К ВЫБОРУ СПОСОБА ДЕКОМПОЗИЦИИ ПРИ ЧИСЛЕННОМ РЕШЕНИИ СИСТЕМ СВЯЗАННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ НА МНОГОПРОЦЕССОРНОЙ ТЕХНИКЕ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ

Е.А. Данилкин, А.В. Старченко

Томский государственный университет

Введение

При использовании параллельных компьютеров с локальной памятью первым по значимости является вопрос о возможности проведения эффективных параллельных вычислений для конкретной решаемой задачи. Речь идет о пересмотре взглядов на стандартные алгоритмы, поскольку некоторые из них становятся неприменимыми при увеличении числа процессоров, другие, наоборот, приобретают большое значение. Определяющим фактором для оценки перспективности алгоритма являются затраты на межпроцессорные обмены, которые необходимы для его реализации. Если алгоритм требует большого числа пересылок, то можно не получить ускорения и тем самым не будет достигнута цель применения параллельных компьютеров – уменьшение времени вычислений. Проблему поможет решить удачное распределение данных по процессорам, при котором затраты на пересылки будут минимальны.

Цель данной работы заключалась в исследовании возможных подходов решения систем связанных линейных дифференциальных уравнений в частных производных методом конечного объёма на многопроцессорной системе с распределённой памятью и в выборе оптимального способа декомпозиции с точки зрения минимизации межпроцессорных обменов.

Математическая постановка задачи

Требуется решить систему двух связанных линейных дифференциальных уравнений в единичном квадрате ($(x, y) \in (0,1) \times (0,1)$, G – граница) с граничными условиями третьего рода

$$\begin{cases} \dfrac{\partial}{\partial x}\left(D_1 \dfrac{\partial u}{\partial x} \right) + \dfrac{\partial}{\partial y}\left(D_1 \dfrac{\partial u}{\partial y} \right) + E_1 \dfrac{\partial u}{\partial x} + B_1 v = F_1; \qquad \alpha_1 \left. \dfrac{\partial u}{\partial n} \right|_G + \beta_1 u|_G = \gamma_1; \\ \dfrac{\partial}{\partial x}\left(D_2 \dfrac{\partial v}{\partial x} \right) + \dfrac{\partial}{\partial y}\left(D_2 \dfrac{\partial v}{\partial y} \right) + E_2 \dfrac{\partial v}{\partial x} + B_2 u = F_2; \qquad \alpha_2 \left. \dfrac{\partial v}{\partial n} \right|_G + \beta_2 v|_G = \gamma_2, \end{cases} \tag{1}$$

где D_i, E_i, B_i, F_i $(i = 1,2)$ – достаточно гладкие функции от x и y, а $\alpha_i, \beta_i, \gamma_i$ $(i = 1,2)$ – константы.

Численный метод решения

Аппроксимация дифференциальной задачи осуществлялась на основе метода конечного объема. 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 каждое уравнение системы по каждому конечному объему. При вычислении интегралов используются интерполяционные квадратурные формулы численного интегрирования для зависимых величин, входящих в (1). В результате такого приближенного интегрирования получается дискретный аналог системы дифференциальных уравнений:

$$\begin{cases} ap_{ij}u_{ij} = ae_{ij}u_{i+1\,j} + aw_{ij}u_{i-1\,j} + an_{ij}u_{ij+1} + as_{ij}u_{ij-1} + ad_{ij}v_{ij} + b_{ij}; & 0 \le i \le Nx; \\ ap'_{ij}v_{ij} = ae'_{ij}v_{i+1\,j} + aw'_{ij}v_{i-1\,j} + an'_{ij}v_{ij+1} + as'_{ij}v_{ij-1} + ad'_{ij}u_{ij} + b'_{ij}; & 0 \le j \le Ny. \end{cases} \quad (2)$$

Матрица полученной системы сеточных уравнений (2) имеет вид, показанный на рис. 1. Здесь U,V – компоненты вектор-столбца неизвестных; B – столбец свободных членов.

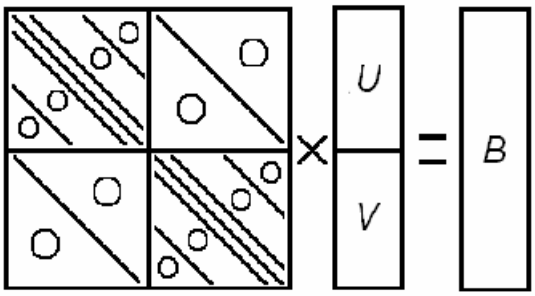


Рис. 1. Представление системы (2) в блочном виде

$$U = \{u_{i\,j}\},\ V = \{v_{i\,j}\},\ B = \begin{pmatrix} \{b_{i\,j}\} \\ \{b'_{i\,j}\} \end{pmatrix}.$$

Матрица системы (обозначим ее через A) имеет большое количество нулевых элементов, а ненулевые выстроены в диагонали, как показано на рис. 1. Матрица A состоит из четырех больших блоков, два из которых пятидиагональные матрицы и два содержат лишь по ненулевой главной диагонали.

Метод решения систем линейных алгебраических уравнений

В качестве метода решения системы сеточных уравнений рассматривается метод BiCGStab [3] с предобусловливанием по Булееву [1, 2]. В настоящее время этот метод является одним из наиболее быстродействующих по времени и количеству итераций. Он относится к семейству методов вариационно-итерационного типа и применим для решения знакоопределенных СЛАУ с несимметричной в общем случае матрицей, для которой неизвестны спектральные свойства.

Практически все современные итерационные алгоритмы решения задач вида $Ax = b$ относятся к методам подпространств Крылова, основанным на минимизации следующего функционала: $F(x) = (Ax, x) - 2(b, x)$. Решение связано с построением системы сопряженных векторов и каждое следующие приближение ищется в направлении нового полученного вектора, ис-

ходя из условия минимума функционала в этом направлении. Ниже представлен алгоритм метода BiCGStab [3].

BiCGStab алгоритм с предобусловливанием:

x_0 – начальное приближение; $r_0 = b - Ax_0$;

$\tilde{r}_0$ – произвольный вектор, для которого $(\tilde{r}_0, r_0) \neq 0$, например, $\tilde{r}_0 = r_0$;

$\rho_0 = \alpha = \omega_0 = 1;\ v_0 = p_0 = 0;$

for $i = 1,\ 2,\ 3,\ \ldots$;

$\rho_i = (\tilde{r}_0, r_{i-1});$

$\beta_i = (\rho_i / \rho_{i-1})(\alpha / \omega_{i-1});$

$p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1} v_{i-1});$

находим y из $Ky = p_i$ $(K \approx A)$;

$v_i = Ay;$

$\alpha = \rho_i / (\tilde{r}_0, v_i);$

$s = r_{i-1} - \alpha v_i;$

находим z из $Kz = s;$

$t = Az;$

$\omega_i = (t, s)/(t, t);$

$x_i = x_{i-1} + \alpha p_i + \omega_i s;$

если x_i достаточно точное, то остановка;

$r_i = s - \omega_i t;$

end;

Как указывается в [3], метод BiCGStab можно считать прямым методом, поскольку при точной арифметике он сойдётся к точному решению за конечное число итераций. Это следует из свойства сопряженности векторов построенной системы.

Предобусловливание

Идея предобусловливания заключается в преобразовании исходной задачи (решение системы линейных уравнений) таким образом, чтобы матрица предобусловленной системы имела лучшие спектральные свойства и полученная система решалась более легко. Под этим понимается, что алгоритм требует меньше вычислительных ресурсов за меньшее число итераций.

Для ускорения процесса сходимости и уменьшения числа итераций в данной работе используется процедура предобусловливания по методу неполной факторизации Булеева, формулы которой получаются из формул явного метода Булеева исключением итерационных членов [1,2].

Алгоритм предобусловливания (решение систем вида $Ky = p_i$ или $Kz = s$, $K \approx A$) состоит из двух этапов: прямого и обратного хода. На первом шаге вычисляются прогоночные коэффициенты, формулы для определения которых имеют рекуррентный вид (где Θ – параметр компенсации $0 \le \Theta < 1$):

$$\begin{cases} P_{ij} = \dfrac{ae_{ij}}{g_{ij}},\ Q_{ij} = \dfrac{an_{ij}}{g_{ij}},\ T_{ij} = \dfrac{(b_{ij} + aw_{ij}T_{i-1\,j} + as_{ij}T_{ij-1})}{g_{ij}}; & i = 1,2,...,Nx-1; \\ g_{ij} = ap_{ij} - aw_{ij}(P_{i-1\,j} + \Theta Q_{i-1\,j}) - as_{ij}(Q_{ij-1} + \Theta P_{ij-1}); & j = 1,2,...,Ny-1. \end{cases} \quad (3)$$

А затем решение $\{y_{i\,j}\}$ находится по формуле

$$y_{ij} = P_{ij}y_{i+1\,j} + Q_{ij}y_{ij+1} + T_{ij};\quad i = Nx-1,...,0;\ \ j = Ny-1,...,0. \quad (4)$$

Отметим, что использующееся в настоящее время предобусловливание по методу Холесского или LU-факторизации для задач вида (1) менее эффективно, чем предобусловливание по Булееву [2].

Параллельная реализация

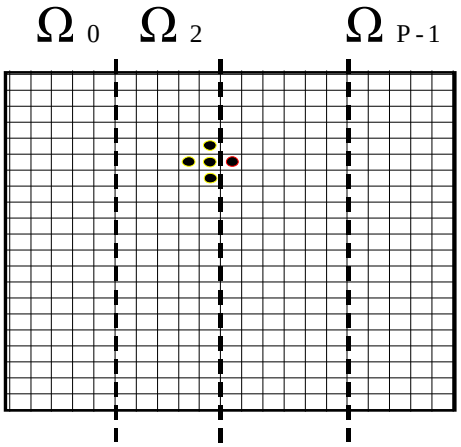


Рис. 2. Декомпозиция расчетной области

Распараллеливание метода BiCG-Stab для случая решения одного дифференциального уравнения осуществляется следующим образом. Используется одномерная декомпозиция сеточной области по индексу j, то есть каждому процессорному элементу выделяется определенное количество узлов сетки Ω, для которых он будет производить вычисления.

При аппроксимации дифференциальной задачи (1) использовался шаблон «крест», поэтому для вычисления очередного приближения в приграничных узлах подобласти Ω_k ($0 \le k \le P-1$, где P – число процессоров) необходимо пересылать значения с соседнего процессора.

Внутренний цикл алгоритма BiCGStab содержит два матрично-векторных произведения, четыре скалярных произведения, шесть операций saxpy и небольшое количество скалярных операций. Все эти алгебраические операции легко распараллеливаются [4].

Структура матрицы A (для случая одного уравнения) позволяет организовать вычисления так, что обмены требуются лишь для пересылки приграничных значений сеточной функции и сбора и рассылки частичных сумм скалярных произведений.

При предобусловливании системы линейных уравнений допустимо применять асинхронный вариант метода Булеева, когда при вычислении значений прогоночных коэффициентов в (3) и нахождении решения в (4) в каждой подобласти декомпозиции на ее границе используются значения сеточной функции $\{y_{ij}\}$ с предыдущей итерации.

Для случая решения системы сеточных уравнений с матрицей, представленной на рис. 1, ситуация меняется, и неудачное распределение данных по процессорным элементам может привести к значительному возрастанию пересылок, что в свою очередь сведёт на нет ускорение параллельной программы.

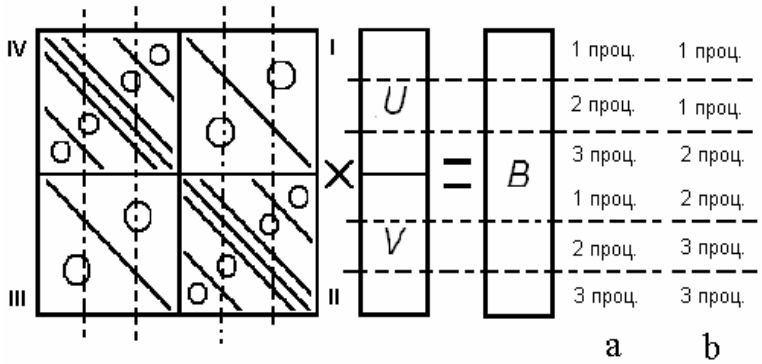


Рис. 3. Два варианта декомпозиции данных по процессорам (a–декомпозиция, b–декомпозиция)

На рис. 3 столбец a представляет оптимальный вариант декомпозиции данных, позволяющий избежать использования дополнительных пересылок. Дело в том, что при численном решении системы дифференциальных уравнений (1) связи накладываются не только на значения сеточной функции в точках шаблона u_{ij} и $u_{i\pm1\,j\pm1}$, но и на значения неизвестных, принадлежащих одной подобласти Ω_k (u_{ij} и v_{ij}).

Идея предложенного способа декомпозиции заключается в следующем. По процессорным элементам последовательно распределяются не неизвестные u_{ij}, v_{ij}, а каждому процессорному элементу выделяется целая сеточная подобласть Ω_k со всеми неизвестными, принадлежащими ей. Таким образом, все связанные значения сеточной функции распределены одному процессорному элементу.

При другом способе декомпозиции (рис. 3, столбец b) потребуется больше времени на межпроцессорную пересылку данных, как, например,

при размещении на одном процессорном элементе только значений $\{u_{ij}\}$ или только $\{v_{ij}\}$.

Были получены теоретические оценки ускорения расчетного метода для различных способов декомпозиции данных.

$$S_p^{(1)} = \frac{P}{1+\frac{P\cdot 24}{61\cdot\sqrt{N}}\cdot\chi}; \qquad S_p^{(2)} = \frac{P}{1+\frac{P\cdot 24}{61\cdot\sqrt{N}}\cdot\chi+\frac{\chi}{61}}.$$

Здесь $S_p^{(1)}$ – ускорение алгоритма при предложенном способе распределения данных по процессорным элементам (рис. 3, столбец a); $S_p^{(2)}$ – теоретическое ускорение для случая, когда возникает необходимость обменов большими частями решения. Например, при декомпозиции, представленной на рис. 3 (столбец b), первому процессору для проведения расчетов потребуются значения неизвестных $\{v_{ij}\}$ со 2-го и 3-го процессоров. Здесь P – число процессоров; N – размерность задачи; χ – отношение времени, затрачиваемого на пересылку одного числа, ко времени одной арифметической операции с плавающей точкой.

Для подтверждения полученных теоретических оценок была выбрана следующая тестовая задача:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial u}{\partial x}+v=1; & u\big|_G=2; \\ \frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial v}{\partial x}+u=2; & v\big|_G=1. \end{cases} \qquad (5)$$

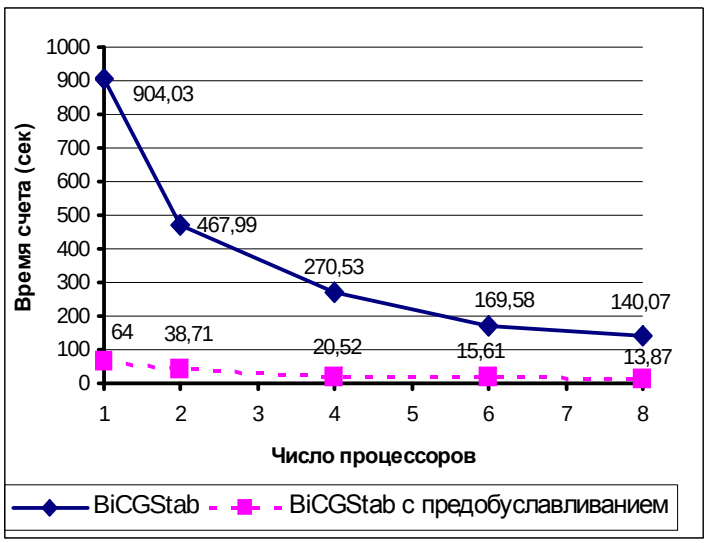


Рис. 4. Зависимость времени счета от числа процессоров

В качестве начального использовалось нулевое приближение. Приближенное решение соответствующих сеточных систем считалось достигнутым, если норма невязки и норма ошибки становились меньше $\varepsilon = 10^{-5}$.

Были проведены расчеты на кластере ИОА СО РАН для сетки 480x480 (http://www.iao.ru/ru/resources/equip/cluster/). На рис. 4 представлен график зависимости времени вычисления от числа используемых процессорных элементов. Результаты численного эксперимента показали, что использование предобусловливателя более чем на порядок сокращает временные затраты на решение системы линейных алгебраических уравнений.

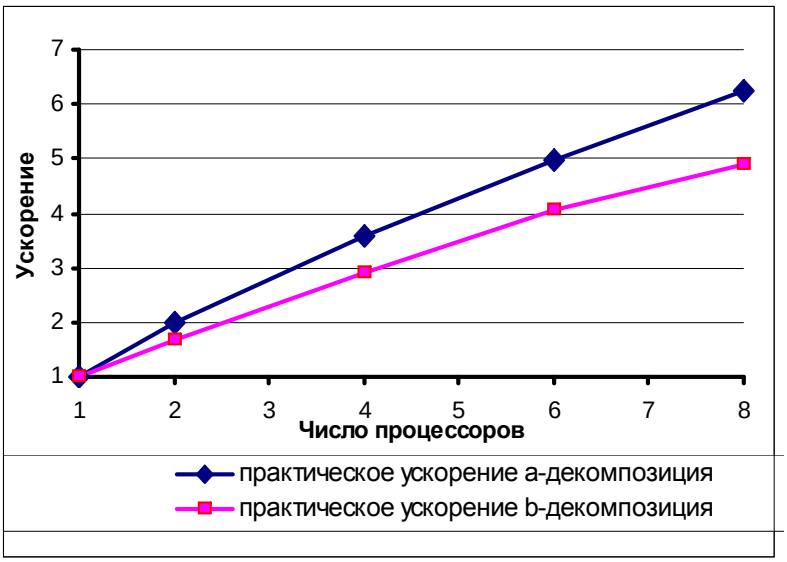


Рис. 5. Ускорение метода для рассматриваемых способов декомпозиции данных

На рис. 5 представлено ускорение для разных способов распределения данных по процессорным элементам. Видно, что при использовании предложенного способа декомпозиции ускорение на единицу больше, чем при альтернативном способе распределения данных, а это даст существенный выигрыш во времени при решении больших задач.

Литература

1. Ильин В.П.. Методы неполной факторизации для решения алгебраических систем. – М.: Наука, 1995. – 287 с.
2. Старченко А.В. // Вестник Томского государственного университета. − 2003. − №10. − С. 70−80.
3. Van der Vorst H.A. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetrical linear systems // SIAM Journal. − 1992. − Vol. 13, № 2. − P. 631−644.
4. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. – М.: Мир, 1991. – 364 с.