

МЕТОД СНЯТИЯ ИНФОРМАЦИОННОЙ ЗАВИСИМОСТИ ВО ВЛОЖЕННЫХ ЦИКЛАХ

Корой В.В., Борзов Д.Б. (Юго-западный государственный университет
г.Курск, Россия)

Тел./Факс: 8 (4712) 58-71-12; E-mail: wkoroy@gmail.com

Abstract: The article presents data on the method of parallelization of sequential programs to areas of communications management for execution on multiprocessor systems.

Keywords: parallelization of sequential programs, parallelizing compilation methods of loop parallelization

В связи с развитием параллельной вычислительной техники стала актуальной задача распараллеливания последовательных участков программ [1], в частности циклов. Не менее важной задачей является и разработка методов алгоритмов распараллеливания последовательных программ для последующей аппаратной реализации [2].

Рассмотрим состав участка программы, а конкретно цикла, исполняемого в однопроцессорной вычислительной системе Фон-Неймановской архитектуры [4]. Как правило, любой цикл (кроме бесконечных циклов) имеет 2 составляющих: условие цикла и тело цикла. При известной области определения переменной-счетчика, определяющего цикл, его можно разложить на последовательные блоки. Если известно, что цикл повторяется максимум N раз, то при условии, что переменная меняется только по определенному закону (без влияния тела цикла), а значения переменных (кроме счетчика для условия) не зависят от предыдущих итераций, то тело цикла можно разложить на последовательные информационно-независимые участки [3]. Для примера приведем следующий участок программы:

```
1. i=0;
2. while(i<40)
{
3.  a=x*r;
4.  d=7+c-2*a+i;
5.  i++;
}
```

Представление этого участка как последовательность операций имеет следующий вид:

```
1. i=0;
2. if(i<40)
{
3.  a=x*r;
4.  d=7+c-2*a+i;
5.  i++;
}
при i=1;
6. if(i<40)
{
7.  a=x*r;
8.  d=7+c-2*a+i;
9.  i++;
}
...
при i=39;
```

```

10. if(i<40)
{
11. a=x*r;
12. d=7+c-2*a+i;
13. i++;
}

```

Чтобы описать используемые в этом участке входные переменные для каждой операции, введем бинарную матрицу входных переменных I (таблица 1), номер строки в которой соответствует номеру операции, а номер столбца – одна из переменных участка программы. Наличие единицы на пересечении строки i и столбца j матрицы означает, что переменная из столбца j используется как входная для оператора i, а наличие 0 означает обратное. Для описания выходных переменных на участке программы введем бинарную матрицу O (таблица 2), номер строки в которой соответствует номеру операции, а номер столбца – одна из переменных участка программы. Наличие единицы на пересечении строки i и столбца j матрицы означает, что переменная из столбца j используется как выходная для оператора i, а наличие 0 означает обратное. Введем также бинарную матрицу F [5] (таблица 3), в которой номер строки i и номер столбца j являются номерами операторов, а наличие единицы на пересечении i и j, будет значить зависимость выполнения оператора i от оператора j, а 0 – независимость этих операторов.

Таблица 1 - Матрица I входных переменных для участка

№ оператора	i	a	x	r	d	c
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	0	1	1	0	0
4	1	1	0	0	1	1
5	1	0	0	0	0	0

В результате расчета по формуле [5]

$$F = (I_m \wedge O_n) \vee (I_n \wedge O_m) \vee (O_m \wedge O_n) \quad (1),$$

где I_m , O_n – это матрицы входных и выходных переменных соответственно (I_m – матрица из таблицы 1, O_n матрица из таблицы 2), получаем результат — матрицу независимых операторов F, представленную в таблице 3.

Таблица 2 - Матрица O

№ оператора	i	a	x	r	d	c
1	1	0	0	0	0	0
2	0	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	0	0	1	0
5	1	0	0	0	0	0

Таблица 3 - Матрица F

№ оператора	1	2	3	4	5
1	1	1	0	1	1
2	1	1	0	0	1
3	1	1	1	1	0
4	1	1	1	1	1
5	1	1	1	1	1

Таким образом (исходя из данных таблицы 2 ,отображающих матрицу F) операторы (1,3);(2,4),(2,3),(3,5) могут исполняться параллельно, т.к. являются независимыми друг от друга. В этом случае операции будут выполняться в порядке, приведенном в таблице 4, т.е. работу можно распределить между двумя процессорами — операторы 1,2,5 исполнены на первом процессоре, а 3,4 — на другом, за 3 шага. При этом известно, что тело цикла исполниться 40 раз, при этом каждая итерация будет происходить с увеличенным на 1 значением переменной i .

Таблица 4 - Параллельные операции цикла

1. $i=0$;	3. $a=x*r$;
2. $\text{while}(i<40)$	4. $d=7+c-2*a+i$;
	5. $i++$;

Если этот алгоритм представить в виде последовательного, с количеством независимых блоков равным 40, то получается следующий набор операторов:

1. $i1=0$;
2. $a1=x1*r1$;
3. $d1=7+c1-2*a1+i1$;
1. $i2=1$;
2. $a2=x2*r2$;
3. $d2=7+c2-2*a2+i2$;

1. $i39=39$;
2. $a39=x39*r39$;
3. $d39=7+c39-2*a39+i39$;

В этом преобразованном в последовательный участке программы, состоящей из 40 блоков, операторы между блоками независимы. Исходя из этого количество параллельных ветвей $36*3=108$, что в 36 раз больше чем в случае обычных циклов , как рассчитано в данном примере.

Таким образом, при наличии участков кода с вложенными и не вложенными циклами, с телом цикла не влияющим на количество проходов, а также при известном количестве проходов цикла, становится возможным преобразовать группу последовательных тела цикла в параллельные блоки по методу [2], что приведет к более оптимальному задействованию многопроцессорной системы в таких вычислениях.

Список литературы: 1. Воеводин В.В., Воеводин Вл.В., Параллельные вычисления // БХВ– Петербург. – Санкт-Петербург, 2002.– 608 с. 2. Дюбрюкс С.А. Метод, алгоритм и устройство обеспечения распараллеливающей компиляции последовательных программ для вычислительных систем: дис. ... канд. технич. наук.:05.13.05: Курск. Гос. Технич. университет, Курск , 2010.-С.34. 3. Трахтенгерц Э.А. Введение в теорию анализа и распараллеливания программ ЭВМ в процессе трансляции. – М.: Наука, 1981. – С. 184-187. 4. Цилькер, Б.Я. Организация ЭВМ и систем [Текст]: учебник для вузов / Б.Я. Цилькер, С.А. Орлов. СПб.: Питер, 2004. 668 с. 5. Корой, В.В., Борзов Д.Б. Выявление параллелизма внутри линейных участков последовательных программ, содержащих рекурсию и вызовы подпрограмм, со связями по управлению.//:Материалы четырнадцатого международного научно-практического семинара. г. Донецк, 15-18 апреля 2013 г. В 3-х томах. Т. 2 – Донецк: ДонНТУ, 2013. - С. 18-19.