

Планирование распределения программных модулей по процессорам вычислительной системы

© Ю.М. Руденко

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Рассмотрены вопросы планирования распределения программных модулей по узлам однородной вычислительной системы с разделяемой памятью. В качестве инструмента планирования использованы временные диаграммы ранних сроков окончания выполнения программных модулей. Это позволяет заранее распределять модули по узлам вычислительной системы и минимизировать время решения поставленной задачи и количество процессоров, используемых для ее решения. При вычислении ранних сроков окончания выполнения программных модулей учитывалось время передачи информации между нитями, составленными из программных модулей. На основе этих диаграмм созданы нити программных модулей, каждая из которых выполнена на своем процессоре. Между некоторыми процессорами существует обмен информацией, требующий определенных временных затрат, которые необходимо учесть, скорректировав соответствующим образом временные диаграммы. В ряде случаев некоторое количество процессоров может быть использовано в качестве передающих звеньев (транзитных вычислительных элементов). Построение плана распределения программных модулей по процессорам рассмотрено на примере структуры вычислительной системы из класса D_n -графов типа «Циркулянт» $G(11, 1, 2, 5)$. Так как диаметр этой вычислительной системы равен двум, то максимальное количество транзитных процессоров для соответствующей передачи равно единице. Алгоритм обработки данных представлен в виде граф-схемы, в которой цикл по счетчику циклов приведен в явном виде, так как реализация этой части алгоритма является наиболее сложной. Вначале нити распределяются по процессорам произвольно, затем осуществляется построение межнитевых связей. Здесь важно, чтобы количество неиспользуемых в заданные отрезки времени процессоров было достаточным для построения этих связей. При этом в случае необходимости можно минимизировать количество транзитных процессоров и время решаемой задачи, осуществляя различные варианты размещения нитей на вычислительных модулях.

Ключевые слова: вычислительная система, распределение модулей, программные модули, вычислительный модуль, граф-схема, матрица следования, матрица дистанций, нить.

Наиболее распространенным способом решения задач на однородной вычислительной системе является построение очередей из программных модулей, которые затем в связи с возникающими ситуациями случайным образом распределяются по процессорам [1, 2, 3]. В настоящее время для вычислительного элемента, находящегося в узле вычислительной сети, используют различные термины: элемен-

тарный процессор, элементарная вычислительная машина, процессор, вычислительный модуль, узел вычислительной системы и др. В этой статье термины «процессор», «узел» и «вычислительный модуль» будут применять в качестве синонимов. В общем случае в связи с появлением более сложных вычислительных элементов, таких как суперскалярные, мультитредовые [1, 2] и другие сложные структуры, требуется учитывать дополнительные особенности создания вычислительных систем, которые в данной статье не рассматриваются.

Как было показано в [4–8], весьма удобным средством для изображения параллельных алгоритмов является граф-схема. В работах [9, 10] был предложен метод построения диаграмм решения параллельной задачи, представленной информационной граф-схемой.

Рассмотрим планирование распределения программных модулей по узлам вычислительной системы для случая, когда каждый вычислительный модуль имеет свою локальную память. Временные диаграммы запуска и работы вычислительных модулей [11] являются удобным средством для построения плана распределения операторов по узлам вычислительной системы. Строки временной диаграммы будем интерпретировать как нити алгоритма решаемой задачи. Для удобства размещения операторов алгоритма на вычислительные модули системы представим ее структуру в виде матрицы дистанций между вычислительными модулями [10], в которой указаны расстояния между узлами. Элементы $R(i, j)$ матрицы дистанций \mathbf{R} вычисляются по формуле

$$R(i, j) = r(i, j),$$

где $r(i, j)$ — расстояние между i -м и j -м вычислительными модулями, определяемое минимальным количеством процессоров, которые необходимо пройти для попадания из i -го в j -й вычислительный модуль, $1 \leq r(i, j) \leq N - 2$ (N — заданное количество вычислительных модулей). Предполагается, что N вычислительных модулей достаточно для решения поставленной задачи указанным методом.

Чтобы определить показатель близости модулей, найдем сумму столбцов $St(j)$ матрицы дистанций:

$$St(j) = \sum_{i=1}^N r(i, j), \quad (1)$$

где $j = 1, \dots, N$.

Упорядочим $St(j)$ по возрастанию. Далее для построения диаграммы ранних сроков окончания выполнения операторов образуем

множество несвязанных между собой пучков нитей $\{P_s\}$, где под нитью понимается непрерывная последовательность программных модулей, расположенных на одном процессоре, $s \in \{0, 1, \dots, q\}$. После чего вычислим ранние сроки окончания выполнения операторов, используя следующий алгоритм [11], модифицированный для учета времени передачи информации:

1. Вычисляется $t_{1j} := 0$, где $j := 1, \dots, M$ (M — размер матрицы следования S).

2. Просматриваются строки матрицы S сверху вниз, выбирается первая необработанная строка и осуществляется переход к следующему шагу; если обработаны все строки, происходит переход в конец алгоритма.

3. Если выбрана j -я строка, не содержащая единичных элементов, вычисляется $t_{1j} := p_j$, где p_j — вес j -го оператора, и осуществляется переход на шаг 5.

4. Если j -я строка содержит единичные элементы, то вычисляется

$$t_{1j} := \max \{t_{1j_q}\} + p_j,$$

где максимум берется по множеству времен t_{1j_q} (j_q — номер элемента j -й строки, равного единице). Если в множестве $\{t_{1j_q}\}$ есть нулевые элементы, то выполняется шаг 6, в противном случае выполняется шаг 5.

5. Строка j метится как обработанная и исключается из рассмотрения. Осуществляется переход на шаг 2.

6. Так как найдены не все значения $\{t_{1j_q}\}$, осуществляется поиск следующей необработанной строки и переход на шаг 3.

Среди множеств $\{P_s\}$ выделим множество $\{P_z\}$, нити которого имеют максимальное число связей с другими нитями этого множества. В множестве $\{P_z\}$ выберем нить T_i и будем выполнять ее на вычислительном модуле с номером $w \in N$. образуем последовательность связей i -й нити с другими нитями множества $\{P_z\}$: $S_1^i, S_2^i, \dots, S_d^i$, где верхний индекс обозначает номер нити, которой принадлежит данная связь, а нижний — порядковый номер связи нити.

Для учета времени передачи данных между нитями пучка вначале определим время передачи информации по связи на расстоянии, равном единице: $t(S_1^i), t(S_2^i), \dots, t(S_d^i)$. Определим также, какая нить из множества $\{P_z\}$ соединена связью S_m^i , имеющей $\max t(S_m^i), m \in \{0, 1, \dots, d\}$.

Предположим, что это будет нить T_{jm} . Связь S_m^i в нити T_i соединяет оператор y с оператором a нити T_{jm} , при этом вес оператора y становится равным

$$p_y = p_y t(S_m^i) r(i, j), \quad (2)$$

а вес оператора a

$$p_a = p_a t(S_m^i) r(i, j). \quad (3)$$

В нити T_i все операторы, начиная с y , сдвинуты по оси времени вправо на величину $t(S_m^i) r(i, j)$. Аналогично в нити T_{jm} все операторы, начиная с a , сдвинуты на величину $t(S_m^i) r(i, j)$. Для нити T_{jm} выделяется вычислительный модуль с номером $w = w + 1$, $w \in N$.

Таким образом, нить T_{jm} будет выполняться на вычислительном модуле с учетом информационных связей с другими нитями. Связь S_m^i исключается из дальнейшего планирования. Выполнение остальных нитей планируется аналогичным образом.

Пример. Рассмотрим планирование распределения программных модулей по узлам вычислительной системы, представленной D_n -графом [1] типа «Циркулянт» $G(11, 1, 2, 5)$ для граф-схемы алгоритма решаемой задачи (рис. 1). Эта структура должна удовлетворять следующим соотношениям:

$$D_n = G(N, q_1, \dots, q_n),$$

$$0 < q_1 < \dots < q_n \leq (N + 1)/2,$$

где N — число вершин в графе (вершины нумеруются от 0 до $N - 1$); q_1, \dots, q_n — такие числа, образующие граф, что множество $\{N, q_1, \dots, q_n\}$ имеет наибольший общий делитель, равный единице.

Вершина i соединяется ребрами с вершинами, для которых справедлива следующая система модулярных соотношений:

$$(i \pm q_1) \bmod N,$$

.....

$$(i \pm q_n) \bmod N.$$

На граф-схеме, приведенной на рис. 1, указаны время решения соответствующего программного модуля и время передачи информации, заданные в относительных единицах.

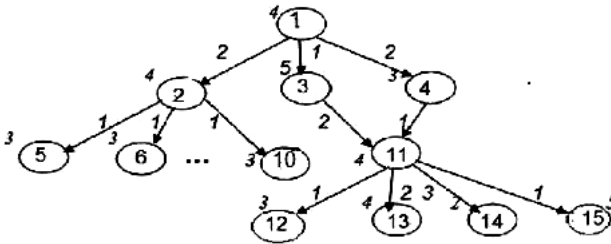


Рис. 1. Граф-схема задачи с указанием веса каждой вершины (времени решения соответствующего программного модуля) и веса каждой дуги (времени передачи информации)

В модифицированной матрице следования (рис. 2) в 16-м столбце указаны веса p_j соответствующих операторов, которые учитывают как время решения программного модуля, так и время передачи информации из рассматриваемого модуля согласно соотношениям (2) и (3). По указанному алгоритму вычисляются ранние сроки окончания выполнения операторов. Эти сроки равны соответственно: $t_{1,1} = 6$, $t_{1,2} = 5$, $t_{1,3} = 7$, $t_{1,4} = 5$, $t_{1,5} = 3$, $t_{1,6} = 3$, $t_{1,7} = 3$, $t_{1,8} = 3$, $t_{1,9} = 3$, $t_{1,10} = 3$, $t_{1,11} = 7$, $t_{1,12} = 3$, $t_{1,13} = 4$, $t_{1,14} = 2$, $t_{1,15} = 5$. Используя эти данные и матрицу следования (см. рис. 2), строим диаграмму размещения программных модулей по процессорам, как показано на рис. 3. Затем программные модули, расположенные на одном процессоре, объединяем в нити.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1																6
2	1															5
3	1															7
4	1															5
5		1														3
6		1														3
7		1														3
8		1														3
9		1														3
10		1														3
11			1	1												7
12											1					3
13											1					4
14											1					2
15											1					5

Рис. 2. Модифицированная матрица следования S для граф-схемы, представленной на рис. 1

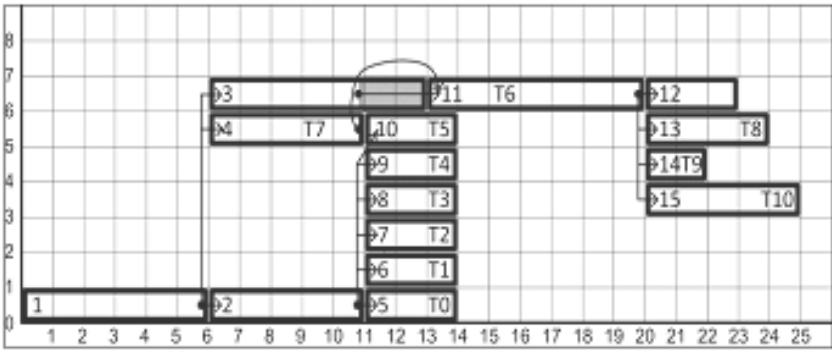


Рис. 3. Диаграмма размещения нитей на поле процессоров вычислительной системы типа «Циркулянт» $G(11, 1, 2, 5)$

Для реализации граф-схемы, представленной на рис. 1, как следует из диаграммы рис. 3, требуется семь процессоров без учета построения связей между ними. Образовались следующие нити:

$$T0 = \{1, 2, 5\}, T1 = \{6\}, T2 = \{7\}, T3 = \{8\}, T4 = \{9\}, T5 = \{10\},$$

$$T6 = \{3, 11, 12\}, T7 = \{4\}, T8 = \{13\}, T9 = \{14\}, T10 = \{15\}.$$

Элементами множеств, образующих нити, являются программные блоки рассматриваемой граф-схемы. Блок 1 нити $T0$ связан с нитями $T6$ (блок 3) и $T7$ (блок 4) в момент времени $t = 6$. Заштрихованная часть блока 3 обозначает простой процессора 7. Этот простой связан с необходимостью одновременного прихода данных в блок 11. Характер построенной диаграммы позволяет объединить нити $T3$ и $T10$, $T4$ и $T9$, а также $T7$, $T5$ и $T8$. Простой вычислительных модулей 4, 5, 6 определяются структурой решаемой задачи и не могут быть исключены.

Рассмотрим структуру вычислительной системы типа «Циркулянт» $G(11, 1, 2, 5)$. Матрица дистанций для этой структуры, вычисленная по формуле (1), представлена на рис. 4.

Покажем размещение нитей на вычислительной системе типа «Циркулянт» $G(11, 1, 2, 5)$, используя рис. 5. Вычислительные модули, загруженные нитями, размещены на графе в виде закрашенных вершин.

Вначале размещаются нити цикла алгоритма $T0 \dots T6$. Затем делается попытка разместить все остальные нити. На отметке времени $t = 6$ (см. рис. 3) осуществляется обмен данными между нитями $T0$ и $T6$, $T7$: вначале между нулевым и шестым вычислительными модулями с помощью прямой связи (см. рис. 5), затем между нулевым и седьмым модулями через транзитный модуль 9. При значении времени $t = 11$ (см. рис. 3) необходимо обеспечить связь между нитями $T0$ и $T1$, $T2$, $T3$, $T4$, $T5$. Соединение нитей $T0$, $T1$, $T2$, $T5$, $T6$ осуществ-

ляется за счет прямых связей (см. рис. 5). Нить T0 подключается к нити T3 через транзитный процессор 9 повторно, так как к моменту времени $t = 11$ он не загружен. Аналогично нить T0 связывается с нитью T4 через транзитный процессор 10. Нить T7 связана с нитью T5 прямой связью. Связь нити T6 с нитями T8, T9, T10 в момент времени $t = 20$ (см. рис. 3) осуществляется следующим образом: между нитями T6 и T8 есть прямая связь (см. рис. 5); нить T6 соединяется с нитью T9 с помощью транзитного процессора 7, а с нитью T10 с помощью транзитного процессора 8.

	0	1	2	3	4	5	6	7	8	9	10
0	0	1	1	2	2	1	1	2	2	1	1
1	1	0	1	1	2	2	1	1	2	2	1
2	1	1	0	1	1	2	2	1	1	2	2
3	2	1	1	0	1	1	2	2	1	1	2
4	2	2	1	1	0	1	1	2	2	1	1
5	1	2	2	1	1	0	1	1	2	2	1
6	1	1	2	2	1	1	0	1	1	2	2
7	2	1	1	2	2	1	1	0	1	1	2
8	2	2	1	1	2	2	1	1	0	1	1
9	1	2	2	1	1	2	2	1	1	0	1
10	1	1	2	2	1	1	2	2	1	1	0

Рис. 4. Матрица дистанций вычислительной системы типа «Циркулянт» $G(11, 1, 2, 5)$

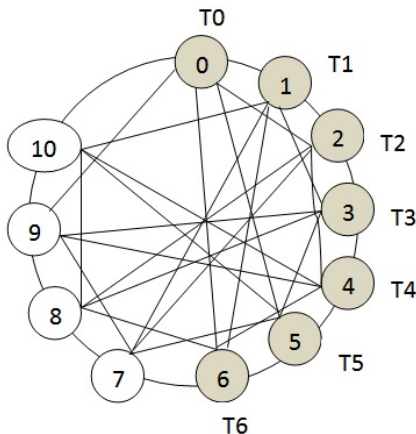


Рис. 5. Распределение нитей по процессорам вычислительной системы типа «Циркулянт»

Таким образом, в данной статье осуществлено планирование распределения программных модулей решаемой задачи по узлам вычис-

лительной системы с разделяемой памятью. Приведены формулы для расчетов и рассмотрен пример планирования распределения программных модулей по узлам вычислительной системы структуры типа «Циркулянт».

ЛИТЕРАТУРА

- [1] Хорошевский В.Г. *Архитектура вычислительных систем*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2005, 511 с.
- [2] Корнеев В.В. *Параллельные вычислительные системы*. Москва, Гелиос АРВ, 2004, 487 с.
- [3] Водяхо А.И., Горнец Н.Н., Пузанков Д.В. *Высокопроизводительные системы обработки данных*. Москва, Высшая школа, 1997, 150 с.
- [4] Воеводин В.В., Воеводин Вл.В. *Параллельные вычисления*. Санкт-Петербург, БХВ-Петербург, 2002, 600 с.
- [5] Руденко Ю.М., Мельдианов П.В. Алгоритм преобразования последовательного алгоритма в параллельный. *Информатика и системы управления в XXI веке. Сб. тр. молодых ученых, аспирантов, студентов*. Москва, 2007, вып. 5, с. 127–131.
- [6] Руденко Ю.М. Представление параллельных алгоритмов в виде граф-схем. *Аэрокосмические технологии. Науч. материалы междунар. науч.-техн. конф. МНТК*. Москва, 2009, с. 179–181.
- [7] Павлов А.Е., Руденко Ю.М. Представление трансляционно-циклического обмена в виде граф-схемы. *Современные информационные технологии. Сб. тр. кафедры «Компьютерные системы и сети»*. Москва, НИИ РЛ МГТУ им. Н.Э. Баумана, 2011, с. 125–129.
- [8] Музина Л.В., Руденко Ю.М. Учет времен задержек на вычислительных модулях при реализации граф-схем. *Современные информационные технологии. Сб. тр. кафедры «Компьютерные системы и сети»*. Москва, НИИ РЛ МГТУ им. Н.Э. Баумана, 2011, с. 133–137.
- [9] Руденко Ю.М. Распределение программных модулей по узлам вычислительной сети с общим полем памяти для граф-схем параллельных алгоритмов. *Наука и образование: электрон. науч.-техн. издание*. 2011. № 10. URL: <http://tehnomag.edu.ru/doc/235842.html>. DOI: 77-30569/235842 (дата обращения 15 марта 2013).
- [10] Руденко Ю.М., Волкова Е.А. *Вычислительные системы*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2010, 211 с.
- [11] Барский А.Б. *Параллельные процессы в вычислительных системах*. Москва, Радио и связь, 1990, 256 с.

Статья поступила в редакцию 28.06.2013

Ссылку на эту статью просим оформлять следующим образом:

Руденко Ю.М. Планирование распределения программных модулей по процессорам вычислительной системы. *Инженерный журнал: наука и инновации*, 2013, вып. 11. URL: <http://engjournal.ru/catalog/it/network/1010.html>

Руденко Юрий Михайлович родился в 1941 г., окончил Харьковский политехнический институт им. В.И. Ленина в 1964 г. Канд. техн. наук, доцент кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Автор 80 научных работ. Область научных интересов: параллельные вычислительные системы. e-mail: Rudenko-yuriy@inbox.ru