

ОБ АВТОМАТИЗАЦИИ ПРОГРАМИРОВАНИЯ ДЛЯ ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Ю.Г. Косарев
(Новосибирск)

1. При программировании для однородных вычислительных систем с программируемой структурой типа "Минск-222" нужно для каждой машины составить программу, включающую в себя операторы системы: настройки, обмена, обобщенных безусловного и условного переходов, с помощью которых достигается функционирование системы и изменяется её конфигурация [1].

В настоящее время разработана и внедрена в практику методика ручного программирования для системы "Минск-222" [2].

Решение очередной проблемы - автоматизация программирования для системы - естественно начать с попытки максимально использовать существующие алгоритмические языки и трансляторы для одиночных машин. В данной работе предлагается методика программирования для системы, которая использует имеющиеся языки и трансляторы без каких-либо изменений.

Многие системы автоматизации программирования в той или иной форме допускают включение в исходную программу блоков, записанных в кодах команд машины. Эти блоки включаются либо в виде стандартных процедур, которым соответствует библиотека стандартных программ (БСП) [3], либо непосредственно (в сопровождении специального слова) [4]. Это позволяет на общих основаниях употреблять стандартные программы, обеспечивающие функциональные системы (СП системы), не выходя за рамки возможностей

Таким образом, если имеется удовлетворительный транслятор для машин, объединяемых в систему, то проблема сводится к за-

писи исходных программ для каждой из машин системы на входном языке, используя процедуры, которые соответствуют СП системы.

Для подавляющего большинства практических задач эта проблема может быть решена путем сравнительно простых преобразований исходной программы для одиночной машины на основе методики, использующей свойства циклов [2]. Различаются три типа циклов:

н е з а в и с и м ы й, каждое повторение которого может выполняться независимо от результатов любого другого повторения цикла;

з а в и с и м ы й, в котором в любой паре повторений цикла, хотя бы одно повторение зависит от результатов другого; **ч а с т и ч н о - з а в и с и м ы й**, к которому относятся все остальные циклы;

Процесс программирования для системы может быть разбит на четыре этапа.

Э т а п 1. Составление программы для одиночной машины на выбранном алгоритмическом языке.

Э т а п 2. Выявление в программе независимых (или частично-зависимых) циклов, с числом (независимых) повторений

$n \geq 1$, охватывающих все или почти все операторы.

l - число машин в системе.

При отсутствии подобных циклов, что практически эквивалентно невозможности эффективного распараллеливания, следует заменить исходную программу другой.

Э т а п 3. Видоизменение программы. l циклов, по которым ведется распараллеливание, число повторений n заменяется для машин с номером s .

$r := \text{entier}(n/l)$; if $s \leq n - r \cdot l$ then $r := r + 1$.

Каждый из массивов, над которым выполняется данный цикл, делится на l массивов, распределяемых между машинами.

Э т а п 4. Введение стандартных процедур для компенсации того, что результаты повторений циклов получаются в различных машинах.

При описанном разделении процесса вычислений, которое называем **распараллеливанием по циклам**, достаточно составить для всех машин системы одну программу. Более того, так как l задается в виде параметра, то данная программа пригодна для любого $1 \leq l$.

2. Иногда выгодно применять схему, при которой между равными по номеру машинами распределяются различные блоки программы, входящие в один независимый (или частично-зависимый) цикл и каждая машина, как правило, обрабатывает данные, полученные в предыдущей по номеру машине (конвейерная схема). Данная схема особенно удобна, когда каждый блок удается выполнить при небольшом числе обращений к вспомогательным памяти и все блоки имеют примерно одинаковое время работы. При конвейерной схеме этап 3 состоит в распределении блоков исходной программы между машинами. При этом исходные программы для машин системы различны, но их сумма, как и в предыдущем случае, отличается от исходной программы для одиночной машины, только системными процедурами.

Практически также выглядит случай, когда некоторые машины работают только при запросе от других машин или извне (режим спящего).

3. Описанные выше схемы распараллеливания базируются на одном важном свойстве - процесс вычислений делится на крупные блоки, определяемые структурой решаемой задачи и применяемым алгоритмом.

Согласованность процесса распараллеливания с особенностями задачи собственно и позволяет сравнительно просто получить эффективную программу для системы.

Данный способ распараллеливания, который назовем крупноблочным, в принципе не универсален. Однако при рассмотрении большого числа специально неотобранных типов задач, пока не удалось обнаружить ни одной задачи с достаточно большим объемом вычислений, для которых он оказался бы неприемлемым.

Подчеркнем, что речь идет не о конкретном алгоритме, а о задаче. Для каждой задачи существует, как правило, множество методов решения, каждый из которых может реализоваться множеством эквивалентных по результатам алгоритмов. Достаточно найти среди эквивалентных алгоритмов хотя бы один, удовлетворяющий требованиям крупноблочного распараллеливания.

Другая методика, основанная на локальных свойствах алгоритма, выраженных в граф-схеме между операторами, позволяет сравнительно просто при заданном быстродействии машин установить минимальное время реализации данного алгоритма [5]. Однако нахождение оптимального варианта распределения операторов между за-

данным числом машин на основе этой методики оказалось весьма сложной задачей. Пока нет других способов её решения, кроме перебора большого числа вариантов, что требует больших затрат времени и не гарантирует отыскание оптимального варианта.

4. Список ОП системы, который нужно включить в БСП, должен удовлетворять следующим естественным требованиям: быть полным, т.е. с его помощью должно быть выполнено любое взаимодействие машин в системе; содержать часто встречающиеся подпрограммы; пополняться в процессе эксплуатации.

В заключение укажем набор стандартных процедур, обеспечивающих полноту, предварительно уточнив основные характеристики системы^{*)}. Будем рассматривать линейную систему, в которой машины взаимодействуют друг с другом с помощью линий связи. Каждая машина соединяется с линией связи с помощью коммутатора. Коммутаторы разбивают линию связи на отрезки. Управляет каждым коммутатором разряд R регистра настройки (РН). При $R = 1$ прерывается линия связи, идущая к машине с большим номером, и система разбивается на изолированные подсистемы. С помощью разрядов РН Q и Q выделяются машины, участвующие соответственно в выполнении команд системы обобщенного безусловного перехода (ОБП) и обобщенного условного перехода (ОУП).

Как указано в [1], вычислительная система обладает полнотой, если в ней могут реализоваться операторы настройки, обмена, обобщенных безусловного и условного переходов. В соответствии с этим, введем четыре следующие процедуры:

1) Настройка (T, N, H).

При $T = 0,2$ изменяется содержимое РН данной машины,

при $T = 1,3$ изменяется содержимое РН машин, номера которых указаны N. РН данной машины не изменяется,

при $T \leq 1$ $RH := H$,

при $T > 1$ $RH := RH \vee H$,

H - информация для изменения РН,

$N = L_1, L_2, \dots, L_{12}, L_1 = 0, 1$.

при $L_1 = 1$ и $T = 1,3$ происходит изменение РН машины с номером 1.

2) Обмен (T, K, A)

$T = 0$ - передача в канал связи K кодов, начиная с A.

$T = 1$ - прием из канала связи K кодов и размещение их, начиная с A.

^{*)}Основные характеристики совпадают с системой "Минск-222" [6]

3) Обобщенный безусловный переход ОБП (Т, Р)

Во все машины, у которых $Q = 1$, засылается оператор Р, который при $T = 1$ должен выполняться немедленно, при $T = 0$ - после окончания очередной операции.

4) Обобщенный условный переход ОУП(Т, L)

При $T = 0$ все машины, принадлежащие той же подсистеме и отмеченные по Ω , ждут друг друга и одновременно приступают к выполнению следующей за ОУП операции (синхронизация).

При $T = 1, 2, 3$ те же машины после синхронизации вырабатывают обобщенный признак $\Omega = \bigcap \omega_{ik}$. Здесь ω_{ik} признак ω_k , вырабатываемый машиной с номером i ; $k = 1, 2, 3$.

При $\Omega = 1$ управление передается по метке L,

при $\Omega = 0$ - следующему оператору.

Указанным набором процедур можно пользоваться для обозначения всех системных процедур. Однако для многих конкретных задач целесообразно расширить список СП системы, записанных на машинном языке, включив в него, например, одновременный обмен информацией между соседними по номеру машинами, засылку подпрограммы, поочередные передачи информации из каждой машины во все остальные и т.п.

ЛИТЕРАТУРА

1. Э.В.Евреинов, Ю.Г.Косарев. Однородные универсальные системы высокой производительности, Новосибирск, "Наука", Сибирское отделение, 1966 г.
2. Ю.Г.Косарев. Распараллеливание по циклам.- Сб."Вычислительные системы", Новосибирск, Изд-во "Наука", Сиб.отд., вып. 24, 1967, с.3-20.
3. Н.В.Шкут. Автоматизация программирования для вычислительной машины "Минск-2(22)", Минск, Изд-во "Наука и техника", 1967.
4. М.Е.Неменман, В.И.Цегальский, И.М.Матюшевская. Автокод для решения инженерных задач на машине "Минск-2", Минск, изд-во ИНТИП, 1965.
5. E.S.Schwartz. An automatic sequencing procedure with application to parallel programming. - J.Assoc.Compt.Machinery, 1961, vol.8, # 4, 513-537.
6. Э.В.Евреинов, Г.П.Лопато. Универсальная вычислительная система "Минск-222".- Вычислительные системы, Новосибирск, Изд-во "Наука", Сиб.отд., 1966, вып.23, с.13-20.