

КРАТКИЕ СООБЩЕНИЯ

УДК 681.3.012

Н.А. Лиходед¹, Ю.Ю. Сокол²**ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ ЭФФЕКТИВНОСТИ
ПРЕОБРАЗОВАНИЯ, УМЕНЬШАЮЩЕГО ОБЪЕМ КОММУНИКАЦИЙ
ПРИ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ**

Экспериментально исследуется влияние применения одного преобразования, уменьшающего обмен данными, на скорость реализации алгоритма на параллельном компьютере с распределенной памятью. Производится сравнение зависимости времени выполнения исходного и преобразованного алгоритмов от размера зерна вычислений. Как следствие формулируются направления возможных теоретических исследований.

Постановка задачи

Экспериментальное исследование эффективности преобразований алгоритмов и сравнение соответствующих параллельных модификаций являются актуальной задачей, так как позволяют сравнивать методы статического распараллеливания и определять направления дальнейших исследований.

Одним из способов повышения эффективности реализации параллельного алгоритма является использование преобразований, уменьшающих число данных, которыми обмениваются процессоры параллельного компьютера. Однако получение таких преобразований и генерация соответствующего кода часто являются довольно сложной задачей. Возникает вопрос: насколько эффективно использование данного вида преобразования, стоит ли выигрывать, полученный в результате применения преобразования, затраченных усилий?

Целью настоящей работы является проведение экспериментального исследования исходного и преобразованного параллельных алгоритмов верхней релаксации приближенного решения двумерной задачи Дирихле для уравнения Пуассона на суперкомпьютере с распределенной памятью СКИФ К-1000. Так как при вычислениях на компьютерах с распределенной памятью обмен данными между процессорами осуществляется пакетами, возникает проблема выбора зерна вычислений [1–4]. Под зерном вычислений понимается множество операций алгоритма, выполняемых атомарно: все вычисления, принадлежащие одному зерну, производятся на одном процессоре и не могут прерываться обменом данными. Выбор зерна определяет объем и частоту передачи пакетов данных.

Из сказанного следует, что основная задача работы – сравнение исходной и преобразованной модификаций алгоритма – естественным образом дополняется задачей исследования влияния размера зерна вычислений на скорость выполнения параллельного алгоритма. Заметим, что задача экспериментального исследования влияния размера зерна вычислений на скорость реализации алгоритма интересна также для однопроцессорного компьютера и связана с проблемой эффективного использования иерархической памяти [5].

Рассмотрим основную часть решения двумерной задачи Дирихле уравнения Пуассона методом верхней релаксации:

```
do m = 1, M
  do i = 1, Nx - 1
    do j = 1, Ny - 1
      u(i, j) = F(u(i - 1, j), u(i, j - 1), u(i, j), u(i, j + 1), u(i + 1, j))
    enddo
  enddo
enddo
```

 (1)

Все зависимости алгоритма, задаваемого гнездом циклов (1), являются однородными и выражаются векторами зависимостей $(0,1,0)$, $(0,0,1)$, $(1,0,0)$, $(1,0,-1)$, $(1,-1,0)$.

Пусть P_1 – число процессоров, предназначенных для реализации трехмерного цикла (1). Единый для каждого из P_1 процессоров псевдокод параллельного алгоритма можно записать следующим образом [6]:

```

if ( $1 \leq p_1 \leq P_1$ )
  do  $m = 1, M$ 
    do  $q = 1, Q$ 
      do  $i = 1 + (p_1 - 1)B_2, \min(p_1B_2, N_x - 1)$ 
        do  $j = 1 + (q - 1)B'_3, \min(qB'_3, N_y - 1)$ 
           $u(i, j) = F(u(i - 1, j), u(i, j - 1), u(i, j), u(i, j + 1), u(i + 1, j))$       (2)
        enddo
      enddo
    enddo
    обмен данными
  enddo
enddo

```

Каждый процессор p_1 на каждой итерации m выполняет операции Q зерен вычислений: одному зерну соответствуют два самых внутренних цикла. Параметры B_2 и B'_3 задают размер зерна. Число операций, выполняемых процессором p_1 на итерации m , задается произведением B_2N_y . Можно показать, что объем данных, который передается при выполнении межпроцессорной коммуникационной операции, характеризует параметр B'_3 . Поэтому далее размер зерна вычислений будем задавать целым положительным числом B'_3 .

Наличие векторов зависимостей $(0,1,0)$ и $(1,-1,0)$ приводит к обмену данными по второй координате, отвечающей за номер процессора, как в прямом, так и в обратном направлениях. Обмен данными между процессорами можно вдвое уменьшить, если к алгоритму (1) применить аффинное преобразование $(m, i, j) \rightarrow (m, m + i, j)$. Такое преобразование называется скашиванием цикла и приводит к гнезду циклов

```

do  $m = 1, M$ 
  do  $i = m + 1, N_x + m - 1$ 
    do  $j = 1, N_y - 1$ 
       $u(i - m, j) = F(u(i - m - 1, j), u(i - m, j - 1), u(i - m, j), u(i - m, j + 1), u(i - m + 1, j))$ 
    enddo
  enddo
enddo

```

Теперь зависимости задаются векторами $(0,1,0)$, $(0,0,1)$, $(1,1,0)$, $(1,1,-1)$, $(1,0,0)$, поэтому по второй координате, отвечающей за номер процессора, обмен данными в обратном направлении отсутствует. Для этого алгоритма, по аналогии с алгоритмом (2), можно записать псевдокод, который далее будем называть преобразованным алгоритмом; алгоритм (2) будем называть исходным алгоритмом.

В настоящей работе ставится задача сравнения времени выполнения на суперкомпьютере СКИФ К-1000 исходного и преобразованного алгоритмов, исследования влияния размера зерна вычислений на скорость выполнения алгоритмов и, как следствие, формулируются направления дальнейших теоретических исследований.

Результаты вычислений

Все вычисления, результаты которых представлены в этом разделе, проводились при следующих значениях параметров алгоритмов: $N_x = N_y = 10\,000$, $M = 100$.

Результаты вычислительных экспериментов при размере зерна, равном 2000, представлены на рис. 1 и 2. Из рисунков видно, что применение к алгоритму преобразования привело к значительному улучшению характеристик времени реализации алгоритма и ускорения.

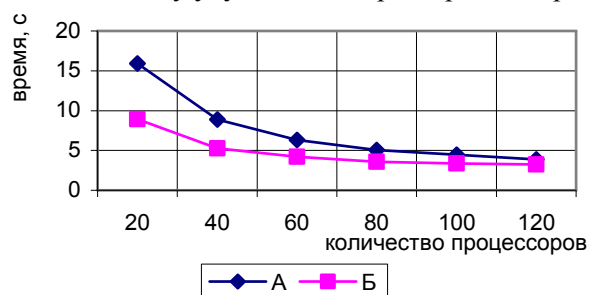


Рис. 1. График зависимости времени решения задачи от количества процессоров (зерно вычислений равно 2000): А – исходный алгоритм; Б – преобразованный алгоритм

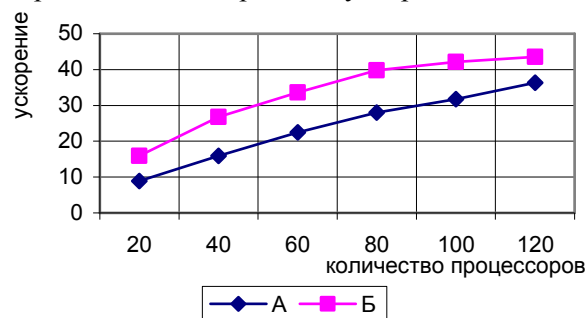


Рис. 2. График зависимости ускорения вычислений от количества процессоров (зерно вычислений равно 2000): А – исходный алгоритм; Б – преобразованный алгоритм

Проведем теперь вычисления с размером зерна, равным 200 (рис. 3 и 4). Видно, что преобразование гнезда циклов в данном случае не дало преимущества. При этом характеристики улучшились по сравнению с использованием зерна размера 2000 как для исходного алгоритма (причем значительно), так и для преобразованного.

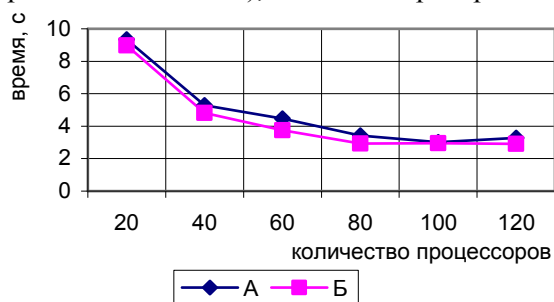


Рис. 3. График зависимости времени решения задачи от количества процессоров (зерно вычислений равно 200): А – исходный алгоритм; Б – преобразованный алгоритм

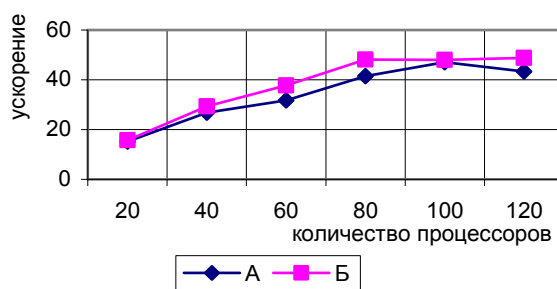


Рис. 4. График зависимости ускорения вычислений от количества процессоров (зерно вычислений равно 200): А – исходный алгоритм; Б – преобразованный алгоритм

Таким образом, размер зерна вычислений может сильно влиять на время вычислений. Было проведено экспериментальное нахождение оптимального размера зерна вычислений при фиксированном (20 и 40) количестве процессоров. Результаты представлены на рис. 5 и 6.

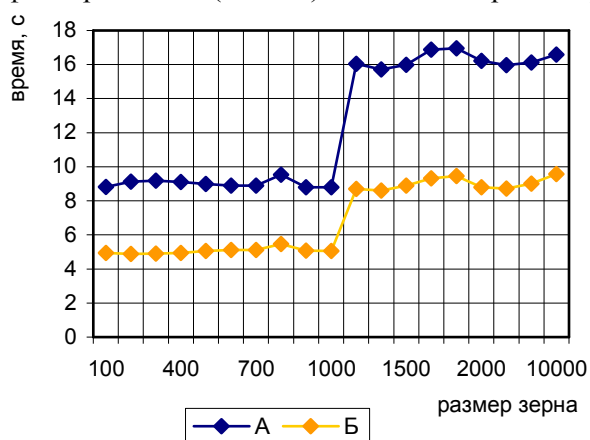


Рис. 5. График зависимости времени реализации исходного алгоритма от размера зерна вычислений: А – на 20 процессорах; Б – на 40 процессорах

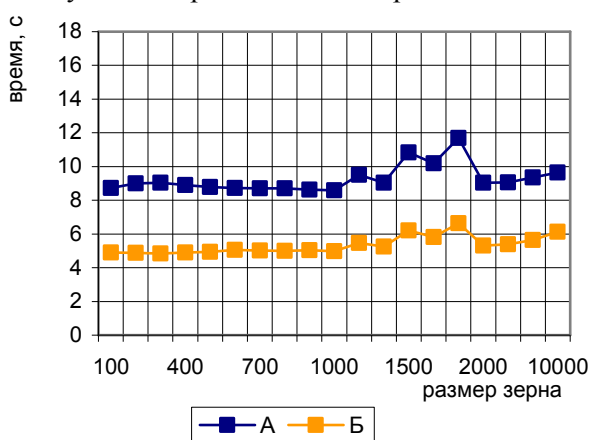


Рис. 6. График зависимости времени реализации преобразованного алгоритма от размера зерна вычислений: А – на 20 процессорах; Б – на 40 процессорах

Для исходного алгоритма существует порог, после которого увеличение размера зерна приводит к резкому увеличению времени вычислений. Для преобразованного алгоритма размер зерна меньше влияет на время вычислений, но в обоих случаях следует выбирать зерно размером до 1000.

Заключение

В работе проведены и проанализированы численные эксперименты на параллельном компьютере СКИФ К-1000, получены во многом неожиданные результаты сравнения исходного и преобразованного (теоретически лучшего для реализации на параллельных компьютерах с распределенной памятью) алгоритмов.

В результате проведения экспериментальных исследований было установлено, что при удачном выборе размера зерна вычислений использование одного из преобразований алгоритмов, уменьшивших число коммуникаций (скачивание цикла), может не дать ощутимого уменьшения времени решения задачи. Однако при не оптимальном выборе размера зерна вычислений выигрыш от использования преобразования может быть значительным.

Таким образом, представляют интерес следующие теоретические исследования в области построения параллельных алгоритмов:

1. Исследование условий, при выполнении которых возможны упрощенная генерация кода параллельной программы для исходного алгоритма и выбор произвольного фиксированного размера зерна вычислений. В этом случае можно надеяться, что для эффективной реализации алгоритма не потребуется прилагать значительных усилий.

2. Разработка методов получения параллельных алгоритмов с уменьшенными коммуникационными затратами. Применение таких методов может быть полезно на практике, по крайней мере в тех случаях, когда невозможно уменьшить размер зерна вычислений до близкого к оптимальному.

Список литературы

1. (Pen)-Ultimate Tiling? / P. Boulet [et al.] // *Integration, The VLSI J.* – 1994. – Vol. 17. – P. 33–51.
2. Xue, J. Time-minimal tiling when rise is larger than zero / J. Xue, W. Cai // *Parallel Computing.* – 2002. – Vol. 28, № 5. – P. 915–939.
3. Баханович, С.В. Отображение алгоритмов на вычислительные системы с распределенной памятью: оптимизация тайлинга для одно- и двумерных топологий / С.В. Баханович, П.И. Соболевский // *Весті НАН Беларусі. Сер. фіз.-мат. навук.* – 2006. – № 2. – С. 106–112.
4. Лиходед, Н.А. О выборе зерна вычислений при реализации алгоритмов на параллельных компьютерах с распределенной памятью / Н.А. Лиходед, А.К. Пашкович // *Весті НАН Беларусі. Сер. фіз.-мат. навук.* – 2008. – № 2. – С. 121–123.
5. Ahmed, N. Synthesizing transformations for locality enhancement of imperfectly-nested loop nests / N. Ahmed, N. Mateev, K. Pingali // *International Journal of Parallel Programming.* – 2001. – Vol. 29, № 2. – P. 493–544.
6. Лиходед, Н.А. Параллельные версии реализации многомерных циклов / Н.А. Лиходед, А.А. Толстикова // *Информатика.* – 2008. – № 2 (18). – С. 129–136.

Поступила 08.09.08

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

²Белорусский государственный университет,
Минск, пр. Независимости, 4
sokoly@cosmostv.by

N.A. Likhoded, Y.Y. Sokol

**PARALLEL IMPLEMENTATION OF ALGORITHMS:
EXPERIMENTAL INVESTIGATION OF THE EFFICIENCY
OF A TRANSFORMATION REDUCING COMMUNICATION**

An experimental study of the influence of a communication-reducing transformation on the execution time is presented. The execution time for original and transformed algorithms depending on the size of computation grain is compared. As a result, the further directions of possible theoretical research are formulated.