

РЕАЛИЗАЦИЯ РЕШЕНИЯ РАЗНОСТНОЙ СХЕМЫ РАСЩЕПЛЕНИЯ ДВУМЕРНОГО УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ В КЛЕТОЧНОЙ МАШИНЕ

Г.В. Заручевская

Северный арктический федеральный университет, Архангельск

Господствующим способом распараллеливания задач до сих пор является крупноблочное. При этом задача разбивается на большие подзадачи (блоки), предназначенные для параллельного решения на небольшом числе процессоров. С ростом числа процессоров блоки измельчаются и вычисления в подавляющем большинстве случаев будут идти медленнее: параллелизм вырождается. Избежать вырождения можно только при условии, что обмены происходят и одновременно, и локально, т.е. физическое расстояние между взаимодействующими процессорами мало и не зависит от размера задачи. Задача должна быть разбита на множество небольших однотипных подзадач, которые будут исполняться параллельно на отдельных вычислительных машинах (ВМ). Данные максимально распределены по системе, а программы в каждой ВМ используют минимально возможные наборы данных. В общем случае число обменов данными между ВМ имеет тот же порядок, что и число вычислительных операций. Таким образом, мелкозернистость, или массовое распараллеливание, означает, что в каждом вычислительном процессе в каждый момент времени содержится минимальное число команд и данных. Такой подход к распараллеливанию алгоритмов носит название мелкозернистого локально-параллельного программирования (МЛПП).

Существуют три обязательных условия, при которых производительность МЛПП не снижается:

1. Локальность взаимодействий, когда обмен данными происходит только в пределах ограниченного физического и структурного радиуса, независимо от размеров задачи и системы.
2. Параллелизм взаимодействий, когда все возможные в данный момент обмены совершаются параллельно и одновременно с процессом счета.
3. Количество глобальных операций не должно влиять на оценку временной сложности задачи.

Опишем клеточную машину, предназначенную для МЛПП-программирования.

Клеточная машина (К-машина) представляет собой *клеточное множество* – массив поименованных клеток, в каждой из которых находится универсальный вычислитель, известный в теории вычислений как РАМ-машина. Клетка может обмениваться информацией с соседями согласно шаблону соседства, задающему доступную для клетки окрестность. Клетка К-машины содержит память и коммутационное устройство для связи с соседями и управляющей машиной. *Шаблон соседства* для данной клетки – это список приращений её координат для получения координат *соседей*. Взаимодействовать должны только ближайшие соседи. *Условие локальности* означает, что радиус шаблона соседства – это константа, намного меньшая размера всего клеточного множества и не зависящая от этого размера. *Топология* К-машины задаётся шаблоном соседства. Одна из них – *тор*, которая получается, если мы склеим попарно и вертикальные, и горизонтальные границы прямоугольника. Тор позволяет масштабировать сетку другой, отличающейся от него размерности. Действительно, пусть сетка $a \times b$ вкладывается

в тор $C \times D$, $a > C$, $b > D$ (его обозначают $E_2\{C, D\}$). Каждой клетке тора (x, y) поставим в соответствие ячейку сетки $(x+ck, y+dn)$, $k, n \in N$. При таком распределении сеточных узлов в тороидальной структуре параллельный алгоритм соответствует МЛП-стилю.

Целью настоящей работы является вложение данных МЛПП-алгоритма решения сеточной задачи Коши для двумерного уравнения теплопроводности по схеме переменных направлений в тороидальную структуру синхронной клеточной машины с иллюстрацией межклеточных обменов.

Решение сеточной задачи для уравнения теплопроводности по схеме переменных направлений сводится к системе линейных уравнений вида:

$$\begin{cases} \tilde{u}_{m,n+1} - 2(1 + \frac{h^2}{\tau})\tilde{u}_{m,n} + \tilde{u}_{m,n-1} = -u_{m+1,n}^p + 2(1 - \frac{h^2}{\tau})u_{m,n}^p - u_{m-1,n}^p \\ u_{m+1,n}^{p+1} - 2(1 + \frac{h^2}{\tau})u_{m,n}^{p+1} + u_{m-1,n}^{p+1} = -\tilde{u}_{m,n+1} + 2(1 - \frac{h^2}{\tau})\tilde{u}_{m,n} - \tilde{u}_{m,n-1} - h^2 f_{mn}^{p+1} \end{cases} \quad (1)$$

Рассмотрим параллельный алгоритм решения задачи.

Этап 1. Хост-машина рассчитывает τ/h^2 , $2(1 + \tau/h^2)$ и $2(1 - \tau/h^2)$, рассылает эти значения в каждую клетку тороидальной структуры; исходные данные u_{mn}^0 и $h^2 \cdot f_{mn}^p$ рассылаются в клетки $(\text{mod}(m, C), \text{mod}(n, D))$. Также машина вычисляет \aleph_k и рассылает в клетки $(\text{mod}(k, C); \text{mod}(n, D))$ и $(\text{mod}(m, C); \text{mod}(k, D))$; $m, n, k=1..(N-1)$.

Этап 2. Вычислим правую часть первого уравнения (1) в подобласти $\omega^{1,1}$. В каждом горизонтальном кольце системе сдвигом вправо передаются значения $u_{m,n}$ соседним клеткам, затем аналогичным сдвигом осуществляется передача этих данных влево. Далее каждая клетка с номером m рассчитывает

$$-u_{m+1,n}^p + 2(1 - \frac{h^2}{\tau})u_{m,n}^p - u_{m-1,n}^p = F_{m,n}^p.$$

Подобным образом вычисляются значения $F_{m,n}^p$ в остальных площадках $\omega^{k,n}$. Заметим, что если рассчитываются значения в приграничных площадках $\omega^{1,s}$ или $\omega^{[(M-1)/C]+1,s}$, то клетки кольцевой структуры с номерами 1 и C друг с другом данными не обмениваются.

Этап 3. В каждом вертикальном кольце клеток, составляющих тор, рассчитываются \tilde{u}_{mn} , $m=\overline{1..C}$. Согласно методу прогонки рассчитываются: \wp_1 в клетках $(\text{mod}(m, C), 1)$; \wp_{N-1} в клетках $(\text{mod}(m, C), \text{mod}(N-1, D))$. Значение \wp_1 передается параллельным сдвигом вверх в каждой кольцевой структуре для расчета \wp_2 по формуле $\wp_k = (\wp_{k-1} - F_{m,k}) \aleph_k$, $k=2..N-2$ в клетках $(\text{mod}(k, C), \text{mod}(m, D))$. Подобно расчету \wp_{N-2} находятся остальные \wp_k , $k=2..N-2$. Затем вычисляются $\tilde{u}_{mN-1} = (\wp_{N-1} + \aleph_{N-1} \cdot \wp_{N-2}) / (1 - \aleph_{N-1} \cdot \aleph_{N-2})$ в клетках $(\text{mod}(m, C), \text{mod}(N-1, D))$ и сдвигом вниз пересылаются соседним клеткам для расчета \tilde{u}_{mN-2} по формуле $\tilde{u}_{mn} = \aleph_n \cdot \tilde{u}_{mN-1} + \wp_n$. Аналогично расчету \tilde{u}_{mN-2} находятся остальные \tilde{u}_{mn} , $n=(N-2)..2$.

Подобным образом находится следующая партия \tilde{u}_{mn} , $m=\overline{(C+1)..(2C)}$; затем при $m=\overline{(2C+1)..(3C)}$ и т.д., пока не будет рассчитана последняя партия при

$$m = \left\lceil \frac{M-1}{C} \right\rceil .. (M-1).$$

Этап 4. Рассчитаем правую часть второго уравнения (1) в подобласти $\omega^{1,1}$. В каждом кольце, образованном клетками P_{mn} с одинаковыми номерами m , сдвигом вверх передаются значения $u_{m,n}$ соседним клеткам, затем аналогичным сдвигом осуществляется передача этих данных в противоположном направлении. Далее каждая клетка P_{mn} выполняет арифметические операции по формуле

$$-\tilde{u}_{m,n+1} + 2\left(1 - \frac{h^2}{\tau}\right)\tilde{u}_{m,n} - \tilde{u}_{m,n-1} - h^2 f_{mm}^{p+1} = \tilde{F}_n^{p+1}.$$

Подобным образом вычисляются значения $F_{m,n}^p$ в остальных площадках $\omega^{k,n}$. При расчете значений на приграничных площадках $\omega^{k,1}$ или $\omega^{k, [(N-1)/D]+1}$ клетки на кольцевых структурах со вторыми номерами 1, D друг с другом данными не обмениваются.

Этап 5. В каждом горизонтальном кольце клеток, составляющих клеточную матрицу, согласно методу прогонки рассчитываются u_{mn} . Сначала в клетках $(1, \text{mod}(n, C))$ вычисляются $\wp_1 = (u_{0,n}^1 - F_{1,n}) \cdot \aleph_1$, а в клетках $(\text{mod}((M-1), C), \text{mod}(n, C)) - \wp_{M-1} = (u_{N,n}^1 - F_{M-1,n}) \cdot \aleph_{M-1}$. Значение \wp_1 передается параллельным сдвигом вправо в каждой кольцевой структуре для расчета \wp_2 по формуле $\wp_k = (\wp_{k-1} - F_{k,n}) \aleph_k$, $k=2..M-2$ в клетках $(\text{mod}(k, C), \text{mod}(n, D))$, аналогично рассчитываются остальные \wp_k . Затем вычисляются $u_{M-1,n} = (\wp_{M-1} + \aleph_{M-1} \cdot \wp_{M-2}) / (1 - \aleph_{M-1} \cdot \aleph_{M-2})$ в клетках $(\text{mod}(M-1, C), \text{mod}(n, D))$, значение сохраняется и передается параллельным сдвигом влево для определения $u_{M-2,n}$ по формуле $u_{k,n} = \aleph_k \cdot u_{k+1,n} + \wp_k$. Аналогично, используя ту же формулу, находятся остальные u_{mn} .

Вычислительный процесс заканчивается, когда выполнен расчет последнего слоя u^T . Цикл повторяется T раз.

В рассмотренном алгоритме только второй и четвертый этапы выполняются в мелкозернистом локально-параллельном стиле. Третий и пятый этапы не являются мелкозернистыми. Легко видеть, что при такой организации параллельных вычислений большинство клеток на этапах 3 и 5 простаивают. Этот недостаток можно исправить, вычисляя прогоночные коэффициенты \wp_k , а затем и значения \tilde{u}_{mn} (u_{mn}) «волной», т.е. каждая i -я вертикальная кольцевая структура на третьем этапе одновременно вычисляет значения $\tilde{u}_{i,n}$, $\tilde{u}_{i+C,n}$, $\tilde{u}_{i+2C,n}$, ..., $\tilde{u}_{i+C(D-1),n}$, затем $\tilde{u}_{i+CD,n}$, $\tilde{u}_{i+C(D+1),n}$, $\tilde{u}_{i+C(D+2),n}$, ..., $\tilde{u}_{i+C(2D-1),n}$ и т.д. Аналогично каждая j -я горизонтальная кольцевая структура на пятом этапе параллельно вычисляет значения u_{mj} , $u_{m,j+D}$, $u_{m,j+2D}$, ..., $u_{m,j+D(C-1)}$, затем $u_{m,j+DC}$, $u_{m,j+D(C+1)}$, $u_{m,j+D(C+2)}$, ..., $u_{m,j+D(2C-1)}$ и т.д. Такой «модернизированный» алгоритм на всех этапах (не связанных с глобальными операциями) отвечает требованиям локальности и мелкозернистости.

При $t_{\text{сдвиг}} \approx 0$ «модернизированный» вариант достигает максимального параллелизма, т.е. $E_{CD}^j \approx 1$.

В заключение отметим, что существуют другие алгоритмы распараллеливания методов решения разностных схем гиперболических уравнений, решаемых с помощью метода прогонки. Однако эти алгоритмы крупноблочные и к ним трудно применима технология мелкозернистого локально-параллельного программирования, и, как правило, они реализуются либо на кластерах, либо на планарных неразрезных процессорных матрицах. Предложенную идею можно использовать для распараллеливания других методов решения разностных схем гиперболических уравнений.