

Обобщение одного алгоритма параллельного умножения матриц в алгебре многомерных матриц

В. И. Мунерман*, Д. В. Мунерман

ФГБОУ ВО «Смоленский государственный университет», г. Смоленск, Российская Федерация

Адрес: 214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4

*vimoon@gmail.com

Аннотация

Многомерные матрицы – это мощный инструмент для решения различных фундаментальных научных проблем и прикладных научно-технических задач. В качестве примеров применения многомерных матриц в различных областях можно привести как работу создателя теории многомерных матриц Н.П. Соколова так и другие работы в том числе и выполненные авторами. Проблема параллельной реализации самой сложной операции алгебры многомерных матриц – (λ, μ) -свернутого произведения относится к числу наиболее важных. Не случайно проблеме параллельного умножения обычных двумерных матриц посвящено много публикаций. В этих работах рассматриваются проблемы выбора способа распределения элементов матриц между процессорами и разработки архитектур программно-аппаратных комплексов для эффективной реализации этой операции. Для параллельного умножения обычных матриц, как правило, используются два вида алгоритмов: ленточные, реализующие поэлементное умножение матриц и блочные, основанные на методе Фробениуса. Многие авторы, проводившие анализ эффективности параллельного умножения матриц, отдают предпочтение блочным алгоритмам, утверждая, что последние обладают высокой степенью масштабируемости. Учитывая тот факт, что масштабируемость – это наиболее важное свойство параллельных алгоритмов и вычислительных комплексов их реализующих, в дальнейшем будет рассматриваться параллельный алгоритм блочного умножения многомерных матриц. В работе рассматривается обобщение алгоритма Кэннона на (λ, μ) -свернутое произведение многомерных матриц. Доказано, что многомерные матрицы-операнды могут быть представлены как совокупность сечений по скоттовым индексам, произведения которых позволяют получить требуемую матрицу-результат. Для разбиения этих сечений на блоки и пересчета индексов блоков в ходе выполнения алгоритма предложен метод, основанный на использовании специфических систем счисления, и определен метод вычисления оснований этих систем счисления. Процесс выполнения обобщенного алгоритма Кэннона проиллюстрирован таблицами распределения блоков на всех этапах выполнения алгоритма. Предложенный метод обобщения может быть распространен и на другие алгоритмы блочного умножения матриц.

Ключевые слова: ориентированные графы, циклы в графе, многомерные матрицы, базы данных, параллельное программирование

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Мунерман В. И., Мунерман Д. В. Обобщение одного алгоритма параллельного умножения матриц в алгебре многомерных матриц // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 3. С. 566-577. doi: <https://doi.org/10.25559/SITITO.18.202203.566-577>

© Мунерман В. И., Мунерман Д. В., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Some Method for Constructing Cycles in a Graph

V. I. Munerman*, D. V. Munerman

Smolensk State University, Smolensk, Russian Federation

Address: 4 Przhevalsky St., Smolensk 214000, Russian Federation

*vymoon@gmail.com

Abstract

Multidimensional matrices are a powerful tool for solving various fundamental scientific problems and applied scientific and technical problems. As examples of the use of multidimensional matrices in various fields, one can cite the work of the creator of the theory of multidimensional matrices N.P. Sokolov and other works, including those performed by the authors. The problem of parallel implementation of the most complex operation of the algebra of multidimensional matrices, the (λ, μ) -convolution product, is one of the most important. It is no coincidence that many publications are devoted to the problem of parallel multiplication of ordinary two-dimensional matrices. In these works, the problems of choosing a method for distributing matrix elements between processors and developing architectures of software and hardware systems for the effective implementation of this operation are considered. For parallel multiplication of ordinary matrices, as a rule, two types of algorithms are used: band algorithms that implement element-wise matrix multiplication and block algorithms based on the Frobenius method. Many authors who have analyzed the efficiency of parallel matrix multiplication prefer block algorithms, arguing that the latter have a high degree of scalability. Taking into account the fact that scalability is the most important property of parallel algorithms and computing systems that implement them, in the future we will consider a parallel algorithm for block multiplication of multidimensional matrices. The paper considers a generalization of the Cannon algorithm to the (λ, μ) -convolution product of multidimensional matrices. It is proved that multidimensional matrices-operands can be represented as a set of sections by Scottish indices, the products of which allow one to obtain the required matrix-result. To split these sections into blocks and recalculate block indices during the execution of the algorithm, a method based on the use of specific number systems is proposed, and a method for calculating the bases of these number systems is determined. The process of executing the generalized Cannon algorithm is illustrated by block distribution tables at all stages of the algorithm execution. The proposed generalization method can be extended to other block matrix multiplication algorithms.

Keywords: directed graphs, cycles in a graph, multidimensional matrices, databases, parallel programming

Conflict of interests: The authors declare no conflict of interest.

For citation: Munerman V.I., Munerman D.V. Some Method for Constructing Cycles in a Graph. *Modern Information Technologies and IT-Education*. 2022;18(3):566-577. doi: <https://doi.org/10.25559/SITITO.18.202203.566-577>



Введение

Многомерные матрицы – это мощный инструмент для решения различных фундаментальных научных проблем и прикладных научно-технических задач. В качестве примеров применения многомерных матриц в различных областях можно привести как работу создателя теории многомерных матриц Н.П. Соколова [1], так и работы [2-7].

Проблема параллельной реализации самой сложной операции алгебры многомерных матриц – (λ, μ) -свернутого произведения относится к числу наиболее важных. Не случайно проблеме параллельного умножения обычных двумерных матриц посвящено много публикаций. В этих работах рассматриваются проблемы выбора способа распределения элементов матриц между процессорами и разработки архитектур программно-аппаратных комплексов для эффективной реализации этой операции¹ [8-23].

Для параллельного умножения обычных матриц, как правило, используются два вида алгоритмов [8]: ленточные, реализующие поэлементное умножение матриц и блочные, основанные на методе Фробениуса². Многие авторы, проводившие анализ эффективности параллельного умножения матриц [17-24], отдают предпочтение блочным алгоритмам, утверждая, что последние обладают высокой степенью масштабируемости. Учитывая тот факт, что масштабируемость – это наиболее важное свойство параллельных алгоритмов и вычислительных комплексов их реализующих, в дальнейшем будет рассматриваться параллельный алгоритм блочного умножения многомерных матриц.

Метод Фробениуса естественно обобщается на многомерные матрицы. Известны различные реализации параллельного умножения матриц: алгоритмы Фокса и Кэннона [8], универсальный алгоритм для параллельных вычислительных комплексов с распределенной памятью PUMMA [19], масштабируемый универсальный алгоритм SUMMA [20].

В работе рассматривается обобщение алгоритма Кэннона на (λ, μ) -свернутое произведение многомерных матриц. Предложенный метод обобщения может быть распространен и на другие алгоритмы блочного умножения матриц.

Умножение многомерных матриц

Определение операции умножения многомерных матриц неоднократно приведено во многих работах³ [1, 3, 25]. Но для лучшего восприятия предложенного в статье метода целесообразно его повторить здесь.

Определение. Пусть даны p -мерная матрица $A = \|a_{i_1 \dots i_p}\|$ и q -мерная матрица $B = \|b_{i_1 \dots i_q}\|$. Можно разбить совокупности индексов i_1, \dots, i_p и i_1, \dots, i_q на четыре группы, содержащие соответственно k, λ, μ и v индексов ($k, \lambda, \mu, v \geq 0$). Причем $k + \lambda + \mu = p$, а $\lambda + \mu + v = q$. Разбиение порождает четыре группы индексов: $l = (l_1, \dots, l_k)$, $s = (s_1, \dots, s_\lambda)$, $c = (c_1, \dots, c_\mu)$ и $m = (m_1, \dots, m_v)$. Индексы разбиений s и c (они называются скоттовыми и кэлиевыми индексами) принадлежат обеим матрицам. Тогда ма-

трицы A и B можно представить в виде $A = \|a_{lsc}\|$ и $B = \|b_{scm}\|$. Очевидно, что если индексу s_ν (c_ν) соответствуют индекс i_α в матрице A и индекс i_β в матрице B , то $n_{i_\alpha} = n_{i_\beta}$.

Матрица $C = \|c_{lsm}\|$, элементы которой вычисляются по формуле $c_{lsm} = \sum a_{lsc} \times b_{scm}$, называется (λ, μ) -свернутым произведением матриц A и B и обозначается ${}^{\lambda, \mu}(A \times B)$.

Особенность, отличающая его от умножения обычных (плоских) матриц и тензоров, заключается в наличии скоттовых индексов, которые присутствуют в обеих матрицах-операндах и в матрице-результате. Эта особенность существенно сказывается на алгоритме умножения многомерных матриц. При отсутствии скоттовых индексов этот алгоритм мало отличается от алгоритма умножения плоских матриц. Далее рассматривается алгоритм параллельного умножения многомерных матриц при наличии скоттовых и кэлиевых индексов ($\lambda > 0, \mu > 0$). Для предложенного метода обобщения алгоритма Кэннона, как и в самом алгоритме, предполагается, что количества значений всех индексов во всех разбиениях l, s, c, m матриц-сомножителей A и B одинаковы, и равны некоторому числу n .

Из определения следует, что матрицы-операнды $A = \|a_{i_1 \dots i_p}\|$, $B = \|b_{i_1 \dots i_q}\|$ и матрицу-результат C можно рассматривать как условно трехмерные матрицы $A = \|a_{lsc}\|$, $B = \|b_{scm}\|$ и $C = \|c_{lsm}\|$, рассматривая индексы разбиений l, s, c и m как мультииндексы.

Разбиение многомерных матриц на блоки по скоттовым индексам

Пусть даны: p -мерная матрица $A (A_{lsc})$, q -мерная матрица $B (B_{scm})$ и r -мерная матрица $C (C_{lsm})$ ($r = p + q - \lambda - 2\mu$) такая, что $C = {}^{\lambda, \mu}(A \times B)$. Если зафиксировать по одному значению каждого из скоттовых индексов, одинаковому во всех трех матрицах ($s_1^0, \dots, s_\lambda^0$), то будут получены: $(p - \lambda)$ -кратное сечение матрицы A , $(q - \lambda)$ -кратное сечение матрицы B и $(r - \lambda)$ -кратное сечение матрицы C . Все эти сечения будут иметь ориентацию (s_1, \dots, s_λ) . Пусть $s_1^1, \dots, s_\lambda^1$ и $s_1^2, \dots, s_\lambda^2$ фиксированные наборы значений скоттовых индексов многомерных матриц A, B и C , отличающиеся друг от друга, по крайней мере, одним значением. Этим наборам значений скоттовых индексов соответствуют два различных сечения каждой из матриц. Элементы соответствующих сечений матрицы C вычисляются по следующим формулам:

$$c_{l_1 \dots l_k s_1^1 \dots s_\lambda^1 m_1 \dots m_v} = \sum_{c_1, \dots, c_\mu=0}^n a_{l_1 \dots l_k s_1^1 \dots s_\lambda^1 c_1 \dots c_\mu} \cdot b_{s_1^1 \dots s_\lambda^1 c_1 \dots c_\mu m_1 \dots m_v}, \quad (1)$$

$$c_{l_1 \dots l_k s_1^2 \dots s_\lambda^2 m_1 \dots m_v} = \sum_{c_1, \dots, c_\mu=0}^n a_{l_1 \dots l_k s_1^2 \dots s_\lambda^2 c_1 \dots c_\mu} \cdot b_{s_1^2 \dots s_\lambda^2 c_1 \dots c_\mu m_1 \dots m_v}. \quad (2)$$

Из приведенных формул вытекает следующее утверждение.

Утверждение. Результат (λ, μ) -свернутого произведения двух многомерных матриц A и B размерностей p и q , при наличии скоттовых индексов ($\lambda > 0$), есть r -мерная матрица C ($r = p + q - \lambda - 2\mu$), составленная из $(r - \lambda)$ -кратных сечений, каждое из которых есть произведение $(p - \lambda)$ -кратного сечения матрицы A

¹ Grama A., Gupta A., Karypis G., Kumar V. Introduction to Parallel Computing. 2nd Ed. Addison Wesley, 2003. 664 p.

² Гантмахер Ф. Р. Теория матриц. 2-е изд., доп. М.: Наука, 1966. 576 с. URL: <http://mathscinet.ru/files/Gantmaher.pdf> (дата обращения: 19.06.2022).

³ Соколов Н. П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.



и $(q-\lambda)$ -кратного сечения матрицы B . При этом наборы значений индексов S_1, \dots, S_λ сечений-сомножителей и сечения-результата совпадают.

Таким образом первое разбиение матриц-операндов и матрицы-результата на блоки можно осуществить, фиксируя значения индексов разбиения s . Следовательно, если рассматривать индексы разбиений l, s, c и t как мультииндексы, то для каждого фиксированного значения мультииндекса $s = s^*$ матрицы, $B = \left\| b_{s^*cm}^* \right\|$ и $C = \left\| C_{ls^*m} \right\|$ можно рассматривать как условно двумерные. Тогда, если каждый индекс из разбиения s принимает значения от 0 до $n-1$, и имеется n^2 процессоров, желательно параллельных, что будет следовать из дальнейшего изложения, то на каждом из этих процессоров будет выполняться $(0, \mu)$ -свернутое произведение вида $C_{ls^*m} = {}^{0,\mu} (A_{ls^*c}^* \times B_{s^*cm}^*)$. Таким образом, в рассматриваемом случае, (λ, μ) -свернутое произведение условно трехмерных матриц сводится к $E^\lambda (0, \mu)$ -свернутых произведений над сечениями ориентации s (по всем наборам значений скоттовых индексов) матриц-операндов.

Обобщение алгоритма Кэннона на $(0, \mu)$ -свернутое произведение

Как и в случае умножения плоских матриц, алгоритм умножения многомерных матриц – простых сечений матриц A и B ориентации (S_1, \dots, S_λ) реализуется параллельно выполняемыми процессами. Поскольку умножения этих сечений не зависят друг от друга, далее рассматривается умножение в рамках только одного сечения. Кроме того, предполагается, что $k=v$, то есть обе матрицы

Сложность обобщения заключается в том, в алгоритме Кэннона, реализующем умножение плоских матриц, операции циклического сложения выполняются над тремя индексами, а в случае многомерных матриц необходимо выполнять эту операцию над $k+\lambda+v$ индексами. Это существенно усложняет как формализацию задачи пересылки блоков матриц операндов между процессами, так и реализацию алгоритма пересчета значений индексов. Поскольку предполагается, что все индексы всех сечений принимают одно и то же количество значений от 0 до $n-1$, то далее рассматривается метод, основанный на представлении каждого набора значений индексов в виде **числа в системе счисления с основанием E** ($E < n$), смысл которого будет определен в дальнейшем.

Пусть T такое число, что n^k, n^μ и n^v кратны T и $E = \frac{n}{T}$. Тогда матрицы A и B могут быть разбиты на $E^{2(k+\mu)}$ и $E^{2(\mu+v)}$ блоков соответственно. Следуя требованиям алгоритма Кэннона предполагается, что $k=v$, то есть обе матрицы содержат одинаковое количество свободных индексов. Следовательно, в этом случае для параллельной реализации $(0, \mu)$ -свернутого произведения требуется $E^{2(k+\mu)} = E^{2(\mu+v)}$ процессов. Каждый процесс последовательно выполняет умножение соответствующих друг другу блоков условно двумерных матриц A и B и складывает произведение с блоком условно двумерной матрицы C . После выполнения последней итерации блок матрицы C принимает окончательное значение. При этом должны выполняться следующие требования:

1. Блок матрицы A – это ее сечение, в котором индексы разбиения l и c принимают E последовательных значений, индексы

разбиения s – единственное значение.

2. Блок матрицы B – это ее сечение, в котором индексы разбиения t и c принимают E последовательных значений, индексы разбиения s – единственное значение.

3. Блок матрицы C – это ее сечение, в котором индексы разбиения l и t принимают n последовательных значений, индексы разбиения s – единственное значение.

Общий индекс процесса можно представить в виде триады: $\langle N_E^l S_1^*, \dots, S_\lambda^* N_E^m \rangle$, где N_E^l и N_E^m – числа в E -ричной системе счисления, принимающие значения от 0 до $E^k-1 = E^v-1$ и соответствующие свободным индексам обеих матриц. Пересчет индексов процессов производится посредством сложения чисел N_E^l и N_E^m с целым числом, которое задается на каждом шаге алгоритма. Фиксированные значения S_1, \dots, S_λ скоттовых индексов в пересчете не участвуют и используются только для идентификации сечения. Таким образом многомерные сечения условно трехмерных матриц сводятся к условно двумерным матрицам и $(0, \mu)$ -свернутое произведение сводится к условно $(0, 1)$ -свернутому произведению, что позволяет упростить обобщение алгоритма Кэннона.

Реализация обобщенного алгоритма Кэннона

В начальном состоянии предполагается, что каждый процесс связан с блоками матриц A и B , индексы которых соответствуют индексам процесса. Это означает, что для сечения ориентации S_1, \dots, S_λ при фиксированных i и j $i = 0, \dots, \underbrace{E-1 \dots E-1}_k, j = \underbrace{E-1 \dots E-1}_v$, где $\underbrace{E-1 \dots E-1}_k = E^k - 1$, $\underbrace{E-1 \dots E-1}_v = E^v - 1$

процессу $P_{i_1 \dots i_k s_1^* \dots s_\lambda^* m_1^* \dots m_k^*}$ будут соответствовать блоки $A_{i_1 \dots i_k s_1^* \dots s_\lambda^* c_1^* \dots c_\mu^*}$ и $A_{s_1^* \dots s_\lambda^* c_1^* \dots c_\mu^* m_1^* \dots m_v^*}$ матриц A и B . Начальная привязка блоков матриц A и B к процессам приведена в таблице 1. Кроме того начальному состоянию соответствуют «обнуленные» (заполненные значениями нейтрального элемента) блоки сечения матрицы C в соответствующих процессах. Распределение блоков сечений матриц A, B и C может быть физическим, если каждый процесс связан с автономным запоминающим устройством, или логическим, если все процессы обрабатывают данные расположенные на одном запоминающем устройстве.

На первом этапе алгоритма выполняются следующие действия:

1. Блок матрицы A с набором значений индексов $(i_1^l \dots i_k^l)_E$ разбиения l связывается с процессом, набор значений индексов разбиения t которого вычисляется по формуле: $(E^k + (i_1^l \dots i_k^l)_E - i) \bmod E^k$ ($i = 0, \dots, \underbrace{E-1 \dots E-1}_k$).
2. Блок матрицы B с набором значений индексов $(m_1^j \dots m_v^j)_E$ разбиения t связывается с процессом, набор значений индексов разбиения t которого вычисляется по формуле: $(E^v + (m_1^j \dots m_v^j)_E - j) \bmod E^v$ ($j = 0, \dots, \underbrace{E-1 \dots E-1}_v$).

Привязка блоков матриц A и B к процессам после завершения первого этапа алгоритма приведена в таблице 2.



$P_{\frac{0...2s_1^* \dots s_2^* 0...0}{\kappa \nu}}$	$P_{\frac{0...2s_1^* \dots s_2^* 0...1}{\kappa \nu}}$	$P_{\frac{0...2s_1^* \dots s_2^* 0...2}{\kappa \nu}}$...	$P_{\frac{0...2s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \nu}}$	$P_{\frac{0...2s_1^* \dots s_2^* E-1...E-1}{\kappa \nu}}$
$A_{\frac{0...2s_1^* \dots s_2^* 0...2}{\kappa \mu}}$	$A_{\frac{0...2s_1^* \dots s_2^* 0...3}{\kappa \mu}}$	$A_{\frac{0...2s_1^* \dots s_2^* 0...4}{\kappa \mu}}$		$A_{\frac{0...2s_1^* \dots s_2^* 0...0}{\kappa \mu}}$	$A_{\frac{0...2s_1^* \dots s_2^* 0...1}{\kappa \mu}}$
$B_{\frac{s_1^* \dots s_2^* 0...20...0}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...30...1}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...40...2}{\mu \nu}}$		$B_{\frac{s_1^* \dots s_2^* 0...0E-1...E-1E-2}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...1E-1...E-1}{\mu \nu}}$
...					
$P_{\frac{E-1...E-1E-2s_1^* \dots s_2^* 0...0}{\kappa \nu}}$	$P_{\frac{E-1...E-1E-2s_1^* \dots s_2^* 0...1}{\kappa \nu}}$	$P_{\frac{E-1...E-1E-2s_1^* \dots s_2^* 0...2}{\kappa \nu}}$...	$P_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \nu}}$	$P_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1}{\kappa \nu}}$
$A_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \mu}}$	$A_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1}{\kappa \mu}}$	$A_{\frac{E-1...E-1E-2s_1^* \dots s_2^* 0...0}{\kappa \mu}}$		$A_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1E-4}{\kappa \mu}}$	$A_{\frac{E-1...E-1E-2s_1^* \dots s_2^* E-1...E-1E-3}{\kappa \mu}}$
$B_{\frac{s_1^* \dots s_2^* E-1...E-1E-20...0}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* E-1...E-10...1}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...00...2}{\mu \nu}}$		$B_{\frac{s_1^* \dots s_2^* E-1...E-1E-4E-1...E-1E-2}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* E-1...E-1E-2E-1...E-1}{\mu \nu}}$
...					
$P_{\frac{E-1...E-1s_1^* \dots s_2^* 0...0}{\kappa \nu}}$	$P_{\frac{E-1...E-1s_1^* \dots s_2^* 0...1}{\kappa \nu}}$	$P_{\frac{E-1...E-1s_1^* \dots s_2^* 0...2}{\kappa \nu}}$...	$P_{\frac{E-1...E-1s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \nu}}$	$P_{\frac{E-1...E-1s_1^* \dots s_2^* E-1...E-1}{\kappa \nu}}$
$A_{\frac{E-1...E-1s_1^* \dots s_2^* E-1...E-1}{\kappa \mu}}$	$A_{\frac{E-1...E-1s_1^* \dots s_2^* 0...0}{\kappa \mu}}$	$A_{\frac{E-1...E-1s_1^* \dots s_2^* 0...1}{\kappa \mu}}$		$A_{\frac{E-1...E-1s_1^* \dots s_2^* E-1...E-1E-3}{\kappa \mu}}$	$A_{\frac{E-1...E-1s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \mu}}$
$B_{\frac{s_1^* \dots s_2^* E-1...E-10...0}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...00...1}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* 0...10...2}{\mu \nu}}$		$B_{\frac{s_1^* \dots s_2^* E-1...E-1E-3E-1...E-1E-2}{\mu \nu}}$	$B_{\frac{s_1^* \dots s_2^* E-1...E-1E-2E-1...E-1}{\mu \nu}}$

Второй этап алгоритма состоит из E итераций, на каждой из которых выполняются три действия:

1. Каждый процесс $P_{\frac{l_1^i \dots l_k^i s_1^* \dots s_2^* m_1^j \dots m_v^j}{\kappa \nu}}$ выполняет умножение связанных с ним блоков матриц A и B и складывает результат умножения с соответствующим ему блоком матрицы C.
2. Блок матрицы A с набором значений индексов $(l_1^i \dots l_k^i)_E$ разбиения l связывается с процессом, набор значений индексов разбиения m которого вычисляется по формуле.

$$(E^k + (l_1^i \dots l_k^i)_E - 1) \bmod E^k.$$

3. Блок матрицы B с набором значений индексов $(m_1^j \dots m_v^j)_E$ разбиения m связывается с процессом, набор значений индексов разбиения t которого вычисляется по формуле. $(E^v + (m_1^j \dots m_v^j)_E - 1) \bmod E^v$.

Последовательности произведений блоков матриц A и B, участвующих в вычислениях блоков матрицы C на всех E итерациях приведены в таблице 3.

Таблица 3. Результаты вычисления блоков сечения матрицы C
Table 3. The results of calculating the blocks of the section of the matrix C

Индекс процесса	Итерация	Слагаемое (произведение блоков матриц A и B)
$\frac{0...00...0}{\kappa \nu}$	0	$A_{\frac{0...0s_1^* \dots s_2^* 0...0}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* 0...00...0}{\mu \nu}}$
	1	$A_{\frac{0...0s_1^* \dots s_2^* 0...1}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* 0...10...0}{\mu \nu}}$
	E-2	$A_{\frac{0...0s_1^* \dots s_2^* E-1...E-1E-2}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* E-1...E-1E-20...0}{\mu \nu}}$
	E-1	$A_{\frac{0...0s_1^* \dots s_2^* E-1...E-1E-1}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* E-1...E-1E-10...0}{\mu \nu}}$
$\frac{0...00...1}{\kappa \nu}$	0	$A_{\frac{0...0s_1^* \dots s_2^* 0...1}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* 0...10...1}{\mu \nu}}$
	1	$A_{\frac{0...0s_1^* \dots s_2^* 0...2}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* 0...20...1}{\mu \nu}}$
	E-2	$A_{\frac{0...0s_1^* \dots s_2^* E-1...E-1}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* E-1...E-10...1}{\mu \nu}}$
	E-1	$A_{\frac{0...0s_1^* \dots s_2^* 0...0}{\kappa \mu}} \times B_{\frac{s_1^* \dots s_2^* 0...00...1}{\mu \nu}}$



$\underbrace{0\dots 0}_{\kappa} \underbrace{E-1\dots E-1}_{\nu} E-2$	0	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2 \underbrace{E-1\dots E-1}_{\nu} E-2$
	1	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} \underbrace{E-1\dots E-1}_{\nu} E-2$
	$E-2$	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-4} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-4 \underbrace{E-1\dots E-1}_{\nu} E-2$
	$E-1$	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3 \underbrace{E-1\dots E-1}_{\nu} E-2$
$\underbrace{0\dots 0}_{\kappa} \underbrace{E-1\dots E-1}_{\nu}$	0	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} \underbrace{E-1\dots E-1}_{\nu}$
	1	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 0}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 0}_{\mu} \underbrace{E-1\dots E-1}_{\nu}$
	$E-2$	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3 \underbrace{E-1\dots E-1}_{\nu} E-2$
	$E-1$	$A_{\underbrace{0\dots 0}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2 \underbrace{E-1\dots E-1}_{\nu} E-2$
$\underbrace{0\dots 10\dots 0}_{\kappa} \underbrace{}_{\nu}$	0	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 10\dots 0}_{\mu} \underbrace{}_{\nu}$
	1	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3 \underbrace{0\dots 0}_{\nu}$
	$E-2$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} \underbrace{E-10\dots 0}_{\nu}$
	$E-1$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 0}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 00\dots 0}_{\mu} \underbrace{}_{\nu}$
$\underbrace{0\dots 10\dots 1}_{\kappa} \underbrace{}_{\nu}$	0	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 2}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 20\dots 1}_{\mu} \underbrace{}_{\nu}$
	1	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 3}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 30\dots 1}_{\mu} \underbrace{}_{\nu}$
	$E-2$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} \underbrace{E-10\dots 1}_{\nu}$
	$E-1$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 10\dots 1}_{\mu} \underbrace{}_{\nu}$
$\underbrace{0\dots 1}_{\kappa} \underbrace{E-1\dots E-1}_{\nu} E-2$	0	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} \underbrace{E-1\dots E-1}_{\nu} E-2$
	1	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* 0\dots 0}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* 0\dots 0}_{\mu} \underbrace{E-1\dots E-1}_{\nu} E-2$
	$E-2$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-3 \underbrace{E-1\dots E-1}_{\nu} E-2$
	$E-1$	$A_{\underbrace{0\dots 1}_{\kappa} s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2} \times B_{s_1^* \dots s_{\lambda}^* E-1\dots E-1}_{\mu} E-2 \underbrace{E-1\dots E-1}_{\nu} E-2$



$\underbrace{E-1 \dots E-1}_{\kappa} \underbrace{0 \dots 0}_{\nu}$	0	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1}_{\mu} \underbrace{E-10 \dots 0}_{\nu}}$
	1	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 0}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 00 \dots 0}_{\mu \nu}}$
	$E-2$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-3}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-30}_{\mu} \underbrace{0}_{\nu}}$
	$E-1$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-20}_{\mu} \underbrace{0}_{\nu}}$
$\underbrace{E-1 \dots E-1}_{\kappa} \underbrace{0 \dots 1}_{\nu}$	0	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 0}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 00 \dots 1}_{\mu \nu}}$
	1	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{0 \dots 10 \dots 1}_{\mu \nu}}$
	$E-2$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-20 \dots 1}_{\mu \nu}}$
	$E-1$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-10 \dots 1}_{\mu \nu}}$
$\underbrace{E-1 \dots E-1}_{\kappa} \underbrace{E-1 \dots E-1 E-2}_{\nu}$	0	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-3}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-3 E-1 \dots E-1 E-2}_{\mu \nu}}$
	1	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2 E-1 \dots E-1 E-2}_{\mu \nu}}$
	$E-2$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-5}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-5 E-1 \dots E-1 E-2}_{\mu \nu}}$
	$E-1$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-4}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-4 E-1 \dots E-1 E-2}_{\mu \nu}}$
$\underbrace{E-1 \dots E-1}_{\kappa} \underbrace{E-1 \dots E-1}_{\nu}$	0	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-2 E-1 \dots E-1}_{\mu \nu}}$
	1	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-1 E-1 \dots E-1}_{\mu \nu}}$
	$E-2$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-4}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-4 E-1 \dots E-1}_{\mu \nu}}$
	$E-1$	$A_{\underbrace{E-1 \dots E-1}_{\kappa} s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-3}_{\mu}} \times B_{s_1^* \dots s_{\lambda}^* \underbrace{E-1 \dots E-1 E-3 E-1 \dots E-1}_{\mu \nu}}$

После выполнения всех итераций каждый процесс содержит соответствующий ему блок сечения матрицы-результата.

Заключение

В статье рассмотрено обобщение алгоритма Кэннона для умножения многомерных матриц. Доказано, что многомерные матрицы-операнды могут быть представлены как совокупность сечений по скоттовым индексам, произведения которых позволяют получить требуемую матрицу-результат. Вычисление этих произведений может выполняться параллельными процессами. Показано, что сечения исходных матриц представляют собой условно двумерные матрицы. Поэтому

для параллельного их умножения может быть использован алгоритм Кэннона. Для разбиения этих сечений на блоки и пересчета индексов блоков в ходе выполнения алгоритма предложен метод, основанный на использовании специфических систем счисления, и определен метод вычисления оснований этих систем счисления. Процесс выполнения обобщенного алгоритма Кэннона проиллюстрирован таблицами распределения блоков на всех этапах выполнения алгоритма. Таким образом можно утверждать, что в статье приведено эффективное обобщение алгоритма Кэннона для параллельного вычисления (λ, μ) -свернутого произведения многомерных матриц.



Список использованных источников

- [1] Sokolov N. P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations // *Ukrainian Mathematical Journal*. 1970. Vol. 22, issue 6. P. 657-674. doi: <https://doi.org/10.1007/BF01086271>
- [2] Ильин П. Л., Мунерман В. И. Рекурсивное вычисление детерминанта многомерной матрицы // *Системы компьютерной математики и их приложения*. 2019. № 20-1. С. 162-167. URL: <https://elibrary.ru/item.asp?id=39103176> (дата обращения: 19.06.2022).
- [3] Мунерман В. И., Мунерман Д. В. О соответствии моделей данных и моделей вычислений // *Системы компьютерной математики и их приложения*. 2021. № 22. С. 146-152. URL: <https://elibrary.ru/item.asp?id=46649891> (дата обращения: 19.06.2022).
- [4] Mukha V. S. Multidimensional Matrix Approach in Parallel Factor Analysis // *Journal of Automation and Information Sciences*. 2006. Vol. 38, issue 10. P. 21-29. doi: <https://doi.org/10.1615/JAutomatInfScien.v38.i10.30>
- [5] Объектно-ориентированный подход к разработке моделей данных / Е. П. Емельченков, В. И. Мунерман, Д. В. Мунерман, Т. А. Самойлова // *Современные информационные технологии и ИТ-образование*. 2020. Т. 16, № 3. С. 564-574. doi: <https://doi.org/10.25559/SITITO.16.202003.564-574>
- [6] Goil S., Choudhary A. PARSIMONY: An Infrastructure for Parallel Multidimensional Analysis and Data Mining // *Journal of Parallel and Distributed Computing*. 2001. Vol. 61, issue 3. P. 285-321. doi: <https://doi.org/10.1006/jpdc.2000.1691>
- [7] Munerman V., Munerman D., Samoilo T. The Heuristic Algorithm For Symmetric Horizontal Data Distribution // *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*. St. Petersburg, Moscow, Russia : IEEE Computer Society, 2021. P. 2161-2165. doi: <https://doi.org/10.1109/ElConRus51938.2021.9396510>
- [8] Ефимов С. С. Обзор методов распараллеливания алгоритмов решения некоторых задач вычислительной дискретной математики // *Математические структуры и моделирование*. 2007. № 17. С. 72-93. URL: <https://elibrary.ru/item.asp?id=21034721> (дата обращения: 19.06.2022).
- [9] Developing parallel programming and soft skills: A project based learning approach / A. A. Younis [и др.] // *Journal of Parallel and Distributed Computing*. 2021. Vol. 158, no. C. P. 151-163. doi: <https://doi.org/10.1016/j.jpdc.2021.07.015>
- [10] Трещев И. А. Построение многопоточных приложений для распараллеливания алгоритмов перебора // *Информатика и системы управления*. 2008. № 1(15). С. 151-159. URL: <https://elibrary.ru/item.asp?id=10366217> (дата обращения: 19.06.2022).
- [11] Иванов Д. А. Параллельное умножение матриц с комплексными числами // *Энергия-2019 : Материалы Четырнадцатой всероссийской (межд.) научно-технической конференции студентов, аспирантов и молодых ученых. В 6-ти томах. Т. 5. Иваново : ИГЭУ, 2019. С. 60. URL: https://elibrary.ru/item.asp?id=41191835&pff=1* (дата обращения: 19.06.2022).
- [12] Fineman J. T. Nearly work-efficient parallel algorithm for digraph reachability // *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'2018)*. New York, NY, USA : Association for Computing Machinery, 2018. P. 457-470. doi: <https://doi.org/10.1145/3188745.3188926>
- [13] van der Klis M. H., Tellings J. L. Multidimensional scaling and linguistic theory // *arXiv:2012.04946*. 2020. doi: <https://doi.org/10.48550/arXiv.2012.04946>
- [14] Overview of Parallel Systems / R. Trobec, B. Slivnik, P. Bulić, B. Robič // *Introduction to Parallel Computing. Undergraduate Topics in Computer Science*. Cham : Springer, 2018. P. 9-44. doi: https://doi.org/10.1007/978-3-319-98833-7_2
- [15] Исследование оптимальной формы разбиения данных для умножения матриц на трех гетерогенных процессорах с полностью связанной топологией и различными пропускными способностями / Е. Г. Ключева, А. А. Адамов, А. Е. Оспанова [и др.] // *Современные наукоемкие технологии*. 2019. № 2. С. 83-88. URL: <https://elibrary.ru/item.asp?id=37034414> (дата обращения: 19.06.2022).
- [16] Performance-Aware Model for Sparse Matrix-Matrix Multiplication on the Sunway TaihuLight Supercomputer / Y. Chen [и др.] // *IEEE Transactions on Parallel and Distributed Systems*. 2019. Vol. 30, no. 4. P. 923-938. doi: <https://doi.org/10.1109/TPDS.2018.2871189>
- [17] Глушань В. М., Красюк О. И., Лозовой А. Ю. Параллельное умножение матриц больших размерностей на множестве заданных процессоров // *Вестник Рязанского государственного радиотехнического университета*. 2020. № 74. С. 42-55. doi: <https://doi.org/10.21667/1995-4565-2020-74-42-55>
- [18] Dorta I., Leon C., Rodriguez C. A comparison between MPI and OpenMP Branch-and-Bound skeletons // *Eighth International Workshop on High-Level Parallel Programming Models and Supportive Environments*, 2003. Proceedings., Nice, France, 2003. P. 66-73. doi: <https://doi.org/10.1109/HIPS.2003.1196496>
- [19] Van De Geijn R. A., Watts J. SUMMA: scalable universal matrix multiplication algorithm // *Concurrency: Practice and Experience*. 1997. Vol. 9, issue 4. P. 255-274. doi: [https://doi.org/10.1002/\(SICI\)1096-9128\(199704\)9:4<255::AID-CPE250>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1096-9128(199704)9:4<255::AID-CPE250>3.0.CO;2-2)
- [20] Choi J., Walker D. W., Dongarra J. J. PUMMA: Parallel Universal Matrix Multiplication Algorithms on distributed memory concurrent computers // *Concurrency: Practice and Experience*. 1994. Vol. 6, issue 7. P. 543-570. doi: <https://doi.org/10.1002/cpe.4330060702>
- [21] Efficient Partitioning Algorithm for Parallel Multidimensional Matrix Operations by Linearization / K. S. Alam [и др.] // *Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies ; ed. by T. Senjyu, P. N. Mahalle, T. Perumal, A. Joshi*. Vol. 195. Singapore : Springer, 2021. P. 141-149. doi: https://doi.org/10.1007/978-981-15-7078-0_13



- [22] Fujiki D., Mahlke S., Das R. In-Memory Data Parallel Processor // ACM SIGPLAN Notices. 2018. Vol. 53, no. 2. P. 1-14. doi: <https://doi.org/10.1145/3296957.3173171>
- [23] Liu J.-Sh., Lin J.-Yu., Chung Y.-C. Efficient parallel algorithms for multi-dimensional matrix operations // Proceedings International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000). Dallas, TX, USA : IEEE Computer Society, 2000. P. 224-229. doi: <https://doi.org/10.1109/ISPAN.2000.900289>
- [24] Pan V. Y., Yu Y., Stewart C. Algebraic and Numerical Techniques for the Computation of Matrix Determinants // Computers & Mathematics with Applications. 1997. Vol. 34, issue 1. P. 43-70. doi: [https://doi.org/10.1016/S0898-1221\(97\)00097-7](https://doi.org/10.1016/S0898-1221(97)00097-7)
- [25] Мунерман В. И. Архитектура программно-аппаратного комплекса для массовой обработки данных на базе многомерно-матричной модели // Системы высокой доступности. 2015. Т. 11, № 2. С. 13-18. URL: <https://www.elibrary.ru/item.asp?id=23819273> (дата обращения: 19.06.2022).

Поступила 19.06.2022; одобрена после рецензирования 05.08.2022; принята к публикации 10.09.2022.

Об авторах:

Мунерман Виктор Иосифович, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, ORCID: <https://orcid.org/0000-0002-9628-4049>, vimoon@gmail.com

Мунерман Даниил Викторович, лаборант-стажер кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), ORCID: <https://orcid.org/0000-0002-5139-6645>, danvmoon@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

- [1] Sokolov N.P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations. *Ukrainian Mathematical Journal*. 1970;22(6):657-674. doi: <https://doi.org/10.1007/BF01086271>
- [2] Iljin P.L., Munerman V.I. Recursive Computation of the Multidimensional Matrix Determinant. *Computer Mathematics Systems and Their Applications*. 2019;(20-1):162-167. Available at: <https://elibrary.ru/item.asp?id=39103176> (accessed 19.06.2022). (In Russ., abstract in Eng.)
- [3] Munerman V.I., Munerman D.V. About the correspondence of data models and calculation models. *Computer Mathematics Systems and Their Applications*. 2021;(22):146-152. Available at: <https://elibrary.ru/item.asp?id=46649891> (accessed 19.06.2022). (In Russ., abstract in Eng.)
- [4] Mukha V.S. Multidimensional Matrix Approach in Parallel Factor Analysis. *Journal of Automation and Information Sciences*. 2006;38(10):21-29. doi: <https://doi.org/10.1615/JAutomatInfScien.v38.i10.30>
- [5] Emelchenkov Ye.P., Munerman V.I., Munerman D.V., SamoiloVA T.A. The Object Oriented Approach to Designing Data Models. *Modern Information Technologies and IT-Education*. 2020;16(3):564-574. (In Russ., abstract in Eng.) doi: <https://doi.org/10.25559/SITITO.16.202003.564-574>
- [6] Goil S., Choudhary A. PARSIMONY: An Infrastructure for Parallel Multidimensional Analysis and Data Mining. *Journal of Parallel and Distributed Computing*. 2001;61(3):285-321. doi: <https://doi.org/10.1006/jpdc.2000.1691>
- [7] Munerman V., Munerman D., SamoiloVA T. The Heuristic Algorithm For Symmetric Horizontal Data Distribution. In: 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). St. Petersburg, Moscow, Russia: IEEE Computer Society; 2021. p. 2161-2165. doi: <https://doi.org/10.1109/ElConRus51938.2021.9396510>
- [8] Efimov S.S. [Review of methods for parallelization of algorithms for solving some problems of computational discrete mathematics]. *Mathematical Structures and Modeling*. 2007;(17):72-93. Available at: <https://elibrary.ru/item.asp?id=21034721> (accessed 19.06.2022). (In Russ.)
- [9] Younis A.A., Sunderraman R., Metzler M., Bourgeois A.G. Developing parallel programming and soft skills: A project based learning approach. *Journal of Parallel and Distributed Computing*. 2021;158(C):151-163. doi: <https://doi.org/10.1016/j.jpdc.2021.07.015>
- [10] Treschev I.A. Construction of multiflow supplements for unparalleling of excess algorithms. *Information Science and Control Systems*. 2008;(1):151-159. Available at: <https://elibrary.ru/item.asp?id=37034414> <https://elibrary.ru/item.asp?id=10366217> (accessed 19.06.2022). (In Russ., abstract in Eng.)
- [11] Ivanov D.A. [Parallel multiplication of matrices with complex numbers]. In: Proceedings of the 14th International Conference on Energy-2019. Vol. 5. Ivanovo: ISPU; 2019. p. 60. Available at: <https://elibrary.ru/item.asp?id=41191835&pf=1> (accessed 19.06.2022). (In Russ.)
- [12] Fineman J.T. Nearly work-efficient parallel algorithm for digraph reachability. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'2018). New York, NY, USA: Association for Computing Machinery; 2018. p. 457-470. doi: <https://doi.org/10.1145/3188745.3188926>



- [13] van der Klis M.H., Tellings J.L. Multidimensional scaling and linguistic theory. arXiv:2012.04946. 2020. doi: <https://doi.org/10.48550/arXiv.2012.04946>
- [14] Trobec R., Slivnik B., Bulić P., Robič B. Overview of Parallel Systems. In: Introduction to Parallel Computing. Undergraduate Topics in Computer Science. Cham: Springer; 2018. p. 9-44. doi: https://doi.org/10.1007/978-3-319-98833-7_2
- [15] Klyueva E.G., Adamov A.A., Ospanova A.E., Snitsar L.R., Kulbaeva L.N. Studying the optimal form of partitioning the data for the matrix multiplication on three fully connected heterogeneous processors with different bandwidths. *Modern high technologies*. 2019;(2):83-88. Available at: <https://elibrary.ru/item.asp?id=37034414> (accessed 19.06.2022). (In Russ., abstract in Eng.)
- [16] Chen Y., Li K., Yang W., Xiao G., Xie X., Li T. Performance-Aware Model for Sparse Matrix-Matrix Multiplication on the Sunway TaihuLight Supercomputer. *IEEE Transactions on Parallel and Distributed Systems*. 2019;30(4):923-938. doi: <https://doi.org/10.1109/TPDS.2018.2871189>
- [17] Glushan V.M., Krasnyuk O.I., Lozovoy A.Yu. Parallel multiplication of large-dimensional matrices on many given processors. *Bulletin of the Ryazan State Radio Engineering University*. 2020;(74):42-55. (In Russ., abstract in Eng.) doi: <https://doi.org/10.21667/1995-4565-2020-74-42-55>
- [18] Dorta I., Leon C., Rodriguez C. A comparison between MPI and OpenMP Branch-and-Bound skeletons. In: Eighth International Workshop on High-Level Parallel Programming Models and Supportive Environments, 2003. Proceedings., Nice, France; 2003. p. 66-73. doi: <https://doi.org/10.1109/HIPS.2003.1196496>
- [19] Van De Geijn R.A., Watts J. SUMMA: scalable universal matrix multiplication algorithm. *Concurrency: Practice and Experience*. 1997;9(4):255-274. doi: [https://doi.org/10.1002/\(SICI\)1096-9128\(199704\)9:4<255::AID-CPE250>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1096-9128(199704)9:4<255::AID-CPE250>3.0.CO;2-2)
- [20] Choi J., Walker D.W., Dongarra J.J. PUMMA: Parallel Universal Matrix Multiplication Algorithms on distributed memory concurrent computers. *Concurrency: Practice and Experience*. 1994;6(7):543-570. doi: <https://doi.org/10.1002/cpe.4330060702>
- [21] Alam K.S., Shishir T.A., Azharul Hasan K.M. Efficient Partitioning Algorithm for Parallel Multidimensional Matrix Operations by Linearization. In: Senjyu T., Mahalle P.N., Perumal T., Joshi A. (eds.). Information and Communication Technology for Intelligent Systems. ICTIS 2020. Smart Innovation, Systems and Technologies. Vol. 195. Singapore: Springer; 2021. p. 141-149. doi: https://doi.org/10.1007/978-981-15-7078-0_13
- [22] Fujiki D., Mahlke S., Das R. In-Memory Data Parallel Processor. *ACM SIGPLAN Notices*. 2018;53(2):1-14. doi: <https://doi.org/10.1145/3296957.3173171>
- [23] Liu J.-Sh., Lin J.-Yu., Chung Y.-C. Efficient parallel algorithms for multi-dimensional matrix operations. In: Proceedings International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN 2000). Dallas, TX, USA: IEEE Computer Society; 2000. p. 224-229. doi: <https://doi.org/10.1109/ISPAN.2000.900289>
- [24] Pan V.Y., Yu Y., Stewart C. Algebraic and Numerical Techniques for the Computation of Matrix Determinants. *Computers & Mathematics with Applications*. 1997;34(1):43-70. doi: [https://doi.org/10.1016/S0898-1221\(97\)00097-7](https://doi.org/10.1016/S0898-1221(97)00097-7)
- [25] Munerman V.I. Construction of hardware-software complexes architecture to improve massively data processing. *Highly Available Systems*. 2015;11(2):13-18. Available at: <https://www.elibrary.ru/item.asp?id=23819273> (accessed 19.06.2022). (In Russ., abstract in Eng.)

Submitted 19.06.2022; approved after reviewing 05.08.2022; accepted for publication 10.08.2022.

About the authors:

Victor I. Munerman, Associate Professor of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), Cand.Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, vimoon@gmail.com

Daniel V. Munerman, Laboratory Assistant, of the Chair of Computer Science, Faculty of Physics and Mathematics, Smolensk State University (4 Przhevalsky St., Smolensk 214000, Russian Federation), **ORCID: <https://orcid.org/0000-0002-5139-6645>**, danvimoon@gmail.com

All authors have read and approved the final manuscript.

