УДК - 004.75

РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ ПУАССОНА ПАРАЛЛЕЛЬНЫМ МНОГОСЕТОЧНЫМ МЕТОДОМ НА ВЫЧИСЛИТЕЛЬНОМ КЛАСТЕРЕ WCCS

Морозов Р.Н., Ладыженский Ю.В.
Донецкий национальный технический университет

В наше время высокого технологического развития компьютерные технологии позволяют решать все более сложные задачи. На данный момент существует большое разнообразие трудно решаемых задач. Для решения таких задач требуется большое количество вычислительных компьютеров и много времени. Одной из актуальных трудно решаемых задач является задача решения дифференциальных уравнений в частных производных. Она включает в себя нахождение численного решения уравнения и оценку полученного решения. Данная задача может быть эффективно решена с помощью вычислительного кластера.

Для ускорения решения таких уравнений необходимо использовать эффективные параллельные алгоритмы. Одним из таких является многосеточный метод[1] решения дифференциальных уравнений в частных производных. Метод основан на использовании последовательности уменьшающихся сеток и операторов перехода от одной сетки к другой.

В данной статье рассматривается решение дифференциального двухмерного уравнения Пуассона с граничными условиями Дирихле параллельным многосеточным методом на вычислительном кластере Microsoft CCS.

Рассмотрим двухмерное уравнение Пуассона в квадратной области:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y) \tag{1}$$

с нулевыми граничными условиями Дирихле на границе области.

Опишем алгоритм одной итерации параллельного многосеточного метода.

Пусть:

k – сеточный уровень, на котором выполняется решение;

$P^{(k)}$ – текущая задача, характеризующаяся правой частью $b^{(k)}$ уравнения, размером сетки уровня k и приближенным решением на этой сетке $u^{(k)}$;

$b^{(k)}$ – правая часть задачи $P^{(k)}$;

$u^{(k)}$ – приближенное решение задачи $P^{(k)}$;

$r^{(k)}$ – невязка задачи $P^{(k)}$.

$$\begin{aligned} r^{(k)} &= b(i,j) - (4 * u(i,j) - u(i-1,j) - u(i+1,j) - \\ &- u(i,j-1) - u(i,j+1)), i,j := \overline{1..n-1} \end{aligned} \tag{2}$$

$d^{(k)}$ – поправка приближенного решения, вычисленная по более грубой сетке;

n – размерность сетки покрывающей заданную область;

MGV() – программная функция, реализующая многосеточный цикл;

$S(u^{(k)}, b^{(k)})$ – оператор сглаживания, который для задачи $P^{(k)}$ с приближенным решением $u^{(k)}$, вычисляет уточненное приближение $u^{(k)}$. Уточнение решения достигается усреднением значения в каждой точке сетки по его ближайшим соседям и вычисляется по формуле,

$$\begin{aligned} S(u^{(k)}, b^{(k)}) &= (1-w) * u(i,j) + (w/4) * (u(i-1,j) + u(i+1,j) + \\ &+ u(i,j-1) + u(i,j+1) + b(i,j)), i,j := \overline{1..n} \end{aligned} \tag{3}$$

где w = 2/3;

$R(b^{(k)})$ – оператор сужения, который отображает невязку $r^{(k)}$ задачи $P^{(k)}$ в $r^{(k-1)}$ на более грубой сетке. Реализация оператора R сводится к взвешенному усреднению по ближайшим соседним узлам сетки и представлена формулой.

$$\begin{aligned}&R(r^{(k)}) = 0.25 * r(i,j) + 0.125 * (r(i-1,j) + r(i+1,j) + \\ &+ r(i,j-1) + r(i,j+1)) + 0.625(r(i-1,j-1) + r(i+1,j-1) \cdot \\ &+ r(i-,j+1) + r(i+1,j+1)), i,j := \overline{3..n-2}\, с\, шагом\, 2\end{aligned} \tag{4}$$

$In(u^{(k-1)})$ – оператор интерполяции, преобразует приближенное решение $u^{(k-1)}$ задачи $P^{(k-1)}$ в приближенное решение $u^{(k)}$ задачи $P^{(k)}$ на более мелкой сетке. Реализация операция сводится к взвешенному усреднению по ближайшим соседним узлам (k-1) сетки. В интерполяции участвуют два узла, если у узла мелкой сетки лишь 2 соседа на грубой сетке, и 4 узла в противном случае.

Алгоритм включает несколько последовательных итераций, результатом которых является уточненное значение приближенного решения $u^{(i)}$.

1) $u^{(k)} = S(b^{(k)}, u^{(k)})$,	Найти улучшенное приближение
2) $r^{(k)} = u^{(k)} - b^{(k)}$	Вычислить невязку
3) $d^{(k)} = In(MGV(4*R(r^{(k)}),0))$	Определить поправку по более грубой сетке
4) $u^{(k)} = u^{(k)} - d^{(k)}$	Уточнить решение на мелкой сетке
5) $u^{(k)} = S(b^{(k)}, u^{(k)})$	Снова уточнить приближенное решение

Теперь опишем способ распараллеливания данной задачи на вычислительном кластере.

Движемся по строкам заданной сетки. Вычисление каждой строки обновленной сетки осуществляется на соответствующем ей процессоре. Способ определения соответствия приведен на рисунке 1, на котором используются два параметра: N – размерность исследуемой сетки, m – количество процессоров, принимающих участие в вычислении.

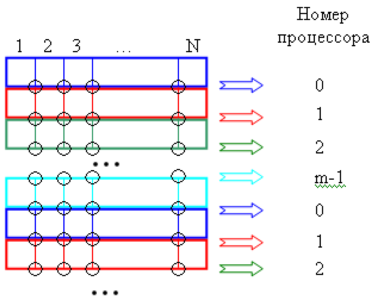

Рис. 1. Схема распределения узлов сетки по процессорам

Из рисунка видно, что сначала строки сетки последовательно по одной распределяются между всеми процессорами, участвующими в вычислении, а затем оставшиеся строки распределяются по процессорам, начиняя с самого первого. И так до тех пор, пока не будут вычислены все узлы обновленной сетки.

Исследования проводились с помощью эмуляции разного количества процессоров в среде Visual Studio 2005 с использованием Compute Cluster Pack. Эксперимент содержит 3 сеточных уровня: мелкая сетка – 9x9 узлов, первая грубая сетка – 5x5 узлов, вторая грубая сетка – 3x3 узлов. Было проэмулированно использование 1 и 2 процессоров.

Для проведения эксперимента необходимо инициализировать правую часть уравнения b(x,y) и задать начальное приближенное решение.

Формула расчета правой части для квадратной сетки размером nxn, где n – количество узлов сетки:

$$b(x,y) = \sin(\frac{(x-1)*3*3.14}{n})*\sin(\frac{(y-1)*3.14}{n}), x := \overline{1..n}, y := \overline{1..n} \quad (5)$$

Формула расчета начального приближенного решения для квадратной сетки размером nxn, где n – количество узлов, зависит от b(x,y):

$$\begin{aligned} npr(i,j) = 4*b(i,j) - b(i-1,j) - b(i+1,j) - \\ - b(i,j-1) - b(i,j+1), i := \overline{1..n}, j := \overline{1..n} \end{aligned} \quad (6)$$

На рисунке 2 приведено полученное решение двухмерного уравнения Пуассона (1) на самой мелкой сетке с уточненным с помощью параллельного многосеточного метода решением.

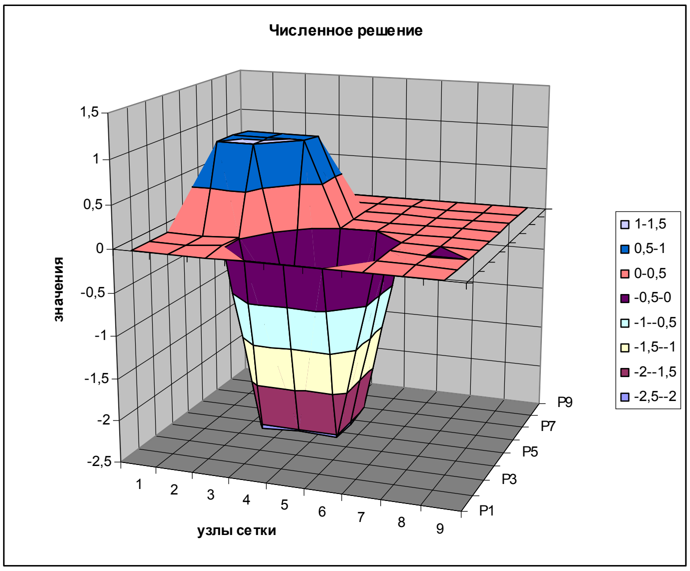

Рис. 2. Численное решение поставленной задачи

В ходе экспериментов было измерено время решения задачи с использованием разного количества процессоров. При использовании 1 процессора время решения задачи составило 1,98 секунды, а при использовании 2 процессоров время решения – 1,26 секунды. Таким образом, при использовании распараллеливания получено ускорение выполнения задачи в 1,57 раза.

Литература

[1] Деммель Джеймс, Икрамов Хаким Вычислительная линейная алгебра. Теория и практика. - Изд-во: Мир, 2001. – 430 с.

[2] Гергель В.П. Теория и практика параллельных вычислений. – СПб.: Бином, 2007. – 424с.: ил.