УДК 681.3.06

С.И. Шматков, Е.Г. Толстолужская, Ю.А. Артюх

Харьковский национальный университет имени В.Н. Каразина, Харьков

АНАЛИЗ РАСПАРАЛЛЕЛИВАНИЯ АЛГОРИТМА ЗАДАЧИ ОПТИМАЛЬНОГО РАЗДЕЛЕНИЯ ГРАФА НА ПОДГРАФЫ

В статье рассмотрена проблема оптимального разделения графа на подграфы, представлены описания последовательного и параллельного алгоритмов, приведены показатели эффективности параллельного алгоритма. Последовательный алгоритм реализован с использованием языка программирования Си, параллельный – с использованием MPI.

Ключевые слова: *оптимальное разделение графа на подграфы, теория графов, матрица смежности, весовые коэффициенты.*

Введение

Актуальность статьи. Одной из часто возникающих проблем при проведении научных исследований, использующих параллельные вычисления в своей реализации, и непосредственно затрагивающей теорию графов является проблема оптимального разделения графов на подграфы [1, 3]. «В качестве примера можно привести задачи обработки данных, в которых области расчетов аппроксимируются двумерными или трехмерными вычислительными сетками. Получение результатов в таких задачах сводится, как правило, к выполнению тех или иных процедур обработки для каждого элемента (узла) сети. Эффективное решение таких задач на многопроцессорных системах с распределенной памятью предполагает разделение сети между процессорами таким образом, чтобы каждому из процессоров выделялось примерно равное число элементов сети, а межпроцессорные коммуникации, необходимые для выполнения информационного обмена между соседними элементами, были минимальными» [2]. Т.о. задачи с разделением сети между процессорами можно представить в виде разделения графа на подграфы. Такое решение проблемы является разумным, т.к. предоставляется возможность использования алгоритмов обработки и работы с графами, а так же получение более легкого решения проблемы хранения и обработки данных.

Постановка задачи исследования. Пусть дан взвешенный полносвязный неориентированный граф G=(V,E), каждой вершине v, принадлежащей V, и каждому ребру e, принадлежащему E, которого приписан вес. Задача оптимального разделения графа состоит в разбиении его вершин на непересекающиеся подмножества с максимально близкими суммарными весами вершин и минимальным суммарным весом ребер, проходящих между полученными подмножествами вершин. Равновесность подмножеств вершин может не соответствовать минимальности весов граничных ребер и наоборот. В большинстве случаев необходимым является выбор того или иного компромиссного решения.

Основная часть

Результаты исследования. Рассмотрим решение задачи оптимального разбиения графов на подграфы с помощью нижеприведенного примера.

Граф состоит из 6 вершин с весами на дугах, представлен на рис. 1. Представим граф в виде симметричной относительно главной диагонали матрицей смежности (рис. 2). Необходимо разбить данный граф на 3 подграфа.

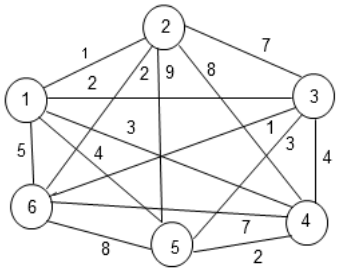

Рис. 1. Граф, с весами на дугах.

7	1	2	3	4	5
1	7	7	8	9	2
2	7	8	4	3	1
3	8	4	8	2	7
4	9	3	2	3	8
5	2	1	7	8	3

Рис. 2. Матрица смежности.

Создаем первоначальное разбиение, которое в последующем будем оптимизировать по весу с максимально близкими суммарными весами вершин и ребер. Первоначальное разбиение имеет следующий вид:

1 подграф: вершины № 1, 2;

2 подграф: вершины № 3, 4;

3 подграф: вершины № 5, 6.

На рис. 2 матрица смежности разделена на 3 части, которые и являются первоначальным разбиением. После получения первоначального разбиения

считаем веса подграфов (на рис. 2 в квадраты обведены веса, которые нас интересуют, т.к. матрица симметрична, нам достаточно использовать только одну строку матрицы в каждом подграфе).

1 подграф: 7+1=8;
2 подграф: 8+4=12;
3 подграф: 3+8=11.

Находим подграфы с максимальным и минимальным весом, т.е. это подграф 1 и 3.

Для оценки максимально близкого суммарного веса вершин и ребер высчитываем средний вес подграфа, т.е. (8+12+11)/3≈10;

Переходим к оптимизации.

ЭТАП 1: в подграфе с минимальным весом находим вершину (А) с минимальным весом (т.е. 1).

В подграфе с максимальным весом существуют 2 варианта:

1) Если разница между подграфами с максимальным и минимальным весом меньше чем половина среднего веса, то в подграфе с максимальным весом находим вершину (В) с минимальным весом.

2) Если разница между подграфами с максимальным и минимальным весом больше чем половина среднего веса, то в подграфе с максимальным весом находим вершину (В) с максимальным весом.

(*) Если разница между подграфами с максимальным и минимальным весом меньше среднего веса разделенный на 3, то прекращаем оптимизацию.

В приведенном примере получен 1 вариант.

Меняем 2 найденные вершины местами, получаем новое разбиение и заново рассчитываем вес подграфов.

1 подграф: вершины № 1, 5;
2 подграф: вершины № 3, 4;
3 подграф: вершины № 2, 6.

1 подграф: 7+3=10;
2 подграф:8+4=12;
3 подграф: 1+8=9.

Возвращаемся к ЭТАПУ 1 и проделываем аналогичные расчеты до тех пор, пока не наступит событие (*).

В приведенном примере событие (*) наступает на 2 итерации, т.к. 12-9=3<10/3.

Программная реализация последовательного алгоритма выполнена на языке С++. Параллельный алгоритм решения задачи оптимального разбиения на подграфы заключается в разделение матрицы на подматрицы, равные количеству подграфов, и распределение полученных подматриц по процессам с дальнейшим вычислением процессами весов соответствующих подграфов. Параллельный алгоритм реализован с использованием MPI [4].

Формулы, используемые для расчета показателей эффективности:

1. Ускорение параллельного алгоритма относительно последовательного.

$$S_p(n) = T_1(n) / T_p(n)$$

2. Эффективность выполнения параллельного алгоритма.

$$E_p(n) = S_p(n) / p,$$

где p – количество процессов;

$T_1(n)$ – время выполнения 1 процессом;

$T_p(n)$ – время выполнения p-процессами.

В табл. 1 приведены результаты эксперимента и показатели эффективности выполнения последовательного и параллельного алгоритмов на базе процессора Intel Pentium T4300 (2.1 GHz). На рис. 3 приведен график ускорение параллельного алгоритма относительно последовательного.

Таблица 1

Результаты выполнения последовательного алгоритма

Количество вершин	Количество подграфов	Время выполнения последовательного алгоритма (сек)	Время выполнения параллельного алгоритма (сек)	Ускорение параллельного алгоритма относительно последовательного	Эффективность выполнения параллельного алгоритма
10	2	10,2068	0,371209	27,49610058	13,74805
	5	5,79033	1,56706	3,695027631	0,739005
40	2	6,62573	0,483073	13,71579451	6,857897
	5	10,9005	0,413821	26,34109917	5,268219
	10	3,1136	5,58603	5,5739049	0,055739
	20	4,86581	0,817696	5,950634466	0,297531
80	10	6,10055	1,80225	3,38496324	0,338496
	20	5,12741	0,180188	28,45589051	1,422794
	40	5,48623	2,57916	2,127138293	0,053178

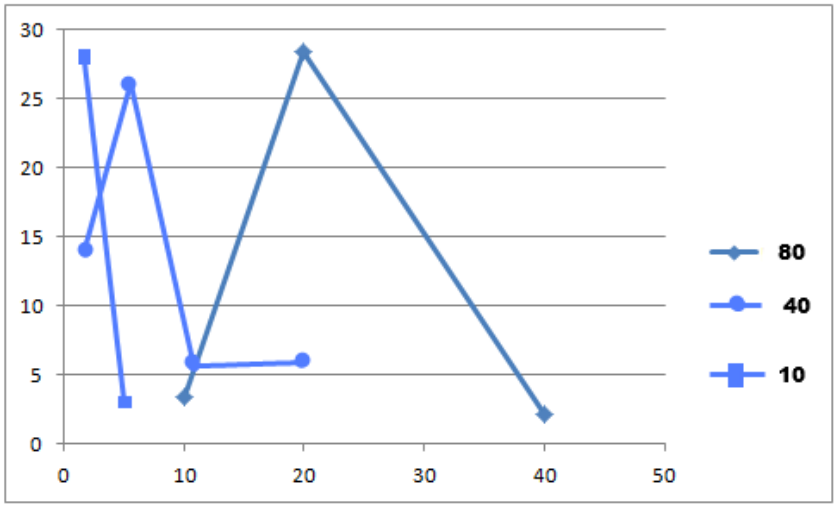

Рис. 3. Ускорение параллельного алгоритма относительно последовательного

Выводы

Графы являются одними из наиболее часто применимых средств представления тех или иных задач и проблем. Одной из часто используемых задач, решаемых теорией графов и используемых для решения разнообразного ряда задач, является оптимальное разбиение графа на подграфы. Т.о. разработка алгоритмов решения данной проблемы является актуальной. Представленные в статье показатели эффективности работы последовательного и параллельного алгоритмов показывают значительный выигрыш параллельного алгоритма по сравнению с последовательным. Т.о. использование параллельного подхода для решения задачи оптимального разбиения графа является целесообразным.

Список литературы

1. Методические указания по проведению лабораторных работ «Задачи декомпозиции графов» по курсу «Эволюционно-генетические алгоритмы решения оптимизационных задач» для студентов факультета ВМК специальности «Прикладная информатика» / Под ред. В.П. Гергеля. – Нижегородский государственный университет, 2001. – 13 с.

2. Гергель В.П. Теория и практика параллельных вычислений [Электронный ресурс] / Интернет университет информационных технологий ИНТУИТ.Р, 2009. – Режим доступа: http://www.intuit.ru/department/calculate/paralltp/ свободный. – Загл. с экрана.

3. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – С.-Пб.: БХВ-Петербург, 2002. – 608 с.

4. Немнюгин С. Параллельное программирование для многопроцессорных вычислительных систем. / С. Немнюгин, О. Стесик. – СПб.: БХВ-Петербург, 2002 – 400 с.

Поступила в редколлегию 19.03.2013

Рецензент: д-р техн. наук, проф. Ю.И. Лосев, Харьковский национальный университет имени В.Н. Каразина, Харьков.

АНАЛІЗ РОЗПАРАЛЕЛЮВАННЯ АЛГОРИТМУ ЗАДАЧІ ОПТИМАЛЬНОГО РОЗПОДІЛУ ГРАФА НА ПІДГРАФИ

С.І. Шматков, О.Г. Толстолузька, Ю.О. Артюх

У статті розглянута проблема оптимального розподілу графа на підграфи, представлені описи послідовного та паралельного алгоритмів, наведені показники ефективності паралельного алгоритму. Послідовний алгоритм реалізован з використанням мови програмування Сі, паралельний - з використанням MPI.

Ключові слова: *оптимальне розділення графа на підграфи, теорія графів, матриця суміжності, вагові коефіцієнти.*

ANALYSIS ALGORITHM'S PARALLELIZATION OF OPTIMAL GRAPH PARTITIONING INTO SUBGRAPHS'S PROBLEMS

S.I. Shmatkov, E.G. Tolslolujskaiia, Yu.A. Artiukh

There are the problem of optimal separation of the graph into subgraphs and descriptions of the serial, parallel algorithms shows the efficiency of the parallel algorithm. in the paper. Sequential algorithm is implemented using the C programming language, the parallel - using MPI.

Keywords: *optimal separation of the graph into subgraphs, graph theory, adjacency matrix, weights.*