УДК 519.67

ПОСТРОЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ НА ОСНОВЕ ГЕКСАГОНАЛЬНОГО ТАЙЛИНГА

С.В. Баханович[1], Н.А. Лиходед[2]

[1]*Институт математики НАН Беларуси,*
ул. Сурганова, 11, 220072, г. Минск, Беларусь, bsv@im.bas-net.by
[2]*Белорусский государственный университет,*
пр. Независимости, 4, 220030 г. Минск, Беларусь, likhoded@bsu.by

Разработан подход к построению вычислительных процессов для реализации алгоритмов решения сеточных задач на суперкомпьютерах с распределенной памятью. Подход использует гексагональный тайлинг. Он позволяет процессам одновременно начинать вычисления, не приводит к перекрытиям вычислений и не требует предварительного преобразования области вычислений.

***Ключевые слова*:** параллельные вычисления; многопроцессорная система с распределенной памятью; гексагонаьный тайлинг; параллельные вычислительные процессы.

CONSTRUCTION OF PARALLEL COMPUTIONAL PROCESSES BASED ON HEXAGONAL TILING

S.V. Bakhanovich[a], N.A. Likhoded[b]

[a]*Institute of Mathematics, National Academy of Science,*
6 Surhanava Street, Minsk 220072, Belarus, bsv@im.bas-net.by
[b]*Belarusian State University,*
4 Niezalieznasci Avenue, Minsk 220030, Belarus, likhoded@bsu.by

An approach to the construction of computational processes for implementation of algorithms for stencil computations on supercomputers with distributed memory has been developed. The approach uses hexagonal tiling, so it allows processes to start calculations at the same time, does not lead to overlapping calculations, does not require a preliminary transformation of the computational domain.

Keywords: parallel computing; distributed memory parallel computer; hexagonal tiling; parallel computing processes.

Введение

Тайлинг (tiling) – выделение макроопераций для получения алгоритмов блочного типа – применяется для построения эффективных параллельных алгоритмов и для уменьшения накладных расходов на использование иерархической памяти [1]. При тайлинге производится разбиение области вычислений (итерационного пространства) алгоритма семействами гиперплоскостей на макрооперации-тайлы.

Для возможности осуществить тайлинг часто необходимо применять предварительное аффинное преобразование алгоритма, например, скос по одной или более координат итерационного пространства.

Без скашивания циклов осуществляется так называемый тайлинг с перекрытием вычислений (overlapped tiling) [2]. Тайлинг с перекрытием (тайлинг с дублированием некоторых вычислений), организованный на одном или нескольких итерационных (или временных) слоях, позволяет все вычисления расширенных тайлов производить без обращения к результатам вычислений других тайлов этих слоев. Основным преимуществом получаемых зернистых алгоритмов считается возможность одновременно начинать выполнение операций многих тайлов, в то время как после скашивания происходит, как правило, разгон и торможение вычислений. Недостатком такого подхода являются накладные расходы, часто значительные, связанные с дублированием вычислений.

Гексагональный тайлинг (hexagonal tiling) – это тайлинг, при котором макрооперации-тайлы в плоскости первых двух координат имеют очертания (форму) шестиугольников [3–5]. Тайлинг осуществляется без перекрытий вычислений и приводит к минимально возможному числу зависимостей между тайлами по первым двум направлениям. Гексагональный тайлинг позволяет получить макрооперации-тайлы (гексагонаьные тайлы), причем по координатам, начиная с третьей, производится традиционный тайлинг (в итоге получается так называемый гибридный тайлинг) или по этим координатам тайлинг вообще не производится. Гибридный тайлинг успешно применяется для реализации сеточных задач на графических ускорителях [6, 7].

В этой работе разработан подход к построению на основе гексагонального тайлинга вычислительных процессов для реализации алгоритмов решения сеточных задач на суперкомпьютерах с распределенной памятью. Отметим некоторые особенности подхода: нет перекрытий вычислений, он позволяет процессам одновременно начинать вычисления, соответственно отсутствует разгон-торможение процессов. Подход не требует предварительного преобразования области вычислений, приводящего к скошенной области вычислений и, соответственно, к дисбалансу вычислительной нагрузки).

1. Параллельный алгоритм с гексагональными тайлами

Будем рассматривать один из классов алгоритмов, возникающих при решении сеточных задач. Пусть алгоритм имеет вид

do $l = 1, r_{it}$ // r_{it} – некоторое фиксированное число итераций

do $i = 1, N_x–1$ // $N_x–1$ – число строк матрицы, задающей внутренние узлы сетки

do $j = 1, N_y–1$ // $N_y–1$ – число столбцов матрицы, задающей внутренние узлы

$$y(i,j)=F((y(i–1,j–1),y(i–1,j),y(i–1,j+1),y(i,j–1), y(i,j),y(i,j+1),$$
$$y(i+1,j–1),y(i+1,j),y(i+1,j+1))$$

Здесь параметр внешнего цикла является номером итерации, функция F считается известной, $y(i,j)$ – значения неизвестной функции в узлах сетки; если i и/или j равны 0, N_x, N_y, то величины $y(i,j)$ известны – они равны значению некоторой функции в граничных узлах сетки. Все зависимости алгоритма являются однородными и выражаются векторами зависимостей (0,1,0), (0,0,1), (1,0,0), (0,1,1), (0,1,–1), (1,–1,1), (1,–1,–1), (1,0,–1), (1,–1,0).

Будем рассматривать две части гексагональных тайлов: Δ-тайлы и ∇-тайлы; рисунок поясняет дальнейшие рассуждения. Эти тайлы в плоскости первых двух координат имеют форму равнобедренных трапеций. Введём параметры h и w, которые имеют следующий смысл: число точек области вычислений на «высоте» Δ-тайлов и ∇-тайлов равно $h+1$, число точек области вычислений на меньшем «основании» Δ- и ∇-тайлов равно $w+1$.

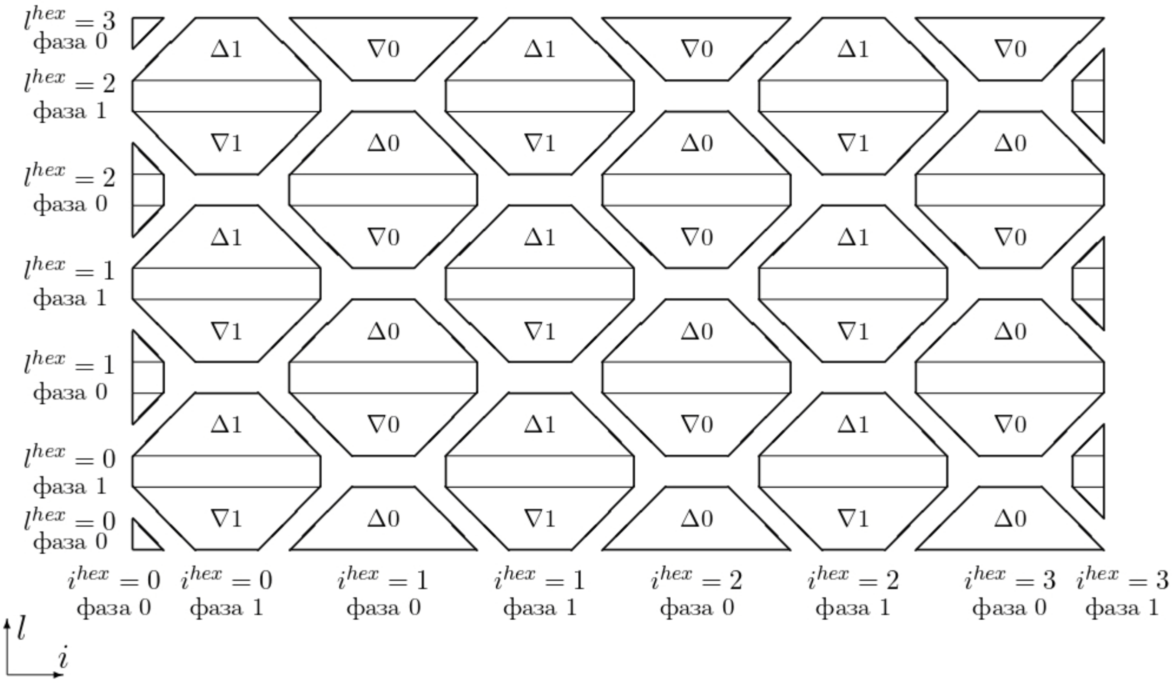

Рисунок – Схематичное изображение гексагональных тайлов (h=2, w=2)

Для простоты изложения будем предполагать, что числа $L^{hex}=\dfrac{r_{it}}{2(h+1)}$ и $I^{hex}=\dfrac{N_x-1-h}{2(h+w+1)}$ являются целыми. Разделяют две фазы вычислений уровня тайлов. Соответственно будем рассматривать Δ- и ∇-тайлы фазы 0 и Δ- и ∇-тайлы фазы 1. Обозначим эти тайлы через $\mathrm{Tile}^{\Delta 0}(l^{hex},i^{hex})$, $\mathrm{Tile}^{\Delta 1}(l^{hex},i^{hex})$, $\mathrm{Tile}^{\nabla 0}(l^{hex},i^{hex})$, $\mathrm{Tile}^{\nabla 1}(l^{hex},i^{hex})$, где параметры l^{hex}, $0\le l^{hex}\le L^{hex}$, и i^{hex}, $0\le i^{hex}\le I^{hex}$, задают нумерацию тайлов.

Если $0\le l^{hex}\le L^{hex}-1$, $i^{hex}=0$, то Δ-тайл фазы 0, обозначим его $\mathrm{Tile}^{\wedge 0}(l^{hex},i^{hex})$, является неполным. Если $1\le l^{hex}\le L^{hex}$, $i^{hex}=0$, то ∇-тайл фазы 0, обозначим его $\mathrm{Tile}^{\vee 0}(l^{hex},i^{hex})$, является неполным. Если $0\le l^{hex}\le L^{hex}-1$, $i^{hex}=I^{hex}$, то Δ-тайл фазы 1, обозначим его $\mathrm{Tile}^{\wedge 1}(l^{hex},i^{hex})$, является неполным. Если $0\le l^{hex}\le L^{hex}-1$, $i^{hex}=I^{hex}$, то ∇-тайл фазы 1, обозначим его $\mathrm{Tile}^{\vee 1}(l^{hex},i^{hex})$, является неполным.

Алгоритм (уровня макроопераций) гексагонального тайлинга:

$l^{hex} = 0$
dopar $i^{hex} = 0, I^{hex}$
  if $i^{hex}=0$ **then** $\mathrm{Tile}^{\wedge 0}(l^{hex},i^{hex})$ **else** $\mathrm{Tile}^{\Delta 0}(l^{hex},i^{hex})$
enddopar
dopar $i^{hex} = 0, I^{hex}$
  if $i^{hex}=I^{hex}$ **then** $\mathrm{Tile}^{\vee 1}(l^{hex},I^{hex})\cup\mathrm{Tile}^{\wedge 1}(l^{hex},I^{hex})$
  else $\mathrm{Tile}^{\nabla 1}(l^{hex},i^{hex})\cup\mathrm{Tile}^{\Delta 1}(l^{hex},i^{hex})$
enddopar
do $l^{hex} = 1, L^{hex}-1$
  dopar $i^{hex} = 0, I^{hex}$
    if $i^{hex}=0$ **then** $\mathrm{Tile}^{\vee 0}(l^{hex},0)\cup\mathrm{Tile}^{\wedge 0}(l^{hex},0)$
    else $\mathrm{Tile}^{\nabla 0}(l^{hex},i^{hex})\cup\mathrm{Tile}^{\Delta 0}(l^{hex},i^{hex})$
  enddopar
  dopar $i^{hex} = 0, I^{hex}$
    if $i^{hex}=I^{he}$ **then** $\mathrm{Tile}^{\vee 1}(l^{hex},I^{hex})\cup\mathrm{Tile}^{\wedge 1}(l^{hex},I^{hex})$
    else $\mathrm{Tile}^{\nabla 1}(l^{hex},i^{hex})\cup\mathrm{Tile}^{\Delta 1}(l^{hex},i^{hex})$
  enddopar
enddo
$l^{hex} = L^{hex}$
dopar $i^{hex} = 0, I^{he}$
  if $i^{hex}=0$ **then** $\mathrm{Tile}^{\vee 0}(l^{hex},i^{hex})$ **else** $\mathrm{Tile}^{\nabla 0}(l^{hex},i^{hex})$
enddopar

2. Псевдокод вычислительного процесса

Пусть параметр цикла i^{hex} определяет ранг (номер) процесса. Для каждого процесса Pr_p, $0 \le p \le I^{hex}$ можно записать следующий псевдокод:

Сформировать матрицу, задающую начальное приближение во внутренних узлах сетки; граничные узлы сетки заполняются точными значениями.

// Итерация $l^{hex} = 0$. Получение данных для вычислений фазы 0 не требуется,

// все необходимые для этой фазы данные уже присвоены

if p=0 **then** $\mathrm{Tile}^{\wedge 0}(l^{hex},0)$ **else** $\mathrm{Tile}^{\Delta 0}(l^{hex},p)$

Send(p-1) // Передача результатов вычислений $\mathrm{Tile}^{\Delta 0}(l^{hex},p)$ процессу p-1;

Recv(p+1) // Получение результатов вычислений $\mathrm{Tile}^{\Delta 0}(l^{hex},p+1)$ от процесса p+1

if $p=I^{hex}$ **then** $\mathrm{Tile}^{\vee 1}(l^{hex},I^{hex}) \cup \mathrm{Tile}^{\wedge 1}(l^{hex},I^{hex})$ **else** $\mathrm{Tile}^{\nabla 1}(l^{hex},p) \cup \mathrm{Tile}^{\Delta 1}(l^{hex},p)$

Send(p+1) // Передача результатов вычислений $\mathrm{Tile}^{\Delta 1}(l^{hex},p)$ процессу p+1

do $l^{hex} = 1, L^{hex}-1$

  Recv (p-1) // Получение результатов вычислений $\mathrm{Tile}^{\Delta 1}(l^{hex}-1,p-1)$

  if p=0 **then** $\mathrm{Tile}^{\vee 0}(l^{hex},0) \cup \mathrm{Tile}^{\wedge 0}(l^{hex},0)$ **else** $\mathrm{Tile}^{\nabla 0}(l^{hex},p) \cup \mathrm{Tile}^{\Delta 0}(l^{hex},p)$

  Send(p-1) // Передача результатов вычислений $\mathrm{Tile}^{\Delta 0}(l^{hex},p)$ процессу p-1;

  Recv(p+1) // Получение результатов вычислений $\mathrm{Tile}^{\Delta 0}(l^{hex},p+1)$

  if $p=I^{he}$ **then** $\mathrm{Tile}^{\vee 1}(l^{hex},I^{hex}) \cup \mathrm{Tile}^{\wedge 1}(l^{hex},I^{hex})$ **else** $\mathrm{Tile}^{\nabla 1}(l^{hex},p) \cup \mathrm{Tile}^{\Delta 1}(l^{hex},p)$

  Send(p+1) // Передача результатов вычислений $\mathrm{Tile}^{\Delta 1}(l^{hex},p)$ процессу p+1

enddo

// Итерация $l^{hex} = L^{hex}$

Recv (p-1) // Получение результатов вычислений $\mathrm{Tile}^{\Delta 1}(L^{hex}-1,p-1)$

if p=0 **then** $\mathrm{Tile}^{\vee 0}(l^{hex},0)$ **else** $\mathrm{Tile}^{\nabla 0}(l^{hex},p)$

Таким образом, предложена методика построения параллельных вычислительных 1D процессов для выполнения сеточных вычислений зейделевского типа. Аналогичная методика разработана в работе [8], но она основана на классическом, а не на гексагональном тайлинге. Поэтому методика работы [8] предполагает предварительное преобразование алгоритма (скос по одной из координат итерационного пространства), требуется разгон и торможение вычислений.

В дальнейшем предполагается предлагаемую методику обобщить для получения 2D структуры вычислительных процессов; координаты процесса будут определять параметры циклов i^{hex} и l^{hex}.

Библиографические ссылки

1. Xue J. Loop tiling for parallelism. Springer Science & Business Media. 2000. 256 p. DOI: 10.1007/978-1-4615-4337-4.
2. Holewinski J., Pouchet L.-N., Sadayappan P. High-performance code generation for stencil computations on GPU architectures // Supercomputing. 2012. P. 311–320. DOI: 10.1145/2304576.2304619.
3. Grosser T., Verdoolaege S., Cohen A., Sadayappan P. The relation between diamond tiling and hexagonal tiling // First International Workshop on High-Performance Stencil Computations. January, 2014. Vienna, Austria. DOI: 10.1142/S0129626414410023.
4. Соболевский П.И., Баханович С.В. Плотное покрытие области вычислений гексагональными тайлами // Доклады НАН Беларуси. 2018. № 62(5). С. 525–530.
5. Соболевский П.И., Баханович С.В. Анализ глобальных зависимостей в гексагональном тайлинге // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. 2020. № 56(1). С. 114–126. DOI: 10.29235/1561-2430-2020-56-1-114-126.
6. Grosser T., Cohen A., Holewinski J., Sadayappan P, Verdoolaege S. Hybrid hexagonal/classical tiling for GPUs // 12th Annual IEEE/ACM International Symposium on Code Generation and Optimization, Orlando, FL, USA, February 15-19, 2014. P. 66–75. DOI: 10.1145/2544137.2544160.
7. Prajapati N, Ranasinghe W, Rajopadhye S., Andonov R, Djidjev H, Grosser T. Simple, accurate, analytical time modeling and optimal tile size selection for GPGPU stencils // Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. January 2017. P. 163–177. DOI: 10.1145/3018743.3018744.
8. Renganarayanan L., Harthikote-Matha M., Dewri R., Rajopadhye S. Towards optimal multi-level tiling for stencil computations // 21st IEEE International Parallel and Distributed Processing Symposium. 2007. DOI: 10.1109/IPDPS.2007.370291.