

Синтез многопроцессорных вычислительных структур на основе сетевых моделей

В.П. Кулагин, Н.Д. Муравьев

Российский технологический университет - МИРЭА, г. Москва

Аннотация — Описан пример синтеза многопроцессорной вычислительной структуры. В основе алгоритмов синтеза лежат сетевые модели и тензорная методология. Описаны основные этапы синтеза вычислительных структур, которые включают декомпозицию исходной модели, вычисление тензора преобразования, формирование программы синтеза новой структуры, а также синтез самой вычислительной структуры. Проанализированы результаты синтеза с точки зрения эффективности загрузки процессорных элементов.

Ключевые слова — матричные структуры, сетевые модели, сети Петри, тензорное преобразование, синтез.

I. ВВЕДЕНИЕ

Стремительный рост объемов обрабатываемой информации, повышение сложности решаемых задач требуют создания методов и алгоритмов построения многопроцессорных вычислительных структур, способных эффективно решать данные задачи. Обработка изображений, видео- и аудио-поток, алгоритмы поиска, сортировки и др. могут быть ускорены за счет распараллеливания процессов обработки данных.

Одной из проблем использования параллельных вычислений является синхронизация данных. Ошибки, возникающие при использовании многопоточной или многопроцессорной обработки, сложно обнаружить. В свою очередь синхронизация данных между параллельными процессами порождает проблему неэффективного использования процессорных элементов, что в свою очередь влияет на производительность всей параллельной системы.

Рассмотрим метод синтеза многопроцессорных вычислительных структур, обладающих различными характеристиками загрузки процессорных элементов.

II. МАТРИЧНАЯ СТРУКТУРА

Для достижения высокой производительности систем обработки данных, можно использовать процессорные матрицы – специализированные вычислительные структуры, состоящие из множества относительно простых устройств (процессорных элементов) [1]. Принцип компьютерных вычислений данных процессоров – SIMD (одиночный поток команд, множественный поток данных). Данный принцип используется во множестве вычислительных

структур. Здесь в качестве сетевых моделей используются сети Петри [2]. В настоящее время сети Петри вносят свой вклад в проекты, связанные с созданием программного обеспечения. Примером может служить удачное применение компонентов сетей Петри в UML 2 (унифицированный язык моделирования), где наиболее заметно использование диаграмм деятельности и их автоматическая трансляция в сети Петри и обратно [6].

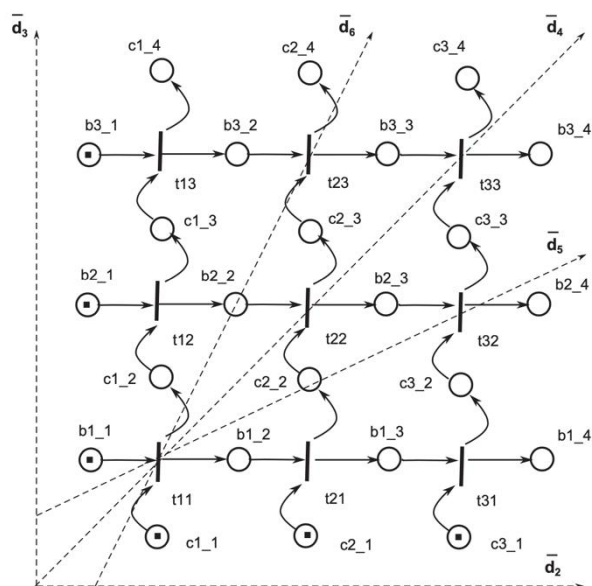


Рис. 1. СП-модель N_1 двумерной параллельной структуры умножения матрицы $A(3 \times 3)$ на вектор $B(3)$

Пусть нам необходимо построить многопроцессорную структуру, умножающую вектор на матрицу. Имеется матрица $A = [a_{ij}]$ и вектор $B = [b_j]$ размерности J , необходимо найти вектор $C = AB$.

Матричное умножение может быть осуществлено за J шагов:

$$c_i^{(k)} = c_i^{(k-1)} + a_i^{(k)} b_j^{(k)},$$

при этом

$$a_i^{(k)} = a_{ik}, b_j^{(k)} = b_k, k = 1, 2, \dots, J; c_i^{(0)} = 0.$$

На рис. 1 представлена СП, моделирующая вычислительную структуру, умножающая матрицу на вектор. При этом переходы $t_{11}, t_{12}, t_{13}, t_{21}, t_{22}, t_{23}, t_{31}, t_{32}, t_{33}$ моделируют процессорные элементы (ПЭ), которые выполняют простейшие операции сложения и умножения. Кроме этого ПЭ содержит регистр для хранения a_{ij} элемента матрицы А, а также регистр для хранения частичной суммы c^n . Структура процессорного элемента представлена на рис. 2.

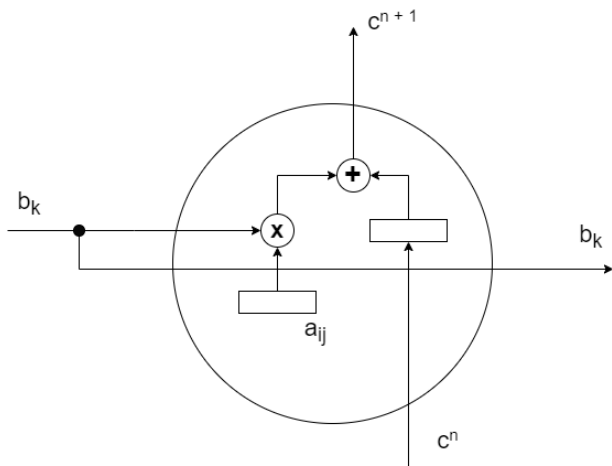


Рис. 2. Структура процессорного элемента (ПЭ)

Для анализа работы модели вычислительной структуры введем понятие такта работы, который определим как интервал времени t , в течение которого происходит передача данных между соседними ПЭ, а также выполнение операций сложения и умножения в самом ПЭ.

Стоит обратить внимание, что в используемой параллельной структуре умножения матрицы на вектор используется асинхронное управление - процессорный элемент приступает к обработке данных только в том случае, когда на его входах сформирован полный набор входных данных (присутствует элемент вектора В, а также сформирована частичная сумма c^n).

Можно заметить, что построенная модель выполнит операцию перемножения заданных матрицы и вектора за 5 тактов. На рис. 3 показана временная диаграмма работы ПЭ. Недостатком данной структуры является то, что каждый ПЭ работает лишь в течении одного такта, что говорит о неэффективной загрузке ПЭ.

Можно ли построить другую вычислительную структуру, умножающую матрицу на вектор и сохраняющую такую же производительность, но в которой ПЭ использовались бы более эффективно? Рассмотрим подход к синтезу одной из таких структур.

III. СИНТЕЗ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР

A. Декомпозиция СП-модели

Синтез новой вычислительной структуры, на основе построенной СП-модели начинается с декомпозиции модели N_1 на множество линейно-

базовых фрагментов (ЛБФ) и с последующим получением примитивной системы [3].

ПЭ/Такты	1	2	3	4	5
t11	■				
t12		■			
t13			■		
t21		■			
t22			■		
t23				■	
t31			■		
t32				■	
t33					■

Рис. 3. Временная диаграмма работы ПЭ в матричной структуре

Процесс разбиения на ЛБФ представляет собой нахождение в заданной СП-модели линейных или линейно-циклических фрагментов. Для СП-модели N_1 декомпозиция на ЛБФ будет иметь вид, представленный на рис. 4. Здесь искусственно добавлены эквивалентные позиции с целью выравнивая размерностей системы, содержащей ЛБФ, и примитивной системы, которая представлена на рис. 5.

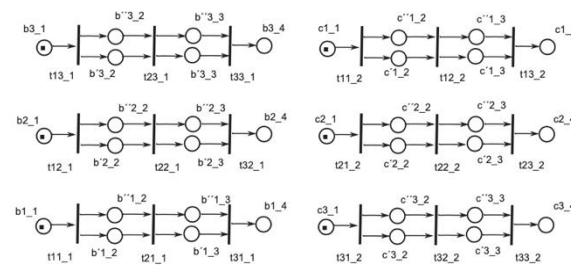


Рис. 4. Линейно-базовые фрагменты СП-модели N_1

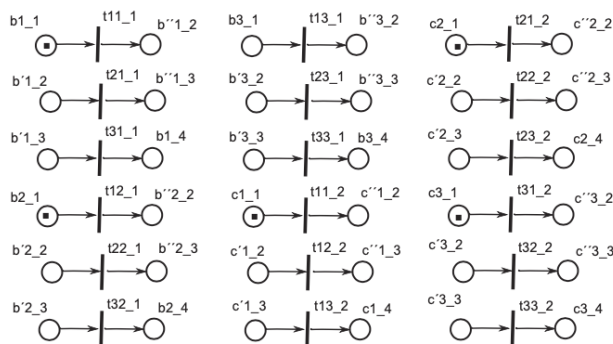


Рис. 5. Примитивная система N_{PR} СП-модели N_1

Следующим этапом синтеза новой СП-модели является построения тензора преобразования.

В. Тензор преобразования

Если мы предположим, что система с ЛБФ и примитивная система, являются представлением одной и той же СП-модели, но выраженной в разных системах координат, то логично найти данный оператор, позволяющий перейти от одной системы координат к другой. Данный оператор называется тензором преобразования (ТП). Под преобразованием системы координат подразумевается переход от одного способа соединения элементарных (примитивных) сетей к другому. В процессе преобразования системы координат выполняются операции разрыва и соединения связей между переходами и позициями СП-структуры [3].

Для вычисления ТП воспользуемся выражением:

$$D_{\beta'}^{\gamma'} = C_{\beta'\gamma'}^{\beta\gamma} \times D_{\beta}^{\gamma} \quad (1)$$

где $D_{\beta'}^{\gamma'}$ – матрица инцидентности системы, содержащей ЛБФ (рис. 8), D_{β}^{γ} – матрица инцидентности примитивной системы (рис. 9), $C_{\beta'\gamma'}^{\beta\gamma}$ – тензор преобразования. Один из возможных ТП, полученных из решения тензорного уравнения (1), представлен на рис. 10.

	t11	t21	t31	t12	t22	t32	t13	t23	t33	t11'	t12'	t13'	t21'	t22'	t23'	t31'	t32'	t33'	
b1_1	-1																		
b1_2	1	-1																	
b1_2	1	-1																	
b1_3		1	-1																
b1_3		1	-1																
b1_4			1																
b2_1				-1															
b2_2				1	-1														
b2_2				1	-1														
b2_3					1	-1													
b2_3					1	-1													
b2_4						1													
b3_1							-1												
b3_2							1	-1											
b3_2							1	-1											
b3_3								1	-1										
b3_3								1	-1										
b3_4									1										
c1_1										-1									
c1_2										1	-1								
c1_2										1	-1								
c1_3											1	-1							
c1_3												1	-1						
c1_4													1						
c2_1																	-1		
c2_2																	1		
c2_2																		-1	
c2_3																		1	
c2_3																			-1
c2_4																			1
c3_1																			-1
c3_2																			1
c3_2																			-1
c3_3																			1
c3_3																			-1
c3_4																			1

Рис. 8. Матрица инцидентности ЛБФ $D_{\beta'}^{\gamma'}$

С. Программа синтеза

Зададим программу синтеза новой СП-модели, которая включает операции объединения позиций и переходов примитивной системы.

Программа синтеза – это некоторая последовательность $V(V_T + V_P)$ размерностью $m + n$, где V_T – последовательность переходов размерностью

m , V_P – последовательность позиций размерностью n , m – число переходов, а n – число позиций в примитивной системе. Будем считать, что элемент последовательности V равен нулю, если соответствующая вершина примитивной системы не принадлежит подмножеству объединяемых вершин и элемент последовательности V больше нуля (имеет положительное значение), если соответствующая вершина принадлежит подмножеству объединяемых вершин [1].

	t11	t21	t31	t12	t22	t32	t13	t23	t33	t11'	t12'	t13'	t21'	t22'	t23'	t31'	t32'	t33'	
b1_1	-1																		
b1_2	1																		
b1_2		-1																	
b1_3		1																	
b1_3			-1																
b1_4				1															
b2_1					-1														
b2_2					1														
b2_2						-1													
b2_3						1													
b2_3							-1												
b2_4							1												
b3_1								-1											
b3_2								1											
b3_2									-1										
b3_3									1										
b3_3										-1									
b3_4										1									
c1_1											-1								
c1_2											1								
c1_2												-1							
c1_3												1							
c1_3													-1						
c1_4													1						
c2_1																		-1	
c2_2																		1	
c2_2																			-1
c2_3																			1
c2_3																			-1
c2_4																			1
c3_1																			-1
c3_2																			1
c3_2																			-1
c3_3																			1
c3_3																			-1
c3_4																			1

Рис. 9. Матрица примитивной системы D_{β}^{γ}

1	2	3	4	5	6	7	...	31	32	33	34	35	36
2	1												
3		1	1										
4		1	1										
5				1	1								
6				1	1								
7						1							
...							...						
31								1					
32									1	1			
33									1	1			
34											1	1	
35											1	1	
36													1

Рис. 10. Тензор преобразования $C_{\beta'\gamma'}^{\beta\gamma}$

Количество программ синтеза зависит от количества переходов и позиций в сети Петри. Также при составлении множества векторов V , могут встречаться вектора, генерирующие эквивалентные программы. В работе [1] описаны свойства, которые

позволяют определять эквивалентные программы синтеза.

Пусть мы имеем следующую программу синтеза искомой СП-модели:

$$\begin{aligned}
 t1 &= t11 + t12 + t13 \\
 t2 &= t21 + t22 + t23 \\
 t3 &= t31 + t32 + t33 \\
 c1 &= c1_1 + c1_4 \\
 c2 &= c2_1 + c2_4 \\
 c3 &= c3_1 + c3_4.
 \end{aligned}$$

Изменение матрицы инцидентности примитивной системы путем объединения позиций и переходов (сложение соответствующих строк и столбцов), а затем, используя ТП, переход к другой системе координат, позволяет получить матрицу инцидентности новой искомой СП-модели, по которой можно построить графовое представление. Данная СП-модель, в которой уже удалены все эквивалентные позиции, представлена на рис.11.

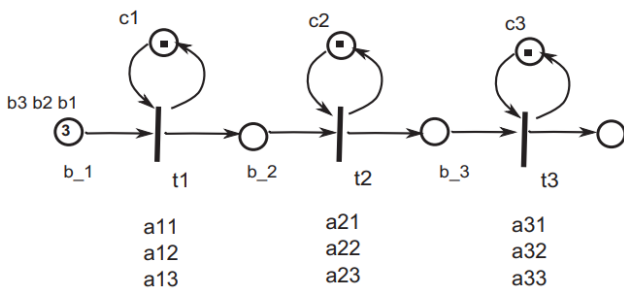


Рис. 11. Синтезированная СП-модель N₂

D. Анализ синтезированной вычислительной структуры

Отличие исходной модели N₁ (рис. 1) от СП-модели N₂, представленной на рис.11, состоит как в количестве используемых ПЭ, так и в структуре самого ПЭ, которая представлена на рис.12.

Вводя метрическое пространство и оценивая характеристики получаемых СП-моделей в соответствие с заданными метриками, можно реализовать процедуры синтеза вычислительных структур с наперед заданными свойствами.

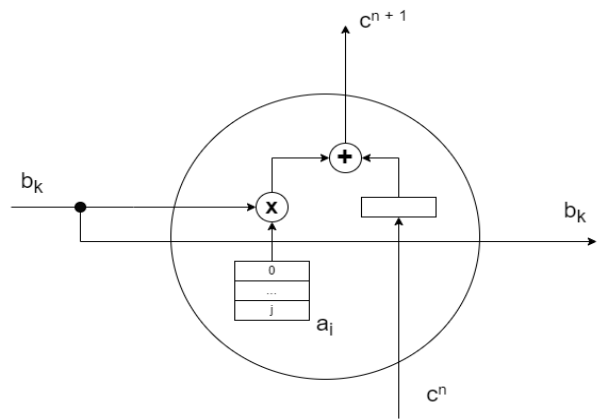


Рис. 12. Структура процессорного элемента синтезированной структуры N₂

ПЭ/Такты	1	2	3	4	5
t1					
t2					
t3					

Рис. 13. Временная диаграмма работы ПЭ синтезированной матричной структуре N₂

IV. ЗАКЛЮЧЕНИЕ

В данной работе продемонстрирован подход к синтезу СП-моделей, отличающихся различными характеристиками. Данный подход позволяет получить все множество возможных вычислительных структур, выполняющих заданную функцию, и может быть реализован программно.

ЛИТЕРАТУРА

- [1] Кулагин В. П. Проектирование матричных вычислительных структур с использованием сетей Петри / В. П. Кулагин, Е. С. Малых // Информационные технологии. – 2019. – Т. 25. – № 5. – С. 271-283. – DOI 10.17587/it.25.271-283.
- [2] Питерсон Дж. Теория сетей Петри и моделирование систем. М.: Мир, 1984. 264 с.
- [3] Кулагин В. П. Тензорные методы исследования структур сетей Петри / В. П. Кулагин // Информационные технологии. – 2015. – Т. 21. – № 2. – С. 83-94.
- [4] Кулагин В. П. Методы построения тензоров преобразования для сетевых моделей сложных систем / В. П. Кулагин // Информатизация образования и науки. – 2015. – № 4(28). – С. 133-147. – EDN UKRKRF.
- [5] Кулагин В. П. Структуры сетей Петри // Информационные технологии. 1997. № 4. С. 17–22.
- [6] Мараховский В. Б., Розенблюм Л. Я., Яковлев А. В. Моделирование параллельных процессов. Сети Петри. – СПб.: Профессиональная литература, 2014. – 400 с.: ил.

Synthesis of Multiprocessor Computing Structures Based on Network Models

V.P. Kulagin, N.D. Muravev

Federal State Budget Educational Institution of Higher Education «MIREA - Russian Technological University», Moscow

Abstract — The article demonstrates an approach to modeling multiprocessor computing structures based on network models, and also describes a method for synthesizing new structures based on tensor transformation. Petri nets (PN) are used as network models, as they are intended for modeling parallel processes. As a computational structure, a processor matrix is used, which consists of the simplest processor elements that support the simplest operations of addition and multiplication. The stages of synthesis of computational structures are also demonstrated: decomposition of the original model into linear-basic fragments and a primitive system, calculation of the transformation tensor, definition of synthesis programs, synthesis of a new structure, the external behavior of which does not differ from the original one. A comparative analysis of the synthesized PN-model with the original one was carried out, in which it was shown that the synthesized structure has a more optimal use of computing resources by reducing the idle time of processor elements and their number.

Keywords — matrix structures, network models, Petri nets, tensor transformation, synthesis.

REFERENCES

- [1] Kulagin V.P. *Proyektirovaniye matrichnykh vychislitel'nykh struktur s ispol'zovaniyem setey Petri (Design of matrix computing structures using Petri nets)* / V.P. Kulagin, E.S. Malykh // *Informatsionnyye tekhnologii*. – 2019. – T. 25. – № 5. – С. 271-283. – DOI 10.17587/it.25.271-283.
- [2] Piterson Dzh. *Teoriya setey Petri i modelirovaniye sistem (Petri Net Theory and System Modeling)*. M.: Mir, 1984. 264 s.
- [3] Kulagin V. P. *Tenzornyye metody issledovaniya struktur setey Petri (Tensor methods of research structures of Petri nets)* / V. P. Kulagin // *Informatsionnyye tekhnologii*. – 2015. – T. 21. – № 2. – S. 83-94.
- [4] Kulagin V. P. *Metody postroyeniya tenzоров preobrazovaniya dlya setevykh modeley slozhnykh sistem (Methods for constructing tensor transformation for network models of complex systems)* / V. P. Kulagin // *Informatizatsiya obrazovaniya i nauki*. – 2015. – № 4(28). – S. 133-147.
- [5] Kulagin V. P. *Struktury setey Petri (Structures of Petri nets)* // *Informatsionnyye tekhnologii*. 1997. № 4. С. 17–22.
- [6] Marakhovskiy V. B., Rozenblyum L. YA., Yakovlev A. V. *Modelirovaniye parallel'nykh protsessov. Seti Petri. (Modeling of parallel processes. Petri nets.)* – Spb.: Professional'naya literatura, 2014. – 400 s.: il.