НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»
ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА №42 «КРИПТОЛОГИЯ И КИБЕРБЕЗОПАСНОСТЬ»

ДОКЛАД ПО ДИСЦИПЛИНЕ
«Параллельное программирование»
НА ТЕМУ:

«Поиск минимального разбиения множества вершин взвешенного графа»

Работу выполнили
студенты группы Б20-505
Панков Дмитрий,
Фёдоров Алексей

Москва, 2022

1 Постановка задачи оптимального разбиения графа.

Естественной и адекватной моделью распределения задач по процессорам кластера является взвешенный неориентированный граф G, который задан множеством вершин V и множеством рёбер E: G(V, E). Кроме того, на множестве вершин определена неотрицательная функция F1(V), а на множестве рёбер также неотрицательная функция F2(E). Значения функции F1 называются весами вершин графа, а значения функции F2 – весами рёбер графа. Каждая вершина этого графа представляет собой одну из задач, а вес этой вершины - число операций, выполняемых при решении этой задачи. Вес ребер моделирует коммуникационные затраты на обмен данными между задачами, соответствующими вершинам, соединенным этими ребрами. С помощью этой модели распределение задач по процессорам представляется как разбиение множества вершин графа V на непересекающиеся подмножества, которое порождает разбиение графа G(V,E) на непересекающиеся подграфы. При этом каждый подграф ассоциируется с некоторым процессором, который будет использоваться для решения всех задач, соответствующих вершинам этого подграфа. Таким образом, оптимальное распределение вычислительной нагрузки между доступными процессорами сводится к оптимальному разделению графа G(V, E). Критериями оптимальности могут быть: 1) равенство сумм весов вершин подграфов, 2) минимальность суммы весов ребер, соединяющих вершины, принадлежащие разным подграфам; 3) число подграфов.

Первый критерий обеспечивает баланс вычислительной нагрузки процессоров. Смысл второго критерия состоит в том, что он обуславливает минимум коммуникационных затрат. Наконец, число подграфов определяет число используемых процессоров. Легко видеть, что указанные выше критерии могут противоречить друг другу. Чаще всего в различных постановках задач оптимального разбиения графов один из критериев считается главным, а остальные рассматриваются как ограничения.

На практике обычно оптимальное разделение графа выполняется для некоторого частного случая, например:

1) веса вершин и ребер графа равны 1 и в качестве приоритетного критерия используется либо условие 1, либо условие 2;

2) веса вершин различны, веса ребер равны 0, и в качестве критерия оптимальности используется только условие 1.

Ниже описывается постановка задачи (модель) оптимального разбиения вершин графа, критерий решения которой 2 предполагает балансировку сумм вычислительных нагрузок и коммуникационных затрат, требуемых для решения задач, ассоциированных с каждым процессором. Целью этого поиска минимального разбиения множества вершин графа является определение минимального количества процессоров, совокупность которых обеспечивает заданное ускорение.

Пусть задан неориентированный взвешенный граф G(V,E) с числом вершин равным n: V = {v 1 ,…,v n }, и числом рёбер равным m: E = {e 1 ,…,e m }. Функция F1 ставит в соответствие каждой вершине графа vi ее вес qi , аналогично функция F2 ставит в соответствие каждому ребру графа ei его вес ri . Такой граф определяет, что при использовании одного процессора все задачи могут быть решены в результате выполнения T1 операций, где T1 представляет собой сумму весов всех вершин множества V: T1= F1(V).

Теперь рассмотрим разбиения множества вершин V графа G(V,E) на k непересекающихся подмножеств P={V1 ,…,Vk }. Этими подмножествами вершин определяются подграфы G1 (V1 ,E 1), …, Gk (Vk ,E k), для которых выполняются следующие соотношения:

1) ∀ i,j ∈{1, 2, …, k} (i≠ j) ⇒ Vi ⋂ Vj =∅; - множества вершин подграфов не пересекаются

2) V1⋃ V2 ⋃ …⋃Vk=V; - вершины всех подграфов составляют множество вершин исходного графа

3) n1 + n2 + …+ nk = n, где n1= |V1|, n2=|V2|, …, nk=|Vk|, n=|V|. – количество вершин всех подграфов равно количеству вершин исходного графа

Будем считать, что при заданном разбиении число операций выполняемых i-тым процессором определяется как сумма весов вершин i-того подмножества: Qi =∑ F1(V i). Пусть C (V i , V j) множество рёбер, соединяющих вершины из множества V i с вершинами множества V j : C (V i , V j) = {(u,w)|u∈V i ,w∈V j }. Тогда сечением Ri по подмножеству V i графа G(V,E) будем называть совокупность ребер, соединяющих вершины принадлежащие множеству V i с вершинами, не принадлежащими этому множеству: E i =⋃j (C (V i , V j)), где i≠j. Тогда затраты на передачу и получение информации i-тым процессором можно определить как сумму весов рёбер i-того сечения: R i = F2 (E i). Исчисляемое в числе операций время решения всех задач для данного разбиения с использованием k процессоров будет равно: Tk=max((Qi + Ri), где i=1, 2, …, k).

Далее пусть задан требуемый коэффициент ускорения S. Тогда поиск минимального разбиения множества вершин графа G(V,E) сводится к поиску разбиения, минимального по числу непересекающиеся подмножеств P={V1 ,…,Vk }, для которого выполняются условия достижения заданного ускорения: max((Qi + Ri), где i=1, 2, …, k)≤ T1/S. (1.1) При этом значение параметра k определяет минимальное число процессоров в многопроцессорной вычислительной системе, которые будут 3 использоваться согласно разбиению P и обеспечат достижение заданного ускорения.

//Проверку монотонности зависимости Tk от числа процессоров k рассказывать не будем

2. Вычислительные схемы метода оптимального разделения графов на основе конструктивного перечисления разбиений множеств их вершин

Минимум функции Tk =max((Qi + Ri), где i=1, 2, …, k), ввиду её монотонности достигается при np=n, номер класса эквивалентности разбиений, а n – число вершин в разделяемом графе. Пусть, как и прежде, задан неориентированный взвешенный граф G(V,E) с числом вершин равным n. Этот граф является математической моделью решения некоторой задачи. В памяти компьютера этот граф задаётся матрицей смежности вершин A[n][n], диагональные элементы которой представляют собой временную сложность подзадач, которые могут быть решены параллельно на нескольких процессорах или последовательно на однопроцессорной платформе, а все остальные элементы этой матрицы моделируют временные сложности коммуникационных операций. При решении задачи с использованием единственного процессора коммуникационные затраты отсутствуют. И временная сложность решения всей задачи T1 определяется суммой диагональных элементов матрицы A[n][n]: mSA=A[0][0]; for(j1 = 1; j1 < n; j1++) mSA += A[j1][j1] положить T1= mSA. Если теперь T1 разделить на заданный коэффициент ускорения S, то мы получим верхнюю

границу для значений функции Tk, соответствующих искомым (допустимым) разбиениям. Добавим к этому ограничению условие «использовать минимальное число процессоров» и получим наиболее практически 5 значимую постановку задачи оптимального разделения взвешенного графа: определить разбиение множества вершин графа P={V 1 ,…,V k }, на минимальное число подмножеств k, такое для которого выполняются условия достижения заданного ускорения.

Монотонное убывание значений функции Tk при возрастании числа подмножеств в разбиениях позволяет использовать базовый алгоритм Eq2_1 как в вычислительной схеме последовательного поиска (будем отождествлять её с алгоритмом Eq2_1), так и в алгоритме двоичного поиска минимального разбиения взвешенного графа, при котором обеспечивается выполнение условия. Ниже будем именовать вычислительную схему двоичного поиска алгоритмом Eq3_1. Сначала рассмотрим псевдокод алгоритма Eq2_1.

Разработка алгоритма Eq3_1 состояла в объединении имеющей общее назначение вычислительной схемы двоичного поиска и поиска экстремального разбиения множества, которая была выше и алгоритма вычисления функции F3. Дальше приводится псевдокод этого алгоритма.

Описанные в данном разделе алгоритмы были реализованы в виде консольных программ, которые обеспечили их экспериментальное исследование. Результаты этого исследования рассматриваются в следующем разделе.

~~Для проведения вычислительного эксперимента с алгоритмами Eq2_1, Eq3_1, их программные реализации были объединены в специальную тестпрограмму. Это позволило выполнять обработку этими алгоритмами одних и тех же матриц смежности взвешенных графов, которые генерировались с использованием равномерно распредел1нных псевдослучайных чисел.~~

К сожалению, в связи с отсутсвием времени, мы не удосужились написать свою реализацию алгоритма. Тем не менее, нашли уже проведенный бэнчмарк.

N	mSA	Последовательный поиск			Двоичный поиск (Eq3_1)		
		np	maxSA	Время, с	np	maxSA	Время, с
1	455,691	11	451,529	10,469	11	451,529	1,828
2	524,902	6	523,995	5,141	6	523,995	5,25
3	477,854	8	471,153	10,187	8	471,153	2,11
4	518,264	6	517,727	4,765	6	517,727	4,672
5	493,197	7	492,011	8,266	7	492,011	4,984
6	501,415	7	498,243	8,141	7	498,243	8,141
7	518,372	6	517,691	7	6	517,691	7
8	504,544	6	501,87	5,422	6	501,87	5,609
9	480,443	8	472,134	10,171	8	472,134	2,094
10	501,598	7	501,512	8,156	7	501,512	4,968
Среднее время поиска =			**7,7718**	**Среднее время поиска =**			**4,6656**

Данные приведённые в таблице свидетельствуют о том, что двоичный поиск оказывается быстрее усовершенствованного последовательного поиска в 1,665766 раз в среднем.

Разработанный метод оптимального разделения взвешенных графов, основанный на использовании алгоритма Eq3_1, обеспечивает эффективное решение задачи

определения минимального числа процессоров, необходимых для реализации параллельных вычислений с заданным ускорением относительно последовательных вычислений.