

МАТЕМАТИКА
MATHEMATICS

УДК 519.67

<https://doi.org/10.29235/1561-8323-2020-64-6-647-656>

Поступило в редакцию 02.11.2020

Received 02.11.2020

Н. А. Лиходед, А. А. Толстикова*Белорусский государственный университет, Минск, Республика Беларусь***МЕТОД ОЦЕНКИ ЛОКАЛЬНОСТИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ,
ОРИЕНТИРОВАННЫХ НА КОМПЬЮТЕРЫ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ***(Представлено членом-корреспондентом Л. А. Яновичем)*

Аннотация. Степень использования памяти с быстрым доступом отражает вычислительное свойство алгоритма, называемое локальностью. Для параллельных компьютеров с распределенной памятью быстрой считается локальная память вычислительного узла. При реализации алгоритмов на многопроцессорных вычислительных устройствах использование локальности играет важнейшую роль для достижения высокой производительности. Основной задачей исследования локальности параллельного алгоритма является оценка числа и объема коммуникационных операций. В этой работе сформулированы и доказаны утверждения, позволяющие получить асимптотические оценки объема коммуникационных операций вычислительных процессов, реализуемых на параллельных компьютерах с распределенной памятью. Получены выражения, характеризующие число данных, для которых требуются коммуникации, и число процессов, вовлеченных в пересылки этих данных. Эти оценки могут быть использованы для сравнения коммуникационных затрат при получении альтернативных вариантов параллельных алгоритмов.

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

Для цитирования. Лиходед, Н. А. Метод оценки локальности параллельных алгоритмов, ориентированных на компьютеры с распределенной памятью / Н. А. Лиходед, А. А. Толстикова // Докл. Нац. акад. наук Беларуси. – 2020. – Т. 64, № 6. – С. 647–656. <https://doi.org/10.29235/1561-8323-2020-64-6-647-656>

Nikolai A. Likhoded, Aliaksei A. Tolstikau*Belarusian State University, Minsk, Republic of Belarus***LOCALITY ESTIMATION OF PARALLEL ALGORITHM FOR DISTRIBUTED MEMORY COMPUTERS***(Communicated by Corresponding Member Leonid A. Yanovich)*

Abstract. Locality is an algorithm characteristic describing a usage level of fast access memory. For example, in case of distributed memory computers we focus on memory of each computational node. To achieve the high performance of algorithm implementation one should choose the best possible locality option. Studying the parallel algorithm locality is to estimate the number and volume of data communications. In this work, we formulate and prove the statements for computers with distributed memory that allow us to estimate the asymptotic volume of data communication operations. These estimation results are useful while comparing alternative versions of parallel algorithms during data communication cost analysis.

Keywords: parallel computing, distributed memory parallel computer, data communication reducing

For citation: Likhoded N. A., Tolstikau A. A. Locality estimation of parallel algorithm for distributed memory computers. *Doklady Natsional'noi akademii nauk Belarusi = Doklady of the National Academy of Sciences of Belarus*, 2020, vol. 64, no. 6, pp. 647–656 (in Russian). <https://doi.org/10.29235/1561-8323-2020-64-6-647-656>

Введение. Локальность алгоритма – это вычислительное свойство алгоритма, отражающее степень использования памяти с быстрым доступом. При многопроцессорной обработке памяти с быстрым доступом считается локальная память процессора, при однопроцессорной – кэш-память. При реализации алгоритмов на многопроцессорных вычислительных устройствах

использование локальности играет важнейшую роль для достижения высокой производительности [1].

В качестве целевого компьютера для реализации алгоритмов будем рассматривать параллельный компьютер с распределенной памятью. Параллельные алгоритмы для компьютеров с распределенной памятью должны быть зернистыми: множество операций алгоритма должно быть разбито на множества, называемые зёрнами вычислений. Можно использовать тайлинг (tiling) – преобразование алгоритма для получения макроопераций-тайлов [2; 3]. Операции одного тайла выполняются атомарно, как одна единица вычислений, а обмен данными происходит массивами.

Локальность параллельного алгоритма, предназначенного для реализации на компьютерах с распределенной памятью, характеризует коммуникационные затраты: чем меньше при заданном числе вычислительных ядер суммарный объем операций обмена данными, тем лучше локальность. Задачей исследования локальности параллельного алгоритма является оценка числа и объема коммуникационных операций. На локальность влияет также структура операций обмена данными: число и топология («близость») процессов, вовлеченных в групповые коммуникации. Хорошей локальности можно достичь путем преобразования алгоритмов и/или удачного распределения операций и данных между процессорами [4–7].

Целью этой работы является разработка метода оценки коммуникационных затрат альтернативных вариантов организации одномерных вычислительных процессов. Используется анализ информационных зависимостей, порождающих коммуникационные операции. Сформулированы и доказаны утверждения, позволяющие получить асимптотические оценки объема коммуникационных операций. Эти оценки могут быть использованы для сравнения коммуникационных затрат при получении альтернативных вариантов параллельных алгоритмов (в том числе и при автоматическом распараллеливании).

В [8] подобные оценки получены для случая, когда целевым вычислительным устройством является графический процессор. Отметим, что получение оценок для случая компьютера с распределенной памятью является задачей более сложной: для каждой из зависимостей, порождающих коммуникационные операции, в исследованиях участвуют не одна, как в случае графического процессора, а две координаты итерационных пространств; кроме того, в исследованиях участвуют две, не обязательно совпадающие, функции назначения операций процессорам.

Предварительные сведения. Приведем некоторые сведения о рассматриваемом классе алгоритмов и об информационных зависимостях между операциями алгоритма.

Будем считать, что алгоритм задан последовательной программой линейного класса [4]. Основную вычислительную часть такой программы составляют циклы произвольной структуры вложенности; границы изменения параметров циклов задаются неоднородными формами, линейными по совокупности параметров циклов и внешних переменных; шаги изменения параметров циклов равны 1. Пусть в гнезде циклов имеется K выполняемых операторов S_β и используется L массивов a_l . Область изменения параметров циклов для оператора S_β и размерность этой области обозначим соответственно V_β и n_β ; через a_l обозначим размерности массивов a_l .

Пусть в гнезде циклов 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 (число циклов, окружающих θ -й набор операторов).

Вхождением (a, S_β, q) будем называть q -е вхождение массива a в оператор S_β . Индексы элементов l -го массива, связанных с вхождением (a_l, S_β, q) , выражаются функцией вида

$$\bar{F}_{a_l, S_\beta, q}(J) = F_{a_l, S_\beta, q} J + f^{a_l, S_\beta, q}, \quad J(j_1, \dots, j_{n_\beta}) \in V_\beta, \quad F_{a_l, S_\beta, q} \in \mathbf{Z}^{n_l \times n_\beta}, \quad f^{a_l, S_\beta, q} \in \mathbf{Z}^{n_l}.$$

Выполнение оператора S_β при конкретных значениях β и вектора параметров цикла J будем называть операцией и обозначать $S_\beta(J)$. Пара вхождений $(a, S_\alpha, 1)$ и (a, S_β, q) порождает истинную зависимость $S_\alpha(I) \rightarrow S_\beta(J)$, если: $S_\alpha(I)$ выполняется раньше $S_\beta(J)$; $S_\alpha(I)$ переопределяет

(изменяет) элемент массива a , а $S_\beta(J)$ использует этот элемент массива в качестве аргумента; между операциями $S_\alpha(I)$ и $S_\beta(J)$ этот элемент не переопределяется. Истинные зависимости и входные данные алгоритма порождают коммуникационные операции.

Зависимости между операциями можно задать функциями вида

$$\bar{\Phi}_{\alpha,\beta}(J) = \Phi_{\alpha,\beta}J + \Psi_{\alpha,\beta}N - \varphi^{\alpha,\beta}, \quad J \in V_{\alpha,\beta}, \quad N \in \mathbb{Z}^s, \quad \Phi_{\alpha,\beta} \in \mathbb{Z}^{n_\alpha \times n_\beta}, \quad \Psi_{\alpha,\beta} \in \mathbb{Z}^{n_\alpha \times s}, \quad \varphi^{\alpha,\beta} \in \mathbb{Z}^{n_\alpha},$$

где $N \in \mathbb{Z}^s$ – вектор внешних переменных алгоритма; s – число внешних переменных. Функция зависимостей $\bar{\Phi}_{\alpha,\beta}(J)$ позволяет для операции $S_\beta(J)$ найти операцию $S_\alpha(I)$, от которой $S_\beta(J)$ зависит.

П р и м е р. Основную вычислительную часть алгоритма, реализующего двухшаговую разностную схему продольно-поперечной прогонки, можно представить в следующем виде (циклы, итерации которых можно выполнять независимо, записаны как *dopar*):

```

do j = 1, j_0
  dopar i_2 = 1, N_2 - 1
    do i_1 = 1, N_1 - 1
      S_1:   alpha(i_1 + 1) = F(alpha(i_1))
      S_2:   beta(i_1 + 1) = F_2(alpha(i_1), beta(i_1), y_(2)(i_1, i_2), y_(2)(i_1, i_2 - 1), y_(2)(i_1, i_2 + 1))
    enddo
    do i_1 = 1, N_1 - 1
      S_3:   y_(1)(N_1 - i_1, i_2) = alpha(N_1 - i_1 + 1) y_(1)(N_1 - i_1 + 1, i_2) + beta(N_1 - i_1 + 1)
    enddo
  enddopar
  dopar i_1 = 1, N_1 - 1
    do i_2 = 1, N_2 - 1
      S_4:   alpha(i_2 + 1) = F_3(alpha(i_2))
      S_5:   beta(i_2 + 1) = F_4(alpha(i_2), beta(i_2), y_(1)(i_1, i_2), y_(1)(i_1 - 1, i_2), y_(1)(i_1 + 1, i_2))
    enddo
    do i_2 = 1, N_2 - 1
      S_6:   y_(2)(i_1, N_2 - i_2) = alpha(N_2 - i_2 + 1) y_(2)(i_1, N_2 - i_2 + 1) + beta(N_2 - i_2 + 1)
    enddo
  enddopar
enddo(j)

```

Здесь $alpha(i)$ и $beta(i)$ – коэффициенты прогонок, возникающих при решении промежуточных систем линейных алгебраических уравнений. Конечные значения $y_{(2)}(i_1, i_2)$ являются выходными данными алгоритма.

Приведем функции истинных зависимостей алгоритма продольно-поперечной прогонки (найти их можно исходя из определения зависимостей или методом работы [4]): $\bar{\Phi}_{1,1}(j, i_2, i_1) = \bar{\Phi}_{1,2}(j, i_2, i_1) = \bar{\Phi}_{2,2}(j, i_2, i_1) = \bar{\Phi}_{3,3}(j, i_2, i_1) = (j, i_2, i_1) - (0, 0, 1)$,

$$\bar{\Phi}_{1,3}(j, i_2, i_1) = \bar{\Phi}_{2,3}(j, i_2, i_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} j \\ i_2 \\ i_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} j_0 \\ N_1 \\ N_2 \end{pmatrix},$$

$$\bar{\Phi}_{6,2}(j, i_2, i_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} j \\ i_2 \\ i_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} j_0 \\ N_1 \\ N_2 \end{pmatrix} - \begin{pmatrix} 1 \\ \varphi_2 \\ 0 \end{pmatrix},$$

где φ_2 равно 0, 1, -1,

$$\bar{\Phi}_{3,5}(j, i_1, i_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} j \\ i_1 \\ i_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} j_0 \\ N_1 \\ N_2 \end{pmatrix} - \begin{pmatrix} 0 \\ \Phi_2 \\ 0 \end{pmatrix},$$

где Φ_2 равно 0, 1, -1.

Функции $\bar{\Phi}_{4,4}(j, i_1, i_2)$, $\bar{\Phi}_{4,5}(j, i_1, i_2)$, $\bar{\Phi}_{5,5}(j, i_1, i_2)$, $\bar{\Phi}_{6,6}(j, i_1, i_2)$ задаются такими же матрицами и векторами, как $\bar{\Phi}_{1,1}$, функции $\bar{\Phi}_{4,6}(j, i_1, i_2)$, $\bar{\Phi}_{5,6}(j, i_1, i_2)$ – как $\bar{\Phi}_{1,3}$.

Организация параллельных зернистых вычислительных процессов. Как уже отмечалось, тайлинг – это преобразование алгоритма для получения макроопераций, называемых также зерном вычислений, или тайлами. 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 неразбиваемый цикл; если все итерации отнести к локальному циклу, то получим локальный неразбиваемый 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. Будем считать, что m_ζ^θ не зависит от внешних переменных, а M_ζ^θ может зависеть от внешних переменных: $M_\zeta^\theta = M^{\theta, N}(\zeta)N + M_\zeta^{\theta, 1}(M^{\theta, N}(\zeta) - \text{вектор-строка})$. Если два набора операторов имеют общий цикл с параметром j_ζ , то $m_\zeta^{\theta_1} = m_\zeta^{\theta_2}$, $M_\zeta^{\theta_1} = M_\zeta^{\theta_2}$.

Размеры тайлов задаются натуральными числами $r_1^\theta, \dots, r_{n^\theta}^\theta$. Параметр r_ζ^θ обозначает число значений параметра цикла 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 $Q_\zeta^\theta = \lceil (M_\zeta^\theta - m_\zeta^\theta + 1) / r_\zeta^\theta \rceil$ (использовано обозначение «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 $V^{\theta, \text{gl}}$, $V_{J^{\text{gl}}}^\theta$ – области изменения параметров соответственно глобальных (уровня тайлов) и локальных (уровня операций тайлов) циклов. Каждый тайл можно обозначить некоторым вектором J^{gl} или, подробнее, вектором $J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n^\theta}^{\text{gl}})$.

Будем рассматривать следующий способ получения зернистых (т. е. уровня макроопераций-тайлов) вычислительных процессов [9; 10]. К одному процессу отнесем вычисления тайлов с одинаковыми значениями функций $\text{Pr}^\theta(J^{\text{gl}})$, $1 \leq \theta \leq \Theta$, $J^{\text{gl}} \in V^{\theta, \text{gl}}$, отображающих тайлы на процессы. Будем полагать

$$\text{Pr}^\theta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n^\theta}^{\text{gl}})) = j_\zeta^{\text{gl}} \quad (1)$$

или

$$\text{Pr}^\theta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n^\theta}^{\text{gl}})) = Q_\zeta^\theta - j_\zeta^{\text{gl}} - 1. \quad (2)$$

Функция Pr^θ вида (1) или (2) задает вычислительный процесс $\text{Pr}^\theta(J^{\text{gl}})$ выполнения операций тайлов J^{gl} с одинаковыми значениями ζ -й координаты j_ζ^{gl} векторов J^{gl} .

Оценка коммуникационных затрат. Приведем сначала сведения о числе фиксированных данных массива, используемых на вхождениях в операторы. Термин «фиксированное данное массива» определяет конкретное, неизмененное содержимое соответствующей ячейки памяти.

Пусть вхождение (a_l, S_β, q) в правую часть некоторого оператора порождает истинную зависимость, $\bar{\Phi}_{\alpha, \beta}$ – функция зависимостей. Обозначим через $e_\zeta^{(n_\beta)}$ вектор-строку размера n_β , у которой координата с номером ζ равна 1, а остальные координаты нулевые,

$$\rho_{a_l, S_\beta, q} = \text{rank } F_{a_l, S_\beta, q}, \rho_{a_l, S_\beta, q}^\zeta = \text{rank} \begin{pmatrix} F_{a_l, S_\beta, q} \\ e_\zeta^{(n_\beta)} \end{pmatrix}, \rho_{a_l, S_\beta, q}^\Phi = \text{rank} \begin{pmatrix} F_{a_l, S_\beta, q} \\ \Phi_{\alpha, \beta} \end{pmatrix}, \rho_{a_l, S_\beta, q}^{\Phi, \zeta} = \text{rank} \begin{pmatrix} F_{a_l, S_\beta, q} \\ \Phi_{\alpha, \beta} \\ e_\zeta^{(n_\beta)} \end{pmatrix}.$$

Всегда имеет место либо $\rho_{a_l, S_\beta, q}^{\Phi, \zeta} = \rho_{a_l, S_\beta, q}^\Phi$, либо $\rho_{a_l, S_\beta, q}^{\Phi, \zeta} - 1 = \rho_{a_l, S_\beta, q}^\Phi$. Если вхождение (a_l, S_β, q) не порождает истинную зависимость, то по определению $\rho_{a_l, S_\beta, q}^\Phi = \rho_{a_l, S_\beta, q}$, $\rho_{a_l, S_\beta, q}^{\Phi, \zeta} = \rho_{a_l, S_\beta, q}^\zeta$.

Обозначим $M^\theta = \max_\zeta (M_\zeta^\theta - m_\zeta^\theta) + 1$ – наибольшее число итераций циклов, участвующих в получении тайлов. Для простоты записи будем использовать обозначение M без индекса θ , где будет подразумеваться набор операторов θ^β при упоминании оператора S_β . Фактически предполагается, что число итераций всех циклов, участвующих в получении тайлов, равно M . Подобного рода предположения накладываются для возможности формулировать и строго доказывать содержательные утверждения.

Отметим, что величина $Q_\zeta^\theta r_\zeta^\theta$ имеет порядок M . Если число частей Q_ζ^θ , на которые при формировании тайлов разбивается область значений параметра j_ζ , фиксировано, то порядок M имеет величина r_ζ^θ ; в этом случае r_ζ^θ зависит от внешних переменных. Если зафиксировать r_ζ^θ , то порядок M имеет Q_ζ^θ .

Л е м м а [8]. Число фиксированных данных, используемых на вхождении (a_l, S_β, q) в правой части оператора S_β , при фиксированном значении цикла с параметром j_ζ , оценивается величиной $O(M^{\rho_{a_l, S_\beta, q}^{\Phi, \zeta}})$. Число фиксированных данных, используемых на вхождении (a_l, S_β, q) , оценивается величиной $O(M^{\rho_{a_l, S_\beta, q}^\Phi})$. Число фиксированных данных, используемых на вхождении (a_l, S_β, q) , при фиксированном значении глобального цикла с параметром j_ζ^{gl} оценивается величиной $O(M^{\rho_{a_l, S_\beta, q}^\Phi})$, если $\rho_{a_l, S_\beta, q}^{\Phi, \zeta} - 1 = \rho_{a_l, S_\beta, q}^\Phi$, и величиной $O(r_\zeta^\beta M^{\rho_{a_l, S_\beta, q}^{\Phi, \zeta}})$, если $\rho_{a_l, S_\beta, q}^{\Phi, \zeta} = \rho_{a_l, S_\beta, q}^\Phi$.

Обозначим через $(\Phi_{\alpha, \beta})_\zeta$ и $(\Psi_{\alpha, \beta})_\zeta$ строки матриц с номером ζ . Не ограничивая общности, будем считать, что операторы S_α и S_β принадлежат группам операторов с номерами α и β (α и β могут и совпадать). Для оператора S_α обозначим ζ -ю координату из равенств (1), (2) через $j_{\zeta^\alpha}^{\text{gl}}$, а для оператора S_β – через $j_{\zeta^\beta}^{\text{gl}}$. Обозначим

$$d_{\alpha, \beta} = m_{\zeta^\beta}^\beta - \varphi_{\zeta^\alpha}^{\alpha, \beta} - m_{\zeta^\alpha}^\alpha.$$

Т е о р е м а 1. Пусть определение элемента некоторого массива a_l происходит на вхождении $(a, S, 1)$ в левой части оператора S_α , а использование – на вхождении (a_l, S_β, q) в правой части оператора S_β ,

$$\text{Pr}^\alpha(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = j_{\zeta^\alpha}^{\text{gl}}, \text{Pr}^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = j_{\zeta^\beta}^{\text{gl}}.$$

Тогда, при реализации алгоритма на компьютере с распределенной памятью, объем коммуникационных операций, связанных с вхождением (a_l, S_β, q) , имеет следующие оценки:

1) если выполняются условия

$$(\Phi_{\alpha, \beta})_{\zeta^\alpha} = e_{\zeta^\beta}^{(n_\beta)}, \tag{3}$$

$$(\Psi_{\alpha, \beta})_{\zeta^\alpha} = 0, \tag{4}$$

$$r_{\zeta^\beta}^\beta = r_{\zeta^\alpha}^\alpha \tag{5}$$

и $d_{\alpha, \beta} = 0$, то коммуникационных операций не требуется;

2) если выполняются условия (3)–(5) и $0 < |d_{\alpha, \beta}| < r_{\zeta^\beta}^\beta$, то процессам требуется получить $O(Q_{\zeta^\beta}^\beta M^{\rho_{a_l, S_\beta, q}^{\Phi, \zeta}})$ данных; коммуникационные операции происходят между процессами, номера которых отличаются на 1;

3) если выполняются условия (3)–(5) и $|d_{\alpha,\beta}| \geq r_{\zeta\beta}^\beta$, или выполняются условия (3), (5), но условие (4) не выполняется, то процессам требуется получить $O(M^{\rho_{al,S\beta,q}^\Phi})$ данных; коммуникационные операции происходят между процессами, номера которых отличаются не менее чем на 1.

В случае, когда вхождение (a_l, S_β, q) не порождает истинной зависимости между итерациями алгоритма (происходит обращение к входным данным) или равенство (3) не выполняется (тогда вычисления требуют групповых коммуникационных операций), оценки следующие:

4) если выполняется условие $\rho_{al,S\beta,q}^{\Phi,\zeta\beta} = \rho_{al,S\beta,q}^\Phi$, то требуется получить $O(M^{\rho_{al,S\beta,q}^\Phi})$ данных;

5) если выполняется условие $\rho_{al,S\beta,q}^{\Phi,\zeta\beta} - 1 = \rho_{al,S\beta,q}^\Phi$, то требуется получить $O(Q_{\zeta\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi})$ данных.

Доказательство. Пусть $I(i_1, \dots, i_{n_\alpha}) \in V_{I^{\text{gl}}}^\alpha$ и $J(j_1, \dots, j_{n_\beta}) \in V_{J^{\text{gl}}}^\beta$ – две зависимые итерации. Определение элемента массива происходит в процессе $\text{Pr}^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n_\alpha}^{\text{gl}})) = j_{\zeta\alpha}^{\text{gl}}$, а использование – в процессе $\text{Pr}^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n_\beta}^{\text{gl}})) = j_{\zeta\beta}^{\text{gl}}$. Тогда

$$\text{Pr}^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n_\alpha}^{\text{gl}})) = j_{\zeta\alpha}^{\text{gl}} = \left\lfloor \frac{1}{r_{\zeta\alpha}^\alpha} \left(j_{\zeta\beta} + \left((\Phi_{\alpha,\beta})_{\zeta\alpha} - e_{\zeta\beta}^{(n_\beta)} \right) J + (\Psi_{\alpha,\beta})_{\zeta\alpha} N + d_{\alpha,\beta} - m_{\zeta\beta}^\beta \right) \right\rfloor.$$

Если выполняются условия (3)–(5) и $d_{\alpha,\beta} = 0$, то

$$\text{Pr}^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n_\alpha}^{\text{gl}})) = \left\lfloor \frac{1}{r_{\zeta\beta}^\beta} (j_{\zeta\beta} - m_{\zeta\beta}^\beta) \right\rfloor = j_{\zeta\beta}^{\text{gl}} = \text{Pr}^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n_\beta}^{\text{gl}})),$$

первое утверждение теоремы справедливо.

Если выполняются условия (3)–(5) и $0 < |d_{\alpha,\beta}| < r_{\zeta\beta}^\beta$, то

$$\text{Pr}^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n_\alpha}^{\text{gl}})) = \left\lfloor \frac{1}{r_{\zeta\beta}^\beta} (j_{\zeta\beta} - m_{\zeta\beta}^\beta + d_{\alpha,\beta}) \right\rfloor.$$

Это выражение отличается от $\left\lfloor \frac{1}{r_{\zeta\beta}^\beta} (j_{\zeta\beta} - m_{\zeta\beta}^\beta) \right\rfloor$ на $|d_{\alpha,\beta}|$ итерациях из каждых $r_{\zeta\beta}^\beta$ подряд идущих итераций $j_{\zeta\beta}$. Для этих итераций $j_{\zeta\beta}$ значения $\text{Pr}^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n_\alpha}^{\text{gl}}))$ и $\text{Pr}^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n_\beta}^{\text{gl}}))$ отличаются на 1, а для других $r_{\zeta\beta}^\beta - |d_{\alpha,\beta}|$ итераций – совпадают; для использования требуются коммуникационные операции между процессами, номера которых отличаются на 1. Множество итераций цикла с параметром $j_{\zeta\beta}$ разбивается на $Q_{\zeta\beta}^\beta$ частей. Из равенства $(\Phi_{\alpha,\beta})_{\zeta\alpha} = e_{\zeta\beta}^{(n_\beta)}$ следует, что выполняется равенство $\rho_{al,S\beta,q}^{\Phi,\zeta\beta} = \rho_{al,S\beta,q}^\Phi$. Поэтому, как следует из леммы, в каждой из $Q_{\zeta\beta}^\beta$ частей требуется получить $|d_{\alpha,\beta}| O(M^{\rho_{al,S\beta,q}^{\Phi,\zeta\beta} - 1})$ данных из общего числа $O(r_{\zeta\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi - 1})$, используемых в каждой части фиксированных данных (остальные данные находятся в локальной памяти). Так как величина $|d_{\alpha,\beta}|$ не зависит от внешних переменных, то суммарный объем получаемых данных по всем $Q_{\zeta\beta}^\beta$ частям определяется оценкой $Q_{\zeta\beta}^\beta O(M^{\rho_{al,S\beta,q}^{\Phi,\zeta\beta} - 1}) = O(Q_{\zeta\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi - 1})$. Отметим, что при выборе малого размера тайла, когда $|d_{\alpha,\beta}|$ близко по значению к $r_{\zeta\beta}^\beta$, полученная оценка близка к $O(M^{\rho_{al,S\beta,q}^\Phi})$. Второе утверждение теоремы доказано.

Если выполняются условия (3)–(5) и $|d_{\alpha,\beta}| \geq r_{\zeta^\beta}^\beta$, или выполняются условия (3), (5), но условие (4) не выполняется, то $\left[\frac{1}{r_{\zeta^\alpha}^\alpha} \left(j_{\zeta^\beta} + (\Psi_{\alpha,\beta})_{\zeta^\alpha} N + d_{\alpha,\beta} - m_{\zeta^\beta}^\beta \right) \right]$ отличается от $\left[\frac{1}{r_{\zeta^\beta}^\beta} (j_{\zeta^\beta} - m_{\zeta^\beta}^\beta) \right]$ не менее чем на 1. Значит, значения $\Pr^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n^\alpha}^{\text{gl}}))$ и $\Pr^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n^\beta}^{\text{gl}}))$ отличаются на 1 или более. Следовательно, в каждой из $Q_{\zeta^\beta}^\beta$ частей, на которые разбивается множество итераций цикла с параметром j_{ζ^β} , требуется получить все $r_{\zeta^\beta}^\beta O(M^{\rho_{al,S\beta,q}^{\Phi,\zeta^\beta}}^{-1})$ данных, используемых в каждой части фиксированных данных. Как было замечено ранее, из равенства $(\Phi_{\alpha,\beta})_{\zeta^\alpha} = e_{\zeta^\beta}^{(n^\beta)}$ следует, что выполняется равенство $\rho_{al,S\beta,q}^{\Phi,\zeta^\beta} = \rho_{al,S\beta,q}^\Phi$. Суммарный объем получаемых данных по всем $Q_{\zeta^\beta}^\beta$ частям определяется оценкой $Q_{\zeta^\beta}^\beta r_{\zeta^\beta}^\beta O(M^{\rho_{al,S\beta,q}^{\Phi,\zeta^\beta}}^{-1}) = O(M^{\rho_{al,S\beta,q}^\Phi})$. Третье утверждение теоремы доказано.

Если равенство (3) не выполняется, то модуль разности чисел

$$\left[\frac{1}{r_{\zeta^\alpha}^\alpha} \left(j_{\zeta^\beta} + \left((\Phi_{\alpha,\beta})_{\zeta^\alpha} - e_{\zeta^\beta}^{(n^\beta)} \right) J + (\Psi_{\alpha,\beta})_{\zeta^\alpha} N + d_{\alpha,\beta} - m_{\zeta^\beta}^\beta \right) \right]$$

и $\left[\frac{1}{r_{\zeta^\beta}^\beta} (j_{\zeta^\beta} - m_{\zeta^\beta}^\beta) \right]$ зависит от J . Поэтому значения $\Pr^\alpha(I^{\text{gl}}(i_1^{\text{gl}}, \dots, i_{n^\alpha}^{\text{gl}}))$ и $\Pr^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_{n^\beta}^{\text{gl}}))$ могут быть, вообще говоря, любыми в диапазонах соответственно от 0 до $Q_{\zeta^\alpha}^\alpha - 1$ и от 0 до $Q_{\zeta^\beta}^\beta - 1$; вычисления требуют групповых коммуникационных операций (на практике, как правило, типа «каждый с каждым»).

Покажем справедливость четвертого и пятого утверждений теоремы. Если на вхождении (a_l, S_β, q) происходит обращение к входным данным или равенство (3) не выполняется (тогда, как было доказано, вычисления требуют групповых коммуникационных операций).

Если выполняется равенство $\rho_{al,S\beta,q}^{\Phi,\zeta^\beta} = \rho_{al,S\beta,q}^\Phi$, то в каждой из $Q_{\zeta^\beta}^\beta$ частей требуется получить, как следует из леммы, $O(r_{\zeta^\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi}^{-1})$ данных. Тогда суммарный объем получаемых данных по $Q_{\zeta^\beta}^\beta$ частям определяется оценкой $Q_{\zeta^\beta}^\beta O(r_{\zeta^\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi}^{-1}) = O(M^{\rho_{al,S\beta,q}^\Phi})$.

Если выполняется равенство $\rho_{al,S\beta,q}^{\Phi,\zeta^\beta} - 1 = \rho_{al,S\beta,q}^\Phi$, то в каждой из $Q_{\zeta^\beta}^\beta$ частей требуется получить $O(M^{\rho_{al,S\beta,q}^\Phi})$ данных. Объем коммуникационных операций по всем $Q_{\zeta^\beta}^\beta$ частям определяется оценкой $Q_{\zeta^\beta}^\beta O(M^{\rho_{al,S\beta,q}^\Phi}) = O(Q_{\zeta^\beta}^\beta M^{\rho_{al,S\beta,q}^\Phi})$. Теорема доказана.

П р и м е р (продолжение). Итерации циклов уровня вложенности 2 можно выполнять независимо. Если распределение операций между процессами задать функциями вида (1), $\zeta = 2$, то получим известный параллельный алгоритм, основанный на естественном параллелизме (при фиксированных j).

Для всех зависимостей имеем $\zeta^\alpha = \zeta^\beta = 2$, $m_{\zeta^\alpha}^\alpha = m_{\zeta^\beta}^\beta = 1$. Пусть $j_0 = N_1 - 1 = N_2 - 1 = M$; положим $r_1^\beta = r_1^\alpha = 1$, $r_{\zeta^\beta}^\beta = r_{\zeta^\alpha}^\alpha$, $r_3^\beta = r_3^\alpha = M + 1$. Применим теорему 1 для оценки объема данных, которые требуется получить процессам. Условия (3)–(5) и $d_{\alpha,\beta} = 0$ выполняются для всех зависимостей, кроме зависимостей, задаваемых функциями $\bar{\Phi}_{6,2}$ и $\bar{\Phi}_{3,5}$.

Коммуникационные операции порождаются только функциями $\bar{\Phi}_{6,2}$ и $\bar{\Phi}_{3,5}$. Имеем: $\rho_{al,S\beta,q}^{\Phi,\zeta} = \rho_{al,S\beta,q}^\Phi = 3$. Требуется получить $O(M^3)$ данных, причем $(\Phi_{\alpha,\beta})_2 = e_3^{(3)} \neq e_2^{(3)}$, поэтому на каждом новом временном слое j вычисления требуют групповых коммуникационных операций

типа «каждый с каждым»: дважды происходит обращение каждого процесса к локальной памяти всех других процессов.

Обозначим $r_{\alpha,\beta} = r_{\zeta^\beta}^\beta Q_{\zeta^\beta}^\beta - (M_{\zeta^\beta}^\beta - m_{\zeta^\beta}^\beta + 1)$ (если число $(M_{\zeta^\beta}^\beta - m_{\zeta^\beta}^\beta + 1) / r_{\zeta^\beta}^\beta$ является целым, то $r_{\alpha,\beta} = 0$), $\bar{d}_{\alpha,\beta} = \varphi_{\zeta^\alpha}^{\alpha,\beta} + m_{\zeta^\alpha}^\alpha + M_{\zeta^\beta}^{\beta,1} + r_{\alpha,\beta}$.

Т е о р е м а 2. Пусть определение элемента некоторого массива a_l происходит на вхождении $(a_l, S_\alpha, 1)$ в левой части оператора S_α , а использование – на вхождении (a_l, S_β, q) в правой части оператора S_β и

$$\Pr^\alpha(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = j_{\zeta^\alpha}^{\text{gl}}, \Pr^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = Q_{\zeta^\beta}^\beta - j_{\zeta^\beta}^{\text{gl}} - 1$$

или

$$\Pr^\alpha(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = Q_{\zeta^\alpha}^\alpha - j_{\zeta^\alpha}^{\text{gl}} - 1, \Pr^\beta(J^{\text{gl}}(j_1^{\text{gl}}, \dots, j_n^{\text{gl}})) = j_{\zeta^\beta}^{\text{gl}}.$$

Тогда, при реализации алгоритма на компьютере с распределенной памятью, объем коммуникационных операций, связанных с вхождением (a_l, S_β, q) , имеет следующие оценки:

1) если выполняются условия

$$(\Phi_{\alpha,\beta})_{\zeta^\alpha} = -e^{\binom{n_\beta}{\zeta^\beta}}, \quad (6)$$

$$(\Psi_{\alpha,\beta})_{\zeta^\alpha} = M^{\beta,N}(\zeta^\beta), \quad (7)$$

$$r_{\zeta^\beta}^{\beta} = r_{\zeta^\alpha}^\alpha, \quad Q_{\zeta^\beta}^\beta = Q_{\zeta^\alpha}^\alpha \quad (8)$$

и $\bar{d}_{\alpha,\beta} = 0$, то коммуникационных операций не требуется;

2) если выполняются условия (6)–(8) и $0 < |\bar{d}_{\alpha,\beta}| < r_{\zeta^\beta}^\beta$, то процессам требуется получить $O(Q_{\zeta^\beta}^\beta M^{\rho_{al,S_\beta,q}^\Phi})$ данных; коммуникационные операции происходят между процессами, номера которых отличаются на 1;

3) если выполняются условия (6)–(8) и $|\bar{d}_{\alpha,\beta}| \geq r_{\zeta^\beta}^\beta$, или выполняются условия (6), (8), но условие (7) не выполняется, то процессам требуется получить $O(M^{\rho_{al,S_\beta,q}^\Phi})$ данных; коммуникационные операции происходят между процессами, номера которых отличаются не менее чем на 1.

В случае, когда вхождение (a_l, S_β, q) не порождает истинной зависимости между итерациями алгоритма (происходит обращение к входным данным) или равенство (6) не выполняется (тогда вычисления требуют групповых коммуникационных операций), оценки следующие:

4) если выполняется условие $\rho_{al,S_\beta,q}^{\Phi,\zeta^\beta} = \rho_{al,S_\beta,q}^\Phi$, то требуется получить $O(M^{\rho_{al,S_\beta,q}^\Phi})$ данных;

5) если выполняется условие $\rho_{al,S_\beta,q}^{\Phi,\zeta^\beta} - 1 = \rho_{al,S_\beta,q}^\Phi$, то требуется получить $O(Q_{\zeta^\beta}^\beta M^{\rho_{al,S_\beta,q}^\Phi})$ данных.

Для доказательств утверждений теоремы используются аналогичные приемы и рассуждения, как и в случае теоремы 1.

Теорему 2 можно использовать, например, для улучшения локальности алгоритмов прогнонок, рассматривая зависимости между операциями прямого и обратного хода.

П р и м е р (продолжение). Рассмотрим альтернативный вариант распределения операций между процессами [11]: для операций, порождаемых операторами S_4, S_5, S_6 , распределение по-прежнему определяется параллельными циклами уровня вложенности 2, а распределение операций, порождаемых операторами S_1, S_2, S_3 , определяется циклами уровня вложенности 3 (все эти циклы имеют параметр i_1 , изменяющийся от 1 до $N_1 - 1$). Для операторов S_1, S_2 используются функции вида (1), $\zeta = 3$; для оператора S_3 используется функция вида (2), $\zeta = 3$; $r_1^1 = r_1^2 = r_1^3 = 1$, $r_2^1 = r_2^2 = r_2^3$, $r_3^1 = r_3^2 = r_3^3$; для операторов S_4, S_5, S_6 используются функции вида (1), $\zeta = 2$; $r_1^\alpha = 1$, $r_2^\alpha = r_3^\alpha$, $r_3^\alpha = N_2 - 1$, $\alpha = 4, 5, 6$.

Применим теоремы 1, 2 (полагая $j_0 = N_1 - 1 = N_2 - 1 = M$) для оценки объема данных, которые требуется получить процессам, и покажем, что суммарный объем коммуникационных операций

уменьшается; кроме того, обмен происходит только между процессами, номера которых отличаются на 1. Как и ранее, функциями $\bar{\Phi}_{4,4}$, $\bar{\Phi}_{4,5}$, $\bar{\Phi}_{5,5}$, $\bar{\Phi}_{4,6}$, $\bar{\Phi}_{5,6}$, $\bar{\Phi}_{6,6}$ коммуникационные операции не порождаются (по теореме 1).

По теореме 2 функции $\bar{\Phi}_{1,3}$ ($\alpha = 1, \beta = 3$), $\bar{\Phi}_{2,3}$ ($\alpha = 2, \beta = 3$) и $\bar{\Phi}_{3,5}$ ($\alpha = 3, \beta = 5$) коммуникационных операций не порождают. Действительно, $m_{\zeta^\alpha}^\alpha = 1$, $M_{\zeta^\beta}^\beta = M^{\beta,N}(\zeta^\beta)(j_0 N_1 N_2)^T + M_{\zeta^\beta}^{\beta,1} = (0 \ 1 \ 0)(j_0 N_1 N_2)^T - 1$. Положим $r_{\zeta^\beta}^\beta = r_{\zeta^\alpha}^\alpha$; тогда удовлетворены условия (6)–(8). Кроме того, $\bar{d}_{\alpha,\beta} = 0$ (в предположении $r_{\alpha,\beta} = 0$): $\varphi_{\zeta^\alpha}^{\alpha,\beta} + m_{\zeta^\alpha}^\alpha + M_{\zeta^\beta}^{\beta,1} + r_{\alpha,\beta} = 0 + 1 - 1 + 0 = 0$.

В случае функций $\bar{\Phi}_{1,1}$, $\bar{\Phi}_{1,2}$, $\bar{\Phi}_{2,2}$, $\bar{\Phi}_{3,3}$ имеем $\zeta^\alpha = \zeta^\beta = 3$, $m_{\zeta^\beta}^\beta = m_{\zeta^\alpha}^\alpha = 1$, $r_3^\beta = r_3^\alpha = M + 1$, $d_{\alpha,\beta} = 1$. По теореме 1 объем данных, которые требуется получить процессам, оценивается величиной $O(Q_{\zeta^\beta}^\beta M^{P_{al,S\beta,q}^\Phi}) = O(Q_{\zeta^\beta}^\beta M^2)$, обмен происходит только между процессами, номера которых отличаются на 1.

Рассмотрим случай функции $\bar{\Phi}_{6,2}$. Положим $r_{\zeta^\beta}^\beta = r_{\zeta^\alpha}^\alpha$ и применим теорему 1 ($\alpha = 6, \beta = 2$). Имеем: $m_{\zeta^\alpha}^\alpha = m_{\zeta^\beta}^\beta = 1$, $\varphi_{\zeta^\alpha}^{\alpha,\beta} = \varphi_2$, где φ_2 равно 0, 1, -1. Условия (6)–(8) выполняются. Условие $m_{\zeta^\beta}^\beta = m_{\zeta^\alpha}^\alpha + \varphi_{\zeta^\alpha}^{\alpha,\beta}$ выполняется в случае $\varphi_2 = 0$; определение и использование элемента массива данных происходит в одном и том же вычислительном процессе. Если $\varphi_2 = 1$ или $\varphi_2 = -1$, то $\left| m_{\zeta^\beta}^\beta - m_{\zeta^\alpha}^\alpha - \varphi_{\zeta^\alpha}^{\alpha,\beta} \right| = 1$. Поэтому коммуникационные операции происходят между процессами, номера которых отличаются на 1.

Таким образом, применение альтернативного варианта параллельного алгоритма приводит к обменам $O(Q_{\zeta^\beta}^\beta M^2)$ данными, для использования требуются коммуникационные операции между процессами, номера которых отличаются на 1. Применение параллельного алгоритма, основанного на естественном параллелизме, приводит к обменам $O(M^3)$ данными, причем на каждом новом временном слое j вычисления требуют групповых коммуникационных операций типа «каждый с каждым».

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. Воеводин, Вл. В. Спасительная локальность суперкомпьютеров / Вл. В. Воеводин, Вад. В. Воеводин // Открытые системы. – 2013. – № 9. – С. 12–15.
2. Xue, J. Time-minimal tiling when rise is larger than zero / J. Xue, W. Cai // Parallel Computing. – 2002. – Vol. 28, N 6. – P. 915–939. [https://doi.org/10.1016/s0167-8191\(02\)00098-4](https://doi.org/10.1016/s0167-8191(02)00098-4)
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, N 2. – P. 1–28. <https://doi.org/10.1145/2948975>
4. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб., 2002. – 608 с.
5. Адуцкевич, Е. В. Согласованное получение конвейерного параллелизма и распределения операций и данных между процессорами / Е. В. Адуцкевич, Н. А. Лиходед // Программирование. – 2006. – Т. 35, № 3. – С. 52–65.
6. Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model / U. Bondhugula [et al.] // Lecture notes in computer science. – 2008. – N 4959. – P. 132–146. https://doi.org/10.1007/978-3-540-78791-4_9
7. Лиходед, Н. А. Достаточные условия определения и использования данных в одном параллельном зернистом вычислительном процессе / Н. А. Лиходед // Журн. вычисл. математики и матем. физики. – 2014. – Т. 54, № 8. – С. 1356. <https://doi.org/10.7868/S0044466914080092>
8. Лиходед, Н. А. Метод ранжирования параметров размера блоков вычислений параллельного алгоритма / Н. А. Лиходед, М. А. Полещук // Докл. Нац. акад. наук Беларуси. – 2015. – Т. 59, № 4. – С. 25–33.
9. Гергель, В. П. Основы параллельных вычислений для многопроцессорных вычислительных систем / В. П. Гергель, Р. Г. Стронгин. – Нижний Новгород, 2003. – 184 с.
10. 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.
11. Баханович, С. В. Улучшение локальности параллельных алгоритмов численного решения двумерных квазилинейных параболических уравнений / С. В. Баханович, Н. А. Лиходед, П. А. Мандрик // Вестн. Рос. ун-та дружбы народов. Сер. Математика. Информатика. Физика. – 2014. – № 2. – С. 211–215.

References

1. Voevodin V. I., Voevodin V. V. The fortunate locality of supercomputers. *Otkrytye sistemy = Open Systems*, 2013, no. 9, pp. 12–15 (in Russian).
2. 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)
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, p. 1–28. <https://doi.org/10.1145/2948975>
4. Voevodin V. I., Voevodin V. V. *Parallel Computing*. Saint Petersburg, 2002. 608 p. (in Russian).
5. Adutskevich E. V., Likhoded N. A. A consistent generation of pipeline parallelism and distribution of operations and data among processors. *Programming and Computer Software*, 2006, vol. 32, no. 3, pp. 166–176. <https://doi.org/10.1134/s0361768806030078>
6. Bondhugula U., Baskaran M., Krishnamoorthy S., Ramanujam J., Rountev A., Sadayappan P. Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model. *Lecture notes in computer science*, 2008, no. 4959, pp. 132–146. https://doi.org/10.1007/978-3-540-78791-4_9
7. Likhoded N. A. Sufficient conditions for the determination and use of data in the same granular parallel computation process. *Computational Mathematics and Mathematical Physics*, 2014, vol. 54, no. 8, pp. 1316–1326. <https://doi.org/10.1134/s0965542514080077>
8. Likhoded N. A., Paliashchuk M. A. Method of ranking tiles size parameters of parallel algorithm. *Doklady Natsional'noi akademii nauk Belarusi = Doklady of the National Academy of Sciences of Belarus*, 2015, vol. 59, no. 4, pp. 25–33 (in Russian).
9. Gergel V. P., Strongin R. G. *Introduction to parallel computing for multiprocessor systems*. Nizhny Novgorod, 2003. 184 p. (in Russian).
10. 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).
11. Bakhanovich S. V., Likhoded N. A., Mandrik P. A. Improvement of locality of parallel algorithms of the numerical solutions of the two-dimensional quasilinear parabolic equation. *Discrete and Continuous Models and Applied Computational Science*, 2014, no. 2, pp. 211–215 (in Russian).

Информация об авторах

Лиходед Николай Александрович – д-р физ.-мат. наук, профессор. Белорусский государственный университет (пр. Независимости, 4, 220030, Минск, Республика Беларусь). E-mail: likhoded@bsu.by.

Толстиков Алексей Александрович – ст. преподаватель. Белорусский государственный университет (пр. Независимости, 4, 220030, Минск, Республика Беларусь). E-mail: tolstikov@bsu.by.

Information about the authors

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

Tolstikov Aliaksei A. – Senior lecturer. Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus). E-mail: tolstikov@bsu.by.