

БУЛЕВЫ МОДЕЛИ СИНТЕЗА ПАРАЛЛЕЛЬНЫХ ПЛАНОВ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ *

Предложен новый подход к построению параллельных асинхронных планов требуемой длины для решения вычислительных задач в мультипроцессорной среде. Условия задачи планирования сформулированы в виде системы булевых уравнений (ограничений), решения которой определяют возможные планы активизации программных модулей. Учитываются ограничения на число используемых узлов мультипроцессорной вычислительной среды, временные задержки, возникающие при выполнении программных модулей, а также множественность установки модулей в узлах.

Ключевые слова: планирование и синтез параллельных планов, булевы модели, ограничения и уравнения.

Введение

Прежде всего, введем рабочее определение мультипроцессорной вычислительной среды (МВС), под которой здесь и далее будем понимать совокупность объединенных сетью вычислительных узлов, каждый из которых, возможно, работает под управлением собственной операционной системы, а расчетные программные модули, установленные в узлах, возможно, имеют своего владельца. Таким образом, мы абстрагируемся от таких характеристик МВС, как используемое коммуникационное оборудование, способ взаимодействия узлов (обмен сообщениями, общая разделяемая память или иные средства), тип адресного пространства (единое или множественное). Введенное определение МВС вполне достаточно для концептуального (схемного) уровня синтеза программ из модулей, где требуются лишь знания о вычислительной модели предметной области и о некоторых ресурсных ограничениях. На этом уровне выполняется синтез параллельного плана решения задачи (или абстрактной программы, по определению С. С. Лаврова). Преобразование абстрактной программы в конкретную с учетом знаний о специфических возможностях аппаратного и программного обеспечения МВС выполняется на процедурном уровне и в данной статье не рассматривается.

Итак, нас интересуют декларативные языки и системы параллельного (в том числе распределенного) модульного программирования [Плакс, 1997; Опарин, Феоктистов, 2002], которые позволяют накапливать в памяти ЭВМ знания о вычислительных модулях предметной области и использовать эти знания при автоматическом решении задач заданного класса. Эффективное использование этой технологии требует решения трудной проблемы построения параллельных планов решения задач. Описания взаимосвязанных по данным программных модулей, включая их распределение по узлам МВС, размещаются в базе знаний планировщика. Допускается множественность установки модулей в узлах МВС.

Необходимо определить, какие модули, в какой последовательности и в каких узлах МВС должны быть выполнены для вычисления требуемого набора целевых параметров B_0 по заданному множеству параметров – исходных данных A_0 . Цель – получить параллельный план решения вычислительной задачи $T = (A_0, B_0)$, который обладает следующими основными свойствами:

* Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 06-01-00340).

- 1) допустимости (модули в плане должны быть упорядочены таким образом, чтобы каждый из них к моменту своего запуска был обеспечен необходимыми входными данными, или, иначе говоря, для любого входного параметра модуля, входящего в план, должен ранее встречаться в плане хотя бы один модуль с одноименным выходным параметром);
- 2) неповторности (каждый модуль может включаться в план не более одного раза);
- 3) неизбыточности (исключение любого модуля из плана приводит к недопустимому плану);
- 4) оптимальности (длина плана должна быть равна заданной величине k).

При этом мы предполагаем, что при построении параллельного плана с перечисленными свойствами число узлов МВС может быть ограничено некоторой величиной; возможны временные задержки, связанные с исполнением программных модулей и зависящие от производительности узла, в котором установлен модуль; время передачи данных между узлами не учитывается; в каждом узле МВС доступен хотя бы один программный модуль; каждый модуль установлен хотя бы в одном узле. При этом допускается случай установки нескольких функционально различных модулей в одном узле, но в каждый момент времени в таком узле может выполняться только один программный модуль.

В данной постановке задача планирования является NP-трудной, ее условия могут быть представлены в виде системы булевых уравнений (ограничений), а собственно план исполнения модулей является решением этой системы. Преимущество булева моделирования (по сравнению с традиционным дедуктивным подходом) состоит в том, что оно позволяет, во-первых, получать параллельные планы требуемой длины, во-вторых, учитывать разнообразные ограничения, предъявляемые к плану, и, наконец, в-третьих, использовать существующие эффективные решатели булевых уравнений (или SAT-решатели), которые в ряде случаев обгоняют по быстродействию специализированные алгоритмы планирования. В целом, как отмечается в [Kautz, Selman, 1992], такой подход не только обеспечивает более гибкую среду для установления различных видов ограничений на параллельные планы, но также более точно отражает последние достижения в построении систем планирования, основанных на ограничениях.

База знаний планировщика

В качестве базы знаний планировщика используется вычислительная модель $KB = (F, Z, Y, In, Out, Com)$, где $F = \{F_1, \dots, F_n\}$ – множество программных модулей, работающих над полем общих транзитных данных $Z = \{Z_1, \dots, Z_m\}$, являющихся входными или выходными параметрами этих модулей; $Y = \{Y_1, \dots, Y_p\}$ – множество узлов МВС; $In \subset F \times Z$, $Out \subset F \times Z$, – отношения, отражающие взаимосвязь модулей с данными соответственно по входу и выходу. Таким образом, с каждым модулем F_i связано два множества параметров $A_i, B_i \subset Z$, называемых соответственно входом и выходом. Вход A_i определяет данные, которые необходимо иметь, чтобы получить результаты, представленные выходом B_i . В дальнейшем для этого используем запись $F_i(A_i, B_i)$. Без потери общности будем считать, что модули F_1 и F_2 из множества F моделируют условия постановки задачи планирования $T = (A_i, B_i)$, а именно: база знаний KB содержит модули $F_1(A_0)$ и $F_2(B_0)$, где $A_0, B_0 \subset Z$. Отсутствие атрибута до или после точки с запятой означает пустоту соответствующего множества. Модуль F_1 назовем модулем начальных данных, а модуль F_2 – целевым модулем. Отношение $Com \subset F \times Y$ (в общем случае «многие ко многим») определяет в базе знаний KB статические связи между программными модулями и узлами МВС. Предполагается, что база знаний KB обладает высоким уровнем внутреннего параллелизма и является избыточной в том смысле, что для решения задачи используется только часть модулей из F и / или задача T имеет несколько альтернативных планов решения.

Отношения In , Out и Com представим в виде трех булевых матриц A , B и C так, что элементы матриц A и B размерности $n \times m$ и матрицы C размерности $n \times p$ формируются следующим образом: $a_{ij} = 1$ ($b_{ij} = 1$), если параметр Z_j является входным (выходным) для модуля F_i ; $c_{ij} = 1$, если модуль F_i установлен в узле Y_j . Через A_i, B_i, C_i обозначим строки этих матриц, а через A'_j, B'_j, C'_j – столбцы. Строки и столбцы матриц A , B , и C являются двоичным представлением соответствующих подмножеств множеств F , Z и Y , связанных отношениями In , Out и Com . Запись $q \in S$ (S – это двоичная строка $A_i, B_i, C_i, A'_j, B'_j$ или C'_j) означает, что q принимает значения номеров единичных элементов в двоичной строке S .

Далее мы рассмотрим три практически важных варианта распределения модулей по узлам.

1. Количество узлов не ограничено в том смысле, что каждый модуль установлен в одном и только в одном узле. Такая ситуация возникает, например, при решении ресурсоемкой задачи в распределенной вычислительной среде в условиях неотчуждаемости программного модуля от владельца узла.

2. Количество узлов является ограниченной величиной, в каждом узле установлены все модули. Эта ситуация является типичной при решении вычислительной задачи на кластере выделенных рабочих станций типа МВС-1000М с использованием, к примеру, DVM-системы.

3. Общий случай множественной установки программных модулей, когда любой модуль может быть установлен сразу в нескольких узлах и в любом узле может быть доступно несколько функционально различных модулей. В первую очередь здесь идет речь о повышении отказоустойчивости решения ресурсоемкой задачи в распределенной вычислительной среде при выходе из строя одного или нескольких узлов.

Булево моделирование при неограниченном числе узлов МВС

Для начала рассмотрим случай построения базовой булевой модели [Опарин, Новопашин, 2004], когда ограничение на число узлов отсутствует, временные задержки при исполнении программных модулей не учитываются, что соответствует синтезу синхронно-параллельного плана в рамках концепции неограниченного параллелизма.

Определим план в виде матрицы X $k \times n$ булевых переменных x_{ij} . Значение x_{ij} , равное единице, означает, что программный модуль F_j выполняется на t -м шаге плана X , где $t \in N = \{1, 2, \dots\}$ играет роль дискретного времени. При такой интерпретации первого индекса переменных x_{ij} естественно предположить, что временные задержки, связанные с исполнением модулей, не учитываются, когда все они одинаковы и равны единице. Общая длина плана равна k , его строка задает множество параллельно исполняемых модулей, а столбцы соответствуют множеству модулей F . Тогда булевы ограничения на элементы матрицы X имеют следующий вид.

Условие 1. Условие постановки задачи планирования $T = (A_i, B_i)$ (модуль начальных данных $F_1(A_0)$ и целевой модуль $F_2(B_0)$) располагаются соответственно в первой и последней строках плана; другие модули в этих строках отсутствуют):

$$\bar{x}_{11} = 0, \bigvee_{j=2}^n x_{1j} = 0, \bar{x}_{k2} = 0, x_{k1} = 0, \bigvee_{j=3}^n x_{kj} = 0.$$

Условие 2. Условие непрерывности (на каждом шаге плана выполняется, по крайней мере, один модуль):

$$\bigvee_{t=2}^{k-1} \bigwedge_{j=1}^n \bar{x}_{tj} = 0.$$

Условие 3. Условие неповторности:

$$\bigvee_{j=1}^n \bigvee_{t=1}^{k-1} \bigvee_{s=t+1}^k (x_{tj} \wedge x_{sj}) = 0.$$

Условие 4. Условие допустимости:

$$\bigvee_{t=2}^k \bigvee_{j=1}^n (x_{tj} \wedge w) = 0,$$

где

$$w = \begin{cases} \bigvee_{q \in A_j} \bigwedge_{s=1}^{t-1} \bigwedge_{r \in B'_q} \bar{x}_{sr}, & \text{if } (A_j \neq 0) \wedge ((\forall q \in A_j)(B'_q \neq 0)); \\ 1, & \text{if } (A_j \neq 0) \wedge ((\exists q \in A_j)(B'_q = 0)); \\ 0, & \text{if } A_j = 0. \end{cases}$$

Условие 5. Условие упорядоченности (если подготовка данных для модуля завершена на шаге $t - 1$, то этот модуль обязательно включается в план на шаге с номером t):

$$\bigvee_{t=2}^k \bigvee_{j=1}^n (x_{tj} \wedge v) = 0,$$

где

$$v = \begin{cases} \bigwedge_{q \in A_j} \bigwedge_{r \in B'_q} \bar{x}_{t-1,r}, & \text{if } (A_j \neq 0) \wedge ((\forall q \in A_j)(B'_q \neq 0)); \\ 1, & \text{if } (A_j \neq 0) \wedge ((\exists q \in A_j)(B'_q = 0)); \\ 0, & \text{if } (A_j = 0) \wedge (t = 2); \\ 1, & \text{if } (A_j = 0) \wedge (t > 2). \end{cases}$$

Условие 6. Условие неизбыточности:

$$\bigvee_{t=2}^{k-1} \bigvee_{j=1}^n (x_{tj} \wedge \bar{u}) = 0,$$

где

$$u = \bigvee_{s=2}^k \bigvee_{r=1}^n (x_{sr} \wedge w),$$

индексы s и r удовлетворяют условию $(s = t) \wedge (t = j)$, а w определяется выражением в условии 4.

Дополнительные условия, задаваемые постановщиком задачи планирования:

$$g(x_{11}, \dots, x_{1n}, \dots, x_{k1}, \dots, x_{kn}) = 0.$$

В качестве примеров дополнительных ограничений можно привести:

– условие обязательного включения в план модуля F_j :

$$h_j = \bigwedge_{t=2}^{k-1} \bar{x}_{tj} = 0;$$

– условие отсутствия в плане модуля F_j :

$$z_j = \bigvee_{t=2}^{k-1} x_{tj} = 0;$$

– условие альтернативного включения в план модулей F_j и F_r :

$$z_j \wedge z_r \vee h_j \wedge h_r = 0.$$

Условие упорядоченности 5 позволяет существенно сократить размер поискового пространства при решении системы булевых уравнений, состоящей из условий 1–6. Исключение условия упорядоченности расширяет список возможных дополнительных условий, формируемых постановщиком задачи планирования. Отказ от условия непрерывности позволяет синтезировать планы длины не больше k .

Количество неизвестных булевых переменных (размерность задачи) в наихудшем случае равно n^2 . В случае большой избыточности множества модулей F для сокращения размерности задачи можно рекомендовать предварительную обработку базы знаний решателя традиционно применяемыми при планировании алгоритмами прямого и обратного поиска.

Булево моделирование с учетом ресурсных ограничений

В этом разделе рассмотрим ситуацию, когда все узлы одинаковы и количество их ограничено величиной p , каждый модуль может выполняться в любом узле, в каждый момент времени в узле может обрабатываться только один модуль. Время τ_j выполнения модуля $F_j \in F$ является дискретной величиной: $\tau_j \in N = \{1, 2, \dots\}$. Кроме того, задан общий директивный срок решения задачи $k \in N$. Необходимо построить асинхронный план, длина которого равна величине директивного срока k или не превышает k . Для начала запишем дополнительное условие, позволяющее строить планы для p узлов МВС без учета временных задержек, возникающих при исполнении модулей (т. е. когда $\tau_j = 1$ для всех $j = \overline{1, n}$).

Условие 7. Ограничение на число узлов МВС без учета временных задержек:

$$\bigvee_{t=2}^{k-1} \left(\bigvee_{1 \leq j_1 < j_2 < \dots < j_{p+1} \leq n} (x_{t,j_1} \wedge x_{t,j_2} \wedge \dots \wedge x_{t,j_{p+1}}) \vee \bigvee_{1 \leq j_1 < j_2 < \dots < j_{n-p+1} \leq n} (\bar{x}_{t,j_1} \wedge \bar{x}_{t,j_2} \wedge \dots \wedge \bar{x}_{t,j_{n-p+1}}) \right) = 0.$$

Для учета временных задержек, связанных с исполнением модулей в узлах МВС, требуется модифицировать представленные выше условия 2, 4, 6 и 7. Кроме того, из системы булевых ограничений необходимо исключить условие упорядоченности 5.

Условие 2.1. Условие непрерывности, учитывающее временные задержки при исполнении модулей:

$$\bigvee_{t=2}^{k-1} \bigwedge_{j=1}^n \bigwedge_{i=t-\tau_j+1}^t \bar{x}_{ij} = 0.$$

Очевидно, что при $\tau_j = 1$ для всех $j = \overline{1, n}$, условие 2.1 совпадает с условием 2.

Условие 4.1. Условие допустимости, учитывающее временные задержки при исполнении модулей:

$$\bigvee_{t=2}^k \bigvee_{j=1}^n (x_{tj} \wedge w) = 0,$$

где

$$w = \begin{cases} \bigvee_{q \in A_j} \bigwedge_{s=1}^{t-\tau_j} \bigwedge_{r \in B'_q} \bar{x}_{sr}, & \text{if } (A_j \neq 0) \wedge ((\forall q \in A_j)(B'_q \neq 0)); \\ 1, & \text{if } (A_j \neq 0) \wedge ((\exists q \in A_j)(B'_q = 0)); \\ 0, & \text{if } A_j = 0. \end{cases}$$

Условие 6.1. Для того чтобы условие избыточности 6 работало с учетом временных задержек, достаточно использовать для него новое определение W из предыдущего условия.

Условие 7.1. Ограничение на число узлов МВС, учитывающее временные задержки при исполнении модулей:

$$\bigvee_{t=2}^{k-1} \left(\bigvee_{1 \leq j_1 < j_2 < \dots < j_{p+1} \leq n} \left(\bigwedge_{i=t-\tau_{j_1}+1}^t x_{i,j_1} \wedge \bigwedge_{i=t-\tau_{j_2}+1}^t x_{i,j_2} \wedge \dots \wedge \bigwedge_{i=t-\tau_{j_{p+1}}+1}^t x_{i,j_{p+1}} \right) \vee \bigvee_{1 \leq j_1 < j_2 < \dots < j_{n-p+1} \leq n} \left(\bigwedge_{i=t-\tau_{j_1}+1}^t \bar{x}_{i,j_1} \wedge \bigwedge_{i=t-\tau_{j_2}+1}^t \bar{x}_{i,j_2} \wedge \dots \wedge \bigwedge_{i=t-\tau_{j_{n-p+1}}+1}^t \bar{x}_{i,j_{n-p+1}} \right) \right) = 0.$$

При малом числе узлов и учете временных задержек при исполнении модулей данная модель содержательно соответствует решению задачи «сжатия параллелизма» и построения асинхронных планов, позволяющих повысить загрузку вычислительных узлов.

Синтез параллельных планов решения задач для систем программирования, не поддерживающих асинхронное исполнение модулей, требует добавления в общую систему ограничений специального условия.

Условие 8. Условие синхронности, учитывающее временные задержки при исполнении модулей (параллельный запуск модулей возможен только после полного завершения работы всех модулей предыдущего шага плана (схема FORK / JOIN)):

$$\bigvee_{t=2}^k \bigvee_{j=1}^n (x_{tj} \wedge e) = 0,$$

где

$$e = \bigvee_{r=1}^n \left(\bar{x}_{tr} \wedge \bigvee_{i=t-\tau_r+1}^t x_{ir} \right).$$

Булево моделирование при множественной установке модулей в узлах МВС

Наконец, рассмотрим общий случай связей между программными модулями и узлами, который допускает, что узлы различаются по производительности, каждый модуль может быть установлен в нескольких узлах, в любом узле может находиться несколько функционально различных модулей, но в каждый момент времени на каждом узле может выполняться только один модуль. Таким образом, матрица C , представляющая отношение Com , не содержит нулевых строк и столбцов, а количество единичных элементов в ней равно l .

Определим план решения задачи в виде матрицы X $k \times l$ булевых переменных $x_{t, \langle j, r \rangle}$, в которых второй индекс элемента матрицы определяется двумя величинами – номером модуля j в множестве F и номером узла r в множестве Y , удовлетворяющими условию $c_{jr} = 1$. Будем считать, что $x_{t, \langle j, r \rangle} = 1$ означает исполнение модуля F_j в узле Y_r на шаге плана X с номером t . Время исполнения модуля обозначим через $\tau_{\langle j, r \rangle}$. Понятно, что для данного варианта распределения модулей по узлам булевы ограничения, полученные в предыдущих разделах, нуждаются в модификации и дополнении. В частности, уточненная редакция условия неповторности имеет вид: каждый модуль вне зависимости от того, в каком количестве узлов он установлен, может включаться в план не более одного раза.

Условие 3.1. Условие неповторности с учетом множественности установки модулей в узлах МВС:

$$\bigvee_{j=1}^n \left(\bigvee_{(r, s \in C_j) \wedge (1 \leq r < s \leq p)} (h_r \wedge h_s) \vee \bigvee_{q \in C_j} h_q \right) = 0,$$

где

$$h_r = \bigvee_{t=1}^k x_{t, \langle j, r \rangle}, \quad h_s = \bigvee_{t=1}^k x_{t, \langle j, s \rangle}, \quad h_q = \bigvee_{t=1}^{k-1} \bigvee_{i=t+1}^k (x_{t, \langle j, q \rangle} \wedge x_{i, \langle j, q \rangle}).$$

Следующие дополнительные булевы ограничения контролируют на любом шаге t плана X исполнение не более одного программного модуля на каждом узле МВС (условие захвата узла модулем):

Условие 9. Условие захвата узла модулем без учета временных задержек при исполнении модулей:

$$\bigvee_{t=1}^k \bigvee_{q=1}^p \bigvee_{(j_1, j_2 \in C'_j) \wedge (1 \leq j_1 < j_2 \leq n)} (x_{t, \langle j_1, q \rangle} \wedge x_{t, \langle j_2, q \rangle}) = 0.$$

Условие 9.1. Условие захвата узла модулем с учетом временных задержек при исполнении модулей:

$$\bigvee_{t=1}^k \bigvee_{q=1}^p \bigvee_{(j_1, j_2 \in C'_j) \wedge (1 \leq j_1 < j_2 \leq n)} \left(\bigvee_{i=t-\tau_{\langle j_1, q \rangle}+1}^t x_{i, \langle j_1, q \rangle} \wedge \bigvee_{i=t-\tau_{\langle j_2, q \rangle}+1}^t x_{i, \langle j_2, q \rangle} \right) = 0.$$

Несложная редакция условий, связанная со специфической индексацией столбцов матрицы X , позволяет в полном объеме представить булевы ограничения при множественной установке модулей в узлах МВС.

Заключение

Представлен новый подход к построению как синхронных, так и асинхронных параллельных планов решения непроцедурных задач на вычислительной модели предметной области для мультипроцессорных вычислительных сред. При этом мы предполагали, что существуют ограничения на длину плана, число узлов МВС и связи между программными модулями и узлами, имеются временные задержки, связанные с исполнением программных модулей в узлах, а время передачи данных между узлами не учитывается.

Для решения полученных в статье систем булевых уравнений мы используем эффективные авторские алгоритмы и программы [Опарин, Богданова, 2001], реализующие технику как хронологического, так и интеллектуального возврата и использующие эвристики распространения булевых ограничений, трехзначные и многозначные логики и ориентированные на общее представление булевых функций в левой части уравнений.

Применение известных SAT-решателей¹ требует приведения булевых функций общего вида к дизъюнктивной нормальной форме, что, как правило, является трудной задачей в классической постановке [Журавлев, Платоненко, 1984].

Разработанные модели и алгоритмы синтеза параллельных планов решения вычислительных задач показали свою эффективность в инструментальных средах организации распределенных вычислений [Опарин, Феоктистов, 2002] и крупноблочного синтеза параллельных программ с использованием DVM-системы [Опарин, Новопашин, 2006].

Список литературы

Журавлев Ю. И., Платоненко И. М. Об экономном умножении булевых уравнений // ЖВМ и МФ. 1984. Т. 24, № 1. С. 164–166.

Опарин Г. А., Богданова В. Г. Алгоритмы решения больших разреженных систем булевых уравнений // Методы оптимизации и их приложения: Тр. XII Байкальской междунар. конф. (Иркутск, июнь 2001 г.). Секция 5: Дискретная математика. Иркутск: ИГУ, 2001. С. 114–118.

Опарин Г. А., Новопашин А. П. Булево моделирование планирования действий в распределенных вычислительных системах // Изв. РАН. Теория и системы управления. 2004. № 5. С. 105–108.

Опарин Г. А., Новопашин А. П. Методы и инструментальные средства крупноблочного синтеза параллельных программ на языке Fortran-DVM // Параллельные вычисления и задачи управления: Избранные доклады III Междунар. конф. PACO-2006 памяти И. В. Прангишвили (Москва, октябрь, 2006). М.: ИПУ РАН, 2006. С. 59–63.

Опарин Г. А., Феоктистов А. Г. Инструментальная распределенная вычислительная САТУРН-среда // Программные продукты и системы. 2002. № 2. С. 27–30.

Плакс Т. П. Синтез параллельных программ на вычислительных моделях // Программирование. 1977. № 4. С. 55–63.

Kautz H., Selman B. Planning as Satisfiability // Proc. 10th European Conf. on Artificial Intelligence (ECAI), 1992. P. 359–363.

Материал поступил в редколлегию 22.03.2008

G. A. Oparin, A. P. Novopashin

Boolean Models for Synthesis of Parallel Plans of Solving Computational Problems

A new approach to constructing parallel asynchronous plans of a required length for solving computational problems in the multiprocessor environment is proposed. Conditions of the planning problem are represented as a system of Boolean equations (constraints). Solutions of such system determine possible plans of starting of the program modules. Constraints on number of nodes of multiprocessor environment, time delays at execution of program modules and plurality of their installation in nodes are taken into consideration.

Keywords: planning and synthesis of parallel plans, Boolean models, constraints and equations.

¹ Simon L. The experimentation web site around the satisfiability problem. <http://www.lri.fr/~simon/satex/satex.php3>.