n) i-го временного среза и подставляя в него выражения (2), получим неявную разностную схему

$$(A^i_{mn} + \varepsilon E^i_{mn})U^{i+1}_{mn} = A^i_{mn}U^i_{mn} -$$

$$- \frac{\varepsilon}{2h}\left[B^i_{mn}(U^{i+1}_{m+1,n} - U^{i+1}_{m-1,n}) + C^i_{mn}(U^{i+1}_{m,n+1} - U^{i+1}_{m,n-1})\right] + \varepsilon F^i_{mn}. \tag{3}$$

Это соотношение порождает на каждом временном срезе систему линейных алгебраических уравнений с клеточно-ленточной матрицей (размеры клеток $w \times w$) для неизвестных U_{mn} $(m, n = 1, 2, ..., w - 1)$ с учетом граничных условий. Точнее говоря, получается v таких систем по числу компонент функционального вектора U.

Чтобы упростить строение матриц, прибегают к расщеплению конечно-разностных аппроксимаций по пространственным направлениям [1], в связи с чем вводится понятие временного полушага $\varepsilon/2$ и дробный индекс $(i = = 1/2)$. Вдоль направления x уравнение (1) заменяется разностным:

$$A^i_{mn}\frac{(U^{i+1/2}_{mn} - U^i_{mn})}{\varepsilon/2} + B^i_{mn}\frac{(U^{i+1/2}_{m+1,n} - U^{i+1/2}_{m-1,n})}{2h} + 0 + E^i_{mn}U^{i+1/2}_{mn} = F^i_{mn}, \tag{4}$$

где частная производная по y вообще не аппроксимируется, а заменяется нулем. Вдоль направления y, наоборот, предполагается, что частная производная по x равна нулю и (1) заменяется выражением

$$A^{i+1/2}_{mn}\frac{(U^{i+1}_{mn} - U^{i+1/2}_{mn})}{\varepsilon/2} + 0 + C^{i+1/2}_{mn}\frac{(U^{i+1}_{m,n+1} - U^{i+1}_{m,n-1})}{2h} + E^{i+1/2}_{mn}U^{i+1}_{mn} = F^{i+1/2}_{mn}. \tag{5}$$

Применяя соотношения (4) и (5) в узлах сетки, получают соответственно систему линейных алгебраических уравнений, где неизвестные имеют дробный верхний индекс $(i + 1/2)$, и систему с целочисленными индексами $(i + 1)$ у неизвестных. Хотя каждая из этих систем сама по себе не аппроксимирует континуальную задачу, попеременное их использование в среднем обеспечивает сходящееся приближение. Благодаря такому расщеплению двумерная задача заменяется на две одномерные. На каждом временном срезе вместо решения одной системы из $w \times w$ уравнений с клеточной матрицей производится решение $2w$ систем по w уравнений и с простыми ленточными матрицами. Помимо сокращения необходимого числа операций, здесь приобретается хорошая возможность распараллеливания процесса.

Параллельное решение двумерных задач наиболее просто осуществляется на процессорной решетке с w^r процессорами, адекватной пространственной разностной сетке, где каждый (m, n)-й узел поставлен в соответствие (m, n)-му процессору. Рассмотрим реализацию неявной схемы по методу дробных шагов

(4) и (5) на этой процессорной решетке. Системы линейных алгебраических уравнений типа (4) на первом полушаге временного среза решаются за время $T_л$ специальными методами независимо в каждой строке процессорной решетки, соответствующей направлению оси x, т. е. при фиксированном n. На втором полушаге временного среза решаются системы типа (5) независимо в каждом столбце решетки, соответствующем направлению оси y при фиксированном m. Затем каждый процессор вычисляет новые значения коэффициентов для следующего временного среза, используя только локальные операции. Операнды для такой операции берутся от ближайших соседей. В этом случае работа по вычислению коэффициентов для каждого процессора состоит в выполнении фиксированного числа K локальных операций, поэтому время такого вычисления есть $T_K \approx K\tau$, где τ — процессорный цикл. Общая временная сложность реализации неявной схемы оценивается как

$$T_н \approx I(2T_л + T_K), \tag{6}$$

где I — число шагов по времени.

Покажем, что оценка (6) достижима и на линейной ВС из w процессоров, каждый из которых содержит блок кольцевой памяти, образованной цепочкой из $2w$ регистров, закольцованной через процессор. В каждом регистре хранится вся информация об (m, n)-м узле разностной сетки. Предполагается, что каждый процессор линейки, взаимодействуя только с двумя непосредственно с ним связанными регистрами, может управлять движением информации по кольцевой памяти в обоих направлениях.

Каждый процессор линейки, имеющий номер n, производит решение системы линейных алгебраических уравнений на первом полушаге временного среза, «прокручивая» данные в своей кольцевой памяти узел за узлом. На втором полушаге временного среза нужно решать системы типа (5) по столбцам пространственной сетки при фиксированных m. Непосредственных связей в этом направлении между регистрами различных блоков памяти не существует, поэтому движение информации возможно только через процессорную линейку. Рассмотрим реализацию некоторого последовательного алгоритма (типа исключения переменных) в этом случае.

В исходном состоянии все процессоры, кроме первого, установлены на прием информации от младших соседей. Пусть в процессорной линейке расположены данные 1-го столбца пространственной сетки. Первый процессор обрабатывает первый элемент столбца, результирующее слово записывает в 1-й регистр своей кольцевой памяти и также передает его второму процессору. После этого первый процессор выбирает из своей кольцевой памяти информацию узла, относящегося ко второму столбцу, и вычисляет первое промежуточное результирующее слово для системы уравнений второго столбца. Далее записывает вычисленное значение во 2-й регистр своей кольцевой памяти и выходит на передачу информации второму процессору. Второй процессор, приняв 1-е промежуточное слово 1-го столбца, вычисляет 2-е слово для этого столбца, записывает его в 1-м регистре своей кольцевой памяти и передает экземпляр вычисленного слова третьему процессору. Затем он выбирает из своей кольцевой памяти информацию узла, относящегося ко второму столбцу, и выходит на прием информации от первого процессора и т. д. Каждый n-й процессор, получив от младшего соседа $(n - 1)$-е результирующее слово m-го столбца, вычисляет n-е слово для этого столбца, записывает его в m-й регистр своей кольцевой памяти и передает один экземпляр этого слова старшему соседу по линейке. Затем выбирает из своей кольцевой памяти информацию следующего узла, относящегося к $(m + 1)$-му столбцу, и выходит на прием информации от младшего соседа.

Информация, относящаяся к строке разностной сетки, обрабатывается на первом полушаге каждым процессором независимо в порядке расположения в кольцевой памяти за время, пропорциональное w. На втором полушаге, один за другим включаясь в работу, процессоры выполняют прямой ход процесса за время, пропорциональное w. Обратный ход с вычислением неизвестных выполняется аналогично начиная с w-го процессора, только направление передач

информации по линейке меняется на противоположное. Таким образом, общее время решения алгебраических систем есть $T_л = cw\tau$, где $c < 10$. Что касается этапа перевычисления коэффициентов для следующего временного среза, то здесь имеется отличие от варианта решения задачи на процессорной решетке. Прокручивая свои кольцевые блоки памяти после того, как найдены все значения U_{mn}^{i+1} для данного временного среза, процессоры последовательно загружаются информацией о столбцах сетки. Каждый процессор вычисляет новые значения коэффициентов, обмениваясь только с непосредственными соседями, и в целом время перевычисления коэффициентов пропорционально числу слов w, хранящихся в кольцевой памяти: $T_K \approx Kw\tau$. Общее время решения задачи по неявной схеме на линейной ВС оказывается того же порядка, что и в (6).

Вышеизложенное легко переносится на трехмерные по пространству задачи, которые описываются уравнениями типа (1) с добавлениями слагаемого $D\frac{\partial U}{\partial z}$. Метод расщепления применительно к таким задачам состоит во введении дробных шагов по времени $\varepsilon/3$ и замене разностного уравнения (3) тремя «урезанными» уравнениями, например, третье из которых имеет вид

$$A_{mnk}^{i+2/3}\frac{(U_{mnk}^{i+1}-U_{mnk}^{i+2/3})}{\varepsilon/3}+0+0+D_{mnk}^{i+2/3}\frac{(U_{mnk}^{i+1}-U_{mn,k-1}^{i+1})}{2h}+$$

$$+E_{mnk}^{i+2/3}U_{mnk}^{i+1}=F_{mnk}^{i+2/3}. \qquad (7)$$

Так называемая экономичная неявная схема с переменными направлениями исключения неизвестных состоит для трехмерного случая в последовательно циклическом решении на каждом временном срезе трех систем типа (7) из w уравнений каждая. Решение их проще всего осуществить параллельно на трехмерной решетке процессоров, каждый из которых поставлен в соответствие одному узлу пространственной сетки. Временная сложность этой реализации отличается от (6) только множителем перед $T_л$. Таким образом, для r-мерной пространственной сетки временная сложность имеет вид

$$T_н \approx I(rT_л + T_K). \qquad (8)$$

Однако трехмерная задача может быть без ущерба во времени решена на ВС, процессоры которой образуют решетку на единицу меньшей размерности. Такая ВС состоит из w процессорных линеек, процессоры которых связаны в поперечном направлении в такие же линейки. Строки полученной таким образом квадратной решетки размером $w \times w$ соответствуют направлению оси y, столбцы — направлению оси z. Направление оси x представлено кольцевыми блоками памяти, в каждом регистре которых хранится информация об (m, n, k)-м узле пространственной сетки. Исключение по направлению оси x осуществляется, как и в двумерной задаче, независимо каждым процессором, взаимодействующим локально с регистрами своей кольцевой памяти. Исключение по y реализуется как в двумерной задаче, только в количестве w экземпляров процесса по числу строк процессорной решетки. Наконец, исключение по оси z выполняется так же, как и процессы по y, только работающими «линейками» здесь становятся столбцы процессорной решетки. Вычисление коэффициентов по аналогии с двумерным случаем идет посредством прокручивания каждым процессором своей кольцевой памяти с выполнением операций в конвейерном режиме. Оценка временной сложности здесь также эквивалентна оценке (8), хотя число процессоров в w раз меньше.

В случае глобальной реализации алгоритма время исполнения операции зависит от размера ВС, работающей под общим тактированием с памятью произвольного доступа, и фактическое время решения равно произведению числа тактов работы на длительность такта, полиномиально растущую с увеличением размера ВС.

Распараллеливание процесса решения систем уравнений с ленточными матрицами. Пусть при четном N имеется система линейных алгебраических уравнений с двудиагональной матрицей:

$$b_1x_1 = f_1,$$
$$a_2x_1 + b_2x_2 = f_2,$$
$$\cdots\cdots\cdots$$
$$a_Nx_{N-1} + b_Nx_N = f_N.$$

Выражая из нечетных уравнений нечетные неизвестные и подставляя их в четные уравнения с номерами, на единицу большими, получим

$$b_2x_2 = f_2 - a_2f_1/b_1,$$
$$-x_2a_3a_4/b_3 + b_4x_4 = f_4 - a_4f_3/b_3,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$-x_{N-2}a_{N-1}a_N/b_{N-1} + b_Nx_N = f_N - a_Nf_{N-1}/b_{N-1},$$

т. е. опять же систему с двудиагональной матрицей, но вдвое меньшего размера. Если $N = 2^P$, то, поступая таким же образом далее, через P этапов найдем неизвестные $x_1, x_2, x_4, x_8, \ldots, x_{N/2}, x_N$. Остальные $(N - P - 1)$ неизвестных вычисляются на $(P - 1)$ возвратных этапах. В этом и состоит метод четно-нечетного исключения, благодаря которому на N-процессорной ВС задача решается за $\log N$ этапов, причем каждый процессор выполняет небольшое число g арифметических операций.

Линейные алгебраические уравнения вида

$$a_ix_{i-1} + b_ix_i + c_ix_{i+1} = f_i,$$

где $i = \overline{1, N}$, $a_1 = c_N = 0$, образуют систему с трехдиагональной матрицей. Выражая из четных уравнений четные неизвестные (можно также выражать нечетные из нечетных уравнений), но подставляя их уже в два уравнения — предыдущее и последующее, получим систему с трехдиагональной матрицей и вдвое меньшим числом неизвестных. Повторяя процесс $\log N$ раз, приходим к одному уравнению с одним неизвестным, которое легко вычисляется, а затем на $(P - 1)$ возвратных этапах вычисляются все неизвестные. Таков метод четно-нечетной редукции.

Строки, над которыми на данном этапе производятся операции, назовем активными. На первом этапе активными являются строки с истинными номерами $i = 1 \cdot 2^0, 2 \cdot 2^0, 3 \cdot 2^0, \ldots, 2^P \cdot 2^0$, на втором этапе активны строки с истинными номерами $i = 1 \cdot 2^1, 2 \cdot 2^1, 3 \cdot 2^1, \ldots, 2^{P-1} \cdot 2^1$, на p-м этапе активны строки с номерами $i = 1 \cdot 2^{p-1}, 2 \cdot 2^{p-1}, 3 \cdot 2^{p-1}, \ldots, 2^{P-p+1} \cdot 2^{p-1}$. На последнем P-м этапе прямого хода активны строки с номерами $i = 1 \cdot 2^{P-1}, 2 \cdot 2^{P-1}$, т. е. $N/2$-я и N-я. Если ввести текущую натуральную нумерацию активных строк $k = 1, 2, \ldots, 2^{P-p+1}$, то их истинные номера для p-го этапа суть $i_k = k \cdot 2^{p-1}$. Разность истинных номеров двух соседствующих в текущей нумерации строк на p-м этапе есть

$$d(p) = (i_k - i_{k-1}) = k2^{p-1} - (k - 1)2^{p-1} = 2^{p-1}.$$

Она не зависит от k и экспоненциально возрастает от 1 до $2^{P-1} = N/2$ с увеличением p. Количество активных строк, наоборот, убывает от этапа к этапу как $N2^{1-p}$.

Если обозначить S_k k-ю активную строку (в текущей нумерации), то операция, производимая над четными активными строками в методе четно-нечетного исключения, приобретает вид

$$S'_k = S_k - (a_k/b_{k-1})S_{k-1}. \qquad (9)$$

В методе четно-нечетной редукции операция производится с участием двух соседних строк:

$$S'_k = S_k - (a_k/b_{k-1})S_{k-1} - (c_k/b_{k+1})S_{k+1}. \qquad (10)$$

Взяв (10) за основу действий над строками и устранив убывание числа активных строк, получим метод четно-нечетного отдаления, основная операция которого имеет вид:

$$S'_i = S_i - (a_i/b_{i-d(p)})S_{i-d(p)} - (c_i/b_{i+d(p)})S_{i+d(p)}, \qquad i = \overline{1, N}. \qquad (11)$$

Операция (11), примененная к каждой строке, преобразует исходную трехдиагональную матрицу таким образом, что элементы главной диагонали меняют значения, а обе побочные (под- и наддиагональ) еще и отодвигаются на 2^{p-1} позиций на каждом p-м этапе вычислений. После P-го этапа сдвиг оказывается равным $1 + 2 + 4 + \ldots + N/2 = (N - 1)$, т. е. матрица становится диагональной и все неизвестные определяются просто делениями.

На каждом p-м этапе четно-нечетного исключения и четно-нечетной редукции список истинных номеров активных строк заканчивается номером $i = 2^{P-p+1} \cdot 2^{p-1} = N$, следовательно, N-я строка остается активной на протяжении всех P этапов. Операция (9) (или (10)), производимая над нею на последнем P-м этапе, использует опосредованную информацию каждой строки исходной матрицы.

Действительно, пусть I есть истинный номер произвольно выбранной строки, отличный от N. Разность $(N - 1)$ обязательно меньше N и больше нуля, поэтому ее двоичное представление содержит не более P значащих цифр $i_p \in \{0, 1\}$, из которых хотя бы одна равна единице:

$$I = i_1 2^0 + i_2 2^1 + i_3 2^2 + \ldots + i_P 2^{P-1}. \qquad (12)$$

В соответствии с (9) и (10) активная строка на p-м этапе либо, если ее несущий номер четен, «принимает» коэффициенты от строки с истинным номером на 2^{p-1} меньшим, либо, если ее текущий номер нечетен, «передает» свои коэффициенты строке с истинным номером на 2^{p-1} большим. В последнем случае разность между N и номером активной строки, несущей информацию I-й строки, убывает именно на 2^{p-1}. Отсюда последовательность $I - i_1 2^1$, $I - (i_2 2^0 + i_2 2^1)$, $I - (i_1 2^0 + i_2 2^1 + i_3 2^2)$, ..., $i_P 2^{P-1}$ есть последовательность истинных номеров строк-посредников, через которые с поэтапной переработкой проходит первоначальная информация I-й строки, пока не достигнет последней N-й строки. Двоичная цифра в разложении (12) указывает характер активности строки-посредника с истинным номером $i = I - (i_1 2^0 + i_2 2^1 + \ldots + \ldots + i_p 2^{p-1})$. При $i_p = 0$ этот номер совпадает с предыдущим в последовательности номером, т. е. эта строка является «принимающей» на данном этапе. При $i_p = 1$ она «передающая».

В рассмотренных алгоритмах с операциями (9) и (10) единственная N-я строка получает информацию каждой из остальных. В алгоритме же четно-нечетного отдаления на каждом p-м этапе операция (11) производится над всеми строками без исключения, поэтому к концу P-го этапа каждая строка имеет первичную информацию обо всех остальных, быть может, в неявной форме. Полное распараллеливание указанных алгоритмов на r-мерной N-процессорной ВС означает, что в каждом процессоре содержится одна строка матрицы. При функционировании ВС процессоры выполняют операции (9), (10) или (11), действительно передают и принимают коэффициенты строк, первоначальные либо преобразованные предшествующими операциями.

Из изложенного вытекает следующий важный вывод. При параллельной реализации указанных алгоритмов (и вообще любых алгоритмов, существенно использующих схему сдваивания данных, например быстрое преобразование Фурье) на N-процессорной ВС, каково бы ни было взаимно однозначное отоб-

ражение множества истинных номеров строк на множество номеров процессоров, необходима передача (с возможной промежуточной переработкой) информации между самыми удаленными процессорами. Поскольку данные распространяются на наибольшее расстояние $\sim N^{1/r}$ за $P = \log_2 N$ этапов, то средняя длина доставки есть величина порядка $N^{1/r}/\log N$, так что соответствующие операции являются глобальными. Таким образом, алгоритмы, базирующиеся на коммуникационной схеме сдваивания, содержат неустранимую глобальную компоненту при их реализации на параллельной ВС и не могут быть выполнены быстрее чем за время $O(N^{1/r})$.

Пусть число P представимо в виде суммы $P = M + Q$, где $M = Q$ при P четном или $|Q - M| = 1$ при P нечетном. Рассмотрим реализацию вышеприведенных алгоритмов на двумерной ВС, представляющей собой решетку из 2^M строк и 2^Q столбцов. Для числа a будем обозначать $]a[$ наименьшее целое из всех больших либо равных a. Отображение истинных номеров i процессорных элементов в узлы (m, q) системной решетки зададим соотношениями

$$m =]i/2^Q[, \qquad q = i - (m-1)2^Q.$$

В каждой строке находится четное число 2^Q процессорных элементов, а поскольку первая строка начинается процессорным элементом с истинным номером $i = 1$, то и каждая строка вообще начинается нечетным i. Согласно (9), операции четно-нечетного исключения предполагают получение процессором с четным текущим номером k информации от ближайшего нечетного $(k-1)$-го. На протяжении первых Q этапов алгоритма все истинные номера $m2^Q$ остаются четными текущими, т. е. процессоры крайнего правого столбца фиксируют результат обработки в своих строках. Начиная с $(Q+1)$-го этапа по $(Q+M)$-й этап вычисляющими являются только процессоры крайнего правого столбца. Таким образом, алгоритм четно-нечетного исключения на прямом ходе требует передач данных сначала только по строкам слева направо и затем только по крайнему правому столбцу сверху вниз. На обратном ходе схема передач повторяется в обратном порядке. Такая схема естественно реализуется на ВС с локальными взаимодействиями, где непосредственный обмен данными производится только между соседними процессорами. Среднее расстояние передачи по строке здесь есть $2^Q/Q$, по столбцу — $2^M/M$, а на одном этапе каждый действующий процессор принимает и обрабатывает по три числа (два элемента матрицы и свободный член). Отсюда временная сложность задачи четно-нечетного исключения, которая решается на локальной ВС, есть

$$T_{\text{ил}} = [(2^Q/Q + g_1)Q\tau + (2^M/M + g_1)M\tau\,] +$$

$$+ [(2^Q/Q + g_2)Q\tau + (2^M/M + g_2)M\tau\,],$$

где арифметические операции прямого и обратного хода различны ($g_1 \neq g_2$).

Для глобальной ВС с временем межпроцессорного обмена, равным Δ, временная сложность реализации этого алгоритма имеет вид

$$T_{\text{иг}} = [(\Delta + g_1\tau)Q + (\Delta + g_1\tau)M\,] + [(\Delta + g_2\tau)Q + (\Delta + g_2\tau)M\,].$$

Асимптотически $T_{\text{ил}} \approx \tau\sqrt{N}$, $T_{\text{иг}} \approx \tau\log N$ (формально здесь следует учитывать рост длины слов как $\log N$, но для практически реальных N в этом нет необходимости). Время Δ, необходимое для распространения сигнала по линейному размеру машины, пропорционально корню квадратному из числа процессоров с коэффициентом, равным задержке δ на линейном размере процессора, т. е. $\Delta = \delta\sqrt{N}$. При $N < (\tau/\delta)^2$ имеем $\Delta = \tau$ и $T_{\text{ил}}/T_{\text{иг}} = \sqrt{N}/\log N$, следовательно, глобальный вариант эффективнее локального в $(\sqrt{N}/\log N)$ раз. При $N > (\tau/\delta)^2$ отношение $T_{\text{ил}}/T_{\text{иг}} = (\tau/\delta)/\log N$, т. е. лишь при больших N локальные вычисления начинают доминировать над глобальными.

Иначе обстоит дело в случае трехдиагональных матриц. Для выполнения операции четно-нечетной редукции, согласно (9), четному процессору нужно получить от двух нечетных ближайших (в текущей нумерации) их данные. При реализации на локальной ВС это потребует встречных потоков по строкам, столбцам и между ними. Поэтому в отличие от локального варианта четно-нечетного исключения коммуникационные взаимодействия на каждом этапе здесь не поддаются конвейеризации. Изложенное тем более справедливо по отношению к методу четно-нечетного отдаления, где количество активных процессоров вовсе не убывает от этапа к этапу. Следовательно, реализация этих алгоритмов на локальной ВС заведомо неэффективна. На глобальной же ВС время работы четно-нечетной редукции есть $T_{\text{РГ}} = 2(\Delta + g\tau)P$, а для алгоритма четно-нечетного отдаления время вычисления $T_{\text{ОГ}} = P(\Delta + g\tau)$, что вдвое меньше за счет отсутствия обратного хода. Асимптотическая временная сложность в обоих случаях $\sim\delta\sqrt{N}\log N$.

Решетка с шинной коммуникационной сетью, где межпроцессорные обмены осуществляются по строчным и столбцевым шинам, не позволяет вести передачу одновременно многих слов по строке (столбцу). При реализации метода четно-нечетного исключения на такой решетке для межпроцессорных обменов по три слова в строках требуется на первом ходе $3 \cdot 2^Q$ глобальных тактов для обмена по строкам и $3 \cdot 2^M$ тактов для обмена по столбцам. Следовательно, время обмена есть

$$T_{\text{об}} = 6\Delta(2^Q + 2^M).$$

Собственно арифметические вычисления ведутся каждым процессором с темпом $1/\tau$ и требуют на прямом ходе времени $T_{\text{в1}} = g_1\tau(2^Q + 2^M)$, а на обратном — $T_{\text{в2}} = g_2\tau(2^Q + 2^M)$. Общее время решения задачи оценивается сверху величиной

$$T_{\text{ИШ}} = T_{\text{об}} + T_{\text{в1}} + T_{\text{в2}} = [6\Delta + (g_1 + g_2)\tau](2^Q + 2^M) \approx 4(3\Delta + g\tau)\sqrt{N}.$$

При постановке на шинную решетку трехдиагональных методов каждая шина используется разнонаправленно при удвоенном количестве передач по 4 слова. Кроме того, на первых Q этапах появляется необходимость обмена между соседними строками у процессоров некоторых пар столбцов. С одновременным использованием всех строчных и всех столбцевых шин решетки такой обмен по одному слову со сдвигом на одну строку между процессорами пары столбцов осуществляется за три глобальных такта. Опуская детали организации взаимодействий и подсчет операций, приведем оценки временной сложности шинной реализации четно-нечетной редукции $T_{\text{РШ}}$ и четно-нечетного отдаления $T_{\text{ОШ}}$:

$$T_{\text{РШ}} \approx (2\Delta + \tau)\sqrt{N} + \Delta\log N, \qquad T_{\text{ОШ}} \approx (3\Delta + \tau)\sqrt{N}.$$

С позиций развиваемой темы интересно соотнести полученные результаты с возможностями такой специализированной структуры, как «гиперкуб». Пусть $N = 2^P$ процессоров с P портами ввода-вывода каждый помещены (топологически) в вершинах P-мерного куба, т. е. объединены в глобальную систему непосредственными связями, соответствующими $P2^{P-1}$ ребрам P-мерного куба. В рамках реально действующей планарной технологии размещение всех процессоров и соединений требует площади, пропорциональной N^2. Время обмена между двумя процессорами в такой системе будет пропорционально N, поэтому реализации рассмотренных алгоритмов, действующих над ленточными матрицами, имеют здесь асимптотическую временную сложность $T_{\text{ГК}} \sim N\log N$. Как видим, гиперкубическая вычислительная структура, ориентированная на вычислительные схемы типа двоичного дерева, в асимптотике проигрывает по эффективности обычной процессорной решетке.

Более перспективна 3-мерная оптико-электронная реализация шинного P-гиперкуба (каждому из $P2^{P-1}$ ребер соответствует процессор, а каждой из 2^P вершин — информационная связь — шина) [1]. Оптическая ВС такой структуры содержит процессорную решетку из $P2^{P-1}$ процессоров с двумя портами ввода-вывода каждый и голографическую решетку из $P2^P$ подголограмм отражательного типа (по две подголограммы на процессор). Каждая из 2^P оптических шин содержит P распределенных по всей решетке подголограмм. С помощью одной такой оптической шины осуществляется размножение (трансляция) световых сигналов от процессора-передатчика к $(P-1)$ процессорам-приемникам. Настройка ВС на матричную или гиперкубическую структуру глобальных межпроцессорных связей осуществляется сменой голограмм.

Обработка ленточных матриц в потоковых вычислениях. До сих пор мы сравнивали между собой однократные реализации алгоритмов, решающих системы линейных алгебраических уравнений с ленточными матрицами, а теперь рассмотрим потоковые вычисления, когда один алгоритм многократно применяется к длинной серии матриц. Оказывается, что в таких условиях соотношение алгоритмов по реализуемой эффективности радикально меняется.

Рассмотрим потоковую реализацию вычислений по методу четно-нечетного отдаления на многоярусной ВС, состоящей из $P = \log N$ процессорных столбцов по N процессоров в каждом, т. е. из $P2^P$ процессоров в целом. Процессоры каждого j-го столбца пронумерованы в естественном порядке номерами i ($j = \overline{1, P}$, $i = \overline{1, N}$) и выполняют операции соответствующего j-го этапа алгоритма. Процессоры первого столбца соединены непосредственными «вертикальными» связями в «линейку». Между одноименными (с одинаковыми i) процессорами j-го и $(j+1)$-го столбцов установлены «горизонтальные» связи в виде цепочек регистров (когда задержкой сигнала в проводнике можно пренебречь, регистры необязательны). Кроме того, между j-м и $(j+1)$-м столбцами установлены аналогичные связи, соединяющие процессоры, разность номеров i' и i'' которых удовлетворяет соотношению $|i' - i''| = 2^j$. Эти «косые» связи соответствуют дугам дерева схемы сдваивания. Каждая связь (как «горизонтальная», так и «косая») между j-м и $(j+1)$-м процессорными столбцами содержит $(2^j - 1)$ регистров, так что всех регистров между этими столбцами не больше чем $3(2^j - 1)N$. Общее количество регистров в такой ВС оценивается величиной

$$3N[(2^1 - 1) + (2^2 - 1) + \ldots + (2^{P-1} - 1)] = 3N(2^P - P - 1) \approx 3N^2.$$

Работа ВС при потоковой реализации метода четно-нечетного отдаления происходит следующим образом. На входы процессоров 1-го столбца поступают строки (одна строка — это 3 элемента матрицы и свободный член). Обменявшись информацией с соседями по столбцу, каждый процессор выполняет над своей строкой операцию (11) первого этапа алгоритма. Результирующую строку он посылает ко 2-му столбцу по всем исходящим от него связующим цепочкам, после чего принимает строку следующей в потоке матрицы и повторяет над ней свои действия. Процессоры других столбцов принимают преобразованные строки из своих входящих цепочек, выполняют операцию (11) соответствующего этапа алгоритма и посылают результаты далее по цепочкам. Процессоры последнего столбца выводят результат решения уравнений во внешнюю среду. Поскольку число g арифметических действий в операции (11) невелико, то решения выдаются из ВС в достаточно быстром темпе $\sim 1/g\tau$. Время прохождения сигнала от входа до выхода ВС здесь пропорционально числу регистров N во всех горизонтальных связях между i-ми процессорами. В данной реализации это время, зависящее от линейного размера ВС, сказывается лишь на величине начальной задержки получения первого результата.

Для алгоритма четно-нечетной редукции подобная потоковая реализация неэффективна, так как из-за наличия в нем обратного хода темп выдачи решений будет определяться не собственным временем процессора, а линей-

ным размером ВС. Этот недостаток в полной мере присущ вычислительной системе «гиперкуб», на которой даже алгоритм четно-нечетного отдаления оказывается неконвейеризуемым и не может быть эффективно реализован в режиме потоковых вычислений.

В потоковых вычислениях у алгоритмов решения «ленточных» систем уравнений по схеме двоичного дерева имеется серьезная альтернатива — классический алгоритм прогонки, позволяющий решать задачу с максимальной эффективностью. Операция, производимая над строками системы на прямом ходе прогонки, имеет вид

$$S_k' = S_k - (a_k/b_{k-1})S_{k-1}', \tag{13}$$

что по форме совпадает с (9), только предыдущая $(k-1)$-я строка участвует уже преобразованной. При реализации прогонки на N-процессорной «линейке» с локальными межпроцессорными связями, в каждом k-м процессоре которой размещена одна k-я строка решаемой системы уравнений, все арифметические операции в операторе (13) зависимые и локальные. То же самое имеет место и для обратного хода, основной оператор которого еще проще. Каждый процессор выполняет по одному оператору прямого и обратного хода, и вся задача решается за время $\sim\tau N$. За счет конвейеризации доставок все эти операции можно выполнить за то же время $\sim\tau N$ на вычислительной машине, имеющей лишь один процессор и магазинную память, куда последовательно записываются на прямом ходе строки, преобразованные оператором (13). На обратном ходе ввод данных прерывается, процессор считывает из памяти строку за строкой, вычисляет и выводит во внешнюю среду компоненты решения. По освобождении магазина вывод прерывается и начинается ввод данных следующей подлежащей решению системы уравнений, т. е. осуществляется двухтактная обработка потока данных.

При одновременной работе N таких машин за время $\sim\tau N$ получаются решения N систем уравнений. Здесь работают N процессоров с памятью, занимающей площадь $O(N^2)$, тогда как на многоярусной ВС, реализующей алгоритм четно-нечетного отдаления, та же производительность достигается на $N\log N$ процессорах и площади ВС $O(N^2\log N)$.

Заключение. При параллельной реализации наиболее важных алгоритмов решения задач математической физики на N-процессорной ВС решетчатой структуры достижимо максимальное N-кратное ускорение от распараллеливания. Сравнительный анализ локальной и глобальной параллельных реализаций этих алгоритмов показывает, что локальная реализация, как правило, эффективнее глобальной. Только при решении задач малого размера, когда можно пренебречь затратами времени на доставку данных к процессорам, глобальная реализация не уступает локальной. Некоторые из алгоритмов, требующие при реализации глобальных связей, эффективно реализуются потоковыми вычислительными схемами. Лишь алгоритмы, реализация которых сопряжена с неизбежными сменами направлений глобальных пересылок, как это имеет место, например, в методе четно-нечетной редукции, являются существенно глобальными и не могут быть эффективно реализованы на локальной ВС. В режиме потоковых вычислений соотношение алгоритмов по реализуемой эффективности вообще радикально меняется. Так, достаточно эффективный алгоритм решения «ленточных» систем по схеме двоичного дерева проигрывает классическому алгоритму прогонки при многократном его применении. Некоторые из глобальных алгоритмов (та же четно-нечетная редукция) позволяют достичь ускорения $\sim\sqrt{N}$ на простейшей матричной ВС, которая, таким образом, превосходит ориентированную на схемы сдваивания вычислительную структуру «гиперкуб». Решетчатая ВС, дополненная глобальными шинами по строкам и столбцам, обладает высокой эффективностью при реализации всех локальных и ряда глобальных алгоритмов решения задач математической физики.

СПИСОК ЛИТЕРАТУРЫ

1. **Мишин А. И.** Оптоэлектронный параллельный компьютер и его потенциальные возможности // Автометрия. 1994. № 4.
2. **Софронов И. Д.** Оценка параметров вычислительной машины, предназначенной для решения задач механики сплошной среды // Численные методы механики сплошной среды. 1975. 6, № 3.
3. **Яненко Н. Н.** Метод дробных шагов решения многомерных задач математической физики. Новосибирск: Наука, 1967.
4. **Ortega J. M., Voigt R. G.** Solution of partial differencial equations on vector and parallel computers // SIAM Rev. 1985. 27, N 2.
5. **Яненко Н. Н., Коновалов А. Н., Бугров А. Н., Шустов Г. В.** Об организации параллельных вычислений и распараллеливании прогонки // Численные методы механики сплошной среды. 1978. 9, № 7.
6. **Кершоу Д.** Решение одной трехдиагональной системы линейных уравнений и векторизация // Параллельные вычисления /Под ред. Г. Родрига, Ю. Г. Дагаева. М.: Наука, 1986.

Поступила в редакцию 10 июня 1994 г.