СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

11 - 9219

О Ен Ир, В.П.Шириков

ОБ ОДНОМ ОПТИМИЗИРУЮЩЕМ ВАРИАНТЕ ОБРАБОТКИ ЦИКЛОВ ПРИ ТРАНСЛЯЦИИ С ЯЗЫКА ФОРТРАН НА ЭВМ БЭСМ-6

1975

ВВЕДЕНИЕ

При программировании на языке ФОРТРАН часто встречающейся проблемой является использование циклов и переменных с индексами в этих циклах.

До сих пор эксплуатировавшийся на БЭСМ-6 транслятор с ФОРТРАНа (Дубна) не мог оптимально программировать циклы. Вследствие этого при составлении некоторых программ сами их авторы должны были применять особые способы программирования для сокращения счетного времени рабочих программ. Например, при программировании задачи, где ведется работа с матрицами, для того, чтобы сократить счетное время рабочих программ, необходимо преобразовать двух- или трехмерные массивы в одномерные /9/. Как правило, это значительно усложняет процесс составления программ. С учетом такой ситуации был реализован описываемый ниже вариант оптимизирующего транслятора с ФОРТРАНа (Дубна). Использование этого варианта облегчает программирование для людей, стремящихся получить эффективные программы, и повышает производительность машины БЭСМ-6.

Постановка задачи

При переводе с алгоритмических языков на объектные языки проблемы оптимизации сводятся к следующим /4,5/:

1) устранение излишних вычислений;

2) выбор общих подвыражений, образование из них последовательностей промежуточных операторов присваивания;

подстановка вместо выбранных общих подвыражений соответствующих им промежуточных переменных в выражения;

3) вынесение из цикла операторов или их частей, не зависящих от цикла;

4 рационализация вычислений индексных функций для определения относительных адресов в массивах переменных с индексами, находящихся внутри цикла, и эффективное использование индексных регистров машины.

При разработке транслятора с ФОРТРАНа (Дубна) в алгоритме транслятора проблема 2 решается для каждого отдельного оператора /2,3,7/. Это делается блоком PRØPI , который производит первичную обработку арифметических выражений. Такой способ оптимизации представляет собой машинонезависимую оптимизацию /5/.

Если программист определенным образом напишет исходную программу, то нужды в решении проблем 1 -3 может не оказаться в процессе трансляции. Но проблема 4 мало связана с процессом программирования пользователем, пишущим на ФОРТРАНе. В основном она связана с алгоритмом транслятора, выбранного разработчиками системы матобеспечения на ЭВМ. Поэтому пользователь в своей программе не может полностью предопределить эффективность при счете.

Таким образом, мы можем сказать, что процесс решения проблемы 4

представляет наибольшую сложность при оптимизации программы. Такая оптимизация, как правило, является машинозависимой.

В настоящей работе мы рассматриваем один вариант (ОРТ-0) оптимизации в трансляторе для решения проблемы 4) и оценку выходных программ, полученных через варианты транслятора, оптимизирующего и не оптимизирующего программу вычисления цикла для задач, написанных на ФОРТРАНе.

В предлагаемом варианте (ОРТ-0) предполагается, что во всех случаях значения параметров цикла DO практически не ограничены. Иными словами, предполагается, что значения параметров в операторе DO на ФОРТРАНе могут быть любым неотрицательным [illegible] используемого на машине БЭСМ-6 диапазона чисел, считывая, что стандартный язык ФОРТРАН не содержит никаких явных ограничений на максимальное значение параметров цикла DO /6/.

Часто возникает такая ситуация, когда необходимо [illegible] на машине БЭСМ-6 по программы (например, биолистинг), [illegible] написаны для других машин, имеющих более длинные [illegible] по сравнению с БЭСМ-6, и максимальное значение параметра цикла выбрано в таких программах в соответствии с этой длиной. Поэтому в предлагаемом варианте транслятора в случаях, когда по крайней мере один из параметров цикла в операторе DO представляет собой переменную, для проверки повторений цикла используется управляющая переменная, а не индексный регистр.

Оптимизация [illegible] процесса повторения тела цикла DO применяется к самому внутреннему циклу.

Процесс оптимизации

Здесь мы описываем процесс реализации оптимизации, связывая его с составлением таблиц .

В неоптимизированном процессе трансляции применяется пооператорный перевод в ØUTLIST -код /2,7/. Однако в процессе оптимизации все операторы, которые находятся в теле самого внутреннего DO -цикла, не переводятся сразу в ØUTLIST -код, а буферизуются в PILIST -коде, являющемся промежуточным языком более высокого уровня, чем ØUTLIST -код.

Список N I=M1,M2,M3 оператора DO переводится на таблицу DØLIST , которая отличается от таблицы DØLIST , получаемой в случае, когда оптимизация не применяется /7/. Ее формат следующий:

9	I5	9	I5
b_1	M_1	000	N
b_2	M_2	010	I
b_3	M_3	020	DN

, где

$$b_i = \begin{cases} 000 : M_j = \text{"константа"} \ (\leq 32767), \\ 200 : M_j = \text{"константа"} \ (> 32767), \\ 400 : M_j = \text{"переменная"}; \end{cases}$$

$$N_j = \begin{cases} M_j : M_j = \text{"переменная"}, \\ \{M_j\} : M_j = \text{"константа"}, \end{cases}$$

DN - номер оператора DO,

$\{M_j\}$ - адрес для M_j .

Для проверки условий оптимизации по PILIST -записи составляются некоторые таблицы и устанавливаются флаги.

Прежде всего, для проверки наличия передачи управления из цикла за его пределы составляется таблица, которая состоит из меток операторов, передающих управление и принимающих его. Если обнаружена передача управления изнутри цикла за его пределы, то оптимизации программирования данного цикла не будет: в этом случае по правилам грамматики фортрановского языка возможна передача управления извне цикла внутрь этого цикла /I/, а это значит, что такой DO -цикл не является в процессе счета замкнутым и содержимое его индексных регистров может быть испорчено после передачи управления за пределы такого цикла.

Для проверки возможных случаев переопределения управляющей переменной и параметров (M2,M3) цикла, если они являются переменными, составляется таблица, которая состоит из простых целых переменных присваивания (т.е. левых частей операторов присваивания) тела DO - цикла.

Составляется также таблица, содержащая необходимую информацию, характеризующую индексные функции в теле DO -цикла (при этом используются общая таблица индексных функций- IFLIST , см. /7/ -обрабатываемой программы, управляющая переменная и таблица простых целых переменных присваивания). Такую таблицу назовем характеристической таблицей индексных функций. Ее формат следующий:

3	3	3	I5	9	I5
TF	TD	S	IFN	0-0	RNF

, где

TF =
- 0 : инвариантная глобальная индексная функция (индексные переменные не меняются в цикле),
- 1: переменная глобальная индексная функция (индексные переменные не являются управляющей переменной, а могут быть переменной присваивания в цикле),
- 2: локальная индексная функция (зависит только от управляющей переменной),
- 4: полулокальная индексная функция (зависит и от управляющей переменной, и от переменных, не меняющихся в цикле).

Тип функции для элемента массива в зависимости от способа описания размерностей соответствующего массива:

TD =
- 1 : A(ℓ), A(L)
- 2 : A(ℓ,m), A(ℓ,M)
- 3 : A(ℓ,m,n), A(ℓ,m,N)
- 4 : A(L,m), A(L,M)
- 5 : A(ℓ,M,n), A(ℓ,M,N)
- 6 : A(L,m,n), A(L,m,N)
- 7 : A(L,M,n), A(L,M,N)

ℓ,m,n – константы, L,M,N – переменные.

S =
- 4 : I-ая индексная переменная – управляющая переменная,
- 2 : 2-ая индексная переменная – управляющая переменная,
- I : 3-я индексная переменная – управляющая переменная.

Предусмотрены и случаи, когда, например, и первая, и вторая индексные переменные совпадают с управляющей (S = 6) и т.п.

IFN – номер индексной функции или адрес индексной переменной (для переменной с одним индексом, коэффициент при котором равен I).

RNF – относительный адрес в таблице IPLIST .

Оптимизация производится в тех случаях, когда число различных локальных и полулокальных индексных функций не больше 6 и нет инвариантных и переменных глобальных индексных функций либо когда число различных локальных и полулокальных индексных функций не больше 5, если глобальные функции имеются.

В случае, когда условие оптимизации удовлетворено, генерируются команды начальной части DO -цикла.

Если нет необходимости переопределения *управляющей переменной* в теле цикла DO и имеется резервный индексный регистр (т.е. имеется возможность использовать индексный регистр для числа повторений цикла), то в случае, когда все параметры цикла используются как константы, число повторений цикла вычисляется в процессе трансляции. Если число повторений цикла больше 32767, то использование индексного регистра для числа повторений цикла невозможно, так что в этом случае для построения цикла применяется способ переопределения значения управляющей переменной по каждому шагу повторения.

С использованием характеристической таблицы индексных функций генерируются команды, посылающие начальные значения индексных функций в индексные регистры. В это же время на базе общей характеристической таблицы составляется характеристическая таблица локальных и полулокальных индексных функций.

Составляется маска, содержащая номера индексных функций, значения которых засылаются в индексные регистры в начальной части. Эта маска обеспечивает связь между индексными функциями и индексными регистрами.

С использованием характеристической таблицы локальных и полулокальных индексных функций и таблицы индексных функций составляется таблица, содержащая необходимую информацию для вычисления приращений индексных функций. Эта таблица называется таблицей приращений индексных функций. Формат таблицы приращений и соответствующие ему коэффициенты (множители) шага управляющей переменной в приращениях индексных функций следующие.

TD	S	Таблица приращений C	ID	N	Коэффициенты шага упр.пер.
⩾1	4	0	Y_1	0	Y_1
2∨3	2	0	Y_2	0	Y_2
3	1	0	Y_3	0	Y_3
4∨6∨7	2	2	0	IFN	≡K⟨IFN⟩
	6	2	Y_1	IFN	Y_1+ ≡K⟨IFN⟩
⩾5	1	1	0	IFN	≡M⟨IFN⟩
	5	1	Y_1	IFN	Y_1+ ≡M⟨IFN⟩
5	2	0	$X_2 \equiv Y_2$	0	$X_2 \equiv Y_2$
	3	1	$X_2 \equiv Y_2$	IFN	$X_2 \equiv Y_2$+ ≡M⟨IFN⟩
	6	0	$Y_1+X_2 \equiv Y_2$	0	$Y_1+X_2 \equiv Y_2$
	7	1	$Y_1+X_2 \equiv Y_2$	IFN	$Y_1+X_2 \equiv Y_2$+≡M⟨IFN⟩
⩾6	3	3	0	IFN	≡K⟨IFN⟩+≡M⟨IFN⟩
	7	3	Y_1	IFN	Y_1+≡K⟨IFN⟩+≡M⟨IFN⟩
		3	6 15	9 15	

Здесь X_i – элемент 2-го столбца и i -ой строки в IFLISTe ,

Y_i – элемент 3-го столбца и i-ой строки в IFLISTe [7];

$$Q = \begin{cases} 0 : \text{все размерности постоянные,} \\ 1 : \text{2-ая размерность переменная,} \\ 2 : \text{1-ая размерность переменная,} \\ 3 : \text{2-ая и 1-ая размерности переменные.} \end{cases}$$

ID – указывает часть, независимую от переменных размерностей в коэффициенте управляющей переменной;

N – указывает номер индексных функций;

*K(IFN)– идентификатор-адрес составляющей (для индексной функции), зависящей только от первой переменной размерности.

*M(IFN) – идентификатор составляющей, зависящей от второй (или первой и второй) переменной размерности.

Эта таблица показывает, что коэффициент управляющей переменной в зависимости от значения Q следующий:

$$CV = \begin{cases} (ID) & : Q=0, \\ (ID) + (*M(N)) & : Q=1, \\ (ID) + (*K(N)) & : Q=2, \\ (ID) + (*K(N)) + (*M(N)) & : Q=3. \end{cases}$$

В случае необходимости на основе таблицы приращений с учетом шага управляющей переменной цикла генерируются команды, которые вычисляют приращения индексных функций. При этом в таблице приращений соответствующие этим функциям строки стираются; таким образом, получается видоизмененная таблица приращений.

Далее составляются команды, соответствующие внутренним операторам тела цикла. При этом, если в них встречаются переменные с индексами, то при генерации команд обращения к таким переменным используется ранее составленная маска связи.

Затем генерируются команды, соответствующие концевой части цикла. Сюда относятся команды, которые добавляют приращения к соответствующим регистрам, используя вышеописанную таблицу приращений. Эти приращения вычисляются в процессе трансляции или выполнения рабочей программы.

Наконец, в зависимости от того, была или не была сделана посылка числа повторений цикла в индексный регистр, генерируются команды для проверки конца повторений цикла.

Анализ и сравнительная оценка неоптимизированной и оптимизированной объектных программ

1) Анализ неоптимизированной объектной программы.

Для определения позиций переменных с индексами, используемых внутри DO -цикла, на каждом шаге цикла производится обращение к программной части, определяющей значения всех индексных функций в данной программе, которые связаны с управляющей переменной рассматриваемого DO -цикла, и посылка значений необходимых индексных функций в индексные регистры.

Для повторения цикла безусловно переопределяется значение управляющей переменной цикла. Ясно, что при этом теряется тем больше машинного времени (при счете), чем больше количество индексных функций и число повторений цикла. При оптимизации программы такой недостаток ликвидирован.

2) Анализ оптимизированной объектной программы

Для вычисления начальных значений индексных функций в теле DO -цикла выполняется обращение к части вычисления индексных функций, связанных с управляющей переменной, и производится посылка необходимых значений индексных функций в индексные регистры. Число

повторений цикла (точнее, – ([(M2-M1)/M3] + 1)), которое вычисляется в процессе трансляции, посылается в индексный регистр. В необходимом случае делаются вычисления приращений индексных функций. Такой процесс есть процесс выполнения начальной части DO -цикла, и он производится только один раз для данного цикла.

Если в теле цикла есть переменные с индексами, то необходимые операции по определению позиции переменных в массивах выполняются с использованием соответствующих регистров.

В завершающей части цикла приращения индексных функций добавляются к соответствующим регистрам и проверяется, требуется ли повторение цикла, и если да, то производится переход к началу тела цикла.

3) Сравнительная оценка

Рассмотрим программу умножения матриц в качестве типичного примера, связанного с применением циклов.

```
      PRØGRAM MUMAT
      DIMENSIØN A(50,50),B(50,50),C(50,50)
          .
          .
          .
      DØ 10 I=1,50
      DØ  5 J=1,50
      C(I,J)=0.
      DØ   1 K=1,50
    1 C(I,J)=C(I,J)+A(I,K)*B(K,J)
    5 CØNTINUE
   10 CØNTINUE
          .
          .
          .
      END
```

(I) Неоптимизированный случай MUMAT

При прохождении начальной части самого внутреннего DO -цикла выполняется 27 команд.

При прохождении остальной части (тело и завершающая часть) требуется выполнение 44 команд.

(2) Оптимизированный случай MUMAT

При прохождении начальной части DO -цикла выполняются 34 команды.

При выполнении остальной части на каждом шаге работает только II команд.

Отсюда следует, что начиная со второго шага при оптимизации число выполняемых на каждом проходе цикла команд сокращено на I/4.

Важно также, что те типы команд, которые имеют сравнительно невысокое быстродействие, в оптимизированном варианте программы находятся в начальной части цикла, а в неоптимизированном варианте - в основной части.

Счетное время для MUMAT в неоптимизированном варианте - 7,04 с , а в оптимизированном - I,50 с (время трансляции и загрузки не учитывается).

В заключение авторы выражают сердечную благодарность Пак Хон Черу за помощь и поддержку, Г.Л.Мазному, И.Н.Силину, Н.С.Заикину, за помощь и замечания, Н.Н.Говоруну, Г.А.Ососкову, В.Г.Иванову, А.И.Салтыкову, А.А.Хошенко за поддержку.

ЛИТЕРАТУРА

I. Язык ФОРТРАН (под редакцией В.П.Ширикова), ОИЯИ, II-4818, Дубна, 1969.

2. З.Броццински и др. Транслятор с языка ФОРТРАН для системы математического обеспечения БЭСМ-6. Труды Первой всесоюзной конференции по программированию, г.Киев, 1968. Изд. ИК АН УССР.

3. О Ен Ир, В.П.Шириков. Новые возможности использования логических выражений при программировании на языке ФОРТРАН для ЭВМ БЭСМ-6. ОИЯИ, II-7396, Дубна, 1973.

4. С.Я.Виленкин, Э.А.Трахтенгерц. Математическое обеспечение управляющих вычислительных машин. Из-во "Энергия", М,1972.

5. Дж.Донован. Системное программирование (пер. с англ. под редакцией Л.Д.Райкова), Изд-во "Мир", М,1975.

6. А.И.Волков. Автокод МАДЛЕН. ОИЯИ, Б4-II-4654, Дубна, 1969.

7. З.Броццински, Н.Н.Говорун и др. Система математического обеспечения ЭВМ БЭСМ-6. Транслятор с языка ФОРТРАН (части I,II,III,IV). ОИЯИ, БI-II-7I60, БI-II-7I62, Дубна, 1972.

8. USA Standard FORTRAN . Издание ANSI, Нью-Йорк, 1966.

9. В.И.Карначук. Язык ФОРТРАН (Оптимизация и большие задачи). Препринт ВЦ СОАН СССР, Новосибирск, 1974.

Рукопись поступила в издательский отдел
9 октября 1975 г.