

Составление многопроцессорного расписания в системе реального времени с ограничениями на связи между процессорами

М.Г. Фуругян¹

¹ФИЦ ИУ РАН, Москва, Россия, rtscas@yandex.ru

Аннотация. Рассматривается задача построения допустимого расписания комплекса программных модулей в многопроцессорной системе реального времени с неполным графом связей между процессорами. Заданы длительности выполнения модулей и их директивные интервалы. При выполнении модулей допускаются прерывания и переключения с одного процессора на другой с учетом ограничений на связи между процессорами и на число переключений в фиксированный момент времени. Параллельное выполнение нескольких модулей одним процессором и одного модуля несколькими процессорами не допускается. Решение задачи основано на ее сведении к многопродуктовой потоковой задаче в сети.

Ключевые слова: многопроцессорная система, допустимое расписание, директивный интервал, многопродуктовый поток в сети, линейная система булевых соотношений

1. Введение

Публикации по вычислительным системам реального времени в нашей стране и за рубежом стали появляться в семидесятых годах прошлого века. В это же время эти системы начали разрабатываться и использоваться для нужд гражданского строительства и в военном деле. Наиболее широкое распространение системы реального времени получили в тех областях, где необходимо было решать задачи планирования вычислений при жестких временных ограничениях. Так, в конце семидесятых годов прошлого века руководством авиационной промышленности СССР перед предприятиями отрасли была поставлена задача сократить сроки испытания самолетов с пяти до полутора лет. Одним из элементов решения этой задачи являлось создание вычислительной системы реального времени для проведения летных экспериментов. Стояла задача обработки в режиме реального времени большого объема информации, поступающей с борта самолета с частотой 50 герц с тем, чтобы успевать передавать летчику корректирующие установки выполнения полета. Подобного рода задачи возникали и возникают в настоящее время и в других областях, как например, при разработке системы космической обороны, при решении сложных логистических задач, в гражданском и военном строительстве, при проектировании и эксплуатации объектов энергетики, в частности, атомных электростанций, при разработке систем экологического, экономического и технического мониторинга различных объектов, во многих других областях.

При создании многопроцессорных вычислительных систем реального времени необходимо решить широкий круг задач, связанных с построением такой структуры связей между процессорами и определением таких производительностей последних, которые обеспечивали бы выполнение определенных вычислений при строгих временных ограничениях. Кроме того, необходимо установить, в какие моменты времени и на каких устройствах нужно выполнять эти вычисления.

Одно из основных требований, предъявляемых к таким системам, заключается в том, что все вычисления должны производиться при заданных временных ограничениях. Для решения этой задачи могут быть использованы различные методы построения оптимальных расписаний [1 – 3]. В [4] рассмотрена задача оптимизации выполнения комплекса работ с нефиксированными длительностями, для решения которой применяется метод ветвей и границ и построении многогранников устойчивости решений. В [5 – 7] разработана методика проверки выполнения ограничений реального времени, заключающихся в том, что каждая работа должна выполняться в заданном директивном интервале. Проведенные исследования выполнены для многоадресной вычислительной системы реального времени и основаны на построении имитационной модели с использованием обобщенных конечных автоматов с остановкой таймера. С помощью этой модели строится временная диаграмма работы системы, позволяющая осуществить непосредственную проверку выполнения ограничений

реального времени. В [8] предложен псевдополиномиальный алгоритм решения задачи построения оптимального по быстродействию расписания исполнения заданий с логическими условиями предшествования. В этой задаче для каждого задания дан список его непосредственных предшественников, а также число завершенных непосредственных предшественников, необходимое для начала его выполнения. Задача сведена к циклической игре. В [9] для решения задач построения оптимальных расписаний с прерываниями в многопроцессорных системах реального времени предложен новый подход, основанный на модификации потоковых алгоритмов в сети. В [10] исследована задача минимизации времени выполнения комплекса непрерываемых заданий. Предложен алгоритм, сочетающий в себе жадную схему с процедурой ограниченного перебора, которая вызывается в случаях, когда жадный выбор не может привести к оптимальному решению.

В настоящей статье рассматривается многопроцессорная система реального времени с неполным графом связей между процессорами. Разработан алгоритм, позволяющий для заданной совокупности программных модулей, допускающих прерывания и переключения, с известными характеристиками (длительности и директивные интервалы) определить, существует ли допустимое расписание их выполнения и определить его в случае положительного ответа. Дополнительно задается ограничение на число прерываний и переключений в каждый момент времени. Алгоритм основан на сведении исходной задачи к многопродуктовой потоковой задачи в сети.

2. Постановка задачи

Имеется p идентичных процессоров, занумерованных от 1 до p , с помощью которых требуется выполнить m программных модулей, занумерованных от 1 до m . Модули выполняются во временном отрезке $[0; T]$, который разделен на T отрезков единичной длины $I_k = [t_{k-1}; t_k]$, $k = \overline{1, T}$, $t_0 = 0$, $t_T = T$. Длительность выполнения i -го модуля составляет τ_i , а его директивный интервал — $[b_i; f_i]$ (т.е. он может начать выполняться не ранее момента времени b_i и должен завершиться не позднее момента времени f_i), $i = \overline{1, m}$. Не допускается параллельное выполнение одного модуля несколькими процессорами. В каждом интервале I_k , $k = \overline{1, T}$, одним процессором может выполняться не более одного модуля.

Между процессорами существует связь, позволяющая переключать выполнение модулей с одного процессора на другой. Эта связь задается

двумерным массивом $C(j_1; j_2)$, $j_1, j_2 = \overline{1, p}$, элементы которого принимают значения 0 или 1. При этом $C(j_1; j_2) = 1$, если при выполнении модуля процессором j_1 в некотором интервале I_{k_1} допускается переключение этого модуля в момент времени t_{k_1} на процессор j_2 , которым он будет выполняться в некотором интервале I_{k_2} ($k_1 < k_2$).

Такое переключение модуля требует суммарных временных затрат процессоров j_1 и j_2 , равных d . Число переключений в каждый момент времени t_k не должно превосходить заданного числа r_k , $k = \overline{1, T-1}$. Предполагается, что $\tau_i, T, d \in \mathbb{N}$, $b_i, f_i \in \{t_0, t_1, \dots, t_T\}$, $i = \overline{1, m}$, $\mathbb{N}$ — множество натуральных чисел.

Требуется определить, существует ли допустимое расписание выполнения m программных модулей и найти его в случае положительного ответа. Допустимое расписание — это такое расписание, при котором:

- 1) суммарное время выполнения процессорами каждого модуля $i = \overline{1, m}$ (за вычетом временных издержек на переключения) составляет τ_i ;
- 2) все переключения выполняются в соответствии с массивом C ;
- 3) не нарушаются ограничения по числу переключений в моменты t_k , $k = \overline{1, T-1}$;
- 4) каждый модуль $i = \overline{1, m}$ выполняется в своем директивном интервале $[b_i; f_i]$.

Как известно [1], такая задача является NP-трудной. Мы сведем ее решение к многопродуктовой потоковой задаче с дополнительными ограничениями.

3. Алгоритм решения задачи

Построим m -продуктовую потоковую сеть $H = (V, A)$ (см. рисунок), где V — узлы, A — ориентированные дуги. Множество V состоит из следующих узлов:

s_i, v_i — соответственно источник и сток i -го продукта;

x_{kj}, y_{kj} — соответственно начало и окончание выполнения некоторого модуля в интервале I_k процессором j .

Множество A состоит из следующих дуг:

(s_i, x_{kj}) — начало выполнения модуля i процессором j в интервале I_k , $i = \overline{1, m}$, $j = \overline{1, p}$, $k = \overline{b_i, f_i}$;

(x_{kj}, y_{kj}) — выполнение некоторого модуля процессором j в интервале I_k , $k = \overline{1, T}$, $j = \overline{1, p}$;

(y_{kj}, v_i) — завершение выполнения модуля i процессором j в интервале I_k , $i = \overline{1, m}$, $j = \overline{1, p}$, $k = \overline{b_i, f_i}$;

$(y_{k_1j_1}, x_{k_2j_2})$ – переключение выполнения некоторого модуля с процессора j_1 в момент времени t_{k_1} на процессор j_2 в момент времени t_{k_2} , $j_1, j_2 = \overline{1; p}$, $C(j_1; j_2) = 1$, $k_1, k_2 = \overline{1; T}$, $k_1 < k_2$. Отметим, что при $j_1 = j_2$, $k_1 = k_2$ переключения не происходит и временные издержки отсутствуют. Издержки на переключение происходят только при $j_1 \neq j_2$.

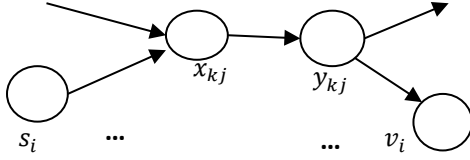


Рисунок. Фрагмент потоковой сети H

Пропускная способность $U(a, b)$ каждой дуги сети H полагается равной единице: $U(a, b) = 1$, $(a, b) \in A$.

В сети H будем рассматривать m -продуктовый поток $g_i(a, b)$, $i = \overline{1; m}$, $(a, b) \in A$, принимающий значения 0 или 1 и обладающий следующими свойствами:

$$\sum_{k=b_i}^{f_i} \sum_{j=1}^p g_i(s_i, x_{kj}) = 1, i = \overline{1; m}, \quad (1)$$

т.е. из источника s_i исходит ровно одна единица i -го продукта;

$$\sum_{k=b_i}^{f_i} \sum_{j=1}^p g_i(y_{kj}, v_i) = 1, i = \overline{1; m}, \quad (2)$$

т.е. в сток v_i входит ровно одна единица i -го продукта.

Единицу потока i -го продукта по дуге (s_i, x_{kj}) будем рассматривать как начало выполнения модуля i в момент времени t_{k-1} процессором j . Далее, модуль i выполняется процессором j в интервале I_k .

Единицу потока i -го продукта по дуге (y_{kj}, v_i) будем рассматривать как завершение выполнения модуля i в момент времени t_k .

Единица потока i -го продукта по дуге (x_{kj}, y_{kj}) означает выполнение i -го модуля в интервале I_k процессором j , в результате чего к суммарному времени его выполнения прибавляется единица.

И наконец, единица потока i -го продукта по дуге $(y_{k_1j_1}, x_{k_2j_2})$ означает переключение выполнения i -го модуля с процессора j_1 в момент времени t_{k_1} на процессор j_2 в момент времени t_{k_2} ,

в результате чего из суммарного времени его выполнения вычитается величина d .

Таким образом, к соотношениям (1), (2) следует добавить следующие соотношения:

$$\sum_{i=1}^m g_i(x_{kj}, y_{kj}) \leq 1, k = \overline{1; T}, j = \overline{1; p}, \quad (3)$$

т.е. в каждом интервале I_k одним процессором выполняется не более одного модуля;

$$\sum_{k=b_i}^{f_i} \sum_{j=1}^p g_i(x_{kj}, y_{kj}) - \sum_{k_1, k_2 = b_i, f_i, k_1 < k_2} \sum_{j_1, j_2 = \overline{1; p}, C(j_1; j_2) = 1} g_i(y_{k_1j_1}, x_{k_2j_2}) = \tau_i, i = \overline{1; m}, \quad (4)$$

т.е. суммарная длительность выполнения модуля i за вычетом затрат на переключения составляет τ_i ;

$$\begin{aligned} & g_i(s_i, x_{kj}) + \sum_{k_1 = b_i, f_i, k_1 < k} \sum_{j_1 = \overline{1; p}, j_1 \neq j, C(j_1; j) = 1} g_i(y_{k_1j_1}, x_{kj}) \\ & = g_i(x_{kj}, y_{kj}), j = \overline{1; p}, k \\ & = b_i, f_i - \end{aligned} \quad (5)$$

сохранение потока i -го продукта в узлах x_{kj} ;

$$\begin{aligned} & g_i(x_{kj}, y_{kj}) = \sum_{k_1 = b_i, f_i, k_1 > k} \sum_{j_1 = \overline{1; p}, j_1 \neq j, C(j; j_1) = 1} g_i(y_{kj}, x_{k_1j_1}) \\ & + g_i(y_{kj}, v_i), j = \overline{1; p}, k \\ & = b_i, f_i - \end{aligned} \quad (6)$$

сохранение потока i -го продукта в узлах y_{kj} ,

$$g_i(a, b) = 0 \vee 1, i = \overline{1; m}, (a, b) \in A. \quad (7)$$

Таким образом, исходная задача о составлении допустимого расписания сведена к системе булевых ограничений (1) – (7), в которой $O(m p T^2)$ переменных $g_i(a, b)$, $i = \overline{1; m}$, $(a, b) \in A$, и $O(m + p T)$ линейных соотношений. А именно, допустимое расписание в исходной задаче существует в том и только том случае, когда система (1) – (7) имеет решение. Если $g_i(x_{kj}, y_{kj}) = 1$, то модуль i выполняется процессором j в интервале I_k . Если $g_i(y_{k_1j_1}, x_{k_2j_2}) = 1$, то модуль i после выполнения процессором j_1 в интервале I_{k_1} переключается на процессор j_2 , которым он будет выполняться в интервале I_{k_2} .

6. Заключение

В данной работе рассмотрена задача построения допустимого расписания комплекса программных модулей в многопроцессорной системе с идентичными процессорами и неполным графом связей между ними. Считаются заданными длительности выполнения модулей и их директивные интервалы. При выполнении модулей допускаются прерывания и переключения с одного процессора на другой с учетом ограниче-

ний на связи между процессорами и на число переключений в фиксированный момент времени. Учитываются временные издержки при выполнении переключений. Параллельное выполнение нескольких модулей одним процессором и одного модуля несколькими процессорами не допускается. Задача сведена к линейной системе булевых соотношений. Решение основано на использовании теории многопродуктовых потоков в сети.

Creation of a Multiprocessor Schedules in Real-Time System with Communication Restrictions Between Processors

Meran Furugyan

Abstract. The problem of constructing an admissible schedule for a complex of program modules in a multiprocessor real-time system with an incomplete graph of connections between processors is considered. The duration of the modules execution and their directive intervals are set. When executing modules, interrupts and switching from one processor to another are allowed, taking into account the restrictions on communications between processors and on the number of switches at a fixed point in time. Parallel execution of several modules by one processor and one module by several processors is not allowed. The solution of the problem is based on its reduction to a multicommodity flow problem in the network.

Keywords: multiprocessor system, admissible schedule, directive interval, multiproduct flow in the network, linear system of Boolean relations

Литература

1. В.С. Танаев, В.С. Гордон, Я.М. Шафранский. Теория расписаний. Одностадийные системы. М.: Наука, 1984, 383 с.
2. P. Brucker. Scheduling Algorithms. Heidelberg: Springer, 2007, 378 с.
3. А.А. Лазарев. Теория расписаний. Оценка абсолютной погрешности и схема приближенного решения задач теории расписаний. М.: МФТИ, 2008, 222 с.
4. А.В. Мищенко, П.С. Кошелев. Оптимизация управления работами логистического проекта в условиях неопределенности. «Изв. РАН. ТиСУ», (2021), № 4, 86 – 101.
5. А.Б. Глонина, В.В. Балашов. О корректности моделирования модульных вычислительных систем реального времени с помощью сетей временных автоматов. «Моделирование и анализ информационных систем», Т. 25 (2018), № 2, 174 – 192.
6. А.Б. Глонина. Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем. «Вестник ЮУрГУ. Сер. Вычисл. математика и информатика», Т. 6 (2017), № 4, 43 – 59.
7. А.Б. Глонина. Инструментальная система проверки выполнения ограничений реального времени для конфигураций модульных вычислительных систем. «Вестн. МГУ. Сер. 15. Вычисл. математика и кибернетика», (2020), № 3, 16 – 29.
8. Д.В. Алифанов, В.Н. Лебедев, В.И. Цурков. Оптимизация расписаний с логическими условиями предшествования. «Изв. РАН. ТиСУ», (2009), № 6, 88 – 93.
9. В.А. Костенко, А.С. Смирнов. Алгоритм построения однопроцессорных статико-динамических расписаний. «Вестн. МГУ. Сер. 15. Вычисл. математика и кибернетика», (2018), № 1, 45 - 52
10. В.А. Костенко. Алгоритмы комбинаторной оптимизации, сочетающие жадные стратегии и ограниченный перебор. «Изв. РАН. ТиСУ», (2017), № 2, 48 - 56.