*УДК 519.6+681.3.012*

*Н. А. ЛИХОДЕД*

# ХАРАКТЕРИСТИКА ЛОКАЛЬНОСТИ ПАРАЛЛЕЛЬНЫХ РЕАЛИЗАЦИЙ МНОГОМЕРНЫХ ЦИКЛОВ

*(Представлено академиком И. В. Гайшуном)*

*Институт математики НАН Беларуси* *Поступило .....*

**Введение.** Локальность при выполнении какой-либо операции означает расположение (хранение) аргумента операции непосредственно перед использованием в памяти с быстрым доступом, где он оказался вследствие участия в более ранней операции. При многопроцессорной обработке памятью с быстрым доступом считается локальная память процессора, при однопроцессорной — кэш. Локальность алгоритма — это вычислительное свойство алгоритма, отражающее совокупность сведений о локальности его данных. Локальность параллельного алгоритма, предназначенного для реализации на компьютерах с распределенной памятью, определяет коммуникационные затраты: чем лучше локальность, тем меньше аргументов операций требуется доставлять в процессоры. Улучшить локальность можно путем преобразования алгоритмов [1–5] или распределения данных [6–11]. Отметим, что строгое понятие локальности не введено.

В данной работе разрабатывается математический аппарат для характеристики локальности параллельных реализаций алгоритмов на компьютерах с распределенной памятью: вводятся величины, характеризующие локальность, получены условия многократности использования в одном процессоре каждого значения элементов массивов и условия, позволяющие определить, в один или многие процессоры требуется доставить элемент массива с таким значением.

**Предварительные сведения.** Пусть алгоритм задан многомерным циклом произвольной структуры вложенности. Будем считать, что индексные выражения элементов массивов и границы изменения параметров циклов задаются неоднородными (в общем случае) формами, линейными по совокупности параметров циклов и внешних переменных. Пусть в многомерном цикле имеется $K$ операторов $S_\beta$ и используется $L$ массивов $a_l$. Обозначим: $V_\beta$ — область изменения параметров циклов для оператора $S_\beta$, $n_\beta$ — число циклов, окружающих оператор $S_\beta$; $W_l \subset \mathbf{Z}^{v_l}$ — область изменения индексов $l$-го массива, где $v_l$ — размерность $l$-го массива; $N \in \mathbf{Z}^e$ — вектор внешних переменных алгоритма, где $e$ — число внешних переменных. Выполнение оператора $S_\beta$ при конкретных значениях $\beta$ и вектора параметров цикла $J$ будем называть операцией и обозначать $S_\beta(J)$.

Вхождением $(l,\beta,q)$ будем называть $q$-е вхождение массива $a_l$ в оператор $S_\beta$. Обозначим через $Q$ множество вхождений $(l,\beta,q)$. Пусть индексы элементов $l$-го массива, связанных с вхождением $(l,\beta,q)$, выражаются функцией $\overline{F}_{l,\beta,q} : V_\beta \to W_l$ вида

$$\overline{F}_{l,\beta,q}(J) = F_{l,\beta,q} J + G_{l,\beta,q} N + f^{(l,\beta,q)},$$

$$J(j_1,...,j_{n_\beta}) \in V_\beta,\ N \in \mathbf{Z}^e, F_{l,\beta,q} \in \mathbf{Z}^{v_l \times n_\beta},\ G_{l,\beta,q} \in \mathbf{Z}^{v_l \times e},\ f^{(l,\beta,q)} \in \mathbf{Z}^{v_l},\ (l,\beta,q) \in Q.$$

Операция $S_\beta(J), J \in V_\beta$, зависит от операции $S_\alpha(I), I \in V_\alpha$, если: 1) $S_\alpha(I)$ выполняется раньше $S_\beta(J)$; 2) $S_\alpha(I)$ и $S_\beta(J)$ используют один и тот же элемент какого-либо массива и по крайней мере одно из использований есть переопределение элемента; 3) между операциями $S_\alpha(I)$ и $S_\beta(J)$ этот элемент не переопределяется. Зависимость называется: истинной, если элемент массива сначала определяется, а затем его значение используется в качестве аргумента. Зависимость операции $S_\beta(J)$ от операции $S_\alpha(I)$ будем обозначать $S_\alpha(I) \to S_\beta(J)$.

Обозначим $P=\{(\alpha,\beta)\mid \exists\, I\in V_\alpha,\ J\in V_\beta,\ S_\alpha(I)\to S_\beta(J)\}$ — множество, которое определяет пары зависимых операторов. Для каждой пары $(\alpha,\beta)\in P$ обозначим $V_{\alpha,\beta}=\{\,J\in V_\beta\mid \exists\, S_\alpha(I)\to S_\beta(J)\}$. Пусть зависимости операций алгоритма задаются функциями $\overline{\Phi}_{\alpha,\beta}: V_{\alpha,\beta}\to V_\alpha$ таким образом, что если $S_\alpha(I)\to S_\beta(J),\ I\in V_\alpha,\ J\in V_{\alpha,\beta}\subseteq V_\beta$, то $I=\overline{\Phi}_{\alpha,\beta}(J)$. Будем называть функции $\overline{\Phi}_{\alpha,\beta}$ функциями зависимостей и предполагать, что они имеют вид

$$\begin{gathered}\overline{\Phi}_{\alpha,\beta}(J)=\Phi_{\alpha,\beta}J+\Psi_{\alpha,\beta}N-\varphi^{(\alpha,\beta)},\\ J\in V_{\alpha,\beta},\ N\in\mathbf{Z}^e,\ \Phi_{\alpha,\beta}\in\mathbf{Z}^{n_\alpha\times n_\beta},\ \Psi_{\alpha,\beta}\in\mathbf{Z}^{n_\alpha\times e},\ \varphi^{(\alpha,\beta)}\in\mathbf{Z}^{n_\alpha},\ (\alpha,\beta)\in P.\end{gathered}\tag{1}$$

Функции зависимостей задают тонкую информационную структуру алгоритма [1].

**Теоретическое обоснование.** Обозначим $\rho_{l,\beta,q}=\operatorname{rank}F_{l,\beta,q}$. Рассмотрим базис пространства $\mathbf{Z}^{n_\beta}$ с базисными векторами $u_i^{\perp},\ 1\le i\le\rho_{l,\beta,q},\ u_i,\ 1\le i\le n_\beta-\rho_{l,\beta,q}$, где $u_i$ — базисные векторы подпространства $\ker F_{l,\beta,q}$; если $\rho_{l,\beta,q}=n_\beta$, то векторы $u_i$ отсутствуют.

Выделим цикл (циклы) уровня вложенности $\xi_1,\ 1\le\xi_1\le n$, где $n=\max\limits_{1\le\beta\le K}n_\beta$. Обозначим $\rho^s_{l,\beta,q}=\operatorname{rank}\begin{pmatrix}F_{l,\beta,q}\\ e^{(n_\beta)}_{\xi_1}\end{pmatrix}$, где $e^{(n_\beta)}_{\xi_1}$ — вектор размера $n_\beta$, у которого координата с номером $\xi_1$ равна 1, а остальные координаты нулевые (если $\xi_1>n_\beta$, то $e^{(n_\beta)}_{\xi_1}$ — нулевой вектор); $u_i^s$ — базисные векторы пересечения подпространства $\ker\begin{pmatrix}F_{l,\beta,q}\\ e^{(n_\beta)}_{\xi_1}\end{pmatrix}$ и $\mathbf{Z}^{n_\beta}$; если $\rho^s_{l,\beta,q}=n_\beta$, то векторы $u_i^s$ не определены. Отметим, что координаты $\xi_1$ векторов $u_i^s$ равны нулю.

Пусть вхождение $(l,\beta,q)$ в правую часть некоторого оператора порождает истинную зависимость, $\overline{\Phi}_{\alpha,\beta}$ — функция зависимостей вида (1). Обозначим $\rho^{\Phi,s}_{l,\beta,q}=\operatorname{rank}\begin{pmatrix}F_{l,\beta,q}\\ \Phi_{\alpha,\beta}\\ e^{(n_\beta)}_{\xi_1}\end{pmatrix}$, $\rho^{\Phi}_{l,\beta,q}=\operatorname{rank}\begin{pmatrix}F_{l,\beta,q}\\ \Phi_{\alpha,\beta}\end{pmatrix}$; $u_i^{\Phi,s}$ — базисные векторы пересечения подпространства $\ker\begin{pmatrix}F_{l,\beta,q}\\ \Phi_{\alpha,\beta}\\ e^{(n_\beta)}_{\xi_1}\end{pmatrix}$ и $\mathbf{Z}^{n_\beta}$, $u_i^{\Phi}$ — базисные векторы пересечения подпространства $\ker\begin{pmatrix}F_{l,\beta,q}\\ \Phi_{\alpha,\beta}\end{pmatrix}$ и $\mathbf{Z}^{n_\beta}$. Если вхождение $(l,\beta,q)$ не порождает истинную зависимость, то по определению $\rho^{\Phi,s}_{l,\beta,q}=\rho^s_{l,\beta,q}$, $\rho^{\Phi}_{l,\beta,q}=\rho_{l,\beta,q}$, множество векторов $u_i^{\Phi,s}$ совпадает с множеством векторов $u_i^s$, множество векторов $u_i^{\Phi}$ совпадает с множеством векторов $u_i$; координаты $\xi_1$ векторов $u_i^{\Phi,s}$ являются нулевыми.

Будем полагать, что координаты виртуальных процессоров, выполняющих операции $S_\beta(J)$ многомерного цикла, задаются функциями вида

$$\begin{gathered}t^{(\beta)}_{\xi_1}(J)=\kappa_{\beta,\xi_1}j_{\xi_1}+b^{(\beta,\xi_1)}N+a_{\beta,\xi_1},\\ J(j_1,\dots,j_{n_\beta})\in V_\beta,\ \kappa_{\beta,\xi_1}\in\{-1;1\},\ b^{(\beta,\xi_1)},\ N\in\mathbf{Z}^e,\ a_{\beta,\xi_1}\in\mathbf{Z},\ 1\le\beta\le K.\end{gathered}\tag{2}$$

Если оператор $S_\beta$ окружает менее $\xi_1$ циклов, т.е. если $\xi_1>n_\beta$, то следует принять $j_{\xi_1}=0$. Заметим, что в реальных вычислениях один процессор выполняет операции, отображаемые на один виртуальный процессор или на некоторый набор виртуальных процессоров.

Обозначим через $\left(\Phi_{\alpha,\beta}\right)_{\xi_1}$ и $\left(\Psi_{\alpha,\beta}\right)_{\xi_1}$ строки матриц с номером $\xi_1$; если $\xi_1>n_\alpha$, то

примем $\left(\Phi_{\alpha,\beta}\right)_{\xi_1}=0$, $\left(\Psi_{\alpha,\beta}\right)_{\xi_1}=0$, $\phi_{\xi_1}^{(\alpha,\beta)}=0$.

Т е о р е м а 1. *Пусть элемент массива, связанный с вхождением* $(l,\beta,q)$ *в правую часть оператора, используется в некотором виртуальном процессоре. Элемент массива определяется в этом же процессоре, если вхождение* $(l,\beta,q)$ *порождает истинную зависимость и выполняются условия*

$$\left(\Phi_{\alpha,\beta}\right)_{\xi_1}=\kappa_{\alpha,\xi_1}\kappa_{\beta,\xi_1}e_{\xi_1}^{(n_\beta)},\ \left(\Psi_{\alpha,\beta}\right)_{\xi_1}=\kappa_{\alpha,\xi_1}\left(b^{(\beta,\xi_1)}-b^{(\alpha,\xi_1)}\right), \tag{3}$$

$$\phi_{\xi_1}^{(\alpha,\beta)}=\kappa_{\alpha,\xi_1}\left(a_{\alpha,\xi_1}-a_{\beta,\xi_1}\right). \tag{4}$$

*Если условия (3) выполняются, но не выполняются условия (4), то элемент массива определяется в процессоре, координата которого отличается от координаты данного процессора на* $a_{\beta,\xi_1}-a_{\alpha,\xi_1}+\kappa_{\alpha,\xi_1}\phi_{\xi_1}^{(\alpha,\beta)}$. *Элемент массива используется в данном процессоре на вхождении* $(l,\beta,q)$ *(возможно, меняя свое значение) на итерациях подпространства итераций размерности* $k_{l,\beta,q}^s$, *где* $k_{l,\beta,q}^s=n_\beta-\rho_{l,\beta,q}^s$.

Д о к а з а т е л ь с т в о. Пусть элемент массива используется на итерации $J$, а определяется на итерации $I$. Тогда $t_{\xi_1}^{(\beta)}(J)$ и $t_{\xi_1}^{(\alpha)}(I)$ задают соответственно координаты процессоров, в которых элемент используется (в качестве аргумента) и определяется. Требуется доказать, что выполнение условий (4) приводит к равенству $t_{\xi_1}^{(\beta)}(J)=t_{\xi_1}^{(\alpha)}(I)$. С учетом $I=\overline{\Phi}_{\alpha,\beta}(J)$, $i_{\xi_1}=\left(\Phi_{\alpha,\beta}\right)_{\xi_1}J+\left(\Psi_{\alpha,\beta}\right)_{\xi_1}N-\phi_{\xi_1}^{(\alpha,\beta)}$ имеем:

$$t_{\xi_1}^{(\beta)}(J)-t_{\xi_1}^{(\alpha)}(I)=\kappa_{\beta,\xi_1}j_{\xi_1}-\kappa_{\alpha,\xi_1}i_{\xi_1}+(b^{(\beta,\xi_1)}-b^{(\alpha,\xi_1)})N+a_{\beta,\xi_1}-a_{\alpha,\xi_1}=\left(\kappa_{\beta,\xi_1}e_{\xi_1}^{(n_\beta)}-\kappa_{\alpha,\xi_1}\left(\Phi_{\alpha,\beta}\right)_{\xi_1}\right)J+$$

$$\left(b^{(\beta,\xi_1)}-b^{(\alpha,\xi_1)}-\kappa_{\alpha,\xi_1}\left(\Psi_{\alpha,\beta}\right)_{\xi_1}\right)N+a_{\beta,\xi_1}-a_{\alpha,\xi_1}+\kappa_{\alpha,\xi_1}\phi_{\xi_1}^{(\alpha,\beta)}=0,$$ если выполняются условия (3), (4).

Если выполняются только условия (3), то $t_{\xi_1}^{(\beta)}(J)-t_{\xi_1}^{(\alpha)}(I)=a_{\beta,\xi_1}-a_{\alpha,\xi_1}+\kappa_{\alpha,\xi_1}\phi_{\xi_1}^{(\alpha,\beta)}$.

Последнее утверждение теоремы вытекает из следующего следствия результатов работы [10] для случая функции $t_{\xi_1}^{(\beta)}$ вида (2): элемент массива используется в фиксированном процессоре на вхождении $(l,\beta,q)$ (возможно, меняя свое значение) на итерациях подпространства итераций, отличающихся линейными комбинациями векторов $u_i^s$, $1\le i\le n_\beta-\rho_{l,\beta,q}^s$, с аннулированной координатой $\xi_1$. Из определения векторов $u_i^s$ следует, что у любого вектора координата $\xi_1$ равна нулю, поэтому размерность подпространства итераций равна числу векторов $u_i^s$, т.е. равна $k_{l,\beta,q}^s$. Если $\rho_{l,\beta,q}^s=n_\beta$, то элемент массива используется в одном процессоре только на одной итерации (т.е. на итерациях подпространства итераций размерности 0).

З а м е ч а н и е 1. Из доказательства теоремы следует, что при выполнении условий (3), (4) любое данное $a_l(\overline{F}_{l,\beta,q}(J))$, $J\in V_{\alpha,\beta}\subset\mathbf{Z}^{n_\beta}$, определяется и используется в одном процессоре. В вырожденных случаях выполнение условий (3), (4) не является обязательным для определения и использования данного в одном процессоре. Например, если $\xi_1>n_\alpha$ и для всех $J(j_1,...,j_{n_\beta})\in V_{\alpha,\beta}$ имеет место $j_{\xi_1}=const$, то достаточно потребовать $\kappa_{\beta,\xi_1}j_{\xi_1}+a_{\beta,\xi_1}-a_{\alpha,\xi_1}=0$, $b^{(\beta,\xi_1)}-b^{(\alpha,\xi_1)}=0$.

Т е о р е м а 2. *Если* $\rho_{l,\beta,q}^{\Phi}=\rho_{l,\beta,q}^{\Phi,s}$, *то на вхождении* $(l,\beta,q)$ *фиксированное данное используется только в одном процессоре; если* $\rho_{l,\beta,q}^{\Phi}<\rho_{l,\beta,q}^{\Phi,s}$, *то данное используется во многих процессорах. На вхождении* $(l,\beta,q)$ *фиксированное данное используется в фиксированном процессоре на итерациях подпространства итераций размерности* $k_{l,\beta,q}^{\Phi,s}$, *где* $k_{l,\beta,q}^{\Phi,s}=n_\beta-\rho_{l,\beta,q}^{\Phi,s}$.

Для доказательства первого утверждения теоремы 2 достаточно убедиться в справедливости следующего утверждения: если $\rho_{l,\beta,q}^{\Phi}<\rho_{l,\beta,q}^{\Phi,s}$, то на вхождении $(l,\beta,q)$ каждое

данное (т.е. элемент массива, не меняющий своего значения) используется виртуальными процессорами, координаты которых отличаются на координату $\xi_1$ векторов $u_i^{\Phi}$, $1 \le i \le n_\beta - \rho_{l,\beta,q}^{\Phi}$; если $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}^{\Phi,s}$, то на вхождении $(l,\beta,q)$ данное используется только одним процессором. Это утверждение есть аналог одного из результатов работы [10] для случая, когда понятие "использование элемента массива" заменяется на понятие "использование элемента массива, не меняющего своего значения" и соответственно во всех выкладках для вхождения, порождающего истинную зависимость, матрицу $F_{l,\beta,q}$ следует заменить на матрицу $\begin{pmatrix} F_{l,\beta,q} \\ \Phi_{\alpha,\beta} \end{pmatrix}$. Кроме того, если вхождение $(l,\beta,q)$ не порождает истинную зависимость, то следует учесть, что по определению $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}$, $\rho_{l,\beta,q}^{\Phi,s} = \rho_{l,\beta,q}^{s}$.

Второе утверждение теоремы непосредственно вытекает из следующего аналога утверждения, использованного при доказательстве теоремы 1: элемент массива используется в фиксированном процессоре на вхождении $(l,\beta,q)$ не меняя своего значения на итерациях, отличающихся линейными комбинациями векторов $u_i^{\Phi,s}$, $1 \le i \le n_\beta - \rho_{l,\beta,q}^{\Phi,s}$, с аннулированной координатой $\xi_1$; если $\rho_{l,\beta,q}^{\Phi,s} = n_\beta$, то элемент массива используется в одном процессоре только на одной итерации.

**Условия, характеризующие локальность использования данных.** Теоремы 1 и 2 позволяют выделить следующие возможные случаи использования данных на вхождении $(l,\beta,q)$ в правую часть оператора.

1. Вхождение $(l,\beta,q)$ порождает истинную зависимость и выполняются условия (3), (4). Каждый элемент массива, связанный с вхождением $(l,\beta,q)$, определяется (может быть, многократно) и используется в одном процессоре. Степень многократности использования определяется величиной $k_{l,\beta,q}^{s}$; если $k_{l,\beta,q}^{s} = 0$, то использование однократное.

2. Вхождение $(l,\beta,q)$ истинную зависимость не порождает или не выполняется хотя бы одно из условий (3), (4), выполняются условия $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}^{\Phi,s}$, $k_{l,\beta,q}^{\Phi,s} \ge 1$. Каждое данное, используемое на вхождении $(l,\beta,q)$, требуется доставить в один процессор для многократного использования. Степень многократности использования определяется величиной $k_{l,\beta,q}^{\Phi,s}$.

3. Вхождение $(l,\beta,q)$ истинную зависимость не порождает или не выполняется хотя бы одно из условий (3), (4), выполняются условия $\rho_{l,\beta,q}^{\Phi} < \rho_{l,\beta,q}^{\Phi,s}$, $k_{l,\beta,q}^{\Phi,s} \ge 1$. Каждое данное, используемое на вхождении $(l,\beta,q)$, требуется доставить во многие процессоры для многократного использования в каждом. Степень многократности использования определяется величиной $k_{l,\beta,q}^{\Phi,s}$.

4. Вхождение $(l,\beta,q)$ истинную зависимость не порождает или не выполняется хотя бы одно из условий (3), (4), выполняются условия $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}^{\Phi,s}$, $k_{l,\beta,q}^{\Phi,s} = 0$. Каждое данное, используемое на вхождении $(l,\beta,q)$, требуется доставить в один процессор для однократного использования.

5. Вхождение $(l,\beta,q)$ истинную зависимость не порождает или не выполняется хотя бы одно из условий (3), (4), выполняются условия $\rho_{l,\beta,q}^{\Phi} < \rho_{l,\beta,q}^{\Phi,s}$, $k_{l,\beta,q}^{\Phi,s} = 0$. Каждое данное, используемое на вхождении $(l,\beta,q)$, требуется доставить во многие процессоры для однократного использования.

Дадим некоторые пояснения.

Случай 1 означает, что данные, используемые на вхождении $(l,\beta,q)$, не требуется доставлять в процессоры. Это наиболее благоприятный случай, если массивы данных не слишком большие и для их хранения достаточно локальной памяти процессора.

Случаи 2–5 есть разбиение вариантов использования данных, используемых на вхождении

$(l,\beta,q)$, по количеству виртуальных процессоров, в которые их требуются доставить, и по числу итераций, на которых они используются в каждом процессоре. Понятно, что предпочтительны случаи 2 и 3.

Говоря о многих процессорах и многократности использования, не учитываются возможные вырожденные (ввиду особенностей многомерного цикла) случаи однократного использования некоторых элементов массивов.

Рассматривать (оценивать) не обязательно все вхождения. Например, можно пренебречь вхождениями констант и массивов размерности 1, вхождениями в операторы, окруженные одним или двумя циклами.

З а м е ч а н и е 2. Пусть каждый процессор вычислительной системы выполняет блок операций, отображаемых на набор соседних виртуальных процессоров. Тогда в случаях 2 и 4 большое значение имеет выполнение или невыполнение условий (3): выполнение этих условий означает многократное уменьшение числа пересылок данных. В случаях 3 и 5 многократность использования данного возрастает во столько раз, сколько виртуальных процессоров, в которые требовалось доставить данное, заменяется на один физический процессор.

П р и м е р. Рассмотрим алгоритм перемножения двух квадратных матриц порядка $N$:

$$\begin{aligned} &do \quad i=1,N \\ &\quad do \quad j=1,N \\ &\quad\quad S_1:\ c(i,j)=0 \\ &\quad\quad do \quad k=1,N \\ &\quad\quad\quad S_2:\ c(i,j)=c(i,j)+a(i,k)b(k,j) \end{aligned} \qquad (5)$$

Истинные зависимости порождает второе вхождение массива $c$ во второй оператор ($a_l=c,\beta=2,q=2$), функции зависимостей имеют вид

$$\overline{\Phi}_{1,2}(i,j,1)=\begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}i\\j\\1\end{pmatrix},\quad \overline{\Phi}_{2,2}(i,j,k)=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}i\\j\\k\end{pmatrix}-\begin{pmatrix}0\\0\\1\end{pmatrix}.$$

Вхождения $(a,2,1)$ и $(b,2,1)$ истинных зависимостей не порождают.

Пусть $\xi_1=1$, $t_1^{(1)}(i,j)=i$, $t_1^{(2)}(i,j,k)=i$.

Для вхождения $(c,2,2)$ условия (4) выполняются: $\left(\Phi_{\alpha,2}\right)_1=(1\,0\,0)$, $\left(\Psi_{\alpha,2}\right)_1=(0\,0\,0)$, $\varphi_1^{(\alpha,2)}=0$. Найдем $k_{c,2,2}^s$: $\rho_{c,2,2}^s=\operatorname{rank}\begin{pmatrix}F_{c,2,2}\\e_1^{(3)}\end{pmatrix}=\operatorname{rank}\begin{pmatrix}1&0&0\\0&1&0\\1&0&0\end{pmatrix}=2$, $k_{c,2,2}^s=n_2-\rho_{c,2,2}^s=1$. Каждый элемент массива $c$ определяется и многократно используется в одном процессоре (случай 1).

Для вхождения $(a,2,1)$ $\rho_{a,2,1}^{\Phi}=\rho_{a,2,1}=\operatorname{rank}\begin{pmatrix}1&0&0\\0&0&1\end{pmatrix}=2$, $\rho_{a,2,1}^{\Phi,s}=\rho_{a,2,1}^s=\operatorname{rank}\begin{pmatrix}1&0&0\\0&0&1\\1&0&0\end{pmatrix}=2$, $k_{a,2,1}^{\Phi,s}=k_{a,2,1}^s=1$. Так как $\rho_{a,2,1}^{\Phi}=\rho_{a,2,1}^{\Phi,s}$, $k_{a,2,1}^{\Phi,s}=1$, то каждый элемент массива $a$ требуется доставить в один виртуальный процессор для многократного использования (случай 2).

Для вхождения $(b,2,1)$ $\rho_{b,2,1}^{\Phi}=\rho_{a,2,1}=\operatorname{rank}\begin{pmatrix}0&0&1\\0&1&0\end{pmatrix}=2$, $\rho_{b,2,1}^{\Phi,s}=\rho_{b,2,1}^s=\operatorname{rank}\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}=3=n_2$, векторы $u_i^{\Phi,s}$ отсутствуют, $k_{b,2,1}^{\Phi,s}=0$. Так как $\rho_{b,2,1}^{\Phi}<\rho_{a,2,1}^{\Phi,s}$, $k_{b,2,1}^{\Phi,s}=0$, то

каждый элемент массива $b$ требуется доставить во многие виртуальные процессоры для однократного использования (случай 5).

Пусть $\xi_1 = 2$, $t_2^{(1)}(i,j) = j$, $t_2^{(2)}(i,j,k) = j$. Для массивов $c, a, b,$ так же, как при $\xi_1 = 1$, соответственно получим случаи использования данных 1, 5, 2.

Пусть $\xi_1 = 3$, $t_3^{(1)}(i,j) = 1$, $t_3^{(2)}(i,j,k) = k$. Для массивов $a$ и $b$ получим многократное использования данных в одном виртуальном процессоре (случаи 2). Исследуем локальность элементов массива $c$. Для вхождения $(c,2,2)$ и зависимости $(\alpha,\beta) = (2,2)$ условия (3) выполняются, условие (4) не выполняются: $\varphi_3^{(2,2)} = 1 \neq 0$. Так как $\rho_{c,2,2}^{\Phi} = \rho_{c,2,2}^{\Phi,s} = n_2 = 3$, $k_{c,2,2}^{\Phi,s} = 0$, то каждое данное (каждое промежуточное значение элемента массива), используемое на вхождении $(c,2,2)$, требуется доставить в один виртуальный процессор для однократного использования (случай 4); можно показать, что происходит трансляция каждого элемента массива $c$ между виртуальными процессорами.

Для всех трех случаев $\xi_1 = 1$, $\xi_1 = 2$, $\xi_1 = 3$ локальность параллельных реализаций трехмерного цикла (5) примерно одинакова: данные двух массивов многократно используются в каждом виртуальном процессоре, данные одного массива используются однократно в каждом процессоре и требуют доставки во многие процессоры. Если в одном процессоре выполняют операции, отображаемые на набор соседних виртуальных процессор (см. замечание 2), то локальность также примерно одинакова: при $\xi_1 = 1$ и $\xi_1 = 2$ улучшается локальность в случае 5, а при $\xi_1 = 3$ — в случае 4. Какой из альтернативных вариантов выбора $\xi_1$ предпочесть зависит от возможности удачного решения задач выбора зерна вычислений, организации коммуникаций, рационального использования иерархической памяти процессоров.

Таким образом, в работе введены и исследованы величины, характеризующие локальность (а следовательно и коммуникационные затраты) параллельных реализаций многомерных циклов произвольной структуры вложенности. Получены условия локализации в виртуальных процессорах входных и промежуточных данных для многократного использования.

Работа выполнена в рамках государственной программы фундаментальных исследований "Математические модели".

**Литература**

1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб., 2002. 600 с.
2. Bau D., Kodukula I., Kotlyar V., Pingali K., Stodghill P. Proceedings of the Seventh Workshop on Languages and Compilers for Parallel Computing. Springer-Verlag, 1994. P. 46–60. Languages and Compilers for Parallel Computing, 7th International Workshop, LCPC'94, Ithaca, NY, USA, August 8-10, 1994, Proceedings 1995
3. Lim A. W., Lam M. S. // Parallel Computing. 1998. Vol. 24. 3-4. P. 645–475.
4. Лиходед Н. А. // Кибернетика и системный анализ. 2003. 3. С. 169–179.
5. Адуцкевич Е. В., Баханович С. В., Лиходед Н. А. Доклады НАН Беларуси. 2006. Т. 50, 1. С. 34–40.
6. Dion M., Robert Y. // Parallel Computing. 1996. Vol. 22. P. 1373-1397.
7. Garcia J., Ayguade E., Labarta J. IEEE Transactions on parallel and distributed systems. 2001. Vol. 12, 4, P. 416–430.
8. Лиходед Н. А. // Программирование. 2003. 3. С. 73–80.
9. Адуцкевич Е. В., Лиходед Н. А. // Программирование. 2006. 3. С. 54–65.
10. Лиходед Н. А. Доклады НАН Беларуси. 2007. Т. 51. 4. С. 19–24.
11. Pan L., Xue J., Lai M.K., Dillencourt M.B., Bic L.F. data distribution for migrating computations. Int. Conf. on Parallel Processing, Xian, China, 2007.

N.A. LIKHODED likhoded@im.bas-net.by

**CHARACTERIZATION OF LOCALITY OF THE PARALLEL IMPLEMENTATIONS OF IMPERFECTLY NESTED LOOPS**

SUMMARY

Amounts for the characterization of locality of imperfectly nested loops are introduced. Conditions for the localization of input and intermediate data in the virtual processors are obtained.