ISSN 1561-2430 (Print)
ISSN 2524-2415 (Online)
УДК 519.67
https://doi.org/10.29235/1561-2430-2022-58-3-292-299

Поступила в редакцию 11.01.2022
Received 11.01.2022

Н. А. Лиходед

Белорусский государственный университет, Минск, Республика Беларусь

УСЛОВИЯ СУЩЕСТВОВАНИЯ БРОДКАСТА И ПРОСТРАНСТВЕННОЙ ЛОКАЛЬНОСТИ В ПОТОКАХ ВЫЧИСЛЕНИЙ

Аннотация. В качестве компьютера, на котором требуется реализовать параллельную версию алгоритма, рассматриваются графические процессоры (GPU). Множество операций алгоритма для выполнения на GPU должно быть разбито на потоки вычислений; потоки должны быть сгруппированы в блоки вычислений, выполняющиеся атомарно на мультипроцессорах. Потоки одного блока выполняются на мультипроцессоре частями-пулами, называемыми варпами (warps); потоки варпа выполняются одновременно. Эффективность параллельного алгоритма зависит от способа размещения данных в памяти GPU. Если все потоки варпа запрашивают при выполнении текущего оператора один и тот же элемент массива, то его желательно размещать в разделяемой или константной памяти GPU; в этом случае его распределение по ядрам мультипроцессора реализуется фактически посредством бродкаста (broadcast). Если потоки варпа запрашивают близко расположенные в памяти данные, то в этом случае имеет место их пространственная локальность, что делает целесообразным размещение этих данных в текстурной памяти GPU. Реализация бродкаста или пространственной локальности за счет размещения данных в памяти соответствующего вида позволяет существенно снизить трафик при обмене ими между уровнями памяти графического процессора. В работе сформулированы и доказаны необходимые и достаточные условия, при которых возможно выполнение бродкаста или имеет место пространственная локальность данных. Условия даны в терминах функций, определяющих использование элементов массивов на вхождениях в операторы алгоритма, и функций, задающих информационные зависимости алгоритма. Полученные результаты могут быть использованы для оптимизации параллельных алгоритмов при их реализации на GPU.

Ключевые слова: параллельные вычисления, графический процессор, бродкаст, пространственная локальность

Для цитирования. Лиходед, Н. А. Условия существования бродкаста и пространственной локальности в потоках вычислений / Н. А. Лиходед // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2022. – Т. 58, № 3. – С. 292–299. https://doi.org/10.29235/1561-2430-2022-58-3-292-299

Nikolai A. Likhoded

Belarusian State University, Minsk, Republic of Belarus

CONDITIONS FOR THE EXISTENCE OF BROADCAST AND SPATIAL LOCALITY IN COMPUTATION THREADS

Abstract. Graphics Processing Units (GPUs) are considered as the target computer for implementing parallel algorithms. The set of algorithm operations to be implemented on the GPU must be split into computation threads; the threads should be grouped into computation blocks that are performed atomically on stream processors. Threads of a single block are executed on a stream processor in parts-pools called warp; warp threads are executed simultaneously. The efficiency of the parallel algorithm depends on the way the data is stored in the GPU memory. If all warp threads request the same datum when executing the current operator, then it is desirable to place it in a shared or constant GPU memory; in this case, its distribution across the cores of the multiprocessor is actually realized by means of broadcast. If warp threads request data located close to the memory, then in this case there is a spatial locality of data, which makes it advisable to place this data in the GPU's memory. The implementation of broadcast or spatial locality by placing data in a memory of the appropriate type allows one to significantly reduce traffic when exchanging data between the memory levels of the GPU. This paper formulates and proves the necessary and sufficient conditions under which it is possible to perform a broadcast or there is a spatial locality of data. The conditions are formulated in terms of functions that determine the use of array elements at occurrences in the algorithm operators and functions that define the information dependencies of the algorithm. The results of the work can be used to optimize parallel algorithms when they are implemented on the GPU.

Keywords: parallel computations, GPU, broadcast, spatial locality

For citation. Likhoded N. A. Conditions for the existence of broadcast and spatial locality in computation threads. *Vestsi Natsyianal'nai akademii navuk Belarusi. Seryia fizika-matematychnykh navuk = Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematics series*, 2022, vol. 58, no. 3, pp. 292–299 (in Russian). https://doi.org/10.29235/1561-2430-2022-58-3-292-299

Введение. В качестве целевого компьютера для реализации алгоритмов будем рассматривать широко используемые в настоящее время графические процессоры (GPU). Множество операций алгоритма для реализации на графическом процессоре должно быть разбито на потоки вычислений. Потоки должны быть сгруппированы в блоки вычислений, выполняющиеся атомарно на потоковых процессорах, которые также называют мультипроцессорами. Потоки одного блока выполняются на мультипроцессоре частями-пулами, называемыми варпами (warps); потоки варпа выполняются одновременно. Потоки разбиваются на варпы автоматически согласно своему номеру: 32 подряд идущих потока образуют варп (2 полуварпа по 16 потоков).

Если все потоки варпа (или, в зависимости от архитектуры GPU, полуварпа) запрашивают на некотором вхождении данных в выполняемый оператор один и тот же элемент массива, то он передается только один раз. Соответственно трафик обмена данными сокращается, поскольку в этом случае происходит так называемый бродкаст данных (broadcast). Другой случай, который может привести к сокращению трафика, – пространственная локальность, т. е. использование потоками одного варпа близко расположенных в памяти данных. Эффективность выполнения бродкаста или реализации пространственной локальности зависит от того, в какой из видов памяти графического процессора размещены данные.

Графические процессоры имеют несколько видов памяти: регистровая (память с самым быстрым доступом), разделяемая (управляемый кэш, общий для ядер мультипроцессора), глобальная (оперативная память GPU, обладает невысокой скоростью доступа), константная (кэшируется), текстурная (кэшируется). Если возникает ситуация бродкаста данных, то для повышения эффективности параллельного приложения их целесообразнее размещать в разделяемой памяти или (если элементы массива только считываются) в константной памяти. При наличии пространственной локальности данных будет более рациональным разместить массив с данными (если элементы массива только считываются) в текстурной памяти.

Бродкаст с точки зрения оптимизации трафика конкурирует с приватизацией данных. Под приватизацией понимается использование элемента массива в вычислениях только одного потока блока вычислений. В случае приватизации массив, содержащий такие элементы, эффективнее размещать в регистровой памяти.

Существование бродкаста, приватизации данных потоками, пространственной локальности очень полезны для эффективного использования памяти с быстрым доступом. Отметим, что какое-либо из отмеченных событий не обязательно происходит; бродкаст не происходит одновременно с приватизацией или пространственной локальностью. Следует совместно рассматривать условия для существования этих событий и выбирать из альтернативных вариантов оптимальные (по выбранным критериям) организацию потоков вычислений и размещение данных в разных видах памяти.

Целью настоящей работы является формулировка и доказательство необходимых и достаточных условий, при которых реализуется бродкаст или имеет место пространственная локальность. Условия формулируются в терминах функций, определяющих использование элементов массивов на вхождениях их элементов в операции алгоритма, и функций, задающих информационные зависимости алгоритма.

Математический аппарат и методика доказательства таких утверждений разработаны в [1, 2] для случая, когда целевым компьютером является суперкомпьютер с распределенной памятью. Для случая графических процессоров эта методика развита в [3–5]. В [3] сформулированы и доказаны утверждения, позволяющие ранжировать параметры размера блоков на основе асимптотических оценок объема коммуникационных операций. Условия эффективного использования параллельными потоками кэшей и разделяемой памяти при наличии пространственной локальности указаны в [4]. В [5] сформулированы и доказаны необходимые условия и достаточные условия приватизации элементов массива потоками вычислений при реализации алгоритма на графическом процессоре. Отмеченные исследования направлены на эффективное использование памяти с быстрым доступом при реализации алгоритмов на GPU.

Функции, определяющие использование элементов массивов на их вхождениях в операции алгоритма. Функции, задающие информационные зависимости алгоритма. Пусть алго-

ритм 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 $V^{\theta}, 1 \le \theta \le \Theta$, – область изменения параметров циклов, окружающих θ-й набор операторов, n^{θ} – размерность этой области, т. е. число циклов, окружающих θ-й набор операторов. Обозначим ν_l – размерности массивов a_l.

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

$$\overline{F}_{a_l,S_\beta,q}(J) = F_{a_l,S_\beta,q}J + f^{a_l,S_\beta,q}, \quad J(j_1,...,j_{n^\theta}) \in V^{\theta}, \quad F_{a_l,S_\beta,q} \in \mathbf{Z}^{\nu_l \times n^\theta}, \quad f^{a_l,S_\beta,q} \in \mathbf{Z}^{\nu_l}.$$

Векторы, в зависимости от ситуации, мы будем считать вектор-столбцами или вектор-строками.

Выполнение оператора S_β при конкретных значениях β и вектора параметров цикла J будем называть операцией и обозначать $S_\beta(J)$. Пара вхождений (a_l,S_α,q) и (a_l,S_β,q) порождает истинную зависимость $S_\alpha(I) \to S_\beta(I)$, если $S_\alpha(I)$ выполняется раньше $S_\beta(J)$; $S_\alpha(I)$ переопределяет (изменяет) элемент массива a_l, а $S_\beta(J)$ использует на вхождении (a_l,S_β,q) в качестве аргумента тот же элемент массива; между операциями $S_\alpha(I)$ и $S_\beta(J)$ этот элемент не переопределяется.

Зависимости между операциями можно задать функциями вида

$$\overline{\Phi}_{\alpha,\beta}(J) = \Phi_{\alpha,\beta}J - \varphi^{\alpha,\beta}, \quad J \in V_{\alpha,\beta}, \quad \Phi_{\alpha,\beta} \in \mathbb{Z}^{n^\alpha \times n^\beta}, \quad \varphi^{\alpha,\beta} \in \mathbb{Z}^{n^\alpha}.$$

Функция зависимостей $\overline{\Phi}_{\alpha,\beta}(J)$ позволяет для операции $S_\beta(J)$ найти операцию $S_\alpha(I)$, от которой $S_\beta(J)$ зависит. Функции зависимостей являются удобным математическим аппаратом для описания информационных связей между операциями алгоритма [6, 7].

Пример. 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) $V^1 = \{(k,i,j) \in Z^3 \mid 1 \le k \le n,\ 1 \le i \le n,\ 1 \le j \le n\}$ для оператора S_1 имеет размерность 3. Для матриц $F_{a,S_1,q}$ на вхождениях (a,S_1,q) имеем

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

Приведем вид функции истинных зависимостей, порождаемой вхождением $(a,S_1,2)$:

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

Это так называемая однородная зависимость – матрица $\Phi_{1,1}$ является единичной.

Разбиение множества операций алгоритма на блоки и потоки вычислений. Обозначим размеры блоков вычислений натуральными числами $r_1^{\theta},...,r_{n^\theta}^{\theta}$; 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 блока – 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 будем обозначать

$V^{\theta}_{J^{gl}}$, где $J^{gl} = \left(j_1^{gl},...,j_{n^{\theta}}^{gl}\right)$, $0 \le j_{\zeta}^{gl} \le Q_{\zeta}^{\theta} - 1$, $1 \le \zeta \le n^{\theta}$. Для разбиения множеств операций алгоритма на блоки можно использовать тайлинг – преобразование алгоритма для получения макроопераций [8].

Далее для простоты записи будем использовать обозначения без индекса θ.

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

Определим в каждом блоке вычислений $V_{J^{gl}}$, $J^{gl} = \left(j_1^{gl},...,j_n^{gl}\right)$, потоки вычислений:

$$\mathrm{Thr}\left(j_{\zeta_1}^{gl2},...j_{\zeta_m}^{gl2}\right), \quad 0 \le j_{\zeta}^{gl2} \le r_{\zeta} - 1, \quad \zeta \in Z_s,$$

где координаты потоков связаны с координатами выполняемых ими итераций алгоритма $J(j_1,...,j_n) \in V$ следующим образом:

$$j_{\zeta} = \min j_{\zeta} + j_{\zeta}^{gl} r_{\zeta} + j_{\zeta}^{gl2}, \quad \zeta \in Z_s, \tag{2}$$

$$\min j_{\zeta} + j_{\zeta}^{gl} r_{\zeta} \le j_{\zeta} \le \min j_{\zeta} - 1 + \left(j_{\zeta}^{gl} + 1\right) r_{\zeta}, \quad \zeta \in Z_t.$$

Таким образом, m координат области изменения параметров циклов определяют m-мерное пространство потоков, а остальные $n - m$ координат соответствуют итерациям, выполняемым фиксированными потоками в лексикографическом порядке. Возникает сетка потоков и сетка блоков вычислений; нумерация потоков и блоков по каждой координате сетки начинается с нуля.

П р и м е р (продолжение). Известны блочные алгоритмы Флойда – Уоршелла с 3D-блоками размера $r \times r \times r$ (одинаковые размеры блока имеют существенное значение) [9, 10]. 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
```

При фиксированном k^{gl}, $i^{gl} \ne k^{gl}$, $j^{gl} \ne k^{gl}$ операции разных блоков не зависят друг от друга.

Пусть m = 2, Z_s = {2,3}, ζ_1 = 2, ζ_2 = 3, Z_t = {1}, ζ_3 = 1. Потоки $\mathrm{Thr}\left(i^{gl2}, j^{gl2}\right)$ блока $V_{k^{gl},i^{gl},j^{gl}}$ имеют следующее представление:

```
i = 1 + i^gl r + i^gl2
j = 1 + j^gl r + j^gl2
if  i ≤ n, j ≤ n
  do k = 1 + k^gl r,  min((k^gl + 1)r, n)
    a(i, j) = min(a(i, j), a(i, k) + a(k, j))
  enddo
```

Условия для осуществления бродкаста. Бродкаст происходит в случае, когда потоки одного варпа (или полуварпа для некоторых архитектур GPU) запрашивают одновременно один и тот же элемент. Бродкаст может происходить только на вхождении данных в правую часть оператора.

Потоки одного блока должны быть помечены как узлы одномерной, двумерной или трехмерной сетки. Если потоки вычислений помечены как узлы одномерной сетки, то варпы формируются из потоков $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}\right)$ при идущих подряд $j_{\zeta_1}^{gl2}$; если потоки помечены как узлы двумерной

сетки, то будем считать, что варпы формируются из потоков $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, j_{\zeta_2}^{gl2}\right)$ при фиксированном $j_{\zeta_2}^{gl2}$ и идущих подряд $j_{\zeta_1}^{gl2}$, а если потоки вычислений помечены как узлы трехмерной сетки, то варпы формируются из потоков $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, j_{\zeta_2}^{gl2}, j_{\zeta_3}^{gl2}\right)$ при фиксированных $j_{\zeta_2}^{gl2}, j_{\zeta_3}^{gl2}$ и идущих подряд $j_{\zeta_1}^{gl2}$.

Обозначим: T_s – матрица, строки которой составлены из векторов $e_\zeta^{(n)}$, $\zeta \in Z_s$, где $e_\zeta^{(n)}$ – вектор-строка размера n, у которой координата с номером ζ равна 1, а остальные координаты нулевые; T_t – матрица, строки которой составлены из векторов $e_\zeta^{(n)}$, $\zeta \in Z_t$.

Напомним, через ζ_1 обозначен первый элемент упорядоченного множества Z_s; если $J = (j_1, ..., j_n)$, то для любого j_ζ либо $\zeta \in Z_s$, либо $\zeta \in Z_t$.

Т е о р е м а 1. *Для того чтобы данные, связанные с вхождением* (a_l, S_β, q)*, порождали бродкаст, необходимо равенство нулю всех элементов* ζ_1*-х столбцов матриц* $F_{a_l, S_\beta, q}$ *и* T_t*. Если вхождение* (a_l, S_β, q) *не порождает в блоке вычислений истинных зависимостей, то указанные необходимые условия являются также и достаточными.*

Д о к а з а т е л ь с т в о. Пусть выполняются условия теоремы, вхождение (a_l, S_β, q) может порождать в блоке вычислений истинные зависимости. Бродкаст происходит только в том случае, когда потоки одного варпа (или полуварпа) запрашивают одновременно один и тот же элемент массива. Формализуем условия, при которых может произойти бродкаст.

Пусть поток $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, ..., j_{\zeta_m}^{gl2}\right)$ некоторого варпа на итерации T (т. е. при фиксированном значении T вектора, составленного из координат j_ζ, $\zeta \in Z_t$) запрашивает элемент массива $a_l\left(\overline{F}_{a_l, S_\beta, q}(J)\right)$, $J = (j_1, ..., j_n)$, $j_\zeta = \min j_\zeta + j_\xi^{gl} r_\zeta + j_\xi^{gl2}$, $\zeta \in Z_s$ (см. (2)), и пусть поток $\mathrm{Thr}\left(h_{\zeta_1}^{gl2}, ..., h_{\zeta_m}^{gl2}\right)$ этого же варпа и на той же итерации T запрашивает тот же элемент массива $a_l\left(\overline{F}_{a_l, S_\beta, q}(H)\right)$, $H = (h_1, ..., h_n)$. Варпы формируются из потоков $\mathrm{Thr}\left(j_{\zeta_1}^{gl2}, ..., j_{\zeta_m}^{gl2}\right)$ при идущих подряд $j_{\zeta_1}^{gl2}$ и фиксированных последующих (если $m = 2$ или $m = 3$) координатах. Поэтому J и H, с учетом равенства (2), отличаются только ζ_1-й координатой j_{ζ_1}.

Выполнение условия $a_l\left(\overline{F}_{a_l, S_\beta, q}(J)\right) = a_l\left(\overline{F}_{a_l, S_\beta, q}(H)\right)$ использования одного и того же элемента массива означает

$$F_{a_l, S_\beta, q} J + f^{a_l, S_\beta, q} = F_{a_l, S_\beta, q} H + f^{a_l, S_\beta, q}.$$

Так как в двух потоках рассматривается одна и та же итерация T, то

$$T_t J = T_t H.$$

Таким образом, для получения бродкаста необходимо, чтобы $F_{a_l, S_\beta, q} J = F_{a_l, S_\beta, q} H$ и $T_t J = T_t H$. Имеем:

$$\begin{pmatrix} F_{a_l, S_\beta, q} \\ T_t \end{pmatrix} J = \begin{pmatrix} F_{a_l, S_\beta, q} \\ T_t \end{pmatrix} H, \quad \begin{pmatrix} F_{a_l, S_\beta, q} \\ T_t \end{pmatrix} (J - H) = 0, \quad \begin{pmatrix} F_{a_l, S_\beta, q} \\ T_t \end{pmatrix} (j_{\zeta_1} - h_{\zeta_1}) e_{\zeta_1}^{(n)} = 0.$$

Последнее равенство означает, что ζ_1-е столбцы матриц $F_{a_l, S_\beta, q}$ и T_t являются нулевыми.

Если вхождение (a_l, S_β, q) не порождает в блоке вычислений истинных зависимостей, то между итерациями J и H элемент массива $a_l\left(\overline{F}_{a_l, S_{\hat{a}}, q}(J)\right)$ переопределяться не может, и для получения бродкаста указанное условие является не только необходимым, но и достаточным. Теорема доказана.

Т е о р е м а 2. *Если вхождение* (a_l, S_β, q) *порождает в блоке вычислений истинные зависимости,* $\overline{\Phi}_{\alpha,\beta}$ *– функция зависимостей, то использование данных, связанных с вхождением* (a_l, S_β, q)*, порождает бродкаст тогда и только тогда, когда* ζ_1*-е столбцы матриц* $\Phi_{\alpha,\beta}$, $F_{a_l, S_\beta, q}$ *и* T_t *являются нулевыми.*

Д о к а з а т е л ь с т в о. Так как ζ_1-е столбцы матриц $F_{a_l, S_\beta, q}$ и T_t являются нулевыми, то по теореме 1 необходимые условия существования бродкаста выполняются.

Для доказательства достаточности покажем, что на вхождении (a_l,S_β,q) на итерациях J и H таких, что у них отличаются только ζ_1-е координаты, содержимое ячеек памяти $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)\right)$ и $a_l\left(\overline{F}_{a_l,S_\beta,q}(H)\right)$ одно и то же. Содержимое ячеек памяти $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)\right)$ и $a_l\left(\overline{F}_{a_l,S_\beta,q}(H)\right)$ определилось на итерациях $\overline{\Phi}_{\alpha,\beta}(J)$ и $\Phi_{\alpha,\beta}(H)$ соответственно. Это одна и та же итерация, так как $\overline{\Phi}_{\alpha,\beta}(J)=\overline{\Phi}_{\alpha,\beta}(H)$ означает

$$\Phi_{\alpha,\beta}J=\Phi_{\alpha,\beta}H,\quad \Phi_{\alpha,\beta}(J-H)=0,\quad \Phi_{\alpha,\beta}(j_{\zeta_1}-h_{\zeta_1})e^{(n)}_{\zeta_1}=0;$$

последнее равенство верно, так как ζ_1-й столбец матрицы $\Phi_{\alpha,\beta}$ является нулевым.

Следствие. *Если вхождение (a_l,S_β,q) в правую часть некоторого оператора S_β порождает однородную истинную зависимость $S_\alpha(I)\to S_\beta(J)$ и операторы S_α, S_β принадлежат одному блоку вычислений, то использование данных, связанных с вхождением (a_l,S_β,q), бродкаст не порождает.*

Действительно, в случае однородной зависимости $\Phi_{\alpha,\beta}$ есть единичная матрица порядка n, поэтому никакой столбец матрицы не может быть нулевым (заведомо $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)\right)$ и $a_l\left(\overline{F}_{a_l,S_\beta,q}(H)\right)$ определяются на разных итерациях, поэтому их значения, вообще говоря, разное).

Пример (продолжение). Рассмотрим блоки вычислений $V_{k^{gl},i^{gl},j^{gl}}$, $i^{gl}\neq k^{gl}$, $j^{gl}\neq k^{gl}$ (тогда вхождения $(a,S_1,3)$ и $(a,S_1,4)$ не порождают зависимостей) и выделенные в каждом из них потоки $\mathrm{Thr}\left(i^{gl2},j^{gl2}\right)$. Так как $Z_t=\{1\}$, $\zeta_3=1$, то $T_t=(1\ 0\ 0)$. Рассмотрим вхождения (a,S_1,q) в правую часть оператора и исследуем, происходит ли бродкаст.

Напомним, $m=2$, $Z_s=\{2,3\}$, $\zeta_1=2$, $\zeta_2=3$. Необходимое условие наличия нулевого столбца с номером $\zeta_1=2$ в матрице T_t выполняется.

Обращение к данным $a(i,j)$ на вхождении $(a,S_1,2)$ к бродкасту не приводит, так как вхождения $(a,S_1,1)$ и $(a,S_1,2)$ порождают однородную истинную зависимость.

Для вхождения $(a,S_1,3)$, на котором считываются данные $a(i,k)$, второй столбец матрицы $F_{a,S_1,3}$ не является нулевым (см. (1)), необходимое условие наличия бродкаста не выполняется (теорема 1).

Для вхождения $(a,S_1,4)$, на котором считываются данные $a(k,j)$, второй столбец матрицы $F_{a,S_1,4}$ является нулевым. Необходимые и достаточные условия наличия бродкаста выполняются (теорема 1).

Условия пространственной локальности данных. В последовательных вычислениях имеет место пространственная локальность на вхождении (a_l,S_β,q), если при изменении на единицу параметра самого внутреннего цикла используются элементы одной строки массива, т. е. элементы, отличающиеся только последним индексом массива. Предполагается, что хранение элементов массива осуществляется по строкам (как при использовании языка программирования С). Пространственная локальность в потоках происходит в том случае, когда подряд идущие потоки одного варпа или полуварпа (в зависимости от технологии) запрашивают одновременно последовательно расположенные в памяти данные.

Теорема 3. *Использование данных, связанных с вхождением (a_l,S_β,q), порождает пространственную локальность в потоках тогда и только тогда, когда ζ_1-й столбец матрицы T_t является нулевым, а в ζ_1-м столбце матрицы $F_{a_l,S_\beta,q}$ ненулевым, равным единице, является только последний элемент.*

Доказательство. Формализуем условия, при которых подряд идущие потоки запрашивают одновременно последовательно расположенные в памяти данные.

Пусть поток $\mathrm{Thr}\left(j^{gl2}_{\zeta_1},\ldots,j^{gl2}_{\zeta_m}\right)$ некоторого варпа на итерации T (т. е. при фиксированном значении T вектора, составленного из координат j_ζ, $\zeta\in Z_t$) запрашивает элемент массива $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)\right)$, $J=(j_1,\ldots,j_n)$, $j_\zeta=m_\zeta+j^{gl}_\zeta r_\zeta+j^{gl2}_\zeta$, $\zeta\in Z_s$, а следующий поток этого же варпа на той же итерации T запрашивает следующий в памяти элемент массива $a_l\left(\overline{F}_{a_l,S_\beta,q}(H)\right)$, $H=(h_1,\ldots,h_n)$.

Поток, следующий за потоком $\mathrm{Thr}\left(j^{gl2}_{\zeta_1},\ldots,j^{gl2}_{\zeta_m}\right)$, есть поток $\mathrm{Thr}\left(j^{gl2}_{\zeta_1}+1,\ldots,j^{gl2}_{\zeta_m}\right)$. Элемент массива, следующий в памяти за $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)\right)$, есть $a_l\left(\overline{F}_{a_l,S_\beta,q}(J)+e^{(\nu_l)}_{\nu_l}\right)$, где, напомним, ν_l – раз-

мерность массива a_l. Следовательно, $a_l\left(\overline{F}_{a_l,S_\beta,q}(H)\right)=a_l\left(\overline{F}_{a_l,S_\beta,q}(J)+e_{\nu_l}^{(\nu_l)}\right)$, причем J и H, с учетом равенства (2), отличаются только ζ_1-й координатой j_{ζ_1}, которые, в свою очередь, отличаются на 1. Имеем:

$$F_{a_l,S_\beta,q}H+f^{a_l,S_\beta,q}=F_{a_l,S_\beta,q}J+f^{a_l,S_\beta,q}+e_{\nu_l}^{(\nu_l)},$$

$$F_{a_l,S_\beta,q}(H-J)=e_{\nu_l}^{(\nu_l)},\quad F_{a_l,S_\beta,q}\left(h_{\zeta_1}-j_{\zeta_1}\right)e_{\zeta_1}^{(n)}=e_{\nu_l}^{(\nu_l)},\quad F_{a_l,S_\beta,q}e_{\zeta_1}^{(n)}=e_{\nu_l}^{(\nu_l)}.$$

Кроме того, как и при бродкасте, выполняется условие $T_tJ=T_tH$, т. е. $T_te_{\zeta_1}^{(n)}=0$ (потоки запрашивают данные на одной и той же итерации).

Таким образом, для наличия пространственной локальности в потоках необходимо и достаточно, чтобы $F_{a_l,S_\beta,q}e_{\zeta_1}^{(n)}=e_{\nu_l}^{(\nu_l)}$ и $T_te_{\zeta_1}^{(n)}=0$. Эти равенства означают, что элементы ζ_1-х столбцов матриц $F_{a_l,S_\beta,q}$ и T_t являются нулевыми, за исключением равного единице элемента с номером ν_l (т. е. последнего элемента) этого столбца матрицы $F_{a_l,S_\beta,q}$. Теорема 3 доказана.

Пример (окончание). Рассмотрим вхождения (a,S_1,q) в правую часть оператора и исследуем, происходит ли пространственная локальность потоков. Необходимое условие наличия нулевого столбца с номером $\zeta_1=2$ в матрице T_t выполняется. Для вхождения $(a,S_1,2)$, на котором считываются данные $a(i,j)$, второй столбец матрицы $F_{a,S_1,2}$ (см. (1)) не имеет требуемый теоремой вид $\begin{pmatrix}0\\1\end{pmatrix}$, пространственная локальность потоков не происходит. Заметим, что на этом вхождении происходит приватизация элементов массива потоками [5]. Для вхождения $(a,S_1,3)$, на котором считываются данные $a(i,k)$, второй столбец матрицы $F_{a,S_1,3}$ не имеет требуемый теоремой вид, пространственная локальность потоков не происходит. Для вхождения $(a,S_1,4)$, на котором считываются данные $a(k,j)$, второй столбец матрицы $F_{a,S_1,4}$ является нулевым, пространственная локальность потоков не происходит. Результаты приведенных исследований существования бродкаста или пространственной локальности можно применять для эффективного использования различных видов памяти графического процессора при построении параллельных алгоритмов Флойда – Уоршелла.

Заключение. Таким образом, в работе формализованы условия, при которых происходит бродкаст (все потоки варпа запрашивают на некотором вхождении массива в выполняемый оператор один и тот же элемент) и условия, при которых возникает пространственная локальность (потоки одного варпа используют на некотором вхождении близко расположенные в памяти данные). Существование бродкаста или пространственной локальности полезны для сокращения трафика обмена данными между уровнями памяти GPU. Полученные результаты могут быть использованы для оптимизации параллельных алгоритмов при их реализации на графических процессорах.

Благодарности. Работа выполнена в рамках Государственной программы научных исследований Республики Беларусь «Конвергенция-2025», подпрограмма «Математические модели и методы».

Acknowledgment. The prepared report was sponsored by the government program of scientific research of the Republic of Belarus “Convergence-2025”, subprogram “Mathematical models and methods”.

Список использованных источников

1. Лиходед, Н. А. Характеристика локальности параллельных реализаций многомерных циклов / Н. А. Лиходед // Докл. Нац. акад. наук Беларуси. – 2010. – Т. 54, № 1. – С. 26–32.

2. Адуцкевич, Е. В. К распараллеливанию последовательных программ: распределение массивов между процессорами и структуризация коммуникаций / Е. В. Адуцкевич, Н. А. Лиходед, А. О. Сикорский // Кибернетика и систем. анализ. – 2012. – № 1. – С. 144–163.

3. Лиходед, Н. А. Метод ранжирования параметров размера блоков вычислений параллельного алгоритма / Н. А. Лиходед, М. А. Полещук // Докл. Нац. акад. наук Беларуси. – 2015. – Т. 59, № 4. – С. 25–33.

4. Лиходед, Н. А. Оценка локальности параллельных алгоритмов, реализуемых на графических процессорах / Н. А. Лиходед, М. А. Полещук // Вестн. Юж.-Урал. гос. ун-та. Сер.: «Вычисл. математика и информатика». – 2016. – Т. 5. № 3. – С. 96–111. https://doi.org/10.14529/cmse160307

5. Лиходед, Н. А. Условия приватизации элементов массива потоками вычислений / Н. А. Лиходед, М. А. Полещук // Журн. Белорус. гос. ун-та. Математика. Информатика. – 2018. – № 3. – С. 59–67.

6. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.

7. Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model / U. Bondhugula [et al.] // Lecture Notes in Computer Science. – Springer, 2008. – P. 132–146. – (LNTCS, vol. 4959). https://doi.org/10.1007/978-3-540-78791-4_9

8. Xue, J. Loop Tiling for Parallelism / Xue J. – Springer Science & Business Media, 2000. – 256 p. – (LNTCS, vol. 575). https://doi.org/10.1007/978-1-4615-4337-4

9. Venkataraman, G. A blocked all-pairs shortest-paths algorithm / G. Venkataraman, S. Sahni, S. Mukhopadhyaya // ACM J. Exp. Algorithm. – 2003. – Vol. 8. – P. 2.2-es. https://doi.org/10.1145/996546.996553

10. Lund, B. D. A multi-stage cuda kernel for floyd-warshall [Electronic Resource] / B. D. Lund, J. W. Smith // Arxiv [Preprint]. – 2010. – Mode of access: https://arxiv.org/abs/1001.4108. https://doi.org/10.48550/arXiv.1001.4108

References

1. Likhoded N. A. Characterization of locality of the parallel implementations of imperfectly nested loops. *Doklady Natsional'noi akademii nauk Belarus = Proceedings of the National Academy of Sciences of Belarus*, 2010, vol. 54, no. 1, pp. 26–32 (in Russian).

2. Adutskevich N. A. Likhoded N. A., Sikorsky A. O. Parallelization of sequential programs: distribution of arrays among processors and structurization of communications. *Cybernetics and System Analysis*, 2012, vol. 48, no. 1, pp. 122–137. https://doi.org/10.1007/s10559-012-9382-2

3. Likhoded N. A., Paliashchuk M. A. Method of ranking tiles size parameters of parallel algorithm. *Doklady Natsional'noi akademii nauk Belarus = Proceedings of the National Academy of Sciences of Belarus*, 2015, vol. 59, no. 4, pp. 25–33 (in Russian).

4. Likhoded N. A., Paliashchuk M. A. 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, vol. 5, no. 3, pp. 96–111 (in Russian). https://doi.org/10.14529/cmse160307

5. Likhoded N. A., Paliashchuk M. A. Conditions for privatizing the elements of arrays by computing threads. *Zhurnal Belorusskogo gosudarstvennogo universiteta. Matematika. Informatika = Journal of the Belarusian State University. Mathematics and Informatics,* 2018, no. 3, pp. 59–67 (in Russian).

6. Voevodin V. V. *Parallel Computing.* Saint Petersburg, BKhV-Petersburg Publ., 2002. 608 p. (in Russian).

7. Bondhugula U., Baskaran M., Krishnamoorthy S., Ramanujam J., Rountev A., Sadayappan P. Automatic transformations for communication-minimized parallelization and locality optimization in the polyhedral model. *Lecture Notes in Computer Science. (LNTCS, vol. 4959)*, 2008, pp. 132–146. https://doi.org/10.1007/978-3-540-78791-4_9

8. Xue J. *Loop Tiling for Parallelism. (LNTCS, vol. 575).* Springer Science & Business Media, 2000. 256 p. https://doi.org/10.1007/978-1-4615-4337-4

9. Venkataraman G., Sahni S., Mukhopadhyaya S. A blocked all-pairs shortest-paths algorithm. *ACM Journal of Experimental Algorithmics*, 2003, vol. 8, pp. 2.2-es. https://doi.org/10.1145/996546.996553

10. Lund B. D., Smith J. W. A multi-stage cuda kernel for floyd-warshall. *Arxiv* [Preprint], 2010. Available at: https://arxiv.org/abs/1001.4108. https://doi.org/10.48550/arXiv.1001.4108

Информация об авторе

Лиходед Николай Александрович – доктор физико-математических наук, профессор, Белорусский государственный университет (пр. Независимости, 4, 220030, Минск, Республика Беларусь). E-mail: likhoded@bsu.by

Information about the author

Nikolai A. Likhoded – Dr. Sc. (Physics and Mathematics), Professor, Belarusian State University (4, Nezavisimosti Ave., 220030, Minsk, Republic of Belarus). E-mail: likhoded@bsu.by