

ISSN 1561-2430 (Print)
ISSN 2524-2415 (Online)
УДК 519.67
<https://doi.org/10.29235/1561-2430-2018-54-4-417-426>

Поступила в редакцию 27.06.2017
Received 27.06.2017

Н. А. Лиходед, М. А. Полещук

Белорусский государственный университет, Минск, Беларусь

ПОСТРОЕНИЕ ДВУМЕРНЫХ ЗЕРНИСТЫХ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Аннотация. Алгоритм, реализуемый на параллельном компьютере с распределенной памятью, имеет, как правило, зернистую структуру: множество операций разбито на подмножества, называемые зёрнами вычислений. Одним из современных подходов к получению зернистых вариантов алгоритмов является тайлинг – преобразование, основанное на информационных разрезах итерационного пространства, в результате которого получаются макрооперации-тайлы. Операции одного тайла выполняются атомарно, как одна единица вычислений, а обмен данными происходит массивами. В настоящей работе для алгоритмов, заданных вложенными многомерными циклами, предложен способ построения зернистых вычислительных процессов, логически организованных в двумерную структуру. По сравнению с одномерными структурами, использование двумерных структур возможно в меньшем числе случаев, но может иметь преимущества при реализации алгоритмов на параллельных компьютерах с распределенной памятью. К числу возможных преимуществ относятся уменьшение объема коммуникационных операций, уменьшение разгона и торможения вычислений, потенциально большее число вычислительных процессов, организация обменных операций только в пределах строк или столбцов процессов. Представленные исследования обобщают на случай двумерной структуры некоторые аспекты метода построения параллельных вычислительных процессов, организованных в одномерную структуру. В частности, исследована возможность организовать полностью загруженные работой параллельные вычислительные процессы. Показано, что при определенных ограничениях на структуру и длину циклов достаточно произвести тайлинг по трем координатам многомерного итерационного пространства. В более ранних теоретических исследованиях параллельность зернистых вычислений гарантировалась при наличии информационных разрезов по всем координатам итерационного пространства, а для более простого случая одномерной структуры – по двум координатам.

Ключевые слова: параллельные вычисления, распараллеливание алгоритмов, параллельный компьютер с распределенной памятью, уменьшение числа обменов данными

Для цитирования. Лиходед, Н. А. Построение двумерных зернистых параллельных вычислительных процессов / Н. А. Лиходед, М. А. Полещук // Вест. Нац. акад. наук Беларусі. Сер. фіз.-мат. навук. – 2018. – Т. 54, № 4. – С. 417–426. <https://doi.org/10.29235/1561-2430-2018-54-4-417-426>

N. A. Likhoded, M. A. Paliashchuk

Belarusian State University, Minsk, Belarus

TILED PARALLEL 2D COMPUTATIONAL PROCESSES

Abstract. The algorithm implemented on a parallel computer with distributed memory has, as a rule, a tiled structure: a set of operations is divided into subsets, called tiles. One of the modern approaches to obtaining tiled versions of algorithms is a tiling transformation based on information sections of the iteration space, resulting in macro-operations (tiles). The operations of one tile are performed atomically, as one unit of calculation, and the data exchange is done by arrays. The method of construction of tiled computational processes logically organized as a two-dimensional structure for algorithms given by multidimensional loops is stated. Compared to one-dimensional structures, the use of two-dimensional structures is possible in a smaller number of cases, but it can have advantages when implementing algorithms on parallel computers with distributed memory. Among the possible advantages are the reduction of the volume of communication operations, the reduction of acceleration and deceleration of computations, potentially a greater number of computation processes and the organization of data exchange operations only within the rows or columns of processes. The results are a generalization of some aspects of the method of construction of parallel computational processes organized in a one-dimensional structure to the case of a two-dimensional structure. It is shown that under certain restrictions on the structure and length of loops, it is sufficient to perform tiling on three coordinates of a multidimensional iteration space. In the earlier theoretical studies, the parallelism of tiled computations was guaranteed in the presence of information sections in all coordinates of the iteration space, and for a simpler case of a one-dimensional structure, in two coordinates

Keywords: parallel computations, parallelization of algorithms, distributed memory parallel computer, data exchange reduction

For citation. Likhoded N. A., Paliashchuk A. A. Tiled parallel 2D computational processes. *Vestsi Natsyianal'nai akademii navuk Belarusi. Seriya fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2018, vol. 54, no. 4, pp. 417–426 (in Russian). <https://doi.org/10.29235/1561-2430-2018-54-4-417-426>

Введение. Для реализации алгоритма на параллельном компьютере с распределенной памятью множество операций алгоритма должно быть, как правило, разбито на подмножества, называемые зёрнами вычислений. Такие алгоритмы называют зёрнистыми. Множество операций алгоритма разбить на зёрна можно путем тайлинга (tiling) – преобразования алгоритма для получения макроопераций-тайлов [1–3]. Операции одного тайла выполняются атомарно, как одна единица вычислений, а обмен данными происходит массивами.

В настоящей работе для алгоритмов, заданных вложенными многомерными циклами, предложен способ построения зёрнистых вычислительных процессов, логически организованных в двумерную структуру. Такие процессы назовем двумерными (2D) процессами. Использование двумерных структур, по сравнению с одномерными, связано с дополнительными ограничениями, но имеет и ряд преимуществ при реализации алгоритмов на параллельных компьютерах с распределенной памятью: позволяет в некоторых случаях (пример, имеющий практическое значение, приведен ниже) уменьшить объем коммуникационных операций, уменьшить разгон и торможение вычислений, получить большее число вычислительных процессов (что важно, если число итераций по одной координате сравнительно небольшое), организовать обменные операции только в пределах строк или столбцов процессов.

Представленные в работе исследования являются обобщением некоторых аспектов метода построения параллельных вычислительных процессов, организованных в одномерную структуру [4, 5], на случай двумерной структуры. Показано, в частности, что для возможности организовать полностью загруженные работой параллельные вычислительные процессы достаточно, при определенных ограничениях на структуру и длину циклов, произвести тайлинг по трем координатам многомерного итерационного пространства, для частного случая одномерной структуры – по двум координатам [4, 5]. В более ранних теоретических исследованиях показана возможность организации зёрнистых параллельных вычислений при возможности произвести информационные разрезы по всем координатам многомерного итерационного пространства [6, с. 407]. Представленные в данной работе исследования являются также развитием и конкретизацией исследований [7] для частного случая 2D-структур: рассмотрен более широкий класс алгоритмов, получены условия загруженности работой всех зёрнистых вычислительных процессов.

Предварительные сведения. Приведем необходимые для дальнейшего изложения сведения о формальном описании алгоритма и зависимостях между операциями алгоритма.

Пусть алгоритм задан гнездом вложенных циклов, в котором имеется Θ наборов выполняемых операторов. Под набором операторов будем понимать один или несколько выполняемых операторов, окруженных одним и тем же множеством циклов. Выполняемые операторы и наборы операторов линейно упорядочены расположением их в записи алгоритма. Обозначим V^θ , $1 \leq \theta \leq \Theta$, – область изменения параметров циклов, окружающих θ -й набор операторов, n^θ – размерность этой области.

Вхождением (a, S_β, q) будем называть q -е вхождение массива a в оператор S_β . Другими словами, вхождение (a, S_β, q) – это q -е обращение в последовательности обращений к элементам массива a при очередном выполнении оператора S_β . Выполнение оператора S_β при конкретных значениях β и вектора параметров цикла J будем называть операцией и обозначать $S_\beta(J)$. Пара вхождений (a, S_α, p) и (a, S_β, q) порождает зависимость $S_\alpha(I) \rightarrow S_\beta(J)$, если $S_\alpha(I)$ выполняется раньше $S_\beta(J)$; $S_\alpha(I)$ и $S_\beta(J)$ используют один и тот же элемент какого-либо массива, и по крайней мере одно из использований есть переопределение элемента; между операциями $S_\alpha(I)$ и $S_\beta(J)$ этот элемент не переопределяется.

Зависимости $S_\alpha(I) \rightarrow S_\beta(J)$ между операциями вложенных циклов будем характеризовать векторами $d^{\alpha, \beta} = J - I$. Если размерности векторов J и I не совпадают, то $J - I$ определяется как раз-

ность векторов меньшей из размерностей. Векторы $d^{\alpha,\beta}$ часто называют векторами зависимостей: $d^{\alpha,\beta}$ определяет для операции $S_\beta(J)$ операцию $S_\alpha(I)$, от которой $S_\beta(J)$ зависит.

Тайлинг. Выделим макрооперации-тайлы путем разбиения циклов, окружающих операторы. При тайлинге каждый цикл разбивается на два цикла: глобальный, параметр которого определяет на данном уровне вложенности порядок вычисления тайлов, и локальный, в котором параметр исходного цикла изменяется в границах одного тайла. Если разбиение не производить и все итерации считать принадлежащими глобальному циклу, то получим так называемый глобальный не разбиваемый цикл; если все итерации отнести к локальному циклу, то получим локальный не разбиваемый цикл. Перестановкой и распределением циклов алгоритм преобразуется таким образом, чтобы глобальные циклы были внешними по отношению к локальным.

Пусть $m_\zeta^\theta = \min_{J(j_1, j_2, \dots, j_{n^\theta}) \in V^\theta} j_\zeta$, $M_\zeta^\theta = \max_{J(j_1, j_2, \dots, j_{n^\theta}) \in V^\theta} j_\zeta$, $1 \leq \zeta \leq n^\theta$, – предельные значения

изменения параметров циклов. Размеры тайлов задаются натуральными числами $r_1^\theta, \dots, r_{n^\theta}^\theta$.

Параметр r_ζ^θ обозначает число значений параметра j_ζ , приходящихся на один тайл θ -го набора операторов. Число частей Q_ζ^θ , на которые при формировании тайлов разбивается область значений параметра j_ζ цикла, окружающего θ -й набор операторов, находится согласно $Q_\zeta^\theta = \lceil (M_\zeta^\theta - m_\zeta^\theta + 1) / r_\zeta^\theta \rceil$ ($\lceil \cdot \rceil$ обозначает ближайшее «сверху» целое число). Тайлы нумеруются по каждой координате в пределах от 0 до $Q_\zeta^\theta - 1$, $1 \leq \zeta \leq n^\theta$.

Обозначим $V^{\theta, gl} = \{J^{gl}(j_1^{gl}, \dots, j_{n^\theta}^{gl}) \mid 0 \leq j_\zeta^{gl} \leq Q_\zeta^\theta - 1, 1 \leq \zeta \leq n^\theta\}$ – область изменения параметров глобальных, т. е. уровня тайлов, циклов. Если цикл является глобальным не разбиваемым, то условимся, что границы изменения этого цикла тайлинг оставляет прежними. Каждый тайл можно обозначить некоторым вектором J^{gl} или, подробнее, вектором $J^{gl}(j_1^{gl}, \dots, j_{n^\theta}^{gl})$. Обозначим $V_{J^{gl}}^\theta = \{J(j_1, \dots, j_{n^\theta}) \in V^\theta \mid m_\zeta^\theta + j_\zeta^{gl} r_\zeta^\theta \leq j_\zeta \leq m_\zeta^\theta - 1 + (j_\zeta^{gl} + 1)r_\zeta^\theta, 1 \leq \zeta \leq n^\theta\}$ – область изменения параметров локальных (уровня операций тайлов) циклов при фиксированных значениях параметров глобальных циклов $J^{gl} \in V^{\theta, gl}$. Если это множество пустое, то тайл J^{gl} называется пустым, если это множество совпадает с множеством $\{J(j_1, \dots, j_{n^\theta}) \in \mathbb{Z}^{n^\theta} \mid m_\zeta^\theta + j_\zeta^{gl} r_\zeta^\theta \leq j_\zeta \leq m_\zeta^\theta - 1 + (j_\zeta^{gl} + 1)r_\zeta^\theta, 1 \leq \zeta \leq n^\theta\}$, то полным, иначе – неполным тайлом. Описанный подход к разбиению множества операций алгоритма на макрооперации-тайлы, при котором допускаются избыточные области изменения параметров глобальных циклов (некоторые множества $V_{J^{gl}}^\theta$ могут быть пустыми), называется методом окаймления [8].

Итерации, порождающие зависимость, принадлежат некоторым тайлам. Таким образом, порождается зависимость уровня тайлов: данное, вычисленное при выполнении операции тайла I^{gl} , используется при выполнении зависимой операции тайла J^{gl} . Зависимость уровня тайлов будем характеризовать векторами глобальных зависимостей: $d^{\alpha,\beta, gl} = J^{gl} - I^{gl}$.

Достаточными условиями допустимости тайлинга циклов произвольной структуры вложенности являются неотрицательность координат всех векторов зависимостей ($d_\zeta^{\alpha,\beta} \geq 0, 1 \leq \zeta \leq n^\theta$) и отсутствие зависимостей операторов с меньшим номером от операторов с большим номером (допускается только $\theta^\beta \geq \theta^\alpha$) [2]. Если рассматривать тайлы произвольного фиксированного размера, то эти достаточные условия являются также и необходимыми; при допущении наличия глобальных и/или локальных не разбиваемых циклов достаточные условия можно ослабить [5].

Рассмотрим алгоритм следующего вида:

```

do t = 1, T
  do j2 = m2, M2
    .....
    do jn = mn, Mn
      Набор операторов
    enddo
  .....
enddo
enddo

```

(1)

Здесь выделен самый внешний цикл, выполняющий некоторое число итераций T . Наличие такого особого цикла типично для сеточных методов численного решения дифференциальных уравнений: параметр цикла t задает номер временного слоя или номер итерации итерационного алгоритма. В записи алгоритма (1) только один набор операторов, поэтому верхний индекс θ в обозначениях не будем указывать.

Следующая лемма является следствием результатов работы [5] и устанавливает для случая алгоритма (1) более слабые, чем известное условие неотрицательности всех координат всех векторов зависимостей, достаточные условия допустимости тайлинга.

Лемма. Пусть алгоритм имеет вид (1), $n \geq 4$, самый внешний цикл является глобальным не разбиваемым, следующие три цикла, обозначим их параметры j_η, j_ξ, j_τ , являются разбиваемыми, остальные (если $n > 4$) – локальными не разбиваемыми. Пусть зависимости вида

$$S_\alpha(I(i_1, \dots, i_n)) \rightarrow S_\beta(J(j_1, \dots, j_n)), \quad i_1 = j_1,$$

либо отсутствуют, либо для них выполняются условия

$$j_\eta \geq i_\eta, j_\xi \geq i_\xi, j_\tau \geq i_\tau. \tag{2}$$

Тогда тайлинг допустим.

Запишем алгоритм (1) после преобразования тайлинга, определяемого предположениями леммы:

```

do t = 1, T
  do jηgl = 0, Qη - 1
    do jξgl = 0, Qξ - 1
      do jτgl = 0, Qτ - 1
        Tile(t, jηgl, jξgl, jτgl)
      enddo
    enddo
  enddo
enddo

```

(3)

Здесь тайл $\text{Tile}(t, j_\eta^{gl}, j_\xi^{gl}, j_\tau^{gl})$ имеет следующий вид:

```

do jη = mη + jηgl rη, min(mη - 1 + (jηgl + 1)rη, Mη)
  do jξ = mξ + jξgl rξ, min(mξ - 1 + (jξgl + 1)rξ, Mξ)
    do jτ = mτ + jτgl rτ, min(mτ - 1 + (jτgl + 1)rτ, Mτ)
      do j = ... // локальный не разбиваемый цикл
        Набор операторов
      enddo
    enddo
  enddo
enddo

```

Указанный в тайле локальный не разбиваемый цикл может быть не один или, напротив, отсутствовать.

Распределение зерен вычислений между двумерными процессами. Рассмотрим следующий способ получения 2D зернистых (т. е. уровня макроопераций-тайлов) вычислительных процессов. Напомним, что каждому тайлу соответствует некоторый набор операторов θ . Зафиксируем два глобальных цикла с уровнями вложенности η, ξ таких, что цикл с параметром j_ξ^{gl} окружен циклом с параметром j_η^{gl} (пока не предполагается, что алгоритм имеет вид (3)), и два таких числа P_η, P_ξ , что $2 \leq P_\eta \leq Q_\eta^0, 2 \leq P_\xi \leq Q_\xi^0$. К одному 2D-процессу с координатами $(j_\eta^{gl} \bmod P_\eta, j_\xi^{gl} \bmod P_\xi)$ отнесем вычисления всех тайлов с одинаковыми значениями функций

$$\text{Pr}^\theta \left(J^{gl} (j_1^{gl}, \dots, j_n^{gl}) \right) = (j_\eta^{gl} \bmod P_\eta, j_\xi^{gl} \bmod P_\xi), \tag{4}$$

если тайл J^{gl} окружен циклами с параметрами j_η^{gl}, j_ξ^{gl} (в этом случае набор операторов θ исходного алгоритма окружен циклами с параметрами j_η и j_ξ). Отметим, что формально можно положить $P_\eta = 1$ или $P_\xi = 1$, но в этом случае структура вычислительных процессов является одномерной.

Определим компоненты функции $\text{Pr}^\theta(J^{gl})$ в случае, когда тайл J^{gl} не окружен циклами с параметрами j_η^{gl}, j_ξ^{gl} . Положим обе компоненты функции $\text{Pr}^\theta(J^{gl})$ равными 0, если тайл расположен выше цикла с параметром j_η^{gl} ; положим первую и вторую компоненты $\text{Pr}^\theta(J^{gl})$ равными $(Q_\eta^0 - 1) \bmod P_\eta$ и $(Q_\xi^0 - 1) \bmod P_\xi$ соответственно, если тайл расположен ниже цикла с параметром j_η^{gl} ; положим вторую компоненту $\text{Pr}^\theta(J^{gl})$ равной 0, если тайл окружен циклом с параметром j_η^{gl} и расположен выше цикла с параметром j_ξ^{gl} ; положим вторую компоненту $\text{Pr}^\theta(J^{gl})$ равной $(Q_\xi^0 - 1) \bmod P_\xi$, если тайл окружен циклом с параметром j_η^{gl} и расположен ниже цикла с параметром j_ξ^{gl} .

Вычислительный процесс с координатами (p_η, p_ξ) будем обозначать $\text{Pr}_{p_\eta, p_\xi}^\theta$.

В случае алгоритма вида (1) (или, после тайлинга, алгоритма вида (3)) единый псевдокод для каждого из 2D вычислительных процессов $\text{Pr}_{p_\eta, p_\xi}^\theta, 0 \leq p_\eta \leq P_\eta - 1, 0 \leq p_\xi \leq P_\xi - 1$, порождаемых функцией (4), можно записать следующим образом:

```
do t = 1, T
do  $j_\eta^{gl} = p_\eta, Q_\eta - 1, P_\eta$  // Цикл с шагом  $P_\eta$ 
do  $j_\xi^{gl} = p_\xi, Q_\xi - 1, P_\xi$  // Цикл с шагом  $P_\xi$ 
do  $j_\tau^{gl} = 0, Q_\tau - 1$ 
Tile( $t, j_\eta^{gl}, j_\xi^{gl}, j_\tau^{gl}$ )
enddo
enddo
enddo
```

Если $P_\eta = Q_\eta$ и $P_\xi = Q_\xi$, то циклы с параметрами j_η^{gl} и j_ξ^{gl} содержат только по одной итерации. В этом случае алгоритм (5) для каждого из 2D вычислительных процессов $\text{Pr}_{p_\eta, p_\xi}^\theta, 0 \leq p_\eta \leq P_\eta - 1, 0 \leq p_\xi \leq P_\xi - 1$, примет вид

```

do  $t = 1, T$ 
  do  $j_\tau^{gl} = 0, Q_\tau - 1$ 
    Tile( $t, p_\eta, p_\xi, j_\tau^{gl}$ )
  enddo
enddo

```

(6)

Параллельность двумерных зернистых вычислительных процессов. Исследуем для алгоритма вида (1) загруженность работой вычислительных процессов, организованных в двумерную структуру $\text{Pr}_{p_\eta, p_\xi}$, $0 \leq p_\eta \leq P_\eta - 1$, $0 \leq p_\xi \leq P_\xi - 1$. Исследования проведем в рамках абстрактной модели вычислений: все макрооперации зернистого алгоритма обеспечены необходимыми данными и одинаковы по длительности выполнения [6], возможные временные затраты на коммуникационные операции и на обработку пустых тайлов не учитываются. Примем, что множество операций любого тайла выполняется за одну единицу времени. Эти предположения, а также ограничения на вид рассматриваемых алгоритмов, накладываются для возможности формулировать и строго доказывать содержательные утверждения.

Определим продолжительность разгона (торможения) вычислений – число единиц времени, когда еще не все процессы начали (закончили) выполняться.

Т е о р е м а. Пусть выполняются предположения леммы и следующие условия:

$$Q_\tau \gg Q_\eta, \quad Q_\tau \gg Q_\xi; \quad (7)$$

$$\frac{Q_\eta}{P_\eta} \in \mathbb{Z}, \quad \frac{Q_\xi}{P_\xi} \in \mathbb{Z}; \quad (8)$$

и

$$\left| d_\tau^{gl} \right| < Q_\tau - (Q_\eta - 1) - (Q_\xi - 1), \quad (9)$$

если для какого-либо вектора глобальных зависимостей $d^{\alpha, \beta, gl} = (d_1^{gl}, d_\eta^{gl}, d_\xi^{gl}, d_\tau^{gl}, \dots)$ имеет место $d_1^{gl} = 1$, $d_\tau^{gl} < 0$, а также $d_\eta^{gl} < 0$ и/или $d_\xi^{gl} < 0$. Тогда, без учета разгона и торможения вычислений, все зернистые вычислительные процессы в каждый момент времени загружены работой; продолжительности разгона и торможения вычислений намного меньше общей продолжительности вычислений.

Д о к а з а т е л ь с т в о. Предположим сначала, что $P_\eta = Q_\eta$, $P_\xi = Q_\xi$. В этом случае каждый из 2D вычислительных процессов $\text{Pr}_{p_\eta, p_\xi}$, $0 \leq p_\eta \leq P_\eta - 1$, $0 \leq p_\xi \leq P_\xi - 1$, выполняет алгоритм (6).

Из леммы следует, что первые четыре координаты всех векторов зависимостей неотрицательны для итераций алгоритма с фиксированным t :

$$d^{\alpha, \beta} = J(t, j_\eta, j_\xi, j_\tau, \dots, j_n) - I(t, i_\eta, i_\xi, i_\tau, \dots, i_n) = (0, j_\eta - i_\eta, j_\xi - i_\xi, j_\tau - i_\tau, \dots).$$

Тогда первые четыре координаты векторов глобальных зависимостей также являются неотрицательными. Отсюда следует, что для каждого t любой процесс $\text{Pr}_{p_\eta, p_\xi}$ выполняет свои Q_τ макроопераций Tile($t, p_\eta, p_\xi, j_\tau^{gl}$) в подряд идущие моменты времени, причем процесс $\text{Pr}_{p_\eta, p_\xi}$ начнет выполнение макроопераций не позже, чем через $p_\eta + p_\xi$ единиц времени по сравнению с процессом $\text{Pr}_{0,0}$.

Условия (7), (9) гарантируют, что процесс $\text{Pr}_{0,0}$ (а значит, и все другие процессы) сразу после выполнения Q_τ макроопераций на текущей итерации t начнет выполнять Q_τ макроопераций при t на следующей итерации.

Таким образом, разгон вычислений происходит при $t = 1$, его продолжительность не превосходит $P_\eta + P_\xi - 2$, торможение вычислений происходит при $t = T$, продолжительность его также равна $P_\eta + P_\xi - 2$. Общее время вычислений алгоритма (6) не превосходит $TQ_\tau + P_\eta + P_\xi - 2$. Минимально возможное время вычислений равно TQ_τ и достигается в том случае, когда зависи-

мости позволяют начать всем процессам выполнять макрооперации одновременно. Вне разгона и торможения вычислений все зернистые вычислительные процессы в каждый момент времени загружены работой. Сравнивая величины $2(P_\eta + P_\xi - 2) = 2(Q_\eta + Q_\xi - 2)$ и $TQ_\tau + P_\eta + P_\xi - 2 = TQ_\tau + Q_\eta + Q_\xi - 2$ заключаем, что суммарное время разгона и торможения вычислений (напомним, по условию $Q_\tau \gg Q_\eta, Q_\tau \gg Q_\xi$) намного меньше общего времени вычислений алгоритма (6).

Предположим теперь, что $P_\eta < Q_\eta$ и $P_\xi < Q_\xi$ («промежуточный» случай $P_\eta < Q_\eta$ или $P_\xi < Q_\xi$ можно рассмотреть аналогично). Тогда 2D вычислительные процессы выполняют алгоритм (5).

В одном процессе производятся вычисления, приписанные $\frac{Q_\eta}{P_\eta} \frac{Q_\xi}{P_\xi}$ процессам (см. условия (8)) рассмотренного случая $P_\eta = Q_\eta$ и $P_\xi = Q_\xi$. Суммарная продолжительность разгона и торможения вычислений (будем считать, что они требуются) равна $2(P_\eta + P_\xi - 2)$, что намного меньше $T \frac{Q_\eta}{P_\eta} \frac{Q_\xi}{P_\xi} Q_\tau + P_\eta + P_\xi - 2$ – общего времени вычислений алгоритма (5).

Теорема доказана.

Условия полной (не считая разгона и торможения вычислений) загруженности всех зернистых вычислительных процессов, организованных в одномерную структуру, исследованы в [4, 5].

З а м е ч а н и е 1. Условия теоремы, как следует из доказательства, могут быть ослаблены, если хотя бы один из циклов, порождающих координаты вычислительных процессов, является параллельным. Например, если параллельным является цикл с параметром j_η , то не требуется выполнения $Q_\tau \gg Q_\eta$, а суммарная продолжительность разгона и торможения вычислений не превосходит $2(P_\xi - 1)$.

З а м е ч а н и е 2. При выполнении условий (2) циклы с параметрами j_η, j_ξ, j_τ переставляемы, поэтому теорема верна и в случае произвольного первоначального порядка следования этих циклов.

З а м е ч а н и е 3. Пусть в алгоритме (1) отсутствует самый внешний цикл. Утверждение теоремы остается верным, если потребовать выполнения условия (2) для каждой зависимости.

П р и м е р. Рассмотрим трехмерную задачу Дирихле для уравнения Пуассона, заданного в области $G = [0 < x_1 < l_1] \times [0 < x_2 < l_2] \times [0 < x_3 < l_3]$ с границей ∂G :

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \frac{\partial^2 u}{\partial x_3^2} - au = f(x_1, x_2, x_3), (x_1, x_2, x_3) \in G, a > 0,$$

$$u(x_1, x_2, x_3)|_{\partial G} = \mu(x_1, x_2, x_3).$$

Для численного решения задачи введем в области $G + \partial G$ сетку узлов $\{(ih_1, jh_2, kh_3) | i = 0, 1, \dots, N_x, N_x h_1 = l_1, j = 0, 1, \dots, N_y, N_y h_2 = l_2, k = 0, 1, \dots, N_z, N_z h_3 = l_3\}$. Используем семиточечный шаблон для аппроксимации значений производных и сеточного представления функций [9]. Основную часть алгоритма численного решения методом последовательной верхней релаксации можно представить в виде (1):

```
do t = 1, T // T – некоторое фиксированное число итераций
  do i = 1, N_x - 1
    do j = 1, N_y - 1
      do k = 1, N_z - 1
        y(i,j,k) = F(y(i-1,j,k), y(i,j-1,k), y(i,j,k-1), y(i,j,k), y(i+1,j,k), y(i,j+1,k), y(i,j,k+1))
      enddo
    enddo
  enddo
enddo
enddo
```

где функция F вычисляется по формуле

$$\begin{aligned}
& F(y(i-1, j, k), y(i, j-1, k), y(i, j, k-1), y(i, j, k), y(i+1, j, k), y(i, j+1, k), y(i, j, k+1)) = \\
& = (1-\omega)y(i, j, k) + \omega \left((y(i+1, j, k) + y(i-1, j, k)) / h_1^2 + (y(i, j+1, k) + y(i, j-1, k)) / h_2^2 + \right. \\
& \quad \left. + (y(i, j, k+1) + y(i, j, k-1)) / h_3^2 - f_{i,j,k} \right) / \left(2 / h_1^2 + 2 / h_2^2 + 2 / h_3^2 + a \right).
\end{aligned}$$

Все зависимости алгоритма верхней релаксации являются однородными и выражаются векторами зависимостей $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$, $(1,0,0,0)$, $(1,-1,0,0)$, $(1,0,-1,0)$, $(1,0,0,-1)$. Осуществим тайлинг. При фиксированном t зависимости между итерациями алгоритма выражаются векторами $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$ с неотрицательными координатами, поэтому достаточное условие допустимости тайлинга (утверждение леммы) выполнено.

Преобразуем циклы с параметрами i, j, k в двумерную циклическую конструкцию. Каждый из указанных циклов разбивается на глобальный и локальный: глобальные циклы определяют порядок вычисления тайлов, локальные циклы определяют порядок вычисления итераций исходного алгоритма в границах одного тайла. Зададим r_2, r_3, r_4 – размеры тайла; тогда

$$Q_2 = \left\lceil \frac{N_x - 1}{r_2} \right\rceil, \quad Q_3 = \left\lceil \frac{N_y - 1}{r_3} \right\rceil, \quad Q_4 = \left\lceil \frac{N_z - 1}{r_4} \right\rceil$$
 – число тайлов по измерениям. После разбиения

и перестановки циклов получим

```

do  $t = 1, T$ 
do  $i^{gl} = 0, Q_2 - 1$ 
do  $j^{gl} = 0, Q_3 - 1$ 
do  $k^{gl} = 0, Q_4 - 1$ 
  // Начало тайла  $\text{Tile}(t, i^{gl}, j^{gl}, k^{gl})$ 
do  $i = 1 + i^{gl}r_2, \min((i^{gl} + 1)r_2, N_x - 1)$ 
do  $j = 1 + j^{gl}r_3, \min((j^{gl} + 1)r_3, N_y - 1)$ 
do  $k = 1 + k^{gl}r_4, \min((k^{gl} + 1)r_4, N_z - 1)$ 
   $y(i, j, k) = F(y(i-1, j, k), y(i, j-1, k), y(i, j, k-1), y(i, j, k), y(i+1, j, k), y(i, j+1, k), y(i, j, k+1))$ 
enddo
enddo
enddo
  // Конец тайла  $\text{Tile}(t, i^{gl}, j^{gl}, k^{gl})$ 
enddo
enddo
enddo

```

Пусть $j_\eta = i, j_\xi = j, j_\tau = k, P_\eta \times P_\xi$ – число процессов, предназначенных для реализации алгоритма. Выберем, следуя теореме, $Q_4 \gg Q_2, Q_4 \gg Q_3, \frac{Q_2}{P_2} \in \mathbb{Z}, \frac{Q_3}{P_3} \in \mathbb{Z}$. Запишем псевдокод параллельного алгоритма с полной (не считая разгона и торможения вычислений) загруженностью зернистых вычислительных процессов. Для каждого процесса $\text{Pr}_{p_\eta, p_\xi}, 0 \leq p_\eta \leq P_\eta - 1, 0 \leq p_\xi \leq P_\xi - 1$:

```

do  $t = 1, T$ 
do  $i^{gl} = p_\eta, Q_2 - 1, P_\eta$  // Цикл с шагом  $P_\eta$ 
do  $j^{gl} = p_\xi, Q_3 - 1, P_\xi$  // Цикл с шагом  $P_\xi$ 
do  $k^{gl} = 0, Q_4 - 1$ 
  получение данных
   $\text{Tile}(t, i^{gl}, j^{gl}, k^{gl})$ 
  отправка данных
enddo
enddo
enddo
enddo

```


Произведем оценку объема передаваемых данных. Объем данных, отправляемых одним процессом на одной итерации t , оценивается (в случае $P_\eta > 1, P_\xi > 1$) величиной $2 \left(r_2 N_z \frac{Q_3}{P_\xi} + r_3 N_z \frac{Q_2}{P_\eta} \right)$.

Объем коммуникаций, осуществляемых всеми процессами, оценивается величиной $2TP_\eta P_\xi N_z \left(\frac{N_x Q_3}{Q_2 P_\xi} + \frac{N_y Q_2}{Q_3 P_\eta} \right)$. Оценим объем данных, отправляемых одним процессом на одной итерации t , в случае $Q_2 = 1$ одномерной вычислительной структуры (тогда имеет место $P_\eta = 1, r_2 = N_x - 1$): $2N_x N_z \frac{Q_3}{P_\xi}$. Объем коммуникационных операций, осуществляемых всеми процессами на всех итерациях t , оценивается величиной $2TN_x N_z Q_3$.

Пусть, например, $N_x = N_y = N_z = N$. Если P^2 процессов организованы в двумерную структуру и $P_\eta \times P_\xi = P \times P = P^2, Q_2 = Q_3 = P$, то объем пересылаемых данных, осуществляемых всеми процессами, оценивается величиной $4TN^2P$. Если P^2 процессов организовать в одномерную структуру и, для определенности, $Q_2 = 1, P_\eta = 1, Q_3 = P_\xi = P^2$, то объем коммуникационных операций оценивается величиной $2TN^2P^2$, что при большом числе процессов P существенно больше $4TN^2P$.

Таким образом, при достаточно большом числе процессов использование 2D-структуры позволило существенно уменьшить объем коммуникационных операций. В этом примере можно отметить и другие преимущества двумерных структур по сравнению с одномерными (при реализации алгоритмов на параллельных компьютерах с распределенной памятью): уменьшение разгона и торможения вычислений, возможность получения большего числа вычислительных процессов при сравнительно небольшом размере сетки узлов.

Благодарности. Работа выполнена в рамках Государственной программы научных исследований Республики Беларусь «Конвергенция-2020», подпрограмма «Методы математического моделирования сложных систем».

Acknowledgments. The prepared report was sponsored by the Government program of scientific research of the Republic of Belarus “Convergence-2020”, subprogram “Methods of mathematical modeling of complex systems”.

Список использованных источников

1. Xue, J. Time-minimal tiling when rise is larger than zero / J. Xue, W. Cai // *Parallel Computing*. – 2002. – Vol. 28, №. 6. – P. 915–939. [https://doi.org/10.1016/s0167-8191\(02\)00098-4](https://doi.org/10.1016/s0167-8191(02)00098-4)
2. Kim, D. Parameterized tiling for imperfectly nested loops / D. Kim, S. Rajopadhye // Technical Report CS-09-101. – Colorado State University, Department of Computer Science, February 2009. – 21 p.
3. Dathathri, R. Compiling Affine Loop Nests for a Dynamic Scheduling Runtime on Shared and Distributed Memory / R. Dathathri, R. T. Mullapudi, U. Bondhugula // *ACM Transactions on Parallel Computing (TOPC)*. – 2016. – Vol. 3, №. 2. – P. 1–28. <https://doi.org/10.1145/2948975>
4. Лиходед, Н. А. Параллельные последовательности зернистых вычислений / Н. А. Лиходед, А. А. Толстикова // Докл. Нац. акад. наук Беларуси. – 2010. – Т. 54, № 4. – С. 36–41.
5. Толстикова, А. А. Корректность разбиений алгоритмов при организации зернистых параллельных вычислительных процессов / А. А. Толстикова, Н. А. Лиходед // Междунар. конгресс по информатике: информационные системы и технологии CSIST'2011, 31 окт. – 3 нояб. 2011 г. – Минск: БГУ, 2011. – Т. 2. – С. 122–126.
6. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.
7. Соболевский, П. И. Двухуровневый тайлинг и его применение при пространственно-временном отображении алгоритмов на параллельные архитектуры / П. И. Соболевский, С. В. Баханович // Вест. Нац. акад. наук Беларуси. Сер. физ.-мат. наук. – 2016. – № 2. – С. 85–97.
8. Parameterized tiled loops for free / L. Renganarayanan [et al.] // *ACM SIGPLAN Conference on Programming Language Design and Implementation*, San Diego, California, USA, June 2007. – [S. l.], 2007. – P. 126–138. <https://doi.org/10.1145/1250734.1250780>
9. Самарский, А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. – М.: Наука, 1978. – 592 с.

References

1. Xue J., Cai W. Time-minimal tiling when rise is larger than zero. *Parallel Computing*, 2002, vol. 28, no. 6, pp. 915–939. [https://doi.org/10.1016/s0167-8191\(02\)00098-4](https://doi.org/10.1016/s0167-8191(02)00098-4)

2. Kim D., Rajopadhye S. *Parameterized Tiling for Imperfectly Nested Loops. Technical Report CS-09-101*. Colorado State University, Department of Computer Science, February 2009. 21 p.
3. Dathathri R., Mullapudi R. T., Bondhugula U. Compiling Affine Loop Nests for a Dynamic Scheduling Runtime on Shared and Distributed Memory. *ACM Transactions on Parallel Computing (TOPC)*, 2016, vol. 3, no. 2, pp. 1–28. <https://doi.org/10.1145/2948975>
4. Likhoded N. A., Tolstikov A. A. Parallel sequences of grain computations. *Doklady Natsional'noi akademii nauk Belarusi = Doklady of the National Academy of Sciences of Belarus*, 2010, vol. 54, no. 4, pp. 36–41 (in Russian).
5. Tolstikov A. A., Likhoded N. A. Correctness of tiling of algorithms for organization of tiled parallel computational processes. *Mezhdunarodnyi kongress po informatike: informatsionnye sistemy i tekhnologii CSIST'2011, 31 oktyabrya – 3 noyabrya 2011 g. T. 2* [International Congress on Computer Science: Information Systems and Technologies CSIST'2011, November 2011. Vol. 2]. Minsk, Belarusian State University, 2011, pp. 122–126 (in Russian).
6. Voevodin V. V., Voevodin V. V. *Parallel Computations*. St. Petersburg, BKhV-Petersburg Publ., 2002. 608 p. (in Russian).
7. Sobolevsky P. I., Bakhanovich S. V. Two-level tiling and its application in the space-time mapping of algorithms onto parallel architectures. *Vestsi Natsyianal'nai akademii navuk Belarusi. Seryia fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2016, no. 2, pp. 85–97 (in Russian).
8. Renganarayanan L., Kim D., Rajopadhye S., Strout M. M. Parameterized tiled loops for free. *Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation - PLDI '07*. San Diego, California, USA, June 2007, pp. 126–138. <https://doi.org/10.1145/1250734.1250780>
9. Samarskii A. A., Nikolaev E. S. *Methods for Solving of the Grid Equations*. Moscow, Nauka Publ., 1978. 592 p. (in Russian).

Информация об авторах

Лиходед Николай Александрович – доктор физико-математических наук, профессор кафедры вычислительной математики факультета прикладной математики и информатики, Белорусский государственный университет (пр. Независимости, 4, 220030, г. Минск, Республика Беларусь). E-mail: likhoded@bsu.by.

Полещук Максим Александрович – ассистент кафедры вычислительной математики факультета прикладной математики и информатики, Белорусский государственный университет (пр. Независимости, 4, 220030, г. Минск, Республика Беларусь). E-mail: poleschuma@bsu.by.

Information about the authors

Nikolai A. Likhoded – D. Sc. (Physics and Mathematics), Professor of the Department of Applied Mathematics, Belarusian State University (4, Nezavisimosti Str., 220030, Minsk, Republic of Belarus). E-mail: likhoded@bsu.by

Maksim A. Paliashchuk – Junior Researcher, Belarusian State University (4, Nezavisimosti Str., 220030, Minsk, Republic of Belarus). E-mail: poleschuma@bsu.by