ОЦЕНКА ОБЪЕМА КОММУНИКАЦИОННЫХ ОПЕРАЦИЙ ПАРАЛЛЕЛЬНОГО ЗЕРНИСТОГО АЛГОРИТМА

М. А. Полещук, Н. А. Лиходед

Белорусский государственный университет
Минск, Беларусь
E-mail: maxipole@gmail.com, likhoded@bsu.by

Множество операций параллельного алгоритма для реализации на графическом процессоре должно быть разбито на блоки вычислений. Один блок выполняется атомарно на одном мультипроцессоре. Мультипроцессоры обмениваются информацией посредством глобальной памяти, причем для хорошей производительности критично минимизировать число обращений к ней. В этой работе сформулированы условия, использование которых позволяет, при рассмотрении возможных вариантов, получать блоки вычислений с меньшим числом обращений к глобальной памяти, т. е. с меньшим объемом коммуникационных операций.

Ключевые слова: распараллеливание алгоритмов, тайлинг, уменьшение обменов данными.

Введение

Время решения задачи на современном компьютере во многом определяется степенью использования памяти с быстрым доступом. В качестве целевого компьютера будем рассматривать графические процессоры (GPU). При многопроцессорной обработке на GPU быстрым является обращение к разделяемой памяти мультипроцессора, оперирующего некоторыми данными алгоритма, но не обращение к глобальной памяти GPU. Чем меньше число обращений к глобальной памяти, тем быстрее выполняется алгоритм.

Для реализации алгоритма на графическом процессоре множество операций алгоритма должно быть разбито на блоки, а блоки – на потоки (нити) вычислений. Множество операций алгоритма разбить на блоки можно путем тайлинга (тайлинг первого уровня); разбить блоки вычислений на потоки (нити) вычислений можно путем повторного применения тайлинга (тайлинг второго уровня) [1]. Тайлинг (tiling) – получение макроопераций-тайлов [2, 3]; операции одного тайла выполняются атомарно, как одна единица вычислений, а обмен данными происходит массивами. В этой работе предлагается подход, позволяющий получать тайлы вычислений с меньшим числом обращений к глобальной памяти. Используется анализ информационных зависимостей, порождающих коммуникационные операции.

Предварительные сведения

Будем считать, что алгоритм задан последовательной программой линейного класса [4]. Основную вычислительную часть такой программы составляют циклы произвольной структуры вложенности; границы изменения параметров циклов задаются неоднородными формами, линейными по совокупности параметров циклов и внешних переменных. Пусть в гнезде циклов имеется K выполняемых операторов S_β

и используется L массивов a_l. Область изменения параметров циклов для оператора S_β и размерность этой области обозначим соответственно V_β и n_β.

Пусть в гнезде циклов имеется Θ наборов выполняемых операторов. Под набором операторов будем понимать один или несколько операторов, окруженных одним и тем же множеством циклов. Операторы и наборы операторов линейно упорядочены расположением их в записи алгоритма. Обозначим: V^ϑ, $1 \le \vartheta \le \Theta$, – области изменения параметров циклов, окружающих наборы операторов, n^ϑ – размерность области V^ϑ (число циклов, окружающих ϑ-й набор операторов). Заметим, что если оператор S_β принадлежит ϑ-му набору операторов, то область V_β может быть уже области V^ϑ.

Выполнение оператора S_β при конкретных значениях β и вектора параметров цикла J будем называть операцией и обозначать $S_\beta(J)$. Зависимости (информационные связи) между операциями можно задать функциями вида

$$\overline{\Phi}_{\alpha,\beta}(J) = \Phi_{\alpha,\beta} J + \Psi_{\alpha,\beta} N - \varphi^{\alpha,\beta},$$

$$J \in V_{\alpha,\beta},\ N \in \mathbf{Z}^e,\ \Phi_{\alpha,\beta} \in \mathbf{Z}^{n_\alpha \times n_\beta},\ \Psi_{\alpha,\beta} \in \mathbf{Z}^{n_\alpha \times e},\ \varphi^{\alpha,\beta} \in \mathbf{Z}^{n_\alpha},$$

где $N \in \mathbf{Z}^e$ – вектор внешних переменных алгоритма, e – число внешних переменных. 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 (другие названия этих функций: покрывающие функции графа алгоритма [4], h-преобразования [5, 6]).

Как уже отмечалось, тайлинг – это преобразование алгоритма для получения макроопераций, называемых также зерном вычислений или тайлами. При тайлинге каждый цикл разбивается на два цикла: глобальный, параметр которого определяет на данном уровне вложенности порядок вычисления тайлов, и локальный, в котором параметр исходного цикла изменяется в границах одного тайла. Допускается вырожденное разбиение цикла, при котором все итерации относятся к глобальному циклу или все итерации относятся к локальному циклу.

Следующие величины и множества используются для формализации тайлинга:

$m_\zeta^\vartheta = \min\limits_{J(j_1, j_2, \ldots, j_{n^\vartheta}) \in V^\vartheta} j_\zeta$, $M_\zeta^\vartheta = \max\limits_{J(j_1, j_2, \ldots, j_{n^\vartheta}) \in V^\vartheta} j_\zeta$, $1 \le \zeta \le n^\vartheta$, – предельные значения изменения параметров циклов;

$r_1^\vartheta, \ldots, r_{n^\vartheta}^\vartheta$ – заданные натуральные числа, определяющие размеры тайла; r_ζ^ϑ обозначает число значений параметра j_ζ, приходящихся на один тайл ϑ-го набора операторов; r_ζ^ϑ может принимать фиксированное значение в пределах от 1 до $r_\zeta^{\vartheta,\max}$ включительно, где $r_\zeta^{\vartheta,max} = M_\zeta^\vartheta - m_\zeta^\vartheta + 1$; если $r_\zeta^\vartheta = 1$, то цикл с параметром $j_\zeta = 1$ является глобальным неразбиваемым; если $r_\zeta^\vartheta = r_\zeta^{\vartheta,\max}$, то цикл с параметром j_ζ является локальным неразбиваемым; если два набора операторов имеют общий цикл с параметром j_ζ, то $r_\zeta^{\vartheta_1} = r_\zeta^{\vartheta_2}$;

$Q_\zeta^\vartheta = \left\lceil (M_\zeta^\vartheta - m_\zeta^\vartheta + 1) / r_\zeta^\vartheta \right\rceil$, $1 \le \zeta \le n^\vartheta$ – число частей, на которые при формировании тайлов разбивается область значений параметра j_ζ цикла, окружающего ϑ-й набор операторов;

$V^{\vartheta,gl} = \left\{ J^{gl}(j_1^{gl}, ..., j_{n^\vartheta}^{gl}) \middle| 1 \le j_\zeta^{gl} \le Q_\zeta^\vartheta,\ 1 \le \zeta \le n^\vartheta \right\}$ – области изменения параметров глобальных, т. е. уровня тайлов, циклов;

$$V_{J^{gl}}^\vartheta = \left\{ J(j_1, ..., j_{n^\vartheta}) \in V^\vartheta \middle| \ m_\zeta^\vartheta + (j_\zeta^{gl} - 1) r_\zeta^\vartheta \le j_\zeta \le m_\zeta^\vartheta - 1 + j_\zeta^{gl} r_\zeta^\vartheta,\ 1 \le \zeta \le n^\vartheta \right\},$$

$J^{gl} \in V^{\vartheta,gl}$, – области изменения параметров локальных (уровня операций тайлов) циклов при фиксированных значениях параметров глобальных циклов. Множество операций, выполняемых на итерациях множества $V_{J^{gl}}^\vartheta$, будем также обозначать $V_{J^{gl}}^\vartheta$. Множества $V_{J^{gl}}^\vartheta$ называются тайлами $V_{J^{gl}}^\vartheta$ или тайлами J^{gl}. Тайлы $V_{J^{gl}}^\vartheta$ будем называть тайлами ϑ-го типа.

Вхождением (l, β, q) будем называть q-е вхождение массива a_l в оператор S_β. Индексы элементов l-го массива, связанных с вхождением (l, β, q), выражаются функцией $\overline{F}_{l,\beta,q} : V_\beta \to W_l$ вида

$$\overline{F}_{l,\beta,q}(J) = F_{l,\beta,q} J + G_{l,\beta,q} N + f^{\,l,\beta,q},$$

$$J(j_1, ..., j_{n_\beta}) \in V_\beta,\ N \in \mathbf{Z}^e,\ F_{l,\beta,q} \in \mathbf{Z}^{\nu_l \times n_\beta},\ G_{l,\beta,q} \in \mathbf{Z}^{\nu_l \times e},\ f^{\,l,\beta,q} \in \mathbf{Z}^{\nu_l}.$$

Обозначим $\rho_{l,\beta,q} = \operatorname{rank} F_{l,\beta,q}$, $\rho_{l,\beta,q}^\xi = \operatorname{rank} \begin{pmatrix} F_{l,\beta,q} \\ e_\xi^{(n_\beta)} \end{pmatrix}$, где $e_\xi^{(n_\beta)}$ – вектор-строка размера n_β, у которой координата с номером ξ равна 1, а остальные координаты нулевые (если $\xi > n_\beta$, то $e_\xi^{(n_\beta)}$ – нулевой вектор). Пусть вхождение (l, β, q) в правую часть некоторого оператора порождает истинную зависимость, $\overline{\Phi}_{\alpha,\beta}$ – функция зависимостей. Обозначим $\rho_{l,\beta,q}^{\Phi,\xi} = \operatorname{rank} \begin{pmatrix} F_{l,\beta,q} \\ \Phi_{\alpha,\beta} \\ e_\xi^{(n_\beta)} \end{pmatrix}$, $\rho_{l,\beta,q}^\Phi = \operatorname{rank} \begin{pmatrix} F_{l,\beta,q} \\ \Phi_{\alpha,\beta} \end{pmatrix}$. Если вхождение (l, β, q) не порождает истинную зависимость, то по определению $\rho_{l,\beta,q}^{\Phi,\xi} = \rho_{l,\beta,q}^\xi$, $\rho_{l,\beta,q}^\Phi = \rho_{l,\beta,q}$.

Термин «фиксированное данное массива» означает конкретное, неизмененное содержимое соответствующей ячейки памяти.

Условия, характеризующие объем коммуникационных операций

Результаты, представленные в этом разделе, определяют приоритеты числа разбиения множеств параметров циклов для минимизации объема коммуникационных операций между тайлами первого уровня (т. е. между блоками вычислений). Более высокий приоритет означает возможность уменьшения r_ζ^ϑ с небольшими коммуникационными издержками. Уменьшать r_ζ^ϑ требуется, если блоки вычислений являются слишком большими для эффективной реализации на мультипроцессорах.

Теорема 1. Пусть определение элемента некоторого массива a_l происходит на вхождении $(l,\alpha,1)$ в левой части оператора S_α, а использование – на вхождении (l,β,q) в правой части оператора S_β, и выполняются условия

$$\left(\Phi_{\alpha,\beta}\right)_\xi = e_\xi^{(n_\beta)}, \quad \left(\Psi_{\alpha,\beta}\right)_\xi = 0 \tag{1}$$

Если

$$\varphi_\xi^{\alpha,\beta} = 0, \tag{2}$$

то определение и использование элемента массива данных происходит при одном и том же значении цикла с параметром j_ξ^{gl}.

Если

$$0 < \left|\varphi_\xi^{\alpha,\beta}\right| < r_\xi^\vartheta, \tag{3}$$

то определение и использование элемента массива данных происходит при одном и том же значении цикла с параметром j_ξ^{gl} на $r_\xi^\vartheta - \left|\varphi_\xi^{\alpha,\beta}\right|$ итерациях из каждых r_ξ^ϑ итераций j_ξ при любых фиксированных значениях других итераций; на $\left|\varphi_\xi^{\alpha,\beta}\right|$ из каждых r_ξ^ϑ итераций j_ξ для использования требуется коммуникационная операция.

Теорема 2. Если $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}^{\Phi,\xi}$, то на вхождении (l,β,q) фиксированное данное используется при одном и том же значении параметра цикла j_ξ^{gl}; если $\rho_{l,\beta,q}^{\Phi} < \rho_{l,\beta,q}^{\Phi,\xi}$, то данное используется при разных значениях параметра цикла j_ξ^{gl}. На вхождении (l,β,q) фиксированное данное используется при фиксированном значении цикла с параметром j_ξ^{gl} на итерациях подпространства итераций размерности $k_{l,\beta,q}^{\Phi,\xi}$, где $k_{l,\beta,q}^{\Phi,\xi} = n_\beta - \rho_{l,\beta,q}^{\Phi,\xi}$.

Обоснование теорем проводится по схеме, предложенной в работе [7].

Теоремы 1 и 2 позволяют определить приоритет разбиения координат для вхождения (l,β,q) в правую часть оператора. Приоритет z_0 является самым высоким, z_4 – самый низкий.

1. Если вхождение (l,β,q) порождает истинную зависимость, и выполняются условия (1), (2), назначим приоритет z_0.
2. Если вхождение (l,β,q) порождает истинную зависимость, и выполняются условия (1), (3), назначим приоритет z_4.
3. Если не выполняются условия (1) или (2), назначим приоритет z_3.
4. Если вхождение (l,β,q) не порождает истинной зависимости (происходит обращение к входным данным) и $\rho_{l,\beta,q}^{\Phi} = \rho_{l,\beta,q}^{\Phi,\xi}$, то назначим приоритет z_1.
5. Если вхождение (l,β,q) не порождает истинной зависимости (происходит обращение к входным данным) и $\rho_{l,\beta,q}^{\Phi} < \rho_{l,\beta,q}^{\Phi,\xi}$, то назначим приоритет z_2.

В пунктах 4 и 5 для вхождений, представляющих собой обращения к входным данным, приоритет выше случаев в пунктах 2 и 3, поскольку элементы массивов не перезаписываются.

Величина многократности использования $k^{\Phi,\xi}_{l,\beta,q}$ определяет значимость данного вхождения для случаев, когда один массив может встречаться в правых частях более одного раза. Малые значения $k^{\Phi,\xi}_{l,\beta,q}$ соответствуют большей значимости.

Направления дальнейших исследований

1. Разработка и программная реализация алгоритмов, позволяющих выбирать соотношения размеров тайла (при координатном тайлинге). Алгоритмы должны учитывать допустимость тайлинга.
2. Получение условий и соотношений, точно характеризующих объем коммуникационных операций и учитывающих, что один массив может в правых частях встречаться более одного раза. Разработка алгоритмов для случая, учитывающего наличие общих циклов для разных групп операторов (приоритеты для выбора соотношения размеров тайла для разных групп операторов могут быть разные).
3. Исследования в рамках предлагаемого подхода для разработки параллельных алгоритмов, ориентированных на компьютеры с распределенной памятью и суперкомпьютеры, использующие многосокетные узлы с многоядерными процессорами в сокетах. В этих случаях следует учитывать также зависимости между операциями тайлов разных типов.
4. Вычислительные эксперименты на GPU, компьютерах с распределенной памятью, многоядерных компьютерах. Разработка параллельных алгоритмов для прикладных задач с использованием предлагаемого подхода.

Работа выполнена при поддержке Белорусского республиканского фонда фундаментальных исследований.

Библиографические ссылки

1. *Baskaran M., Ramanujam J., Sadayappan P.* Automatic C-to-CUDA code generation for affine programs // Proceedings of the Compiler Construction, 19th International Conference. Part of the Joint European Conferences on Theory and Practice of Software. Paphos, Cyprus, March 20–28, 2010.
2. *Xue J., Cai W.* Time-minimal tiling when rise is larger than zero // Parallel Computing. 2002. V. 28. № 5. P. 915–939.
3. *Kim D., Rajopadhye S.* Parameterized tiling for imperfectly nested loops // Technical Report CS-09-101, Colorado State University, Department of Computer Science, February, 2009.
4. *Воеводин В. В.* Параллельные вычисления. СПб. : БХВ-Петербург, 2002.
5. *Feautrier P.* Some efficient solutions to the affine scheduling problem Part 1 // International Journal of Parallel Programming. 1992. V. 21. № 5. P. 313–348.
6. *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. 2008. № 4959. P. 132–146.
7. *Лиходед Н. А.* Характеристика локальности параллельных реализаций многомерных циклов // Доклады НАН Беларуси. 2010. № 1. С. 26–32.