В. А. Евстигнеев

МНОГОЧЛЕНЫ ЭРХАРТА*

ВВЕДЕНИЕ

Исследование различных свойств программ и преобразований часто сводится к определению числа целочисленных точек (т.е. точек с целочислеными координатами) в многогранниках (в частности, в многоугольниках). К такой задаче, например, сводится определение мощности множества итераций гнезда циклов, заданного в виде многогранника, в том числе и *параметризованного многогранника*, в котором число целочисленных точек есть функция от этих параметров. Такие функции называются *энумераторами* многогранника.

В 50-е годы прошлого столетия французский математик Эжен Эрхарт (Eugène Ehrhart) обнаружил, что некоторое расширение полиномов, которые потом стали называть *полиномами Эрхарта*, даёт возможность описать энумератор любого многогранника. При этом некоторые свойства таких полиномов находятся в непосредственной связи со свойствами многогранника.

В настоящей статье излагаются основы теории полиномов Эрхарта, следуя работе [1].

1. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

1.1. Периодические числа

Определим вначале *периодические числа*, которые будут служить коэффициентами для *псевдополиномов*.

Определение 1. *Одномерное периодическое число* есть выражение

$$u(n) = [u_0, \ldots, u_{p-1}] := u_{n \bmod p} = \begin{cases} u_0, \text{ если } n \bmod p = 0, \\ \vdots \\ u_{p-1}, \text{ если } n \bmod p = p-1 \end{cases},$$

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (гранты № 02-07-90409 и 04-07-90441).

равное тому значению, индекс которого равен $n \bmod p$. При этом p называется *периодом* для $u(n)$.

Определение 2. *q-мерное периодическое число* $u(n_1, \ldots, n_q)$ определяется через q-мерное поле U величины $p_1 \times \cdots \times p_q$:

$$u(n_1, \ldots, n_q) := U_{(n_1 \bmod p_1, \ldots, n_q \bmod p_q)}.$$

При этом q есть *размерность* поля U. Вектор $p = (p_1, \ldots, p_q)$ называется *мультипериодом* u. Наименьшее общее кратное чисел p_i называется *периодом* u. i-й коэффициент мультипериода называется *периодом размерности* i.

Двумерные периодические числа могут изображаться матрицами. Для более высоких размерностей используется способ представления, известный как "вектор векторов". Примером периодического числа служит

$$[[[0,1]_k, [2,3]_k]_l, [[4,5]_k, [6,7]_k]_l, [[8,9]_k, [10,11]_k]_l]_m.$$

Его размерность равна 3, а его мультипериод равен $\{2,2,3\}$. Следовательно, его период — 6; период размерности 1 есть 2 и т.д.

Период периодического числа может быть увеличен многократно. При этом компоненты соответствующим образом сцепляются. Это же справедливо для периода размерности i многомерного периодического числа. Например, период размерности 2 двумерного периодического числа $[[1,2,3]_k, [3,4,5]_k]_l$ может быть удвоен:

$$[[1,2,3]_k, [3,4,5]_k, [1,2,3]_k, [3,4,5]_k]_l.$$

Значение числа при этом не изменится. Соответственно, периодическое число может быть *редуцировано*, если составляющие элементы в размерности повторяются.

Различные периодические числа одинаковой размерности могут складываться. Пусть p_i и q_i — периоды размерности i двух чисел. Периоды должны быть заменены на наименьшее общее кратное $kgV(p_i, q_i)$, после чего числа складываются покомпонентно. Например, нам нужно сложить два числа $[2,3]_n$ и $[4,5,6]_n$. Наименьшее общее кратное периодов равно 6; следовательно, числа преобразуются к виду $[2,3,2,3,2,3]_n$ и $[4,5,6,4,5,6]_n$. Результатом будет число $[6,8,8,7,7,9]_n$. Периодические числа могут также покомпонентно перемножаться.

Определение 3. *Псевдополином* есть функция $f:\ Z^q \to Z$

$$(n_1,\ldots,n_q) \mapsto \sum_{i_1=0}^{k} \sum_{i_2=0}^{k-i_1} \cdots \sum_{i_q=0}^{k-i_1-\cdots-i_{q-1}} c_{i_1,\ldots,i_q} n_1^{i_1} \cdots n_q^{i_q},$$

где $c_{i_1,\ldots,i_q}$ — коэффициенты периодического числа, наибольшая размерность которого равна q.

Псевдополином характеризуется размерностью, степенью и псевдопериодом. *Размерность* равна q, *степень* есть наибольший показатель k, и *псевдопериод* есть наименьшее общее кратное периодов коэффициентов.

1.2. Многогранники

Введём ряд определений, касающихся многогранников и необходимых в дальнейшем.

Определение 4. *Полиэдром* называется пересечение конечного числа полупространств. Ограниченный полиэдр называется *многогранником* (*политопом*). *Размерностью* полиэдра $\mathcal{P}$ называется размерность аффинного подпространства, целиком содержащего $\mathcal{P}$.

Как альтернатива может быть использовано определение многогранника как выпуклой оболочки конечного множества точек.

Когда речь идёт о двух- или трёхмерных полиэдрах, то интуитивно ясно, что подразумевается под "вершинами" или "рёбрами". Следующее определение формализует эти понятия и обобщает их на полиэдры любой размерности.

Определение 5. Пусть $\mathcal{P} \subseteq Q^n$ — полиэдр и $\mathcal{H} \subseteq Q^n$ — гиперплоскость. Пусть далее $\mathcal{F} := \mathcal{P} \cap \mathcal{H}^+$. Если $\mathcal{F} \neq \emptyset \wedge \mathcal{F} \subseteq \mathcal{H}$, то $\mathcal{H}$ называется *опорной гиперплоскостью*, а $\mathcal{F}$ — *гранью* многогранника $\mathcal{P}$. Каждая грань в $\mathcal{P}$ — снова многогранник.

Грань размерности 0 называется *вершиной*, размерности 1 — *ребром*, размерности $dim(\mathcal{P}) - 1$ — *собственно гранью*.

Если отдельные неравенства, которые описывают различные полупространства, собраны вместе и их коэффициенты сведены в матрицу, то полиэдр может быть представлен в следующем виде:

$$\mathcal{P} = \{\vec{x} |\ A\vec{x} + \vec{b} \geq 0\}.$$

Прежде чем приступить к выводу компактной формулы для подсчёта числа целочисленных точек в полиэдре, определим области допустимости параметров.

Введённые многогранники используются, например, для описания пространства итераций гнезда циклов, причём границы циклов описываются аффинными функциями. Так как эти описания часто зависят от имеющихся в программе параметров, то введём в рассмотрение *вектор параметров* $\vec{n}$, символически охватывающий все имеющиеся параметры.

С помощью этого вектора можно будет параметрически описывать полиэдры. Коэффициенты матрицы A при этом остаются неизменными, изменяться в зависимости от параметров будет только смещение $\vec{b}$.

Определение 6. *Параметрический полиэдр*, соотнесённый с вектором параметров $\vec{n} = (n_1, \ldots, n_r)$, есть полиэдр формы $\mathcal{P} = \{\vec{x} |\ A\vec{x} + C\vec{n} + \vec{b} \geq 0\}$. *Параметрическое пространство* есть подмножество $\mathcal{D}$ пространства N_0^r, которое содержит только допустимые значения $\vec{n}$.

Ясно, что параметры никогда не принимают отрицательные значения.

В исследуемом классе регулярных программ границы массивов суть константы, а границы циклов зависят от параметров программы.

Определение 7. *Параметрическая константа* f размерности m есть m-мерная параметризованная функция

$$f(\vec{x}) = F\vec{x} + F_n\vec{n} + \vec{f},$$

для которой $F = 0$. Так как значение функции при этом больше не зависит от $\vec{x}$, то применяется следующее обозначение:

$$f = F_n\vec{n} + \vec{f}.$$

1.3. Пример

Пусть дан следующий многогранник с вектором параметров $\vec{n} = (P, Q)$:

$$\begin{aligned}\mathcal{P} &= \left\{ \begin{pmatrix} i \\ j \end{pmatrix} | \begin{pmatrix} 1 & 0 \\ -2 & 0 \\ -1 & 1 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \vec{n} \geq 0 \right\} = \\ &= \left\{ \begin{pmatrix} i \\ j \end{pmatrix} | \, 0 < i \leq \tfrac{1}{2}P \wedge i \leq j \leq \tfrac{1}{2}Q \right\}.\end{aligned}$$

Конфигурация этого многогранника зависит от параметров P и Q. Возможны два случая (см. рис. 1).

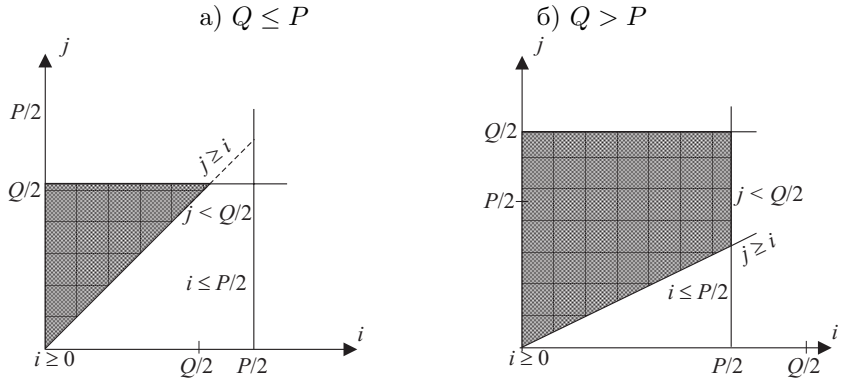


Рис. 1.

2. ЦЕЛОЧИСЛЕННЫЕ ТОЧКИ

Чтобы вывести компактную формулу для подсчёта числа целочисленных точек в многограннике, надо вначале определить область допустимых значений параметров.

Определение 8. *Областью допустимости* параметризованного полиэдра называется максимальное подмножество пространства параметров, для которых множество вершин фиксированно. Пространство параметров $\mathcal{D}$ разлагается на непересекающиеся области допустимости

$$\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_n.$$

Множесство вершин для области допустимости $\mathcal{D}_i$ обозначается через $\mathcal{E}_i$.

Определение 9. *Энумератор* параметризованного многогранника в области $\mathcal{D}_i$ есть функция $E : \mathcal{D}_i \mapsto N_0$, которая каждому значению параметра сопоставляет число целочисленных точек в многограннике.

В дальнейшем с понятием многогранника всегда будет связываться некоторая область допустимости параметризованного многогранника.

Определение 10. Многогранник называется *целочисленным*, если все его вершины для всех значений параметров имеют целочисленные коэффициенты. Иначе он — *рациональный*. *Знаменатель* рациональных точек есть наименьшее общее кратное знаменателей их координат. *Знаменатель* рационального многогранника есть наименьшее общее кратное знаменателей его вершин.

Следующее предложение, доказательство которого опускается, утверждает, что знаменатель рационального полиэдра не зависит от параметров как от констант.

Предложение 1. Пусть $\mathcal{P}(\vec{n})$ — параметризованный многогранник. Координаты его параметризованных вершин допускают представление в виде аффинных функций параметров.

2.1. Пример (продолжение)

Для заданного выше многогранника общее пространство параметров имеет вид $\mathcal{D} = \{(P,Q)|\ P \geq 0, Q \geq 0\}$, что означает, что возможные значения параметров ничем больше не ограничены. Чтобы можно было принять во внимание различные формы многогранника, это пространство разбивается на области допустимости. Первая область допустимости есть $\mathcal{D}_1 = \{(P,Q)|\ Q \leq P\} \cap \mathcal{D}$, вторая — $\mathcal{D}_2 = \{(P,Q)|\ Q > P\} \cap \mathcal{D}$. Имеем $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$.

Для этих областей допустимости множества параметризованных вершин определяются как:

$$\begin{aligned} \mathcal{E}_1 &= \left\{(0,0), \left(\tfrac{Q}{2}, \tfrac{Q}{2}\right), \left(0, \tfrac{Q}{2}\right)\right\}, \\ \mathcal{E}_2 &= \left\{(0,0), \left(\tfrac{P}{2}, \tfrac{P}{2}\right), \left(\tfrac{P}{2}, \tfrac{Q}{2}\right), \left(0, \tfrac{Q}{2}\right)\right\}. \end{aligned}$$

Только две вершины $(0,0)$ и $\left(0, \frac{Q}{2}\right)$ входят в обе области допустимости.

3. ВЫЧИСЛЕНИЕ ПОЛИНОМОВ

Следующее фундаментальное утверждение Эрхарта даёт основу для алгоритма вычисления полиномов.

Предложение 2. Пусть $\vec{n} = (n_1, \ldots, n_p)$ — вектор параметров. Каждый энумератор $E(\vec{n})$ d-многогранника $\mathcal{P}(\vec{n})$ со знаменателем q есть

псевдополином размерности d и степени d. Период его коэффициентов есть максимум q.

Размерность многогранника определяет степень полинома, знаменатель — максимальный период, а число параметров — размерность. Тем самым определяется схематический вид полинома и нужно только определить его коэффициенты.

Знание структуры делает возможным следующий подход к определению многогранника. Нужно придавать параметрам различные конкретные значения и определять число целочисленных точек. Тем самым будут получены значения энумераторов в определенных точках. Отсюда вытекает возможность определять коэффициенты, для чего нужно решить систему уравнений, соответствующих выбранным конкретным значениям.

3.1. Пример (продолжение)

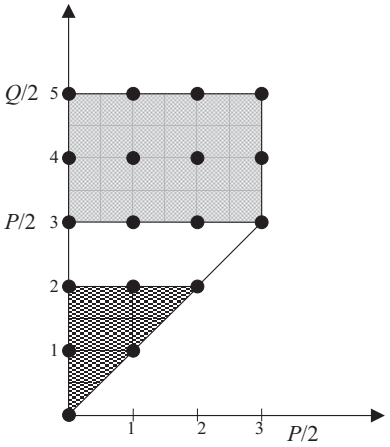
В качестве примера снова возьмём наш многогранник $\mathcal{P}$ с областью допустимости $\mathcal{D}_2$. Размерность многогранника равна 2; кроме того, степень разыскиваемого полинома также равна 2. Знаменатель также равен 2 (см. формулы для величин $\mathcal{E}_1$ и $\mathcal{E}_2$). Отсюда следует, что максимальный период коэффициентов равен 2. Так как число параметров равно 2, размерность полинома должна быть равной 2. Полином теперь имеет следующий вид:

$$\begin{aligned} E_P(P,Q) = {} & [[c_1,c_2]_Q,[c_3,c_4]_Q]_P P^2 + [[c_5,c_6]_Q,[c_7,c_8]_Q]_P Q^2 \\ & +[[c_9,c_{10}]_Q,[c_{11},c_{12}]_Q]_P PQ + [[c_{13},c_{14}]_Q,[c_{15},c_{16}]_Q]_P P \\ & +[[c_{17},c_{18}]_Q,[c_{19},c_{20}]_Q]_P]Q + [[c_{21},c_{22}]_Q,[c_{23},c_{24}]_Q]_P. \end{aligned}$$

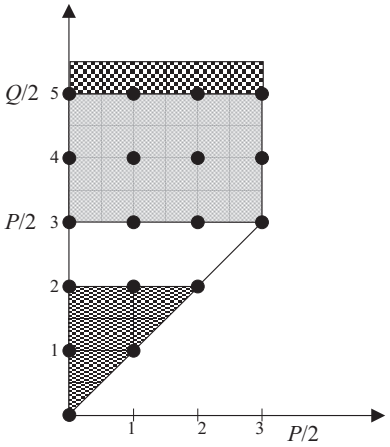
В совокупности нужно определить 24 неизвестных от c_1 до c_{24}. Будем определять периодические числа, исходя из известных четырех частных случаев, указанных на рис. 2. Для случая (а) Q и P — чётные, и мы имеем следующий энумератор:

$$E_P(P,Q) = c_1P^2 + c_5Q^2 + c_9PQ + c_{12}P + c_{17}Q + c_{21}.$$

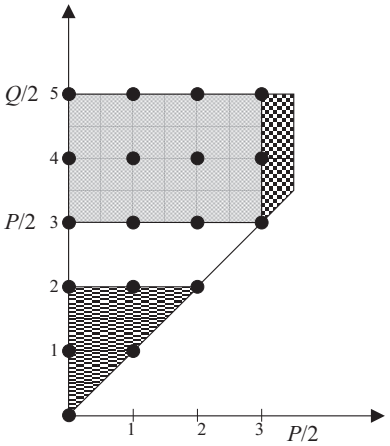
Аналогично получаем ещё три энумератора для оставшихся трёх случаев (б)–(г). Для рассматриваемого случая 6 неизвестных определяются путём подсчёта числа точек для 6 пар конкретных значений для P и Q. Результаты приведены в табл. 1. Из этой таблицы извлекается система из 6 линейных уравнений:



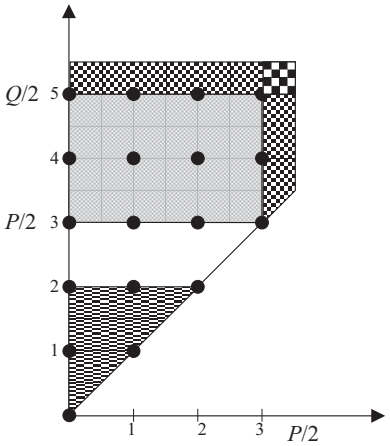
а) P, Q — четные



б) Q нечетно



в) P нечетно



г) P, Q — нечетные

Рис. 2.

Таблица 1

Результаты вычисления многогранника $\mathcal{P}$

P	Q	# точек	$E_P(P,Q)$
0	0	1	c_{21}
0	2	2	$4c_5 + 2c_{17} + c_{21}$
2	2	3	$4c_1 + 4c_5 + 4c_9 + 2c_{13} + 2c_{17} + c_{21}$
2	4	5	$4c_1 + 16c_5 + 8c_9 + 2c_{13} + 4c_{17} + c_{21}$
4	6	9	$16c_1 + 36c_5 + 24c_9 + 4c_{13} + 6c_{17} + c_{21}$
4	8	12	$16c_1 + 64c_5 + 32c_9 + 4c_{13} + 8c_{17} + c_{21}$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 4 & 0 & 0 & 2 & 1 \\ 4 & 4 & 4 & 2 & 2 & 1 \\ 4 & 16 & 8 & 4 & 2 & 1 \\ 16 & 36 & 24 & 4 & 6 & 1 \\ 16 & 64 & 32 & 4 & 8 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_5 \\ c_9 \\ c_{13} \\ c_{17} \\ c_{21} \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 5 \\ 9 \\ 12 \end{pmatrix}.$$

Решение этой системы даёт следующие значения для коэффициентов

$$c_1 = -\frac{1}{8},\ c_5 = 0,\ c_9 = \frac{1}{4}.\ c_{13} = \frac{1}{4},\ c_{17} = \frac{1}{2},\ c_{21} = 1.$$

В остальных трёх случаях также имеем дело с 6 неизвестными. После упрощения получим следующий полином:

$$E(P,Q) = \tfrac{1}{4}QP - \tfrac{1}{8}P^2 + \tfrac{1}{4}\left[[1,0]_Q, [2,1]_Q\right]_P P + \tfrac{1}{4}[2,1]_P Q + \tfrac{1}{8}\left[[1,4]_Q, [5,3]_Q\right]_P.$$

4. ПРИЛОЖЕНИЯ К NUMA-АРХИТЕКТУРАМ

4.1. Введение

Высоко масштабируемые параллельные компьютеры, например SCI-сцепленные кластеры рабочих станций, относятся к NUMA-архитектурам [2]. Таким образом, хорошая статическая локальность существенна для высокой производительности и масштабируемости параллельных программ на этих машинах. В этом разделе, следуя работе [3], описывается

новая техника для оптимизации статической локальности во время компиляции путём применения преобразования и распределения данных. Метрика, с помощью которой оценивается оптимизация, использует полиномы Эрхарта и допускает точное вычисление количества статической локальности. Эффективность новой техники подтверждается экспериментами, которые проводились на SCI-сцепленных кластерах рабочих станций в университете в Падерборне.

Мы используем ограниченную версию блочно-циклической модели распределения, начиная с гнезда параллельных циклов, которые включают аффинные границы циклов и аффинные индексные функции для многомерных массивов. Предполагается, что гнезда содержит только один параллельный цикл. Мы предполагаем, что массивы нарезаны блоками вдоль одной размерности, которые распределены по обрабатывающим узлам. В лучшем случае любой такой блок доступен исключительно для обрабатывающего узла, который владеет блоком. Следовательно, обязанностью преобразований данных является выявление регулярного шаблона блоков, которые доступны из отдельных узлов. Последующее распределение данных должно определить назначение блоков обрабатывающим узлам, которое совместимо с шаблоном. Мы сохраняем распределение вычислений по процессорам, которое было использовано на предыдущих шагах компиляции.

В итоге, для каждого массива в регулярной программе мы автоматически исполняем унимодулярные преобразования данных, которые преобразуют массив, и блочно-циклическое преобразование данных, которое распределяет элементы массива по обрабатывающим узлам. Распределение использует некоторую длину цикла и некоторое смещение.

4.2. Геометрический подход

Мы будем использовать многосеточное приложение из области гидродинамики для иллюстрации нашего подхода. Вычислительное ядро — один из вариантов SOR-цикла.

```
DO I = 2, M+N-2
  FORALL J = MAX(1, I-M+1), MIN(I-1, N-1)
    U(S, I-J, J-1) = (F(S, I-J, J) + U(S, I-J, J-1)
                     + U(S, I-J-1, J) + U(S, I-J, J+1)
                     + U(S, I-J+1,J)/4.0
  ENDFORALL
END DO
```

Это двумерное гнездо циклов с пятью ссылками на массив U и одной ссылкой на массив F. Мы сосредоточимся на ссылках на массив U. Параллельная программа приведённой версии SOR-цикла была получена автоматическим параллелизатором нашего прототипа компилятора. Гнездо циклов показывает параллелизм в самом внутреннем уровне и представляет собой субъект для последующей оптимизации локальности данных.

Теперь мы введём некоторые удобные сокращения и определим фундаментальные геометрические абстракции, приспособленные для классификации преобразований и распределений.

Гнездо циклов N определяет *итерационное пространство* $\mathcal{I}_N$. Массив X, который доступен через ссылку R_l, $l \geq 1$, определяет *индексное пространство* $\mathcal{D}_X$. Через f_l мы сошлёмся на индексную функцию ссылки R_l. Кроме того, P обозначает число процессоров, d — размерность распределения, B — размер блока, j — параллельную размерность. Отныне мы опускаем индексы, если это не вызовет недоразумения. Хорошо известно, что пространство итераций $\mathcal{I}$ и индексное пространство $\mathcal{D}$ оба определяют выпуклые многогранники. Обычно мы представляем выпуклый многогранник $\mathcal{P}$ множеством неравенств, т.е. имеем:

$$\mathcal{P} = \{\vec{x} \in Z^k |\ A \cdot \vec{x} + C \cdot \vec{n} + \vec{b} \geq 0\}.$$

Полином Эрхарта E параметризованного выпуклого многогранника $\mathcal{P}$ есть функция от параметров многогранника, отображающая число целочисленных точек, содержащихся внутри $\mathcal{P}$. Фундаментальная идея нашего подхода — закодировать итерационные точки, которые вызывают локальные доступы через выпуклые многогранники. Тогда полиномы Эрхарта обеспечивают средства для оценки качества преобразования и распределения данных. Мы используем обычную нотацию для полиномов Эрхарта. Обозначение $E(M, N) = [10, 5]_N \cdot M$ есть сокращение, позволяющее различать два случая: $E(M, N) = 10 \cdot M$, если $N \bmod 2 = 0$, и $E(M, N) = 5 \cdot M$, если $N \bmod 2 = 1$, соответственно. Эта вычислительная схема расширяет циклические коэффициенты больших размерностей. Более того, многогранник может иметь множество полиномов Эрхарта. В этом случае его полиномы определяются для выпуклых подмножеств пространства параметров, называемых областями правильности (validity domains).

Теперь наши два примитива для распределения данных — агрегирование блоков (block aggregation) и свёртка блоков (block convolution) —

могут быть оттранслированы в терминах выпуклых многогранников. Мы начнём с примитива, который адресует агрегирование блоков. Он собирает эффект от распределения данных с блоками одинакового размера.

Пусть $\vec{x}' = f_l(\vec{x}) \in \mathcal{D}$ обозначает индексную точку, доступную через ссылку R_l в итерационной точке $\vec{x} \in \mathcal{I}$. Так как распределение данных применяется к размерности d, блок, идентифицируемый как $x'_d = \lfloor x_d/B \rfloor$, доступен в $\vec{x}$. Нелинейное выражение $\lfloor x_d/B \rfloor$ может быть преобразовано в линейное за счёт дополнительной неизвестной b и ограничения C_d. Если мы встретим уравнение, содержащее $\lfloor x_d/B \rfloor$, то заменим его на новую свободную переменную b и дополнительное ограничение на допустимые границы изменения x_d для удовлетворения C_d: $B \cdot b \le x_d \le B \cdot (b+1)$.

Мы продолжим с примитивом для свёртки (конволюции) блоков. Он суммирует эффект циклического распределения данных. Поэтому пусть $b = f_l(\vec{x})$ обозначает выражение, которое вычисляет в блоке, доступном через ссылку R_l в итерационной точке $\vec{x} \in \mathcal{I}_N$. Потом этот блок будет распределён процессору $b \bmod P$. Выражение $b \bmod P$ должно быть преобразовано в линейное выражение для подстановки в наш линейный шаблон. Мы заменяем его на $(b - P \cdot z)$, где z — новая свободная переменная, и дополнительно ограничим выражение b для удовлетворения C_b: $P \cdot z \le b \le P \cdot (b+1)$.

Таким образом, мы можем использовать приведённые выше примитивы для описания множеств итерационных точек, не покидая области выпуклых многогранников.

4.3. Преобразование данных

Предлагаемый метод для выбора преобразований данных может быть разделён на две фазы. Первая фаза вычисляет множество оптимальных преобразований и распределений для каждой ссылки отдельно. Этот подход гарантирует достижение цели в случае инъективных индексных функций, и оптимальность совпадает с отсутствием отдалённых доступов. Вторая фаза упорядочивает эти преобразования-кандидаты; она сравнивает сопоставленные им полиномы Эрхарта, рассматривая все (!) ссылки совместно и выбирая лучшее преобразование из всех преобразований-кандидатов.

На рис. 3 показаны три главных шага в генерации преобразований-кандидатов. В течение первого шага выбираются базисные векторы, ко-

торые стягивают подпространства итерационного пространства так, что эти подпространства доступны только одному процессору (а). Потом эти векторы отображаются в индексное пространство, где они стягивают подпространства, которые доступны уже более чем одному процессору (б). Внутри следующего шага определяется унимодулярное преобразование, которое делает эти подпространства ортогональными одной из осей (в). Полученный массив нарезается на блоки вдоль выбранной оси. Каждый из этих блоков либо не используется, либо используется только одним процессором, что ведёт к некоторому шаблону использования (utilization pattern) блоков памяти. Наконец, определяется смещение для перемещения шаблона так, чтобы он отображал распределение данных. Результатом является преобразование, которое возвращает все (!) доступы, исполняемые одной ссылкой в локальных доступах.

4.3.1. Упорядочение ссылок

Вначале покажем, как упорядочить преобразование данных относительно отдельной ссылки R. Мы начинаем с выпуклого многогранника итерационного пространства $\mathcal{I}$ и разлагаем его на подпространства $\mathcal{I}_p$ так, что подпространство $\mathcal{I}_p$ исполняется на процессоре P_p. Таким образом,

$$\mathcal{I} = \{\vec{x} \in Z^k |\ A \cdot \vec{x} + C \cdot \vec{n} + \vec{b} \geq 0\}$$

для подходящих матриц A, C и вектора $\vec{b}$. Полином Эрхарта I для $\mathcal{I}$ показывает число итераций, которые могут исполняться всеми параллельными процессорами. В случае примера с мультирешётками получаем полином следующего вида:

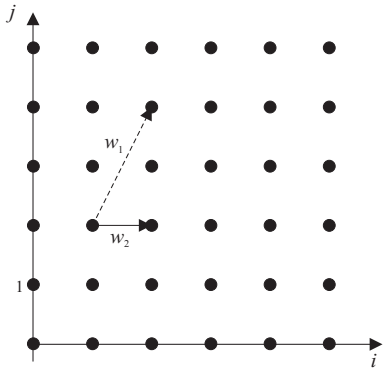
$$I(M, N) = M \cdot N - M - N + 1.$$

Так как мы предположили циклическое отображение итерационных точек в параллельной размерности j в множество из P параллельных процессоров, то получаем, что

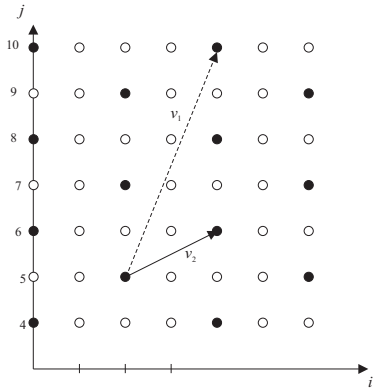
$$\mathcal{I}_p = \{\vec{x} |\ \vec{x} \in I \wedge x_j \bmod P = p\}.$$

Таким образом, каждое множество $\mathcal{I}_p$ также есть выпуклый многогранник, и существует полином Эрхарта I_p для $\mathcal{I}_p$. Например, для рассматриваемого примера

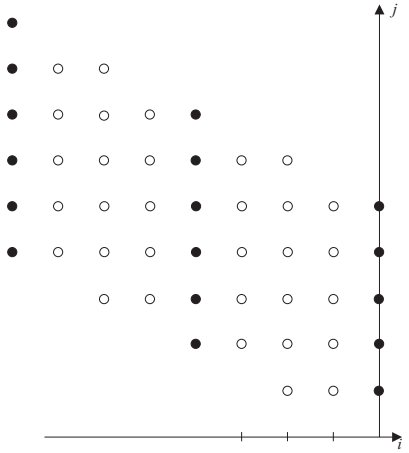
$$I_p(M, N) = \frac{1}{2}(M \cdot N - M - \left[\begin{array}{cc} 2 & 1 \\ 0 & 1 \end{array}\right]_{p,M} \cdot N + \left[\begin{array}{cc} 2 & 1 \\ 0 & 1 \end{array}\right]_{p,M}).$$



а) итерационное пространство



б) индексное пространство



в) новое индексное пространство

Рис. 3.

Чтобы вычислить индексную точку внутри преобразованного индексного пространства, нам нужно применить преобразование

$$\vec{x} \mapsto T \cdot \vec{x} + T_n \cdot \vec{n} + \vec{t}$$

для индексной точки $f(\vec{x})$. Потом мы можем исследовать блочно циклическое распределение массива, для того чтобы выявить, является ли элемент массива $t(f(\vec{x})) = \vec{x}''$, к которому имеется доступ из итерационной точки $\vec{x}$, локальным элементом массива процессора P_p. Согласованное ограничение C_p, таким образом, имеет вид:

$$C_p : \; B \cdot p \leq \pi_d(\vec{x}'') - (B \cdot P) \cdot z_2 < B \cdot (p+1).$$

Заметим, что $\pi_d(\vec{x})$ обозначает проекцию на компоненту x_d. Таким образом, множество итерационных точек $\mathcal{L}_p$, которые порождают доступы к элементам локального массива, равно

$$\mathcal{L}_p = \{\vec{x} \in {}_p |\; C_p(\vec{x} = true\}.$$

Заметим, что множество $\mathcal{L}_p$ — также выпуклый многогранник. Множество $\mathcal{R}_p$, которое порождает удалённые доступы, равно разности $\mathcal{R}_p := \mathcal{I}_p - \mathcal{L}_p$ и в общем случае не является выпуклым многогранником. Тем не менее полином для $\mathcal{R}_p$ существует и даёт число удалённых доступов. Для ссылки $\mathcal{R}_1 = \mathtt{U(I-J, J-1}$ нашего примера и для $B = 1$, $t = id$, получаем:

$$L_0(M,N) = \frac{1}{4}(M \cdot N - [2,1]_N \cdot M - [2,1]_M \cdot N + \left[\begin{array}{cc} 4 & 2 \\ 2 & 1 \end{array}\right]_{N,M}).$$

Отметим, что L_0 есть специализация полинома $L(M,N,p)$ для $p = 0$. Следовательно, мы можем вычислить полином $|L_0(M,N) - L_1(M,N)|$, который будет означать *небаланс* доступов к удаленной памяти для задействованных процессоров. Ниже мы рассмотрим подробнее действие этого небаланса.

4.3.2. Ранжирование преобразований

Покажем, что представленная выше конструкция $\mathcal{L}_p$ позволяет определить отношение локальных (удалённых) доступов любой ссылки $\mathcal{R}_l$. Мы начнём с итерационного пространства как параметризованного многогранника, включаем новый параметр p для отбора итерационных точек, исполняемых процессором P_p, и далее ограничиваем этот многогранник так, чтобы он содержал только точки с локальным доступом.

Если мы опустим параметр p, который представляет процессор P_p, то мы получим желаемый многогранник $\mathcal{L}_l$.

Объём многогранника $\mathcal{L}_l$ представляется полиномом Эрхарта L_l, который служит в качестве метрики для ранжирования преобразований относительно ссылки R_l. Таким образом, сумма $\sum_l L_l$ по всем ссылкам, дополненная полиномом G, который изображает полное число доступов к памяти, отражает отношение локальных (удалённых) доступов исходного гнезда циклов N и обеспечивает желаемую метрику для ранжирования комбинаций линейного преобразования данных и блочно-циклического распределения данных.

Мы не рассматриваем случай кратных областей правильности для многогранников $\mathcal{L}_l$. В этом случае нужно больше информации относительно параметров программы, для того чтобы выбрать правильную область правильности.

4.3.3. Заключительный отбор

Чтобы окончательно выбрать преобразование, которое хорошо реализуется для входного гнезда циклов, мы *символически сравним* полиномы Эрхарта для различных преобразований и сохраним лучшее из всех преобразований-кандидатов. Для заданного конечного множества преобразований, которое будет сконструировано ниже в разд. 4.4, сравним полиномы L этих преобразований. Без дальнейших знаний о параметрах программы мы вначале упростим периодические коэффициенты, т.е. мы заменим их арифметическими средними, унифицируем параметры программы и потом сравним коэффициенты полиномов в нисходящем порядке. Следующие полиномы Эрхарта E_M и E_N суть полиномы для нашего текущего примера для двух параллельных процессоров:

$$E(M,N,p) = \frac{1}{4}\cdot(5\cdot M\cdot N - \left[\begin{array}{cc} 10 & 5 \\ 0 & 5 \end{array}\right]_{p,N}\cdot M - \left[\begin{array}{cc} 6 & 5 \\ 4 & 5 \end{array}\right]_{p,M}\cdot N + \left[\left[\begin{array}{cc} 12 & 10 \\ 6 & 5 \end{array}\right]_{N,M}, \left[\begin{array}{cc} 0 & 0 \\ 4 & 5 \end{array}\right]_{N,M}\right]_p),$$

$$E(M,N,p) = \frac{1}{4}\cdot(6\cdot M\cdot N - \left[\begin{array}{cc} 12 & 6 \\ 0 & 6 \end{array}\right]_{p,N}\cdot M - 6\cdot N + \left[\begin{array}{cc} 12 & 6 \\ 0 & 6 \end{array}\right]_{p,N}).$$

Полином E_M представляет собой дефолт-распределение вдоль оси M, в то время как E_N представляет распределение вдоль оси N. Оба полинома отображают число локальных доступов, откуда следует, что распределение данных вдоль оси N "старше// распределения вдоль оси M, поскольку $\frac{6}{4} \cdot M \cdot N > \frac{5}{4} \cdot M \cdot N$ для значащих параметров M, N задачи. Более того, эти члены не зависят от процессорной координаты p. Распределение массива U вдоль оси M (E_M) показано на рис. 3.

4.4. Перечисление преобразований

Хотя результаты предыдущего подраздела позволяют ранжировать преобразования данных или заданную формулировку программы и поэтому обеспечивают точный результат сами по себе, нас будет интересовать перечисление *преобразований-кандидатов*, которое обеспечивает локальность, для того чтобы выбрать лучшее с помощью метрики L.

Вначале мы выберем $n-1$ векторов $\vec{w}_1, \ldots, \vec{w}_{n-1}$ внутри n-мерного итерационного пространства $\mathcal{I}$, которые стягивают непересекающиеся подпространства размерности $n-1$. Если $\mathcal{I}_{\vec{o}}$ — такое подпространство, идентифицируемое некоторым началом $\vec{o}$, т.е. $\mathcal{I}_{\vec{o}} = \{\vec{x} \in \mathcal{I} |\ \vec{x} = \vec{o} + \sum_{i=1}^{n-1} k_i \cdot \vec{w}_i,\ k_i \in Q\}$, то следующая импликация должна выполняться:

$$\vec{x}, \vec{x}' \in \mathcal{I}_{\vec{o}} \Rightarrow x_p \bmod P = x'_p \bmod P.$$

Таким образом, мы предназначаем подпространство $\mathcal{I}_{\vec{o}}$ отдельному процессору. В терминах порождающих векторов $\vec{w}_i$ мы требуем для произвольных итерационных точек $\vec{x},\ \vec{x}' \in \mathcal{I}$, чтобы

$$\vec{x}' = \vec{x} + \sum_{i=1}^{n-1} (k_i \cdot \vec{w}_i) \Rightarrow x_p \bmod P = x'_p \bmod P.$$

Теорема 4.1 даёт достаточное условие, которое позволяет выбор $\vec{w}_i$. Заметим, что ниже p обозначает параллельную размерность гнезда циклов.

Теорема 4.1. Пусть $\vec{w}_1, \ldots, \vec{w}_{n-1}$ — линейно независимые порождающие векторы из Z^n такие, что $\vec{w}_i = (w_{1,i}, \ldots, w_{n-1,i})^t$. Тогда эти векторы удовлетворяют сформулированному выше ограничению, если:

i) $\forall i:\ \gcd(w_{1,i} \ldots, w_{n-1,i}) = 1$ и

ii) $\forall j$: существует самое большее одно $i:\ w_{j,i} \neq 0$ и

iii)$\forall i:\ w_{p,i} \bmod P = 0$.

Следующая импликация применяется к индексному пространству.

Теорема 4.2. Пусть $\vec{w}_1, \ldots, \vec{w}_{n-1}$ суть линейно независимые векторы из Z^n, удовлетворяющие приведённому выше ограничению. Пусть $f(\vec{x}) = F \cdot \vec{x} + F_n \cdot \vec{n} + \vec{b}$ обозначает индексную функцию, имеющую квадратную обратимую матрицу доступа F. Пусть далее $\vec{v}_i = F \cdot \vec{w}_i$ обозначают образы векторов $\vec{w}$ относительно линейной части индексной функции f. Тогда:

$$f(\vec{x}') = f(\vec{x}) + \sum_{i=1}^{n-1} k_i \cdot \vec{v}_i \Rightarrow x_p \bmod P = x'_p \bmod P.$$

Таким образом, векторы $\vec{v}_i$, упоминаемые в теореме 4.2, стягивают подпространства индексного пространства, которые доступны из самое большее одного процессора. Формулировка теоремы 4.2 влечёт, что включаются только те индексные точки, которые имеют копию в итерационном пространстве. Рис. 3(а) иллюстрирует порождающие векторы $\vec{w}_i$, в то время как рис. 3(б) иллюстрирует векторы $\vec{v}_i$ для обеих теорем.

Остаётся вычислить преобразование T. Пусть $\vec{v}'_i = T \cdot \vec{v}_i$ обозначает образ $\vec{v}_i$ при преобразовании T. Если существует индекс j такой, что все векторы $\vec{v}'_i$ имеют 0-позицию в их j-й компоненте, тогда достаточно распределить индексные точки вдоль этой размерности. Мы вычисляем такое преобразование T, применяя гауссовское исключение, комбинируя векторы $\vec{v}_i$ в матрицу из n строк и $n-1$ столбцов. Её ранг равен $n-1$, потому что векторы $\vec{v}_i$ линейно независимы. Теперь мы исключаем элементы последней строки гауссовским исключением и замещаем эту строку на желаемом уровне. Алгоритм исключения даёт нам преобразование T.

4.5. Распределение данных

Пока мы вычислили матрицу преобразования. Остаётся определить параметры распределения. Это делается в два этапа. Вначале мы определяем окончательный шаблон использования (utilization pattern), а потом будет вычислено смещение для функции преобразования.

4.5.1. Шаблон использования

Вначале мы введём важное понятие вырезки массива (array slice).

Определение 11. *Вырезкой* массива X относительно размерности d называется подпространство индексного пространства $\mathcal{D}_X$, которое есть результат оценки фиксированной координаты в размерности d. Говорят, что процессор *владеет* вырезкой S_k, если он имеет доступ к элементам внутри вырезки, но никакой другой процессор такого доступа не имеет.

Шаблон пользования состоит из вырезок, которыми владеет специфицированный процессор, и из неиспользуемых вырезок. Используя $'*'$ для обозначения неиспользуемых вырезок, мы можем описать шаблон для преобразований-кандидатов на рис. 3(в) как $'0, *, *, *, 1, *, *, *'$. Имеем вырезку, принадлежащую процессору 0, за ней следуют три неиспользуемые вырезки, затем вырезка, принадлежащая процессору 1 и т.д. Этот шаблон повторяется циклически. Блоки с неиспользуемыми вырезками всегда имеют один и тот же размер: в нашем случае 3.

Поэтому для вычисления этого шаблона может быть использован простой итеративный алгоритм. Должны быть выделены три случая. В первом случае шаблон непосредственно встраивается в циклическое разбиение. Во втором случае к массиву должно быть применено обратное преобразование, чтобы сделать возможным включение шаблона в распределение, что, например, верно для шаблона $'2, *, 1, *, 0, *'$. В третьем случае шаблон не может быть встроен в распределение. Следовательно, эти пребразования удаляются из множества законных кандидатов.

4.5.2. Смещение

Вплоть до данного момента мы точно знали порцию T преобразования $t(\vec{x}) = T \cdot \vec{x} + T_n \cdot \vec{n} + \vec{t}$. Смещение $T_n \cdot \vec{n} + \vec{t}$ должно быть выбрано так, чтобы каждый процессор имел доступ только к тем вырезкам, которыми он сам владеет. Это свойство удовлетворяется для индексной функции без смещения. Мы должны определить смещение преобразования t так, чтобы оно компенсировало бы смещение f.

В контексте нашего простого процессорного отображения итерационная точка $\vec{0}$ исполняется на процессоре 0. Более того, вырезка S_0 всегда находится во владении процессора 0. Таким образом, достаточно выбрать смещение так, чтобы итерационная точка $\vec{0}$ вызывала бы доступ к вырезке S_0. Начиная с $t(f(\vec{0})) = \vec{0}$, мы получаем $T_n = -T \cdot F_n \wedge \vec{t} = -T \cdot \vec{f}$.

5. ЗАКЛЮЧЕНИЕ

Из экспериментов следует, что техника, основанная на полиномах Эрхарта, значительно улучшает обработку регулярных программ на NUMA-архитектурах. Она годится для улучшения распределения данных в программах с явным параллелизмом и для руководства оптимизацией распределения данных в распараллеливающих компиляторах.

СПИСОК ЛИТЕРАТУРЫ

1. Heine F. Optimierung der Datenverteilung für SCI-gekoppelte Workstation-Cluster. Master's thesis, Universität-GH Padeborn, May 1999.
2. Евстигнеев В.А., Мирзуитова И.Л. Развитие NUMA-архитектуры: текущее состояние // Современные проблемы конструирования программ. — Новосибирск, 2002. —- С. 139–154.
3. Heine F., Slowik A. Volume driven data distribution for NUMA-machines // Lect. Notes Comp. Sci. — Vol. 1900. — 2000. — P. 415 – 424.