УДК 681.3-181.48

© 2003 г. **В.Г. Кобак,** канд. тех. наук
(Донской филиал Центра тренажеростроения и подготовки персонала, Новочеркасск)

МОДИФИКАЦИЯ АЛГОРИТМА ОБСЛУЖИВАНИЯ ПО «КРИТИЧЕСКОМУ ПУТИ» ДЛЯ СИСТЕМ С ИЗБИРАТЕЛЬНЫМИ СВОЙСТВАМИ ПРИБОРОВ

Рассматривается поведение алгоритма «критического пути» в изменившихся условиях. Приведены результаты эксперимента.

Введение

В последние годы широко и все более быстрыми темпами осуществляются разработка, выпуск, внедрение в системы обработки информации разнообразных многопроцессорных и многомашинных вычислительных систем. Такие вычислительные системы отличаются повышенной гибкостью, а также хорошими показателями производительности, надежности. Программирование для многопроцессорных и многомашинных систем связано с организацией выполнения параллельных задач. Одной из таких задач является построение расписания выполнения программ (зависимых, независимых) на многопроцессорных и многомашинных системах обслуживания.

В данной работе будут рассматриваться только независимые задания, которые могут выполняться на технически, но не функционально однородных приборах обслуживания, и построение расписания будет вестись статично, т.е. в период подготовки работы программ на вычислительной системе.

В качестве основного критерия оптимальности расписания в данной работе будем рассматривать минимаксный критерий, который минимизирует максимальное время окончания работы программ при заданном числе приборов. Формализуем задачу планирования при использовании минимаксного критерия [1]:

$$\max T_j \to \min, \qquad (1)$$

где $T_j = \sum_{i \in N_j} t_{i,j}$ – общее время загрузки j-го канала при условиях

$$t_{i,j} \geq 0; i = 1,...,m, j = 1,...,n; N_k \cap N_l = \phi; k,l \neq 1,...,n, k \neq l. \ \bigcup_{j=1}^{n} N_j = M,$$

где M – множество заданий, которые необходимо распределить между приборами; N – количество однородных приборов, которые могут параллельно обслуживать; t_{ij} – длительность выполнения i-го задания j-м прибором.

Задача раскраски взвешенного графа

Для решения минимаксной задачи для однородных систем обслуживания существуют точные и приближенные методы. В данной работе будем рассматривать эвристический метод, основанный на идее «критического пути», который описан в [2]. В этой же работе дается аналитическое решение проблемы, показывающее, насколько максимально отстоит метод «критического пути» от оптимального решения. В то же время при практической работе однородной системы часто встречается ситуация, когда одна программа может выполняться на одной машине, а на другой не может быть выполнена (не установлено соответствующее программное обеспечение). Кроме того, в работе [3] впервые была решена задача раскраски обыкновенного взвешенного графа и наиболее четко рассмотрена подобная задача. Формальная постановка раскраски обыкновенного взвешенного графа имеет следующий вид.

Задан граф $G = (X,U)$, где X – множество вершин графа G, а U – множество ребер. Граф G является обыкновенным (нет петель, нет мультиребер, ребра не ориентированны), однако каждой вершине поставлено в соответствие целое число, которое характеризует либо время выполнения данной вершины, либо объем памяти ($V(x)$). Необходимо раскрасить вершины графа G таким образом, чтобы никакие две смежные вершины не были окрашены одинаково и выполнялось условие

$$\max_{j \in (1,...,P)} \sum_{X \in B_j} V(X) \to \min, \qquad (2)$$

где B_j – подмножество несмежных вершин графа G, окрашенных в j-й цвет; P – хроматическое число графа G. В работе [3] предлагалась схема решения, которая включала три этапа.

На первом этапе по методу Магу [4] находят все максимально внутренне устойчивые подмножества C_ℓ, $L=\overline{1,K}$, где K – число вершин в таком подмножестве. На втором этапе методом ветвей и границ решается задача о покрытии, т.е. определяются кортежи T_ε ($\varepsilon = \overline{1,l}$), где ℓ -число таких кортежей, элементами которых являются подмножества C_ℓ – такие, что их объединение дает множество A, а количество C_ℓ равно хроматическому числу графа G. На третьем, заключительном этапе ℓ раз решается задача распределения функциональных программ по однородным вычислительным машинам. Размеры матрицы загрузки составляются следующим образом: число строк в матрицах равно $|A|$, а число столбцов P. Содержимое матрицы загрузки заполняется по следующему правилу: если вершина $a_m \in C_\ell$, а C_ℓ является элементом кортежа T_j, то на пересечении m-й строки и номера, под которым стоит подмножество C_ℓ в кортеже T_j, ставится значение $V(a_m)$.

Если $a_m \notin C_\ell$, а C_ℓ не является элементом кортежа T_j, то в этом случае ставится заведомо большее число, – например, на порядок превосходящее любое из $V(a_m)$. Это делается для того, чтобы в результате работы алгоритма распределения (планировщика) оператор a_m не мог быть назначен другому процессору.

Модификация алгоритма распределения, основанная на идее «критического пути»

Как видно из построения, строка матрицы загрузки имеет либо значение $V(a_m)$, либо, условно, $V(a_m)=\infty$. Таким образом, применение метода «критического пути» напрямую невозможно, так как он требует упорядочения строк матрицы загрузки в порядке убывания элементов строки, а почти в каждой строке присутствует бесконечный элемент. Следовательно, для решения задачи планирования для матриц загрузки такого вида предлагается три различные модификации метода, основанного на идее «критического пути».

Первая модификация:

все операторы, имеющие время $a_i > 0$,но не равное бесконечности, записываются под тем же индексом в матрицу-строку $A1[1,m]$, где m – число строк в матрице A;

элементы матрицы $A1$ упорядочиваются в порядке убывания, формируя список приоритетов;

строки матрицы A упорядочиваются в соответствии со списком приоритетов;

решается задача распределения согласно минимаксному критерию.

Вторая модификация:

в каждой строке матрицы A ищется количество бесконечностей и записывается в матрицу-строку $A1$;

элементы матрицы $A1$ упорядочиваются в порядке убывания количества бесконечностей, формируя список приоритетов;

строки матрицы A упорядочиваются в соответствии с получившимся списком приоритетов;

решается задача распределения согласно минимаксному критерию.

Третья модификация:

в каждой строке матрицы A ищется количество бесконечностей и записывается в матрицу-строку $A1$;

элементы матрицы $A1$ упорядочиваются в порядке убывания количества бесконечностей;

строки матрицы A упорядочиваются в соответствии с матрицей $A1$;

если строки матрицы A имеют одинаковое количество бесконечностей, то происходит упорядочение в порядке убывания элементов, отличных от бесконечности;

решается задача распределения согласно минимаксному критерию.

Проиллюстрируем приведенные модификации алгоритма, основанного на идее «критического пути», на примере. Пусть исходная матрица имеет следующий вид:

$$A = \begin{vmatrix} 4 & \infty & 4 \\ 3 & 3 & 3 \\ 19 & 19 & \infty \\ 4 & 4 & 4 \\ 8 & \infty & \infty \\ \infty & 17 & 17 \end{vmatrix}.$$

Рассмотрим первую модификацию. Матрица $A1$ представляет собой соотношение $A1= |4{,}3{,}19{,}4{,}8{,}17|$. Модифицированная матрица имеет вид $A1` = |19{,}17{,}8{,}4{,}4{,}3|$, а упорядоченная матрица –

$$A` = \begin{vmatrix} 19 & 19 & \infty \\ \infty & 17 & 17 \\ 8 & \infty & \infty \\ 4 & \infty & 4 \\ 4 & 4 & 4 \\ 3 & 3 & 3 \end{vmatrix}.$$

Решая задачу распределения, получаем следующее решение: на первый прибор распределяются 3-е и 5-е задания, на второй прибор распределяется 6-е задание, на третий прибор распределяются все оставшиеся задания. Получившаяся загрузка приборов равна $T_1 = 27$, $T_2 = 17$, $T_3 = 11$.

Рассмотрим вторую модификацию, где модифицированная матрица $A1`=|8{,}4{,}19{,}17{,}3{,}4|$, а упорядоченная матрица

$$A` = \begin{vmatrix} 8 & \infty & \infty \\ 4 & \infty & 4 \\ 19 & 19 & \infty \\ \infty & 17 & 17 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{vmatrix}$$

Решая задачу распределения, получаем конечную загрузку приборов: T_1=15, T_2=19, T_3=21.

Рассмотрим третью модификацию, где модифицированная матрица $A1` = |8,4,19,17,3,4|$, а упорядоченная матрица

$$A` = \begin{vmatrix} 8 & \infty & \infty \\ 19 & 19 & \infty \\ \infty & 17 & 17 \\ 4 & \infty & 4 \\ 4 & 4 & 4 \\ 3 & 3 & 3 \end{vmatrix}$$

Решая задачу распределения, получаем конечную загрузку приборов $T_1 = 19$, $T_2 = 19$, $T_3 = 17$.

Попробуем решить эту задачу «в лоб», без всякого предварительного упорядочения. Получим следующее решение: $T_1 = 12$, $T_2 = 22$, $T_3 = 21$.

Как видно из результатов примера, наиболее перспективной модификацией с точки зрения точности результата является третья.

Для определения, какая из модификаций точнее, был проведен вычислительный эксперимент, результаты которого отражены в табл. 1 – 3.

Таблица 1

Сравнение модификаций для двух приборов

Кол-во работ, шт.	$T_{ср}$. max, у.е.			
	1-й алгоритм	2-й алгоритм	3-й алгоритм	4-й алгоритм
5	48,37	47,53	46,12	48,44
10	84,09	83,30	80,49	86,78
15	126,83	126,98	123,41	131,53
20	165,42	165,45	162,31	170,09
25	196,75	199,00	194,94	202,49
30	244,60	245,89	241,73	250,96
35	280,96	281,91	277,88	288,64
40	324,50	326,36	321,85	332,08
45	367,39	368,75	364,70	374,89
50	405,05	406,99	402,23	411,85
55	443,51	447,05	442,31	451,33
60	484,91	487,11	482,76	491,63
65	526,66	530,27	525,45	534,66
70	564,48	567,00	562,77	570,38
75	605,55	609,75	604,59	613,24
80	637,82	641,45	636,62	647,01
85	678,07	680,80	676,84	685,81
90	730,10	732,46	728,43	738,56
95	761,46	764,58	760,21	769,58
100	806,86	810,14	805,69	814,78

Из [5] известно, что наиболее часто для вычислительных систем количество используемых приборов колеблется от двух до четырех, поэтому в таблицах приведены результаты для такого количества приборов. Кроме того, количество операторов, подлежащих распределению, последовательно увеличивается от 5 до 100 у.е., а время выполнения каждого оператора распределено по равномерному закону в интервале от 2 до 30.

Таблица 2

Сравнение модификаций для трех приборов

Кол-во работ, шт.	$T_{ср}$. max, у.е.			
	1-й алгоритм	2-й алгоритм	3-й алгоритм	4-й алгоритм
5	38,24	36,90	35,55	39,86
10	63,79	61,98	59,68	66,15
15	87,93	88,73	84,82	95,14
20	112,12	114,38	109,72	119,06
25	139,67	141,55	136,73	147,28
30	166,71	169,59	164,57	175,07
35	192,93	196,61	191,08	200,81
40	215,48	219,63	213,63	224,50
45	243,31	247,58	241,77	254,35
50	268,16	273,01	266,78	276,74
55	297,97	302,22	296,06	307,77
60	322,74	327,55	321,39	331,43
65	349,20	353,83	347,23	358,04
70	377,60	383,20	376,42	388,86
75	401,58	406,81	400,66	413,35
80	424,56	430,23	423,57	436,59
85	446,51	451,87	445,32	456,10
90	480,24	485,76	479,63	491,68
95	507,18	512,47	505,98	518,48
100	536,42	542,41	535,43	548,27

Таблица 3

Сравнение модификаций для четырех приборов

Кол-во работ, шт.	$T_{ср}$. max, у.е.			
	1-й алгоритм	2-й алгоритм	3-й алгоритм	4-й алгоритм
5	31,88	30,07	29,05	33,10
10	49,10	48,82	46,46	52,23
15	66,34	67,95	65,12	72,89
20	87,33	88,51	85,20	93,73
25	105,78	108,03	104,20	114,09
30	126,13	130,35	124,84	134,57
35	145,02	148,42	143,01	153,98
40	164,28	168,13	162,86	173,85
45	184,14	188,85	182,42	193,74
50	204,22	208,90	202,59	213,07
55	223,22	227,92	221,72	234,34
60	241,38	246,15	239,78	252,01
65	260,47	266,34	259,61	271,23
70	282,02	287,46	281,01	293,87
75	305,95	311,62	304,97	316,88
80	323,65	330,52	322,32	334,48
85	343,47	350,25	342,68	356,11
90	359,05	365,22	358,60	372,09
95	380,91	387,45	379,96	391,72
100	403,76	411,48	402,82	415,23

После генерации матрицы загрузки для каждой строки генерировалось случайное число, равномерно распределенное от 0 до n, которое показывало число бесконечностей в строке. После этого для каждого полученного числа генерировались равномерно распределенные числа, отвечающие за номер бесконечностей в строке. Для каждой конкретной размерности генерировались 100 матриц, и каждая такая матрица решалась тремя модификациями и алгоритмом распределения без предварительного упорядочения.

Заключение

По результатам вычислительного эксперимента можно сделать следующие выводы:

все три модификации дают лучшее решение, чем распределение без предварительного упорядочения;

наилучшие результаты по всем таблицам дает третья модификация;

вторая модификация дает результаты лучшие, чем первая, для малого количества заданий, но при увеличении количества заданий первая дает все же лучшие результаты, нежели вторая.

ЛИТЕРАТУРА

1. *Алексеев О.Г.* Комплексное применение методов дискретной оптимизации. М.: Наука, 1987.
2. *Кофман Э.Г.* Теория расписаний и вычислительные машины. М.: Наука. 1987.
3. *Букин В.В., Кобак В.Г.* Алгоритм раскраски взвешенного графа. //Известия СКНЦВШ. Технические науки. 1988. №3.
4. *Магу* (*K. Maghout*)/ Applications de e'Algebre de Bool a la Theorie des Graphes, Cahiers du Centie d'Etudes de Recherche Operationnelle. Bruxelles 5. №1-2 (1963). 21.
5. *Ларионов А.М., Майоров С.А., Новиков Г.И.* Вычислительные комплексы, системы и сети. Л.: Энергоатомиздат. 1987.

Статья представлена к публикации членом редколлегии А.В. Бушмановым.