Вычислительная математика

Computational mathematics

УДК 519.67

УСЛОВИЯ ПРИВАТИЗАЦИИ ЭЛЕМЕНТОВ МАССИВА ПОТОКАМИ ВЫЧИСЛЕНИЙ

Н. А. ЛИХОДЕД[1)]***, М. А. ПОЛЕЩУК***[1)]

[1)]*Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь*

Множество операций параллельного алгоритма для реализации на графическом процессоре должно быть разбито на потоки (нити) вычислений. Потоки следует сгруппировать в блоки вычислений, выполняющиеся атомарно на потоковых процессорах, называемых также мультипроцессорами. Для хорошей производительности графического процессора важно, чтобы как можно больше данных умещались в быстрых регистровой и разделяемой памяти, иначе используются медленные глобальная и локальная память. Степень использования памяти с быстрым доступом отражает вычислительное свойство алгоритма, называемое локальностью. При реализации алгоритмов на многопроцессорных вычислительных устройствах применение локальности играет важнейшую роль для достижения высокой производительности. В данной работе сформулированы и доказаны необходимые условия и достаточные условия, использование которых позволяет получать потоки с приватизированными данными, т. е. такие потоки вычислений, что элемент массива используется только одним потоком, и поэтому его целесообразно разместить в регистре.

Ключевые слова: параллельные вычисления; графический процессор; тайлинг; приватизация элементов массива; регистры.

Благодарность. Работа выполнена в рамках государственной программы научных исследований Республики Беларусь «Конвергенция-2020» (подпрограмма «Методы математического моделирования сложных систем»).

Образец цитирования:
Лиходед НА, Полещук МА. Условия приватизации элементов массива потоками вычислений. *Журнал Белорусского государственного университета. Математика. Информатика.* 2018;3:59–67.

For citation:
Likhoded NA, Paliashchuk MA. Conditions for privatizing the elements of arrays by computing threads. *Journal of the Belarusian State University. Mathematics and Informatics.* 2018;3: 59–67. Russian.

Авторы:
Николай Александрович Лиходед – доктор физико-математических наук, профессор; профессор кафедры вычислительной математики факультета прикладной математики и информатики.
Максим Александрович Полещук – ассистент кафедры вычислительной математики факультета прикладной математики и информатики.

Authors:
Nikolai A. Likhoded, doctor of science (physics and mathematics), full professor; professor at the department of computational mathematics, faculty of applied mathematics and computer science.
likhoded@bsu.by
Maksim A. Paliashchuk, assistant at the department of computational mathematics, faculty of applied mathematics and computer science.
poleschuma@bsu.by

Журнал Белорусского государственного университета. Математика. Информатика. 2018;3:59–67
Journal of the Belarusian State University. Mathematics and Informatics. 2018;3:59–67

CONDITIONS FOR PRIVATIZING THE ELEMENTS OF ARRAYS BY COMPUTING THREADS

N. A. LIKHODED[a]***, M. A. PALIASHCHUK***[a]

[a]*Belarusian State University, 4 Niezaliežnasci Avenue, Minsk 220030, Belarus*
Corresponding author: N. A. Likhoded (likhoded@bsu.by)

The set of operations of the parallel algorithm for implementation on the GPU must be split into computation threads. The threads must be grouped into computation units that run atomically on stream processors, also called multiprocessors. For good GPU performance, it is important that as much data as possible can fit into fast register and shared memory, otherwise slow global and local memory are used. The degree of memory usage with fast access reflects the computational property of the algorithm, called locality. When implementing algorithms on multiprocessor computing devices, the use of locality plays a crucial role in achieving high performance. In this paper, necessary conditions and sufficient conditions have been formulated and proved, the use of which allows receiving threads with privatized data, i. e. it allows to receive such computation threads that the array element is used only by one thread and therefore it is advisable to place it in the register.

Key words: parallel computations; GPU; tiling; array privatization; registers.

Acknowledgements. 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»).

Введение

В качестве целевого компьютера для реализации алгоритмов будем рассматривать графические процессоры (GPU) – довольно мощные и в настоящее время широко используемые многоядерные устройства. При вычислениях на GPU быстрым является процесс обращения к регистрам (самый быстрый вид памяти), разделяемой памяти мультипроцессора и кешам, но не обращение к глобальной и локальной памяти GPU. Регистры мультипроцессора делятся поровну между потоками блока вычислений. Особенно эффективно используется регистровая память, если элемент массива приватизирован потоком. Под приватизацией понимается использование элемента массива в вычислениях только одного потока блока вычислений. Если элемент массива приватизирован, то он может быть размещен в регистрах, выделенных потоку. Цель данной работы – формулировка и доказательство необходимых условий и достаточных условий приватизации элементов массива потоками вычислений.

Степень использования памяти с быстрым доступом отражает вычислительное свойство алгоритма, называемое локальностью. При реализации алгоритмов на многопроцессорных устройствах локальность играет важнейшую роль для достижения высокой производительности [1; 2].

Отметим некоторые исследования локальности алгоритмов для реализации на GPU. В работе [2] на примере решения задачи поиска кратчайших путей показана особая для GPU эффективность использования свойства локальности. В работе [3] сформулированы и доказаны утверждения, позволяющие ранжировать параметры размера блоков на основе асимптотических оценок объема коммуникационных операций; в [4] подобные оценки получены с использованием более сложного (но и лучше приспособленного для автоматизации) математического аппарата. Известен способ получения более точной оценки количества элементов, к которым осуществляется доступ при выполнении операций блока вычислений, но практическое применение этого результата довольно трудоемкое и требует привлечения специализированных средств автоматизации [5; 6]. В работе [7] сформулированы необходимые условия использования элемента массива только одним потоком вычислений и, соответственно, размещения в регистрах.

Разбиение множества операций алгоритма на блоки и потоки вычислений

Пусть алгоритм задан гнездом вложенных циклов, в котором 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} – область изменения параметров циклов, окружающих θ-й набор операторов, $1 \le \theta \le \Theta$; n^{θ} – размерность этой области, число циклов,

окружающих θ-й набор операторов; ν_l – размерности массивов a_l. Также обозначим размеры блоков вычислений натуральными числами $r_1^\theta, \ldots, r_{n^\theta}^\theta$; r_ζ^θ – число значений параметра j_ζ, приходящихся на один блок θ-го набора операторов; Q_ζ^θ – число блоков θ-го набора операторов по координате с номером ζ (обозначение этой координаты – j_ζ^{gl}). Нумеровать блоки вычислений будем по каждой координате в пределах от 0 до $Q_\zeta^\theta - 1$, $1 \le \zeta \le n^\theta$. Блоки обозначим $V_{J^{gl}}^\theta$, где $J^{gl} = \left(j_1^{gl}, \ldots, j_{n^\theta}^{gl}\right)$, $0 \le j_\zeta^{gl} \le Q_\zeta^\theta - 1$, $1 \le \zeta \le n^\theta$. Далее для простоты записи будем часто использовать обозначения без индекса θ.

Зададим в блоках вычислений потоки вычислений посредством выделения блоков второго уровня. Зададим размеры потоков натуральными числами $r_{1,2}, r_{2,2}, \ldots, r_{n^\theta,2}$; $r_{\zeta,2}$ – число значений параметра j_ζ, приходящихся на один поток; $Q_{1,2}, Q_{2,2}, \ldots, Q_{n^\theta,2}$ – число таких потоков; $Q_{\zeta,2} = \left\lceil \dfrac{r_\zeta}{r_{\zeta,2}} \right\rceil$.

В общем случае разбить множество операций алгоритма на блоки и потоки можно путем тайлинга – преобразования алгоритма для получения макроопераций [8].

Вхождением $\left(a_l, S_\beta, q\right)$ будем называть q-е вхождение массива a_l в оператор S_β. Индексы элементов l-го массива, связанных с вхождением $\left(a_l, S_\beta, q\right)$, выражаются функцией вида

$$\bar{F}_{a_l, S_\beta, q}(J) = F_{a_l, S_\beta, q} J + f^{a_l, S_\beta, q}, \; J\left(j_1, \ldots, j_{n^\theta}\right) \in V^\theta, \; F_{a_l, S_\beta, q} \in \mathbb{Z}^{\nu_l \times n^\theta}, \; f^{a_l, S_\beta, q} \in \mathbb{Z}^{\nu_l}.$$

Пример 1. Рассмотрим основную часть алгоритма Флойда – Уоршелла поиска кратчайших путей между всеми парами вершин графа:

```
do k = 1, n
 do i = 1, n
  do j = 1, n
   a(i, j) = min(a(i, j), a(i, k) + a(k, j))
  enddo
 enddo
enddo
```

В гнезде циклов имеется один выполняемый оператор S_1 (один набор операторов), и используется один массив a размерности 2. Область изменения параметров циклов (область итераций) $V^1 = \left\{(k, i, j) \in \right.$ $\left. \in \mathbb{Z}^3 \middle| 1 \le k \le n, \; 1 \le i \le n, \; 1 \le j \le n \right\}$ для оператора S_1 имеет размерность 3. Для матриц $F_{a, S_1, q}$ на вхождениях $\left(a, S_1, q\right)$ имеем

$$F_{a, S_1, 1} = F_{a, S_2, 2} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; F_{a, S_1, 3} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \; F_{a, S_1, 4} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Известны блочные алгоритмы с 3D-блоками размером $r \times r \times r$ для реализации на GPU [9; 10]. Заметим, что одинаковые размеры блока имеют существенное значение, а не выбраны для простоты. Пусть k^{gl}, i^{gl}, j^{gl} – номера частей, на которые при формировании блоков разбиваются области значений параметров k, i, j циклов. Блоки $V_{k^{gl}, i^{gl}, j^{gl}}$ имеют следующий вид:

```
do k = 1 + k^gl r, min((k^gl + 1)r, n)
 do i = 1 + i^gl r, min((i^gl + 1)r, n)
  do j = 1 + j^gl r, min((j^gl + 1)r, n)
   a(i, j) = min(a(i, j), a(i, k) + a(k, j))
  enddo
 enddo
enddo
```

Обозначим через $r_{k,2}$, $r_{i,2}$ и $r_{j,2}$ размеры блоков второго уровня. Числа $Q_{k,2} = \left\lceil \frac{r}{r_{k,2}} \right\rceil$, $Q_{i,2} = \left\lceil \frac{r}{r_{i,2}} \right\rceil$ и $Q_{j,2} = \left\lceil \frac{r}{r_{j,2}} \right\rceil$ задают количество итераций в новых циклах с параметрами k^{gl2}, i^{gl2} и j^{gl2} соответственно. Пусть $r_{k,2} = r$, тогда $Q_{k,2} = 1$. Блок $V_{k^{gl},\, i^{gl},\, j^{gl}}$ с выделенными потоками $\mathrm{Thr}\left(i^{gl2},\ j^{gl2}\right)$ имеет следующее представление:

$$
\begin{aligned}
&\text{do } i^{gl2} = 0,\ Q_{i,2} - 1\\
&\ \text{do } j^{gl2} = 0,\ Q_{j,2} - 1\\
&\ \ \mathrm{Thr}\left(i^{gl2},\ j^{gl2}\right):\\
&\ \ \text{do } k = 1 + k^{gl} r,\ \min\left(\left(k^{gl}+1\right)r,\ n\right)\\
&\ \ \ \text{do } i = 1 + i^{gl} r + i^{gl2} r_{i,2},\ \min\left(i^{gl} r + \left(i^{gl2}+1\right)r_{i,2},\ \left(i^{gl2}+1\right)r,\ n\right)\\
&\ \ \ \text{do } j = 1 + j^{gl} r + j^{gl2} r_{j,2},\ \min\left(j^{gl} r + \left(j^{gl2}+1\right)r_{j,2},\ \left(j^{gl}+1\right)r,\ n\right)\\
&\ \ \ \ a(i,\ j) = \min\left(a(i,\ j),\ a(i,\ k) + a(k,\ j)\right)\\
&\ \ \ \text{enddo}\\
&\ \ \text{enddo}\\
&\ \text{enddo}\\
&\text{enddo}\\
&\text{enddo}
\end{aligned}
\qquad (1)
$$

Необходимое условие приватизации элементов массива

Сформулируем и докажем условия, налагаемые на матрицу $F_{a_l,\, S_\beta,\, q}$, без выполнения которых невозможна приватизация элементов массива, используемых на вхождении $\left(a_l,\ S_\beta,\ q\right)$, в любом из трех возможных способов задания потоков вычислений (потоки могут помечаться как узлы одномерной, двумерной или трехмерной сетки).

Введем в рассмотрение множество $Z_s = \left\{\zeta_1, \ldots, \zeta_m\right\} \subset \left\{1, \ldots, n^\theta\right\}$, составленное из одного ($m = 1$), двух ($m = 2$) или трех ($m = 3$) произвольных элементов множества $\left\{1, \ldots, n^\theta\right\}$, и множество $Z_t = \left\{\zeta_{m+1}, \ldots, \zeta_{n^\theta}\right\} = \left\{1, \ldots, n^\theta\right\} \setminus Z_s$. Для удобства будем считать множества Z_s и Z_t упорядоченными: $\zeta_i < \zeta_j$, если $i < j$ (отдельно для любых $\zeta_i, \zeta_j \in Z_s$ и $\zeta_i, \zeta_j \in Z_t$).

Полагаем, что $j_{\zeta_1}^{gl2}, \ldots, j_{\zeta_m}^{gl2}$ задают координаты потоков, и если $\zeta \in Z_s$, то $1 \le r_{\zeta,2} < r_\zeta$, а если $\zeta \in Z_t$, то $r_{\zeta,2} = r_\zeta$. Таким образом, m координат задают m-мерное пространство потоков, а остальные $n^\theta - m$ координат – итерации, выполняемые потоками в лексикографическом порядке.

Введем обозначения: T_s – матрица, строки которой составлены из векторов $e_\zeta^{(n^\theta)}$, $\zeta \in Z_s$, где $e_\zeta^{(n^\theta)}$ – вектор-строка размером n^θ, у которого координата с номером ζ равна 1, а остальные координаты нулевые; $\rho_{a_l,\, S_\beta,\, q} = \mathrm{rank} F_{a_l,\, S_\beta,\, q}$, $\rho^s_{a_l,\, S_\beta,\, q} = \mathrm{rank}\begin{pmatrix} F_{a_l,\, S_\beta,\, q} \\ T_s \end{pmatrix}$; u_i – базисные векторы (размерности n^θ) подпространства $\ker F_{a_l,\, S_\beta,\, q}$ пространства Z^{n^θ}, $1 \le i \le n^\theta - \rho_{a_l,\, S_\beta,\, q}$. Если $\rho_{a_l,\, S_\beta,\, q} = n^\theta$, то подпространство $\ker F_{a_l,\, S_\beta,\, q}$ имеет нулевую размерность.

Теорема 1. *Пусть зафиксирован блок вычислений, потоки вычислений* $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, \ldots, j_{\zeta_m}^{gl2}\right)$ *помечены как узлы одномерной, или двумерной, или трехмерной сетки и для вхождения* $\left(a_l,\ S_\beta,\ q\right)$ *массива* a_l *в оператор* S_β *выполнено условие*

$$\rho_{a_l,\, S_\beta,\, q} = \rho^s_{a_l,\, S_\beta,\, q}. \qquad (2)$$

Тогда каждый элемент массива a_l используется на вхождении $\left(a_l,\, S_\beta,\, q\right)$ только одним потоком. Если $\rho_{a_l,\, S_\beta,\, q} < \rho^s_{a_l,\, S_\beta,\, q}$, то элемент массива используется, вообще говоря, более чем одним потоком.

Доказательство. Сперва рассмотрим случай $r_{\zeta,\,2} = 1$, $\zeta \in Z_s$. Тогда один поток $\mathrm{Thr}\left(j_{\zeta_1},\, \ldots,\, j_{\zeta_m}\right)$ составляют все вычисления блока при фиксированных значениях $j_{\zeta_1},\, \ldots,\, j_{\zeta_m}$, причем $\left(j_{\zeta_1},\, \ldots,\, j_{\zeta_m}\right)^{\mathrm{T}} = T_s\left(j_1,\, \ldots,\, j_{n^\theta}\right)^{\mathrm{T}}$.

Одним из результатов работы [11] (частный случай следствия 2) является следующее утверждение: для каждого элемента массива, связанного с вхождением $\left(a_l,\, S_\beta,\, q\right)$, значения координат потоков, в которых используется этот элемент, отличаются линейными комбинациями векторов $T_s u_i$, $1 \le i \le n^\theta - \rho_{a_l,\, S_\beta,\, q}$.

Если $\rho_{a_l,\, S_\beta,\, q} = \rho^s_{a_l,\, S_\beta,\, q}$, то $\ker\begin{pmatrix} F_{a_l,\, S_\beta,\, q} \\ T_s \end{pmatrix} = \ker F_{a_l,\, S_\beta,\, q}$, поэтому $T_s u_i = 0$ для всех u_i. Следовательно, в этом случае каждый элемент массива используется только одним потоком. Если $\rho_{a_l,\, S_\beta,\, q} < \rho^s_{a_l,\, S_\beta,\, q}$, то $\ker\begin{pmatrix} F_{a_l,\, S_\beta,\, q} \\ T_s \end{pmatrix} \ne \ker F_{a_l,\, S_\beta,\, q}$, поэтому для каких-то u_i имеет место $T_s u_i \ne 0$. Тогда элемент массива используется (если только этот элемент не используется однократно на границе блока вычислений) более чем одним потоком.

Пусть теперь $1 < r_{\zeta,\,2} < r_\zeta$ для каких-то (возможно, всех) $\zeta \in Z_s$. Тогда один поток $\mathrm{Thr}\left(j^{gl2}_{\zeta_1},\, \ldots,\, j^{gl2}_{\zeta_m}\right)$ составляют вычисления с более чем одним значением таких j_ζ. Если элемент массива используется только одним потоком при $r_{\zeta,\,2} = 1$, то тем более он будет использоваться только одним потоком при $r_{\zeta,\,2} > 1$. Если элемент массива используется более чем одним потоком при $r_{\zeta,\,2} = 1$, то при $r_{\zeta,\,2} > 1$ он будет использоваться меньшим числом потоков, но с учетом $r_{\zeta,\,2} < r_\zeta$ все же более чем одним (исключение, как и при $r_{\zeta,\,2} = 1$, возможно на границе блока вычислений).

Частные случаи теоремы 1 ($m = 1$, $m = 2$) сформулированы в работе [7].

Пример 1 (продолжение). Пусть блоки имеют вид (1). Наиболее интересными с точки зрения эффективного использования быстрой регистровой памяти являются такие варианты выбора матрицы T_s, при которых условие (2) выполняется для вхождений $\left(a,\, S_1,\, 1\right)$ и $\left(a,\, S_1,\, 2\right)$: на этих вхождениях осуществляется считывание и запись данного $a\left(i,\, j\right)$ (на других вхождениях – только считывание $a\left(i,\, k\right)$ или $a\left(k,\, j\right)$). Нетрудно видеть, что условия $\rho_{a,\, S_1,\, 1} = \rho^s_{a,\, S_1,\, 1}$, $\rho_{a,\, S_1,\, 2} = \rho^s_{a,\, S_1,\, 2}$ выполняются, если матрица T_s не содержит строки $e^{(3)}_1 = \left(1\ 0\ 0\right)$. Например, если $Z_s = \left\{2,\, 3\right\}$, то

$$\rho_{a,\, S_1,\, 1} = \mathrm{rank}\, F_{a,\, S_1,\, 1} = \mathrm{rank}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 2,\ \rho^s_{a,\, S_1,\, 1} = \mathrm{rank}\begin{pmatrix} F_{a,\, S_1,\, 1} \\ T_s \end{pmatrix} = \mathrm{rank}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 2.$$

Пример 2. Основную часть алгоритма численного решения двумерной задачи Дирихле для уравнения Пуассона методом последовательной верхней релаксации можно представить (после аффинного преобразования итерационной области, применяемого для возможности получать блочные версии) в следующем виде:

```
do t = 1, T
  do i = t + 1, t + N_x − 1
    do j = t + 1, t + N_y − 1
     y(i − t, j − t) = F(y(i − t − 1, j − t), y(i − t, j − t − 1), y(i − t, j − t), y(i − t + 1, j − t), y(i − t, j − t + 1))
    enddo
  enddo
enddo
```

Здесь T – некоторое фиксированное число итераций метода релаксации, N_x и N_y характеризуют число частей, на которые разбивается итерационная область.

Для любого вхождения (y, S_1, q) и для любого выбора матрицы T_s имеем

$$F_{a_l, S_\beta, q} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \ \rho_{a_l, S_\beta, q} = 2, \ \rho^s_{a_l, S_\beta, q} = 3$$

(ранг матрицы $F_{a_l, S_\beta, q}$ равен 2, но после добавления в матрицу хотя бы одной из строк $(1\,0\,0)$, $(0\,1\,0)$, $(0\,0\,1)$ ранг полученной матрицы станет равен 3). Поэтому для любого варианта организации потоков выполняется $\rho_{a_l, S_\beta, q} < \rho^s_{a_l, S_\beta, q}$, каждый элемент массива используется более чем одним потоком, и элементы массива y не могут размещаться в регистрах.

Достаточные условия приватизации элементов массива

Условие (2) гарантирует использование элемента массива a_l на вхождении (a_l, S_β, q) только одним потоком. Для исследования возможности приватизации элемента массива надо принять во внимание и другие вхождения элементов этого массива в операторы данного блока вычислений.

Обозначим через $\bar{F}_{a_l, S_\beta, q}\left(V^\theta_{J^{gl}}\right)$ образ множества $V^\theta_{J^{gl}}$ после применения отображения $\bar{F}_{a_l, S_\beta, q}$. По определению

$$\bar{F}_{a_l, S_\beta, q}\left(V^\theta_{J^{gl}}\right) = \left\{ F \in \mathbb{Z}^{v_l} \,\middle|\, F = \bar{F}_{a_l, S_\beta, q}(J),\ J \in V^\theta_{J^{gl}} \right\}.$$

Теорема 2. *Пусть зафиксирован блок вычислений $V^\theta_{J^{gl}}$, потоки вычислений $\mathrm{Thr}\left(j^{gl2}_{\zeta_1}, \ldots, j^{gl2}_{\zeta_m}\right)$ помечены как узлы одномерной, или двумерной, или трехмерной сетки и для вхождения (a_l, S_β, q) массива a_l в оператор S_β выполнено условие (2). Для того чтобы каждый элемент массива a_l, возникающий на вхождении (a_l, S_β, q), использовался только одним потоком, достаточно выполнения условий*

$$\bar{F}_{a_l, S_\beta, q}\left(V^\theta_{J^{gl}}\right) \cap \bar{F}_{a_l, S_\gamma, y}\left(V^\theta_{J^{gl}}\right) = \varnothing$$

или

$$F_{a_l, S_\gamma, y} = F_{a_l, S_\beta, q}, \ f^{a_l, S_\gamma, y} = f^{a_l, S_\beta, q},$$

где (a_l, S_γ, y) есть любое, отличное от (a_l, S_β, q) вхождение массива a_l в операторы блока вычислений $V^\theta_{J^{gl}}$.

Доказательство. Пусть имеет место условие (2) и пусть сначала $r_{\zeta, 2} = 1$, $\zeta \in Z_s$. Тогда один поток $\mathrm{Thr}\left(j_{\zeta_1}, \ldots, j_{\zeta_m}\right)$ составляют все вычисления блока при фиксированных значениях $j_{\zeta_1}, \ldots, j_{\zeta_m}$ и на вхождении (a_l, S_β, q) каждый элемент массива a_l используется только одним потоком.

Если $\bar{F}_{a_l, S_\beta, q}\left(V^\theta_{J^{gl}}\right) \cap \bar{F}_{a_l, S_\gamma, y}\left(V^\theta_{J^{gl}}\right) = \varnothing$, то на вхождении (a_l, S_β, q) и на любом другом вхождении (a_l, S_γ, y) используются разные элементы массива, поэтому другие вхождения не влияют на использование массива a_l на вхождении (a_l, S_β, q) только одним потоком.

Пусть $F_{a_l, S_\gamma, y} = F_{a_l, S_\beta, q}$, $f^{a_l, S_\gamma, y} = f^{a_l, S_\beta, q}$. Покажем, что на вхождениях (a_l, S_β, q) и (a_l, S_γ, y) один и тот же элемент массива может появиться только в одном потоке, т. е. если $\bar{F}_{a_l, S_\beta, q}(J) = \bar{F}_{a_l, S_\gamma, y}(I)$, $I, J \in V^\theta_{J^{gl}}$, то вычисления итераций I и J выполняются одним потоком.

Так как $F_{a_l, S_\beta, q} J + f^{a_l, S_\beta, q} = F_{a_l, S_\gamma, y} I + f^{a_l, S_\gamma, y}$, $F_{a_l, S_\beta, q}(J - I) = 0$, то $J - I \in \ker F_{a_l, S_\beta, q}$. Поэтому векторы I и J отличаются линейными комбинациями базисных векторов u_i, $1 \le i \le n^\theta - \rho_{a_l, S_\beta, q}$, подпространства $\ker F_{a_l, S_\beta, q}$. Тогда координаты потоков $T_s I$ и $T_s J$, которые выполняют операции итераций I и J, отличаются линейными комбинациями векторов $T_s u_i$. Так как $\rho_{a_l, S_\beta, q} = \rho^s_{a_l, S_\beta, q}$, то $T_s u_i = 0$ для всех u_i, вычисления итераций I и J выполняются одним потоком.

Пусть теперь $1 < r_{\zeta,2} < r_\zeta$ для каких-то (или всех) $\zeta \in Z_s$. Тогда один поток $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, \ldots, j_{\zeta_m}^{gl2}\right)$ составляют вычисления с более чем одним значением таких j_ζ. Если элемент массива используется только одним потоком при $r_{\zeta,2} = 1$, то тем более и при $r_{\zeta,2} > 1$ он будет использоваться лишь одним потоком.

Пример 1 (окончание). Рассмотрим блоки вычислений $V_{k^{gl},\, i^{gl},\, j^{gl}}$, для которых $i^{gl} \neq k^{gl}, j^{gl} \neq k^{gl}$. Покажем, что если $T_s = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}$, или $T_s = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$, или $T_s = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, то на вхождениях $(a, S_1, 1)$ и $(a, S_1, 2)$ происходит приватизация потоками данных $a(i, j)$. В предыдущем разделе установлено, что при таком выборе матрицы T_s условие (2) выполнено. Осталось показать, что выполнены и остальные условия теоремы 2. Имеем

$$\bar{F}_{a,S_1,1}(k, i, j) = \bar{F}_{a,S_1,2}(k, i, j) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}(k, i, j)^{\mathrm{T}} = (i, j)^{\mathrm{T}},$$

$$\bar{F}_{a,S_1,1}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) = \bar{F}_{a,S_1,2}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) =$$

$$= \left\{(F_1, F_2) \in \mathbb{Z}^2 \,\middle|\, (F_1, F_2) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}(k, i, j)^{\mathrm{T}} = (i, j), (k, i, j) \in V_{k^{gl},\, i^{gl},\, j^{gl}}\right\} =$$

$$= \left\{(F_1, F_2) \in \mathbb{Z}^2 \,\middle|\, 1 + i^{gl} r \le F_1 \le \left(i^{gl} + 1\right), 1 + j^{gl} r \le F_2 \le \left(j^{gl} + 1\right)\right\}.$$

Для вхождений $(a, S_1, 3)$, $(a, S_1, 4)$, на которых осуществляется считывание данных $a(i, k)$, $a(k, j)$, имеем соответственно:

$$\bar{F}_{a,S_1,3}(k, i, j) = (i, k)^{\mathrm{T}},$$

$$\bar{F}_{a,S_1,3}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) = \left\{(F_1, F_2) \in \mathbb{Z}^2 \,\middle|\, 1 + i^{gl} r \le F_1 \le \left(i^{gl} + 1\right), 1 + k^{gl} r \le F_2 \le \left(k^{gl} + 1\right)\right\};$$

$$\bar{F}_{a,S_1,4}(k, i, j) = (k, j)^{\mathrm{T}},$$

$$\bar{F}_{a,S_1,4}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) = \left\{(F_1, F_2) \in \mathbb{Z}^2 \,\middle|\, 1 + k^{gl} r \le F_1 \le \left(k^{gl} + 1\right), 1 + j^{gl} r \le F_2 \le \left(j^{gl} + 1\right)\right\}.$$

Так как $i^{gl} \neq k^{gl}$, $j^{gl} \neq k^{gl}$, то $\bar{F}_{a,S_1,1}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) \cap \bar{F}_{a,S_1,3}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) = \varnothing$ (пересечений нет по второй координате), $\bar{F}_{a,S_1,1}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) \cap \bar{F}_{a,S_1,4}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right) = \varnothing$ (пересечений нет по первой координате). Следовательно, выполняются достаточные условия приватизации.

Замечание 1. В рассмотренном примере множества $\bar{F}_{a,S_1,q}\left(V_{k^{gl},\, i^{gl},\, j^{gl}}\right)$ получить довольно просто. Для построения $\bar{F}_{a_l,S_\beta,q}\left(V_{J^{gl}}^{\theta}\right)$ в общем случае можно воспользоваться методом получения отображений выпуклых ограниченных многогранников [12; 13].

Замечание 2. Пусть, как и ранее, $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, \ldots, j_{\zeta_m}^{gl2}\right)$ – потоки вычислений, для вхождения (a_l, S_β, q) массива a_l в оператор S_β выполнено условие (2), а для любого другого вхождения (a_l, S_γ, y) справедливо $F_{a_l,S_\gamma,y} = F_{a_l,S_\beta,q}$, m строк матрицы $F_{a_l,S_\beta,q}$ совпадают со строками матрицы T_s и соответствующие координаты вектора $f^{a_l,S_\beta,q} - f^{a_l,S_\gamma,y}$ равны нулю. Тогда каждый элемент массива a_l, возникающий на вхождениях (a_l, S_β, q), (a_l, S_γ, y), используется только одним потоком.

Действительно, покажем, что если $\bar{F}_{a_l,S_\beta,q}(J) = \bar{F}_{a_l,S_\gamma,y}(I)$, $I, J \in V_{J^{gl}}^{\theta}$, то $T_s I = T_s J$. Так как $F_{a_l,S_\beta,q} J + f^{a_l,S_\beta,q} = F_{a_l,S_\gamma,y} I + f^{a_l,S_\gamma,y}$, $F_{a_l,S_\gamma,y} = F_{a_l,S_\beta,q}$, то $F_{a_l,S_\beta,q}(J - I) = f^{a_l,S_\beta,q} - f^{a_l,S_\gamma,y}$. Отсюда, поскольку

m строк матрицы $F_{a_l, S_\beta, q}$ совпадают со строками матрицы T_s и соответствующие координаты вектора $f^{a_l, S_\beta, q} - f^{a_l, S_\gamma, y}$ равны нулю, получим $T_s(J - I) = 0$.

Достаточные условия, приведенные в замечании 2, можно применить, например, в случае итераций (k, i, j), оператора вида $a(i, j) = a(i, j-1) + \ldots$, $T_s = (0 \quad 1 \quad 0)$ (потоки определяет координата i).

Заключение

В работе сформулированы и доказаны необходимые условия и достаточные условия приватизации элементов массива потоками вычислений при реализации алгоритма на графическом процессоре. Приватизация элементов массива позволяет разместить их в самой быстрой памяти процессора – регистрах.

Направления дальнейших исследований: формализация условий, при которых также происходит эффективное использование быстрой памяти графического процессора – бродкаст (потоки запрашивают одновременно один и тот же элемент массива) и пространственная локальность (потоки используют близко расположенные в памяти данные); применение сформулированных условий для автоматизации получения алгоритмов, реализуемых на графических процессорах; использование результатов работы при разработке параллельных алгоритмов для решения прикладных задач.

Библиографические ссылки

1. Воеводин ВлВ, Воеводин ВадВ. Спасительная локальность суперкомпьютеров. *Открытые системы.* 2013;9:12–15.

2. Buluc A, Gilberta JR, Budak C. Solving path problems on the GPU. *Parallel Computing.* 2010;36(5–6):241–253.

3. Лиходед НА, Полещук МА. Оценка локальности параллельных алгоритмов, реализуемых на графических процессорах. *Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика.* 2016:5(3):96–111. DOI: 10.14529/cmse160307.

4. Лиходед НА, Полещук МА. Метод ранжирования параметров размера блоков вычислений параллельного алгоритма. *Доклады Национальной академии наук Беларуси*. 2015;59(4):25–33.

5. Kandemir M, Ramanujam J, Irwin M, Narayanan V, Kadayif I, Parikh A. A compiler based approach for dynamically managing scratch-pad memories in embedded systems. *IEEE Transactions on Computer-Aided Design.* 2004;23(2):243–260.

6. Baskaran M, Bondhugula U, Krishnamoorthy S, Ramanujam J, Rountev A, Sadayappan P. Automatic data movement and computation mapping for multi-level parallel architectures with explicitly managed memories. In: *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming; 2008 February 20–23; Salt Lake City, USA.* New York: ACM; 2008. p. 1–10. DOI: 10.1145/1345206.1345210.

7. Полещук МА, Лиходед НА. Приватизация элементов массивов потоками вычислений. В: *CSIST'2016. Международный конгресс по информатике: информационные системы и технологии; 24–27 октября 2016 г.; Минск, Беларусь.* Минск: БГУ; 2016. с. 883–888.

8. Baskaran M, Ramanujam J, Sadayappan P. Automatic C-to-CUDA code generation for affine programs. In: *Compiler Construction. 19th International Conference. Part of the Joint European Conferences on Theory and Practice of Software; 2010 March 20–28; Paphos, Cyprus.* Berlin, Heidelberg: Springer-Verlag; 2010. DOI: 10.1007/978-3-642-11970-5_14.

9. Katz GJ, Kider J. All-pairs shortest-paths for large graphs on the GPU. In: *Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware; 2008 June 20–21; Sarajevo, Bosnia and Herzegovina.* Geneve: Eurographics Association; 2008. p. 47–55.

10. Lund BD, Smith JW. A Multi-Stage {CUDA} Kernel for Floyd-Warshall. CoRR [Internet]. 2010;abs/1001.4108. Available from: http://arxiv.org/abs/1001.4108.

11. Адуцкевич ЕВ, Лиходед НА, Сикорский АО. К распараллеливанию последовательных программ: распределение массивов между процессорами и структуризация коммуникаций. *Кибернетика и системный анализ.* 2012;48(1):144–163.

12. Воеводин ВВ. *Информационная структура алгоритмов.* Москва: МГУ; 1997.

13. Лиходед НА, Толстиков АА. Формализация коммуникационных операций многомерных циклов. *Известия Национальной академии наук Беларуси. Серия физико-математических наук.* 2010;3:109–114.

References

1. Voevodin VlV, Voevodin VadV. [The fortunate locality of supercomputers]. *Otkrytye sistemy* [Open systems]. 2013;9:12–15. Russian.

2. Buluc A, Gilberta JR, Budak C. Solving path problems on the GPU. *Parallel Computing.* 2010;36(5–6):241–253.

3. Likhoded NA, Paliashchuk MA. [Estimate of locality of parallel algorithms implemented on GPUs]. *Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika* [Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering]. 2016:5(3):96–111. Russian. DOI: 10.14529/cmse160307.

4. Likhoded NA, Poleshchuk MA. Method of ranking tiles size parameters of parallel algorithm. *Doklady of the National Academy of Sciences of Belarus.* 2015;59(4):25–33. Russian.

5. Kandemir M, Ramanujam J, Irwin M, Narayanan V, Kadayif I, Parikh A. A compiler based approach for dynamically managing scratch-pad memories in embedded systems. *IEEE Transactions on Computer-Aided Design.* 2004;23(2):243–260.

6. Baskaran M, Bondhugula U, Krishnamoorthy S, Ramanujam J, Rountev A, Sadayappan P. Automatic data movement and computation mapping for multi-level parallel architectures with explicitly managed memories. In: *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming; 2008 February 20–23; Salt Lake City, USA.* New York: ACM; 2008. p. 1–10. DOI: 10.1145/1345206.1345210.

7. Poleshchuk MA, Likhoded NA. [Array privatization by computing threads]. In: *CSIST'2016. Mezhdunarodnyi kongress po informatike: informatsionnye sistemy i tekhnologii; 24–27 oktyabrya 2016 g.; Minsk, Belarus'* [CSIST'2016. International Congress on Computer Science: Information Systems and Technologies; 2016 October 24–27; Minsk, Belarus]. Minsk: Belarusian State University; 2016. p. 883–888. Russian.

8. Baskaran M, Ramanujam J, Sadayappan P. Automatic C-to-CUDA code generation for affine programs. In: *Compiler Construction. 19th International Conference. Part of the Joint European Conferences on Theory and Practice of Software; 2010 March 20–28; Paphos, Cyprus.* Berlin, Heidelberg: Springer-Verlag; 2010. DOI: 10.1007/978-3-642-11970-5_14.

9. Katz GJ, Kider J. All-pairs shortest-paths for large graphs on the GPU. In: *Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware; 2008 June 20–21; Sarajevo, Bosnia and Herzegovina.* Geneve: Eurographics Association; 2008. p. 47–55.

10. Lund BD, Smith JW. A Multi-Stage {CUDA} Kernel for Floyd-Warshall. CoRR [Internet]. 2010;abs/1001.4108. Available from: http://arxiv.org/abs/1001.4108.

11. Adutskevich EV, Likhoded NA, Sikorsky AO. [Parallelization of sequential programs: distribution of arrays among processors and structurization of communications]. *Kibernetika i sistemnyi analiz* [Cybernetics and System Analysis]. 2012;48(1):144–163. Russian.

12. Voevodin VV. *Informatsionnaya struktura algoritmov* [Informational structure of algorithms]. Moscow: MGU; 1997. Russian.

13. Likhoded NA, Tolstikov AA. Formalization of communication operations of multidimensional loops. *Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics Series.* 2010;3:109–114. Russian.

Статья поступила в редколлегию 12.06.2018.
Received by editorial board 12.06.2018.