

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

УДК 681.3.012

Н.А. Лиходед¹, А.А. Толстиков²ПАРАЛЛЕЛЬНЫЕ ВЕРСИИ РЕАЛИЗАЦИИ
МНОГОМЕРНЫХ ЦИКЛОВ

Формулируются и доказываются условия, при выполнении которых параллельные версии алгоритмов, заданных вложенными циклами, можно получить незначительной модификацией исходного последовательного алгоритма. Исследуются загруженность процессоров и задача выбора зерна вычислений.

Введение

Для реализации алгоритма на многопроцессорной вычислительной технике в первую очередь необходимо выявить зависимые операции алгоритма и распределить операции между процессорами таким образом, чтобы они могли работать параллельно [1, 2]. Затем возникает проблема генерации кода: для каждого процессора требуется получить программную запись выполнения операций. Отметим, что многие методы распараллеливания (в частности, рассмотренные в [1, 3–6]) не могут в настоящее время широко применяться на практике из-за трудности генерации кода.

Для вычислений на параллельных компьютерах с распределенной памятью возникают также проблемы, связанные с организацией коммуникаций между процессорами. Одной из основных задач является получение зерна вычислений [7–10]. Под зерном вычислений понимается множество операций алгоритма, выполняемых атомарно: все вычисления, принадлежащие одному зерну, производятся на одном процессоре и не могут прерываться синхронизацией, а обмен данными, связанными с вычислениями одного зерна, не может выполняться на фоне этих вычислений. Выбор зерна вычислений влияет на эффективность параллельной реализации алгоритма, так как определяет объем и частоту передачи пакетов данных, пересылаемых процессорами.

В настоящей работе предлагается метод распараллеливания алгоритмов, заданных вложенными циклами, который приводит к простому решению задач генерации кода и выбора зерна вычислений. Исследуется, в каких случаях метод применим и насколько он эффективен с точки зрения загруженности процессоров.

1. Разбиение операций на блоки и зерна вычислений

Рассмотрим гнездо вложенных циклов

```

do  $j_1 = l_1, u_1$ 
  do  $j_2 = l_2(j_1), u_2(j_1)$ 
    .....
    do  $j_n = l_n(j_1, \dots, j_{n-1}), u_n(j_1, \dots, j_{n-1})$ 
      операторы
    enddo
  .....
enddo
enddo

```

(1)

Циклы не обязательно вложены тесно: между операторами do и между операторами enddo допускается наличие выполняемых операторов.

Пусть целевой компьютер содержит P_1 процессоров, P_1 – заданное число. Выделим цикл с параметром j_{ξ_1} , $\xi_1 > 1$.

Введем следующие обозначения:

V – множество итераций $J = (j_1, \dots, j_n)$ многомерного цикла (1);

$J_i = (j_1, \dots, j_i)$, $1 \leq i \leq n$;

$m_i = \min_{J \in V} j_i$, $M_i = \max_{J \in V} j_i$, $1 \leq i \leq n$;

$B_{\xi_1} = \left\lceil \frac{M_{\xi_1} - m_{\xi_1} + 1}{P_1} \right\rceil$;

$V_{p_1} = \{J(j_1, \dots, j_n) \in V \mid m_{\xi_1} + (p_1 - 1)B_{\xi_1} \leq j_{\xi_1} \leq m_{\xi_1} - 1 + p_1 B_{\xi_1}\}$, $1 \leq p_1 \leq P_1$;

$V_{p_1, \bar{q}} = \{J(q_1, \dots, q_{\xi_1-1}, j_{\xi_1}, \dots, j_n) \in V_{p_1}\}$, $\bar{q} = (q_1, \dots, q_{\xi_1-1})$.

Будем полагать, что множество V_{p_1} является блоком итераций, выполняемых на процессоре с номером p_1 ; обозначим такой процессор $\text{Pr}(p_1)$. Для любой операции, выполняемой оператором S_β на итерации J (операция $S_\beta(J)$), будем считать значение j_{ξ_1} равным m_{ξ_1} , если S_β расположен в записи многомерного цикла выше цикла с параметром j_{ξ_1} (цикла J_{ξ_1}), и равным $m_{\xi_1} - 1 + P_1 B_{\xi_1}$, если S_β расположен ниже цикла J_{ξ_1} . Операторы, не принадлежащие телу цикла J_{ξ_1} , выполняются процессором $\text{Pr}(1)$, если оператор расположен до цикла, и процессором $\text{Pr}(P_1)$, если оператор расположен после цикла. Множество операций, выполняемых на итерациях множества $V_{p_1, \bar{q}}$, будем обозначать также $V_{p_1, \bar{q}}$.

Будем считать, что итерации первых ξ_1 циклов соответствуют точкам ξ_1 -мерного параллелепипеда $\{(j_1, \dots, j_{\xi_1}) \in Z^{\xi_1} \mid m_\xi \leq j_\xi \leq M_\xi, 1 \leq \xi \leq \xi_1\}$. Этим, возможно, добавляются фиктивные итерации, но такого рода предположения позволяют получать содержательные результаты.

Л е м м а 1. Множества операций $V_{p_1, \bar{q}}$ могут являться зерном вычислений.

Действительно, итерации множества $V_{p_1, \bar{q}}$ являются B_{ξ_1} последовательными итерациями цикла J_{ξ_1} . Поэтому эти множества могут выполняться атомарно: все необходимые данные с предыдущих итераций уже получены, данные с последующих итераций не могут потребоваться.

Исследуем возможность уменьшения зерна вычислений. Пусть $\xi_1 < n$. Рассмотрим разбиение множества итераций $V_{p_1, \bar{q}}$ на заданное число Q частей $V_{p_1, \bar{q}, q}$, $1 \leq q \leq Q$, осуществленное посредством разбиения цикла J_{ξ_1+1} .

Л е м м а 2. Множества операций $V_{p_1, \bar{q}, q}$ могут являться зерном вычислений, если

$$\min_{\substack{I, J \\ i_k = j_k, 1 \leq k < \xi_1}} (j_{\xi_1+1} - i_{\xi_1+1}) \geq 0, \quad (2)$$

где $\min_{\substack{I, J \\ i_k = j_k, 1 \leq k < \xi_1}}$ вычисляется по всем $I(i_1, \dots, i_n)$ и $J(j_1, \dots, j_n)$, таким, что операция $S_\beta(J)$ зависит от операции $S_\alpha(I)$ и $i_1 = j_1, \dots, i_{\xi_1-1} = j_{\xi_1-1}$.

Действительно, можно предположить, что для вычисления итераций множества $V_{p_1, \bar{q}, q}$ требуются результаты итераций следующий типов:

- 1) $V_{p_1, \bar{q}', q}$, $\bar{q}' <_{lex} \bar{q}$ (неравенство понимается в лексикографическом смысле), $V_{p_1', \bar{q}, q}$, $p_1' < p_1$, $V_{p_1, \bar{q}, q'}$, $q' < q$; итерации этих зерен уже вычислены;
- 2) $V_{p_1, \bar{q}', q}$, $\bar{q}' >_{lex} \bar{q}$, $V_{p_1', \bar{q}, q}$, $p_1' > p_1$; результаты этих итераций не могут потребоваться в силу лексикографической упорядоченности итераций многомерного цикла;
- 3) $V_{p_1, \bar{q}, q'}$, $q' >_{lex} q$; зависимости, приводящие к такой ситуации, запрещены условием (2).

Отметим, что в каждом зерне вычислений $V_{p_1, \bar{q}}$ и $V_{p_1, \bar{q}, q}$ есть операции $S(j_1, \dots, j_n)$, для которых $j_{\xi_1} = m_{\xi_1} + (p_1 - 1)B_{\xi_1}$, $j_{\xi_1} = m_{\xi_1} - 1 + p_1 B_{\xi_1}$. Из результатов таких операций блока V_{p_1} формируются, как правило, пакеты данных для использования в процессорах, реализующих блоки вычислений V_{p_1-1} и V_{p_1+1} .

2. Основные результаты

Обозначим:

$\delta_{\xi_1}^{(\alpha, \beta)} = \min_{I, J} (j_{\xi_1} - i_{\xi_1})$, где \min вычисляется по всем $I(i_1, \dots, i_n)$ и $J(j_1, \dots, j_n)$, таким, что

операция $S_\beta(J)$ зависит от операции $S_\alpha(I)$;

$$d_{\xi_1}(I(i_1, \dots, i_{\xi_1-1}), J(j_1, \dots, j_{\xi_1-1})) = \sum_{k=1}^{\xi_1-1} (j_k - i_k) \prod_{l=k+1}^{\xi_1-1} (M_l - m_l + 1), \quad \text{при } k = \xi_1 - 1 \text{ запись}$$

$\prod_{l=k+1}^{\xi_1-1} (M_l - m_l + 1)$ означает величину, равную 1;

$$\delta_{\xi_1}^{(\alpha, \beta)} = \begin{cases} \min_{I, J, j_{\xi_1} < i_{\xi_1}} d_{\xi_1}(I_{\xi_1-1}, J_{\xi_1-1}), & \text{если } \delta_{\xi_1}^{(\alpha, \beta)} < 0, \\ 1, & \text{если } \delta_{\xi_1}^{(\alpha, \beta)} \geq 0, \end{cases}$$

где $\min_{I, J, j_{\xi_1} < i_{\xi_1}}$ вычисляется по всем $I(i_1, \dots, i_n)$ и $J(j_1, \dots, j_n)$, таким, что $j_{\xi_1} < i_{\xi_1}$ и операция $S_\beta(J)$ зависит от операции $S_\alpha(I)$, $I_{\xi_1-1} = I(i_1, \dots, i_{\xi_1-1})$, $J_{\xi_1-1} = J(i_1, \dots, i_{\xi_1-1})$;

$$\Delta = \max \left\{ 0, \max_{\alpha, \beta} \left[\left| \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right| / \delta^{(\alpha, \beta)} \right] \right\}.$$

Величины $d_{\xi_1}(I_{\xi_1-1}, J_{\xi_1-1})$ определяют число итераций первых $\xi_1 - 1$ циклов, выполняемых между итерациями I_{ξ_1-1} и J_{ξ_1-1} . Для фиксированной пары (α, β) числа $\left[\frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right]$ определяют

время, необходимое при формировании для фиксированного процессора $\text{Pr}(p_1)$ данных процессорами $\text{Pr}(p_1)$, $p_1' > p_1$, после получения ими результатов вычисления процессором $\text{Pr}(p_1)$ очередного зерна. Числа $\delta_{\xi_1}^{(\alpha, \beta)} - 1$ определяют время, в течение которого может работать процессор $\text{Pr}(p_1)$ после выполнения им очередного зерна до формирования для него данных процессорами $\text{Pr}(p_1)$, $p_1' > p_1$. В случае $\delta_{\xi_1}^{(\alpha, \beta)} \geq 0$ никакой процессор $\text{Pr}(p_1)$ не использует результатов вычислений процессоров $\text{Pr}(p_1)$, $p_1' > p_1$. Отметим, что числа $\delta_{\xi_1}^{(\alpha, \beta)}$ положительны.

Теорема 1. Пусть зерном вычислений являются множества операций $V_{p_1, \bar{q}}$. Многомерный цикл (1) реализуется на P_1 процессорах с асимптотической загруженностью, большей или равной $\frac{1}{\Delta + 1}$.

Доказательство. Примем за единицу времени время выполнения операций одного зерна вычислений, т. е. время выполнения B_{ξ_1} итераций цикла J_{ξ_1} .

Покажем, что можно установить следующий режим выполнения многомерного цикла: процессор $P_1(p_1)$, $1 \leq p_1 \leq P_1$, выполняет итерации в моменты времени

$$T_{\xi_1}(\bar{q}, p_1) = (1 + \Delta)d_{\xi_1}(\bar{m}, \bar{q}) + p_1 - 1, \quad (3)$$

где \bar{q} – итерации первых $\xi_1 - 1$ циклов, $\bar{m} = (m_1, \dots, m_{\xi_1-1})$. Достаточно доказать, что порядок выполнения зависимых операций не изменяется.

Пусть операция $S_\beta(J)$ зависит от операции $S_\alpha(I)$. Обозначим $d_J = d_{\xi_1}(\bar{m}, J_{\xi_1-1})$, $d_I = d_{\xi_1}(\bar{m}, I_{\xi_1-1})$, $p_I = \left\lfloor \frac{i_{\xi_1} - m_{\xi_1}}{B_{\xi_1}} \right\rfloor + 1$ – номер процессора, выполняющего итерацию $I(i_1, \dots, i_n)$, $p_J = \left\lfloor \frac{j_{\xi_1} - m_{\xi_1}}{B_{\xi_1}} \right\rfloor + 1$ – номер процессора, выполняющего итерацию $J(j_1, \dots, j_n)$.

Рассмотрим возможные случаи.

Пусть $J_{\xi_1-1} = I_{\xi_1-1}$. Если $S_\beta(J)$ и $S_\alpha(I)$ принадлежат одному зерну вычислений, то $S_\beta(J)$ выполняется после $S_\alpha(I)$, так как в зерне вычислений сохраняется такой же порядок выполнения операций, как и в исходном алгоритме (1). Если операции принадлежат разным зернам вычислений, то $T_{\xi_1}(J_{\xi_1-1}, p_J) > T_{\xi_1}(I_{\xi_1-1}, p_I)$ в силу $d_J = d_I$, $p_J > p_I$ (следует из равенства $J_{\xi_1-1} = I_{\xi_1-1}$).

Пусть $J_{\xi_1-1} >_{lex} I_{\xi_1-1}$, $j_{\xi_1} \geq i_{\xi_1}$. Тогда $T_{\xi_1}(J_{\xi_1-1}, p_J) > T_{\xi_1}(I_{\xi_1-1}, p_I)$ в силу $d_J > d_I$, $p_J \geq p_I$.

Пусть $J_{\xi_1-1} >_{lex} I_{\xi_1-1}$, $j_{\xi_1} < i_{\xi_1}$. Покажем, что $T_{\xi_1}(J_{\xi_1-1}, p_J) > T_{\xi_1}(I_{\xi_1-1}, p_I)$. Предположим про-

тивное: $(1 + \Delta)d_J + p_J - 1 \leq (1 + \Delta)d_I + p_I - 1$. Тогда $1 + \Delta \leq \frac{p_I - p_J}{d_J - d_I}$, $1 + \Delta \leq \frac{\left\lfloor \frac{-(j_{\xi_1} - i_{\xi_1})}{B_{\xi_1}} \right\rfloor}{d_{\xi_1}(I_{\xi_1-1}, J_{\xi_1-1})}$ (ис-

пользовано свойство $\lfloor a \rfloor - \lfloor b \rfloor \leq \lceil a - b \rceil$), $1 + \Delta \leq \frac{\left\lfloor \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rfloor}{\delta^{(\alpha, \beta)}}$. С другой стороны, с учетом оп-

ределения Δ имеем: $\Delta \geq \left\lceil \frac{\left\lfloor \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rfloor}{\delta^{(\alpha, \beta)}} \right\rceil$, $\Delta \geq \left(\left\lfloor \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rfloor - \delta^{(\alpha, \beta)} + 1 \right) / \delta^{(\alpha, \beta)}$. Получено про-

тиворечие: $\left(\left\lfloor \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rfloor - \delta^{(\alpha, \beta)} + 1 \right) / \delta^{(\alpha, \beta)} \leq \Delta \leq \frac{\left\lfloor \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rfloor}{\delta^{(\alpha, \beta)}} - 1$, следовательно, предположение

$T_{\xi_1}(J_{\xi_1-1}, p_J) \leq T_{\xi_1}(I_{\xi_1-1}, p_I)$ неверно.

Таким образом, выполнение каждого зерна вычислений происходит не позднее момента времени, задаваемого равенством (3). Процессоры простаивают не более Δ единиц времени после выполнения операций одного зерна вычислений, а затем могут начать выполнение очередного зерна. Теорема доказана.

Следствие. Если $\Delta = 0$, в частности, если все $\delta_{\xi_1}^{(\alpha, \beta)}$ неотрицательны, то использование зерна вычислений $V_{p_1, \bar{q}}$ приводит к реализации многомерного цикла (1) на P_1 процессорах с асимптотической загруженностью 1.

Заметим, что если цикл j_{ξ_1} является параллельным, то на каждой фиксированной итерации \bar{q} первых $\xi_1 - 1$ циклов все процессоры могут выполнять вычисления параллельно (с загруженностью 1) независимо от значения величины Δ .

Использование зерна вычислений $V_{p_1, \bar{q}}$ (теорема 1) не создает проблем с генерацией кода. В каждом процессоре выполняется многомерный цикл (1) с незначительными изменениями: в процессоре $\text{Pr}(p_1)$ в цикле j_{ξ_1} выполняются итерации при условии $m_{\xi_1} + (p_1 - 1)B_{\xi_1} \leq j_{\xi_1} \leq m_{\xi_1} - 1 + p_1 B_{\xi_1}$.

Рассмотрим случай использования зерна вычислений, определяемого леммой 2.

Обозначим:

$$B'_{\xi_1+1} = \left\lceil \frac{M_{\xi_1+1} - m_{\xi_1+1} + 1}{Q} \right\rceil;$$

$$\delta_{\xi_1+1}^{(\alpha, \beta)} = \min_{I, J} (j_{\xi_1+1} - i_{\xi_1+1}), \text{ где } \min_{I, J} \text{ вычисляется по всем } I(i_1, \dots, i_n) \text{ и } J(j_1, \dots, j_n), \text{ та-}$$

ким, что операция $S_\beta(J)$ зависит от операции $S_\alpha(I)$;

$$\Delta' = \max \left\{ 0, \max_{\alpha, \beta} \left[\frac{\left\lceil \frac{-\delta_{\xi_1}^{(\alpha, \beta)}}{B_{\xi_1}} \right\rceil + \left\lceil \frac{-\delta_{\xi_1+1}^{(\alpha, \beta)}}{B'_{\xi_1+1}} \right\rceil}{\delta^{(\alpha, \beta)} Q} \right] \right\}.$$

Теорема 2. Пусть выполняются условия (2) и зерном вычислений являются множества операций $V_{p_1, \bar{q}, q}$, $1 \leq q \leq Q$. Многомерный цикл (1) реализуется на P_1 процессорах с асимптотической загруженностью, большей или равной $\frac{1}{\Delta' + 1}$.

Доказательство теоремы 2 аналогично доказательству теоремы 1. Укажем на необходимые изменения в обозначениях. За единицу времени принимается время выполнения операций зерна вычислений $V_{p_1, \bar{q}, q}$. Процессор $\text{Pr}(p_1)$, $1 \leq p_1 \leq P_1$, выполняет q -е, $1 \leq q \leq Q$, зерно вычислений итерации \bar{q} в момент времени $T_{\xi_1}(\bar{q}, p_1, q) = (1 + \Delta')Qd_{\xi_1}(\bar{m}, \bar{q}) + (p_1 - 1) + (q - 1)$;

$q_J = \left\lceil \frac{j_{\xi_1+1} - m_{\xi_1+1}}{B'_{\xi_1+1}} \right\rceil + 1$ – номер зерна вычислений на итерации $J(q_1, \dots, q_{\xi_1-1}, j_{\xi_1}, \dots, j_n)$, вычисляемого процессором $\text{Pr}(p_J)$.

Использование зерна вычислений $V_{p_1, \bar{q}, q}$ требует изменения порядка выполнения итераций циклов ξ_1 и $\xi_1 + 1$. Корректность кода при таких изменениях гарантируется условиями (2).

3. Пример

Рассмотрим трехмерный цикл, представляющий основную часть решения двумерной задачи Дирихле методом Зейделя:

```

do m = 1, M
  do i = 1, Nx - 1
    do j = 1, Ny - 1
      S1: U(i, j) = F(U(i - 1, j), U(i, j - 1), U(i, j + 1), U(i + 1, j))
    enddo
  enddo
enddo

```

(4)

Все зависимости гнезда циклов являются однородными и выражаются векторами зависимостей $\varphi^{(1,1)_1} = (0,1,0)$, $\varphi^{(1,1)_2} = (0,0,1)$, $\varphi^{(1,1)_3} = (1,0,-1)$, $\varphi^{(1,1)_4} = (1,-1,0)$.

Пусть $\xi_1 = 2$. Тогда $\delta_2^{(1,1)_1} = 1$, $\delta_2^{(1,1)_2} = 0$, $\delta_2^{(1,1)_3} = 0$, $\delta_2^{(1,1)_4} = -1$, $\delta^{(1,1)_1} = 1$, $\delta^{(1,1)_2} = 1$, $\delta^{(1,1)_3} = 1$, $\delta^{(1,1)_4} = 1$, $\Delta = 1$. Так как $\Delta > 0$ и цикл по i не является параллельным (из-за наличия векторов зависимостей $(0,1,0)$, $(1,-1,0)$), то согласно теореме 1 трехмерный цикл (4) можно реализовать с асимптотической загруженностью $\frac{1}{2}$. Единый для каждого из D_1 процессоров код можно записать следующим образом:

```

if (1 ≤ p1 ≤ P1)
  do m = 1, M
    do i = 1 + (p1 - 1)B2, min(p1B2, Nx - 1)
      do j = 1, Ny - 1
        S1: U(i, j) = F(U(i - 1, j), U(i, j - 1), U(i, j + 1), U(i + 1, j))
      enddo
    enddo
    обмен данными
  enddo

```

Пусть теперь $\xi_1 = 3$. Тогда $\delta_3^{(1,1)_1} = 0$, $\delta_3^{(1,1)_2} = 1$, $\delta_3^{(1,1)_3} = -1$, $\delta_3^{(1,1)_4} = 0$, $\delta^{(1,1)_1} = 1$, $\delta^{(1,1)_2} = 1$, $\delta^{(1,1)_3} = N_x - 1$, $\delta^{(1,1)_4} = 1$, $\Delta = 0$. Так как $\Delta = 0$, то согласно теореме 1 трехмерный цикл (4) можно реализовать с асимптотической загруженностью 1. Запишем единый для каждого процессора код:

```

if (1 ≤ p1 ≤ P1)
  do m = 1, M
    do i = 1, Nx - 1
      do j = 1 + (p1 - 1)B3, min(p1B3, Ny - 1)
        S1: U(i, j) = F(U(i - 1, j), U(i, j - 1), U(i, j + 1), U(i + 1, j))
      enddo
    enddo
    обмен данными
  enddo
enddo

```

Теперь воспользуемся теоремой 2.

Пусть $\xi_1 = 2$. Тогда $\delta_2^{(1,1)_1} = 1$, $\delta_2^{(1,1)_2} = 0$, $\delta_2^{(1,1)_3} = 0$, $\delta_2^{(1,1)_4} = -1$, $\delta_3^{(1,1)_1} = 0$, $\delta_3^{(1,1)_2} = 1$, $\delta_3^{(1,1)_3} = -1$, $\delta_3^{(1,1)_4} = 0$, $\delta^{(1,1)_1} = 1$, $\delta^{(1,1)_2} = 1$, $\delta^{(1,1)_3} = 1$, $\delta^{(1,1)_4} = 1$. Если $Q = 1$, то имеем $\Delta' = 1$, как и при использовании теоремы 1. Пусть $Q > 1$, тогда $\Delta' = 0$. Так как $\Delta' = 0$ и выполнено условие (2), то согласно теореме 2 трехмерный цикл (4) можно реализовать с асимптотической загруженностью 1. Единый для каждого из P_1 процессоров код можно записать следующим образом:

```

if (1 ≤ p1 ≤ P1)
  do m = 1, M
    do q = 1, Q
      do i = 1 + (p1 - 1)B2, min(p1B2, Nx - 1)
        do j = 1 + (q - 1) B'3, min(q B'3, Ny - 1)
          S1: U(i, j) = F(U(i - 1, j), U(i, j - 1), U(i, j + 1), U(i + 1, j))
        enddo
      enddo
    enddo
  обмен данными
  enddo
enddo

```

Использование теоремы 2 позволяет задавать фиксированные размеры зерна вычислений.

Заключение

В статье разработан метод получения параллельных версий многомерных вложенных циклов для реализации на многопроцессорных вычислительных системах с распределенной памятью. Отличительной особенностью получаемых параллельных реализаций является простота генерации кода: в каждом процессоре реализуется незначительно модифицированный исходный многомерный цикл.

Дальнейшие исследования могут проводиться в следующих направлениях.

- обобщение метода на случай двухмерной ($P_1 \times P_2$ процессоров) и большей размерности архитектуры целевого компьютера;
- выбор циклов, используемых для разбиения на блоки итераций, с учетом условий минимизации коммуникаций между процессорами;
- исследование возможности и эффективности организации блоков итераций и зерен вычислений с использованием несложных преобразований, таких, как слияние, сдвиг, перестановка циклов;
- разработка метода для случая многомерного цикла произвольной структуры вложенности;
- программная реализация алгоритмов организации блоков итераций и зерен вычислений.

Работа выполнена в рамках государственной комплексной программы научных исследований «Инфотех».

Список литературы

1. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.
2. Воеводин В.В. Вычислительная математика и структура алгоритмов / В.В. Воеводин // МГУ [Электронный ресурс] – 2006. – Режим доступа: <http://parallel.ru/info/parallel/voevodin/>. – Дата доступа: 26.12.2007.
3. Darte, A. Mathematical tools for loop transformations: from systems of uniform recurrence equations to the polytope model / A. Darte // Algorithms for Parallel Processing, IMA Volumes in Mathematics and its Applications. – 1999. – Vol. 105. – P. 147–183.

4. Feautrier, P. Some efficient solutions to the affine scheduling problem. Part 2 / P. Feautrier // *Int. J. of Parallel Programming*. – 1992. – Vol. 21, № 6. – P 389–420.
5. Lim, A.W. Maximizing parallelism and minimizing synchronization with affine partitions / A.W. Lim, M.S. Lam // *Parallel Computing*. – 1998. – Vol. 24, № 3, 4. – P. 445–475.
6. Adutskevich, E.V. Affine transformations of loop nests for parallel execution and distribution of data over processors / E.V. Adutskevich, S.V. Bakhanovich, N.A. Likhoded. – Минск, 2005. – 10 с. – (Препринт / НАН Беларуси, Ин-т математики; № 3 (574)).
7. (Pen)-Ultimate Tiling? / P. Boulet [et al.] // *Integration, The VLSI J.* – 1994. – Vol. 17. – P. 33–51.
8. Hodzic, E. On-Time Optimal Supernode Shape / E. Hodzic, W. Shang // *IEEE Transactions on Parallel and Distributed Systems*. – 2002. – Vol. 13, № 12. – P. 1220–1223.
9. Xue, J. Time-minimal tiling when rise is larger than zero / J. Xue, W. Cai // *Parallel Computing*. – 2002. – Vol. 28, № 5. – P. 915–939.
10. Баханович, С.В. Отображение алгоритмов на вычислительные системы с распределенной памятью: оптимизация тайлинга для одно- и двумерных топологий / С.В. Баханович, П.И. Соболевский // *Весці НАН Беларусі. Сер. фіз.-мат. навук*. – 2006. – № 2. – С. 106–112.

Поступила 15.01.08

¹*Институт математики НАН Беларуси,
Минск, Сурганова, 11
e-mail: likhoded@im.bas-net.by*

²*Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: NAXART@yandex.ru*

N.A. Likhoded, A.A. Tolstikov

PARALLEL VERSIONS OF MULTI-DIMENSIONAL LOOP NESTS

Conditions for the parallelization of loop nests by slight modification of initial codes are stated. Processor capacity and a problem of computation grain choice are investigated.