ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УДК 681.3
DOI: 10.17586/0021-3454-2015-58-2-104-108

П. Ю. Ткачев, Д. Б. Борзов

МЕТОД РАСПАРАЛЛЕЛИВАНИЯ ЦИКЛОВ СО СЧЕТЧИКОМ

Разработан метод распараллеливания циклов со счетчиком внутри последовательных программ. Распараллеливание достигается за счет определения операторов или итераций цикла, которые могут быть выполнены одновременно на разных процессорах.

Ключевые слова: *метод, распараллеливание, цикл, программа, многопроцессорная система, оператор.*

Автоматическое распараллеливание — это процесс оптимизации программы компилятором, предполагающий ее автоматическое преобразование для возможности работы на параллельном компьютере. Целью автоматизации процесса является освобождение программиста от рутинного процесса ручного распараллеливания, однако полное автоматическое распараллеливание последовательных программ остается слишком сложной задачей, требующей сложнейших видов анализа.

Автоматический параллелизатор обычно работает с такими управляющими конструкциями, как циклы, поскольку в общем случае большая часть программы выполняется внутри каких-то циклов. Задача распараллеливающего компилятора — разделить цикл так, чтобы отдельные итерации могли исполняться на разных процессорах одновременно [1].

Любую программу можно представить в виде последовательности операторов [2]. Основной задачей при выявлении параллелизма между операторами (т.е. возможности параллельного выполнения двух последовательно идущих операторов) является определение информационной независимости, поскольку не должно одновременно инициироваться более одной операции записи в ячейку памяти, а также операции чтения и записи в одну ячейку [3]. Математически это можно проверить, вычислив функцию:

$$F(i,k) = (I_i \wedge O_k) \vee (I_k \wedge O_i) \vee (O_i \wedge O_k), \qquad (1)$$

где I_i, O_k — строки матриц входных/выходных переменных соответственно. В ячейках этих матриц ставится единица, если переменная является входной/выходной для оператора i; F — результат проверки возможности распараллеливания. Если F=0, то операторы могут выполняться параллельно, поскольку обрабатывают разные данные.

На таком подходе основан метод поиска и определения информационно независимых циклов, предполагающий нахождение циклов, которые можно выполнить параллельно на нескольких процессорах, обрабатывающих разные данные. Однако задача распараллеливания линейных участков внутри циклов в этом методе не решена.

Настоящая работа является продолжением исследований [4, 5]. Предлагаемый метод определения параллелизма внутри циклов основан на раскрутке (размотке) цикла. Линейный участок внутри цикла копируется столько раз, сколько в цикле итераций.

Например, исходный цикл

```
for(i=1; i<=3; i++)
{
 a=b+c;
 c=d+e;
 b=a+f;
}
```

сводится к линейному участку

```
a=b+c;
c=d+e;
b=a+f;
a=b+c;
c=d+e;
b=a+f;
a=b+c;
c=d+e;
b=a+f.
```

Далее задача сводится к распараллеливанию линейного участка согласно выражению (1). Матрица входных переменных имеет следующий вид:

$$I = \begin{bmatrix} A & B & C & D & E & F \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

матрица выходных переменных имеет следующий вид:

$$O = \begin{bmatrix} A & B & C & D & E & F \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

Далее по формуле (1) определяем параллельность операторов. Получаем следующий блок:

```
a=b+c;
c=d+e;      b=a+f;
a=b+c;
c=d+e;      b=a+f;
a=b+c;
c=d+e;      b=a+f.
```

Операторы, находящиеся в первом столбце, выполняются на первом процессоре, операторы во втором столбце выполняются на втором процессоре.

Таким образом, алгоритм распараллеливания циклов по предложенному методу следующий.

1. Выделить тело цикла.
2. Определить количество итераций.
3. Скопировать тело цикла столько раз, сколько в цикле итераций.
4. Для полученного блока составить матрицы входных и выходных переменных.
5. Определить по формуле (1) возможность параллельного выполнения операторов внутри блока.
6. Модифицировать цикл с учетом данных, полученных на предыдущем шаге.
7. Выполнить полученный цикл.

Если в исходной части программы есть несколько вложенных циклов, приведенный алгоритм применяется для каждого из циклов по убыванию уровня вложенности.

Процедура поиска информационно независимых итераций заключается в нахождении циклов, тело которых на (i+1)-й итерации информационно не зависит от тела цикла на i-й итерации, т.е. на каждой итерации циклом обрабатываются разные данные. Для определения возможности такого распараллеливания предлагается следующая методика.

Пусть исходный цикл имеет следующий вид:

```
for (i=1;i<10;i++)
{
 A=0;
 B=0;
 C=D+1;
}.
```

Выделим тело цикла и составим для всего набора операторов, входящих в тело, векторы входных и выходных переменных.

Тело цикла

```
A=0;
B=0;
C=D+1;
I=I+1.
```

В этом наборе операторов есть четыре переменные (`A`, `B`, `C`, `D`), а также счетчик цикла `I`, который всегда является и входной и выходной переменной в циклах такого вида, поэтому в эти векторы счетчик цикла не включается. Рассмотрим подробнее способ составления этих векторов. Так как в рассматриваемом случае в теле цикла используются четыре оператора, векторы входных и выходных переменных будут содержать по четыре элемента, т.е. каждому элементу ставится в соответствие определенная переменная. Элемент вектора I принимает значение „1“, если переменная используется хотя бы в одном операторе в теле цикла справа от знака „=“, иначе ставится „0“. Вектор выходных переменных заполняется так же, как и вектор входных переменных, за исключением того, что элемент вектора O принимает значение „1“, если переменная используется хотя бы в одном операторе в теле цикла слева от знака „=“, иначе ставится „0“.

Вектор входных переменных для итерации имеет следующий вид:

$$I = \{0,0,0,1\},$$

выходных:

$$O = \{1,1,1,0\}.$$

Вычислив конъюнкцию этих векторов, получим

$$I \wedge O = \{0,0,0,0\}. \quad (4)$$

Наличие в векторе (4) хотя бы одной единицы говорит о том, что какая-то переменная используется в теле цикла и в качестве входной, и в качестве выходной, следовательно, распараллелить тело цикла по этому методу невозможно. Если все элементы вектора будут равны нулю, то необходимо составить векторы (условия) использования счетчика цикла для каждого оператора. В нашем примере четыре оператора, значит, эти векторы будут иметь по четыре элемента, по одному на каждый оператор. Элемент этих векторов принимает значение „1“, если счетчик используется в операторе в качестве входной переменной, либо в качестве выходной переменной, иначе элемент принимает значение „0“.

Составим вектор использования счетчика цикла в качестве входной переменной:

$$I_i = \{0,0,0,1\},$$

выходной

$$O_i = \{0,0,0,1\}.$$

Для выявления использования счетчика цикла в каких-либо операторах, кроме оператора инкремента/декремента, лучше всего подходит операция сложения по модулю 2. В нашем случае:

$$I_i \oplus O_i = \{0,0,0,0\}. \quad (5)$$

Наличие хотя бы одной единицы в векторе (5) — показатель того, что счетчик цикла используется не только в операторе инкремента/декремента. В этом случае возможны два варианта развития событий. В первом счетчик используется в качестве входной переменной в одном или нескольких операторах, можно заменить его константой в зависимости от номера итерации. Во втором варианте счетчик используется в качестве выходной переменной в одном или нескольких операторах, т.е. изменяется этими операторами, и распараллелить цикл таким способом не получится. Определить возможность распараллеливания можно, вычислив функцию:

$$(I_i \oplus O_i) \wedge O_i = \{0,0,0,0\}. \quad (6)$$

В итоге функцию определения информационной зависимости итераций можно описать следующим образом:

$$F = \max(I \wedge O) \vee \max((I_i \oplus O_i) \wedge O_i). \quad (7)$$

Функция max() определяет максимальный элемент вектора, т.е. является показателем того, содержит вектор хотя бы одну единицу или нет. Если $F = 0$, итерации могут независимо друг от друга параллельно выполниться на разных процессорах. F=1 означает, что некоторые операторы используются как в качестве входных, так и в качестве выходных внутри итерации, либо счетчик цикла используется в качестве выходной переменной не только в операторе инкремента/декремента. В этом случае предложенную методику распараллеливания тела цикла применить нельзя.

Таким образом, алгоритм распараллеливания циклов по предложенному методу следующий.

1. Выделить тело цикла.
2. Вычислить векторы I, O, I_i, O_i.
3. Вычислить функцию F по формуле (5).
4. Если F=0, перейти к шагу 5, иначе — выйти из алгоритма.

5. Отделить оператор инкремента/декремента и вычислить значение счетчика цикла для каждой итерации.

6. Подставить значения счетчика, вычисленные на предыдущем шаге, в операторы тела цикла.

7. Выполнить цикл.

Сложность программных вычислений в предложенных алгоритмах достаточно высока, поэтому они могут быть взяты за основу при построении специализированного вычислительного устройства, обеспечивающего распараллеливание циклических участков в то время, когда основные процессоры заняты выполнением своих непосредственных задач. Таким образом, получим выигрыш по времени выполнения циклов.

Работа выполнена в рамках гранта Президента РФ для поддержки ведущих научных школ НШ-2357.2014.8.

СПИСОК ЛИТЕРАТУРЫ

1. *Воеводин В. В.* Параллельные вычисления. СПб: БХВ-Петербург, 2002. 608 с.

2. *Цилькер Б. Я.* Организация ЭВМ и систем: Учеб. для вузов. СПб: Питер, 2004. 668 с.

3. *Трахтенгерц Э. А.* Введение в теорию анализа и распараллеливания программ ЭВМ в процессе трансляции. М.: Наука, 1981. 254 с.

4. *Ткачев П. Ю.* Способы выявления параллелизма внутри циклических участков последовательных программ // „Инновация-2014“: Сб. матер. II регионального науч.-техн. семинара. Курск: ЮЗГУ, 2014.

5. *Ткачев П. Ю.* Способы выявления параллелизма внутри циклических участков последовательных программ // Изв. ЮФУ–ДонНТУ. Матер. 15-й Междунар. науч.-практ. конф. „Практика и перспективы развития партнерства в сфере высшей школы“. Кн. 1. Таганрог: ЮФУ, 2014. № 14. С. 165—169.

Сведения об авторах

Павел Юрьевич Ткачев — аспирант; Юго-Западный государственный университет, кафедра вычислительной техники, Курск; E-mail: amdathlon64@yandex.ru

Дмитрий Борисович Борзов — канд. техн. наук, доцент; Юго-Западный государственный университет, кафедра вычислительной техники, Курск; E-mail: borzovdb@kursknet.ru

Рекомендована Юго-Западным государственным университетом

Поступила в редакцию 10.09.14 г.