

## МЕТОД АНАЛИЗА СТРУКТУРЫ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ, РЕАЛИЗУЕМЫХ НА ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Д.Н.Кикнадзе

(Москва)

Одной из причин создания вычислительных систем (ВС) [1] является наличие большого числа задач, решение которых на одной электронно-вычислительной машине (ЭВМ) невозможно или требует значительных затрат машинного времени. Будем называть задачи такого типа б о л ь ш и м и .

Для реализации большой задачи на ВС необходимо представить вычислительный алгоритм в параллельной форме. Последний должен обладать возможностью нахождения распределения, при котором обеспечивается минимальное суммарное время простоя ЭВМ ВС.

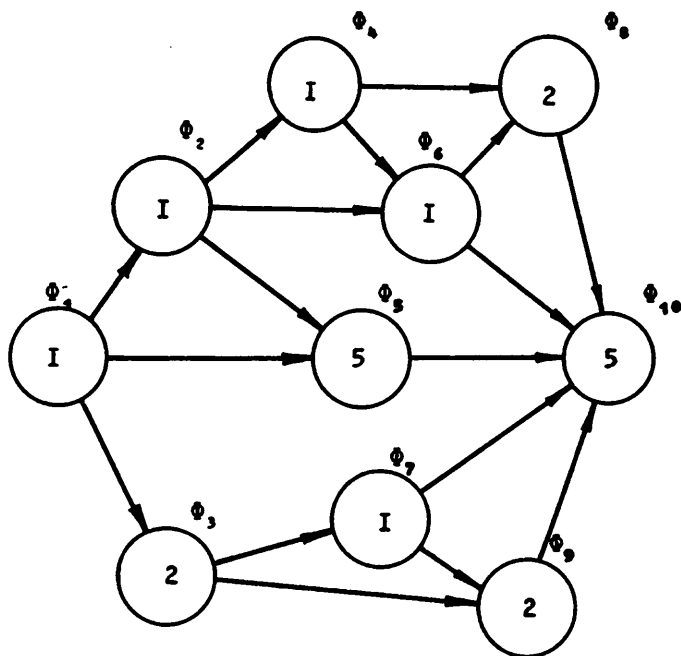
В этой работе предлагается способ определения числа ЭВМ, близкого к наименьшему достаточному для реализации данного алгоритма на ВС за некоторое заранее фиксированное время  $T$ , что, в свою очередь, обеспечивает уменьшение суммарного времени простоя ЭВМ ВС.

Всякий алгоритм, реализуемый на ВС, можно представить в виде временной функциональной граф-схемы [2]. Всем вершинам такого графа, которые являются функциональными операторами (ФО), поставлен в соответствие некоторый вес, выражающий в условных единицах время выполнения оператора на данной ЭВМ. Вершины соединены между собой ребрами двух типов: функциональными и управляющими. Функциональные ребра определяют связь по величинам между отдельными операторами. Функциональное ребро, идущее из оператора  $\Phi_1$  к оператору  $\Phi_2$ , показывает, что результат работы оператора  $\Phi_1$  служит аргументом при реализации оператора  $\Phi_2$ . Управляющие ребра определяют порядок

выполнения операторов. Управляющее ребро, идущее из оператора  $\Phi_1$  к операторам  $\Phi_{j_1}, \Phi_{j_2}, \dots, \Phi_{j_k}$ , показывает, что после выполнения оператора  $\Phi_1$  должен быть выполнен один и только один из операторов  $\Phi_{j_1}, \Phi_{j_2}, \dots, \Phi_{j_k}$ .

Будем считать, что имеются ребра только одного типа, а именно - функционального. Задача о сведении структуры алгоритма с ребрами двух типов к структуре алгоритма с ребрами одного типа будет рассмотрена в специальной работе автора.

На рисунке приведен пример параллельного алгоритма, структуру которого определяют ребра функционального типа. Числа, вписанные в вершины показывают длительность выполнения оператора  $\Phi_i$  в условных единицах времени.



По граф-схеме известными методами [ 3 ] можно определить минимально возможное время реализации алгоритма -  $T_{кр}$ .

Для приведенного примера на рисунке  $T_{кр} = 12$  условным единицам. Критический путь включает  $\Phi_1, \Phi_2, \Phi_5, \Phi_{10}$ .

Можно также определить моменты времени  $T_{\Phi_i}^1$  - самого раннего начала реализации и  $T_{\Phi_i}^1$  - самого позднего окончания реализации оператора  $\Phi_i$ .

Данные для граф-схемы приведены в таблице.

	$t_1$	$T_{\Phi_1}^0$	$T_{\Phi_1}^1$
$\Phi_1$	I	0	I
$\Phi_2$	I	I	2
$\Phi_3$	2	I	4
$\Phi_4$	I	2	4
$\Phi_5$	5	2	7
$\Phi_6$	I	3	5
$\Phi_7$	I	3	5
$\Phi_8$	2	4	7
$\Phi_9$	2	4	7
$\Phi_{10}$	5	7	I2

Представим каждый  $\Phi_i$  строкой вида:

$$a_0, a_1, a_2, \dots, a_{n-1}, \quad (I)$$

где  $n$  - длина критического пути в условных единицах, а

$a_\mu$  ( $\mu = 0, 1, \dots, n-1$ ) принимает следующие значения:

$$a_\mu = I, \text{ если } T_{\Phi_i}^{\text{нач}} \leq \mu < T_{\Phi_i}^{\text{нач}} + t_1;$$

$$a_\mu = 0 - \text{в противном случае.}$$

Здесь  $T_{\Phi_i}^{\text{нач}}$  - один из возможных моментов времени начала  $\Phi_i$ .

Например,  $\Phi_8$  можно начать в момент времени  $t = 4$ . Тогда соответствующая ему строка выглядит так:

$$\begin{array}{cccccccccccc} 0 & I & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & IO & II \\ 0 & 0 & 0 & 0 & I & I & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Для тех операторов, которые не находятся на критическом пути, справедливо следующее соотношение:  $T_{\Phi_i}^0 + t_1 < T_{\Phi_i}^1$ .

Поэтому реализацию такого оператора можно начать в моменты времени  $\tau_{\Phi_i}^1 = T_{\Phi_i}^0$  или  $\tau_{\Phi_i}^2 = T_{\Phi_i}^0 + I$ , или  $\tau_{\Phi_i}^3 = T_{\Phi_i}^0 + 2$  и т.д. до тех пор, пока не станет справедливым равенство  $\tau_{\Phi_i}^k + t_1 = T_{\Phi_i}^1$ . Таким образом, для каждого оператора, не лежащего на критичес-

ком пути, получим множество строк вида (I), каждую из которых ( $\Phi_1^j$ ) назовем j-ой модификацией оператора  $\Phi_1$ .

Например, для  $\Phi_8$  существуют две модификации  $\Phi_8^1$  и  $\Phi_8^2$ :

$$\Phi_8^1 - 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad I \quad I \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$\Phi_8^2 - 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad I \quad I \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

Все функциональные операторы и их модификации, представленные строками вида (I), можно расположить в виде матрицы.

Для приведенной выше граф-схемы (см. рисунок) такая матрица выглядит следующим образом:

	0	1	2	3	4	5	6	7	8	9	10	11
$\Phi_1^1$	<u>I</u>	0	0	0	0	0	0	0	0	0	0	0
$\Phi_2^1$	0	<u>I</u>	0	0	0	0	0	0	0	0	0	0
$\Phi_5^1$	0	0	<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>	0	0	0	0	0
$\Phi_{10}^1$	0	0	0	0	0	0	0	<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>
-----												
$\Phi_3^1$	0	<u>I</u>	<u>I</u>	0	0	0	0	0	0	0	0	0
$\Phi_3^2$	0	0	<u>I</u>	<u>I</u>	0	0	0	0	0	0	0	0
-----												
$\Phi_4^1$	0	0	<u>I</u>	0	0	0	0	0	0	0	0	0
$\Phi_4^2$	0	0	0	<u>I</u>	0	0	0	0	0	0	0	0
-----												
$\Phi_6^1$	0	0	0	<u>I</u>	0	0	0	0	0	0	0	0
$\Phi_6^2$	0	0	0	0	<u>I</u>	0	0	0	0	0	0	0
-----												
$\Phi_7^1$	0	0	0	<u>I</u>	0	0	0	0	0	0	0	0
$\Phi_7^2$	0	0	0	0	<u>I</u>	0	0	0	0	0	0	0
-----												
$\Phi_8^1$	0	0	0	0	<u>I</u>	<u>I</u>	0	0	0	0	0	0
$\Phi_8^2$	0	0	0	0	0	<u>I</u>	<u>I</u>	0	0	0	0	0
-----												

	$\Phi_9^1$	0	0	0	0	1	1	0	0	0	0	0	0
	$\Phi_9^2$	0	0	0	0	0	1	1	0	0	0	0	0
0	м - вес	1	2	3	5	5	3	3	1	1	1	1	1
	п - вес	1	1	2	1	1	3	1	1	1	1	1	1
I	м - вес	1	2	3	4	5	3	3	1	1	1	1	1
	п - вес	1	2	2	1	1	3	1	1	1	1	1	1
II	м - вес	1	2	3	4	4	3	3	1	1	1	1	1
	п - вес	1	2	2	1	1	3	2	1	1	1	1	1
III	м - вес	1	2	2	3	4	3	3	1	1	1	1	1
	п - вес	1	2	2	2	2	3	2	1	1	1	1	1
IV	м - вес	1	2	2	3	3	3	3	1	1	1	1	1
	п - вес	1	2	2	3	2	3	2	1	1	1	1	1

Введем некоторые определения.

Просматривая  $\mu$ -ый столбец матрицы, можно утверждать, что количество машин, работающих в  $\mu$ -ый промежуток времени, не превосходит числа разных  $\Phi_i$ , входящих в этот столбец. Назовем это число  $m_\mu$  - вес  $\mu$ -го столбца матрицы, а совокупность всех  $m_\mu$  -весов -  $m$  - вес  $m$  матрицы.

В то же время количество машин, работающих одновременно в  $\mu$ -ый промежуток времени, не может быть меньше  $n$  числа разных  $\Phi_i$ , входящих в  $\mu$ -ый столбец, за исключением тех, у которых имеется хотя бы одна модификация, не пересекающая данный столбец. Это число назовем  $n_\mu$  - вес  $\mu$ , а совокупность  $n_\mu$  - весов для всех столбцов -  $n$  - вес  $n$  матрицы.

$M_i^j$  - вес  $\mu$   $j$ -ой модификации  $i$ -го функционального оператора называется невозрастающая последовательность  $m_\mu$  - весов тех столбцов, которые пересекает данная модификация  $\Phi_i^j$ .

$N_i^j$  - вес  $\mu$   $j$ -ой модификации  $i$ -го оператора называется невозрастающая последовательность  $n_\mu$  - весов тех столбцов, которые пересекает данная модификация  $\Phi_i^j$ .

Для граф-схемы (см. рисунок) вычислены  $m$ - и  $n$ - веса

матрицы и приведены в строках, помеченных знаком 0.  $M_8^2$  - вес 2-ой модификации  $\Phi_8$ , равен 33, а  $N_8^2$  - вес - 31.

Путем в граф-схеме из  $\Phi_{i_1}$  в  $\Phi_{i_r}$  будем называть всякую такую последовательность  $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_r}$ , что для каждого  $\Phi_{i_v}$  ( $v = 1, 2, \dots, r$ ) выполняются условия: входом  $\Phi_{i_v}$  является выход  $\Phi_{i_{v-1}}$ , а выход  $\Phi_{i_v}$  является входом  $\Phi_{i_{v+1}}$ .  $\Phi_{i_1}$  и  $\Phi_{i_r}$  являются соответственно началом и концом пути.

Путем в матрице  $\Phi_{i_1}^{j_1}, \Phi_{i_2}^{j_2}, \dots, \Phi_{i_r}^{j_r}$  будем называть множество тех строк, соответствующие функциональные операторы которых образуют путь в граф-схеме, а также выполняются условия  $\tau_{i_{v-1}}^{j_{v-1}} + t_{i_{v-1}} = \tau_{i_v}^{j_v}$ .

Например, модификации  $\Phi_7^1$  и  $\Phi_8^1$  образуют путь в матрице.

Модификация  $\Phi_1^{j_1}$  называется крайней слева, если не существует другой такой модификации  $\Phi_1^{j_2}$ , для которой момент времени начала  $\tau_1^{j_2}$  был бы меньше, чем  $\tau_1^{j_1}$ . Для модификации, крайней слева, выполняется неравенство  $\tau_1^{j_1} < \tau_1^{j_2}$ .

Аналогично определяется модификация, крайняя справа. Только в этом случае  $\tau_1^{j_1} > \tau_1^{j_2}$ . Теперь перейдем к самому алгоритму.

Общая идея алгоритма состоит в следующем: получив матрицу приведенного выше вида, последовательно вычеркиваем некоторые модификации так, что  $n_\mu$  - веса столбцов растут по возможности меньше, а  $m_\mu$  - веса уменьшаются по возможности больше. Уменьшив  $m_\mu$  - вес, тем самым уменьшим и количество ЭВМ, работающих одновременно в  $\mu$ -ый промежуток времени. Алгоритм заканчивает работу тогда, когда

$$(\max m_\mu) - n_\mu = 0 \quad (\mu = \mu_1, \mu_2, \dots, \mu_r; r \leq n).$$

Это означает, что дальнейшее уменьшение числа ЭВМ, работающих в  $\mu$ -ый промежуток времени, невозможно.

### А л г о р и т м

1. Определяем  $m_\mu$  - вес и  $n_\mu$  - вес матрицы и переходим к 2.

2. Ищем все  $\max(m_\mu)$  такие, что  $[(\max m_\mu) - n_\mu]$  минимально. Возможны два случая:

а) если  $\min [(\max m_{\mu}) - n_{\mu}] = 0$ , то - конец;  
 б) если  $\min [(\max m_{\mu}) - n_{\mu}] \neq 0$  в столбцах  $\mu_1, \mu_2, \dots, \mu_r$ ,  
 то отмечаем столбец  $\mu_1 < \mu_2 < \dots < \mu_r$  и переходим к 3.

3. Выбираем некоторую модификацию  $\Phi_{i_1}^{j_1}$ , входящую в данный  $\mu_1$ -ый столбец, и сравниваем  $M_{i_1}^{j_1}$  и  $N_{i_1}^{j_1}$  - веса с  $M_{i_1}^{j_2}$  и  $N_{i_1}^{j_2}$  - весами другой модификации  $\Phi_{i_1}^{j_2}$ . Возможны следующие случаи:

$$\text{а) } M_{i_1}^{j_2} > N_{i_1}^{j_2} \text{ и } N_{i_1}^{j_1} > N_{i_1}^{j_2}.$$

Вычеркиваем  $\Phi_{i_1}^{j_1}$  и переходим к 4.

$$\text{б) } M_{i_1}^{j_1} > M_{i_1}^{j_2} \text{ и } N_{i_1}^{j_1} = N_{i_1}^{j_2}.$$

Вычеркиваем  $\Phi_{i_1}^{j_1}$  и переходим к 4.

$$\text{в) } M_{i_1}^{j_1} = M_{i_1}^{j_2} \text{ и } N_{i_1}^{j_1} > N_{i_1}^{j_2}.$$

Вычеркиваем  $\Phi_{i_1}^{j_1}$  и переходим к 4.

$$\text{г) } M_{i_1}^{j_1} > M_{i_1}^{j_2} \text{ и } N_{i_1}^{j_1} < N_{i_1}^{j_2}.$$

Вычеркиваем  $\Phi_{i_1}^{j_2}$  и переходим к 4.

$$\text{д) } M_{i_1}^{j_1} = M_{i_1}^{j_2} \text{ и } N_{i_1}^{j_1} = N_{i_1}^{j_2}.$$

Вычеркиваем  $\Phi_{i_1}^{j_1}$  и переходим к 4.

4. Возможны следующие случаи:

а) Если вычеркнутая модификация не является крайней, то переходим к I.

б) Если вычеркнутая модификация является крайней слева и если новой крайней слева стала  $\Phi_{i_1}^{j_1}$ , то вычеркиваются также все модификации  $\Phi_{i_1}^{j_1}, \Phi_{i_2}^{j_2}, \dots, \Phi_{i_r}^{j_r}$ , образующие в матрице путь, который начинается раньше, чем  $\tau_{i_1}^{j_1} + t$ , и началом которого является одна из модификаций  $\Phi_{i_1}$ , затем переходим к I.

в) Если вычеркнутая модификация является крайней справа и если новой крайней справа стала  $\Phi_{i_1}^{j_1}$ , то вычеркиваются также все модификации  $\Phi_{i_1}^{j_1}, \Phi_{i_2}^{j_2}, \dots, \Phi_{i_r}^{j_r}$ , образующие путь в матрице, который кончается не позже чем  $\tau_{i_1}^{j_1}$  и концом которого является одна из модификаций  $\Phi_{i_1}$ . Переходим к I.

К матрице, приведенной выше, алгоритм применяется четыре

раза. Вычисленные на каждом из четырех этапов  $m$ - веса и  $n$  - веса матрицы приведены в строках, помеченных знаками I, II, III, IV. Из последней пары строк видно, что алгоритм, представленный граф-схемой на рисунке, не может быть реализован менее чем на трех ЭВМ за время, равное  $T_{кр} = 12$  условным единицам.

Подчеркнутые в матрице модификации означают, что именно они должны быть выполнены на какой-нибудь ЭВМ. К тем модификациям, с которыми работает алгоритм на  $i$ -ом этапе, приписаны цифры, равные  $i$ .

### Л и т е р а т у р а

1. Э.В.Евреинов, Ю.Г.Косарев. О вычислительных системах высокой производительности.- Известия АН СССР, Техническая кибернетика, 1963, № 4, стр.3-25.
2. Д.А.Поспелов. Об одном способе построения вычислительной системы.- Семинар "Теория автоматов", Киев, 1964.
3. С.И.Зуховицкий, И.Д.Радчик. Математические методы сетевого планирования. М., Изд-во "Наука", 1965.