

ИССЛЕДОВАНИЕ МЕТОДОВ ОРГАНИЗАЦИИ ДАННЫХ В ЗАДАЧАХ РАЗБИЕНИЯ ГРАФОВ БОЛЬШИХ РАЗМЕРНОСТЕЙ

Краснокутская М.В., Костин М.В.
Донецкий национальный технический университет

Параллельные вычисления позволяют значительно повысить эффективность и скорость обработки информации при решении сложных задач. Решаемую задачу удобно представлять в виде графа потоков данных (ГПД), где программа представляется набором вычислительных узлов-подзадач, которые имеют фиксированное количество информационных входов и выходов. Каждая подзадача выполняется на отдельном процессоре. Узлы-подзадачи – вершины графа потоков данных, а информационные потоки между ними – ребра графа.

Одна из основных проблем, которая возникает в каждом параллельном вычислении, это распределение обработки данных между процессорами. Оптимальное распределение минимизирует время выполнения всех вычислений. Задача распределения обработки данных на процессоры сводится к задаче разбиения графа. Необходимо разбить граф потоков данных так, чтобы количество связей между подграфами было минимальным.

Так как разбиение графа является NP-трудной задачей, то все известные алгоритмы дают приближение к оптимальному решению[1]. Среди наиболее известных можно выделить следующие: алгоритм Kernighan-Lin / Fiduccia-Mattheyses (KL/FM), уровневое ячеечное разбиение, многоуровневые схемы, алгоритм спектральной бисекции.

При программной реализации любого из этих алгоритмов встает задача выбора типа данных для представления информации о графе. Существуют различные способы внутреннего представления графов в оперативной памяти ЭВМ, в том числе в виде списков (массивов) вершин и ребер, списков (массивов) смежности, матриц смежности, а также в виде комбинаций этих структур хранения. Выбор внутреннего представления оказывает решающее влияние на эффективность выполнения различных операций над графами [2].

Целью данной работы является исследование методов организации данных в задачах разбиения графов больших размерностей методом спектральной бисекции.

Алгоритм спектральной бисекции. По этому алгоритму в соответствии каждой вершине графа ставится переменная x , равная $+1$ или -1 , таким образом, что сумма всех x -ов равна 0 . Первое условие подразумевает разбиение на два различных набора. Второе требует, чтобы наборы были равного размера, учитывая четность исходного количества. Вектор x является вектором-индикатором, так как он показывает принадлежность каждой вершины к набору. Затем определяется функция от x , задающая число граней, пересекающихся между наборами. Теперь, когда имеется функция для минимизации, преобразуем ее к матричной форме. Полученная функция содержит матрицу Лапласа.

Так как разбиение графа - NP-трудная задача, необходимо ослабить ограничения дискретности на x и сформулировать новую непрерывную задачу. Эта непрерывная задача - только приближение к дискретной, и значения, определяющие ее решение, должны быть отображены назад к ± 1 согласно некоторой соответствующей схемой. Идеально, когда решение близко к ± 1 .

Используя собственные вектора и значения матрицы Лапласа, выразим X в терминах собственных векторов L : $x = \sum \alpha_i U_i$, где α_i – вещественные константы, такие, что $\sum (\alpha_i)^2 = n$. Заменой для x мы получаем нижнюю границу значений $f_{\min} = n \lambda_2 / 4$. Этому значению соответствует $x = \sqrt{n} U_2$. Полученный вектор x – решение непрерывной задачи. Остается решить задачу отображения вектора x к дискретному разделению. Для этого находится медиана значений x_i , и затем отображаются вершины больше значения медианы в один набор, меньше – в другой [3].

Задание графов с помощью матриц удобно для алгоритмов, использующие матричные вычисления (например, алгоритм спектральной бисекции). Однако, при обработке графа большой размерности ($n=1000, 10000$), матрицы занимают слишком много памяти. Следует учесть, что матрицы графов потоков данных довольно разрежены, т. е. матрицы содержат много нулей.

Для алгоритма спектральной бисекции и алгоритмов для нахождения собственных значений основной операцией является умножение матрицы на вектор. В данной работе рассмотрены два представления матрицы при программной реализации алгоритма. Первый способ – двумерный массив, второй – массив динамических массивов (первый элемент динамического массива – число ненулевых элементов в строке матрицы)

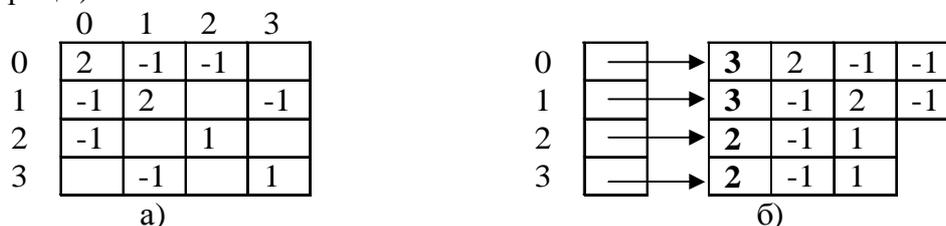


Рис.1.1. Способы представления матрицы:
а) двумерный массив; б) массив динамических массивов

Была разработана программная реализация этих двух способов, построены диаграммы зависимости времени умножения матрицы на вектор от размерности матрицы и степени ее разреженности.

Представление матрицы двумерным массивом проще в реализации. Объем занимаемой памяти и время умножения матрицы на вектор при такой реализации – постоянны. Такой способ оптимально подходит для неразреженных матриц с большим числом ненулевых элементов. Разреженные матрицы лучше представлять массивом динамических массивов, так как при такой реализации объем занимаемой памяти и время умножения матрицы на вектор зависит от размерности и степени разреженности графа.

Литература

[1] Bradford L. Chamberlain. Graph Partitioning Algorithms for Distributing Workloads of Parallel Computations. 1998
 [2] <http://www.caravan.ru/~alexch/AGraph>
 [3] Multidimensional Spectral Load Balancing. Bruce Hendrickson and Robert Leland Sandia, National Laboratories NM 87185 1993