4. *Новиков Д.А.* Методология управления. – М.: Либроком, 2011. – 128 с.

5. *Новиков Д.А.* Теория управления организационными системами. – М.: Ленанд, 2022. – 500 с.

6. Теория управления (дополнительные главы) / Под ред. Д.А. Новикова. – М.: Ленанд, 2019. – 552 с.

7. *Новиков Д.А.* Исследовательские принципы теории управления организационно-техническими системами / Труды 13 Международной конференции «Управление развитием крупномасштабных систем» MLSD'2020 (Москва, 28-30 сентября 2020 года). – Москва: ИПУ РАН, 2020. – С. 79-83.

DOI: 10.25728/iccss.2022.37.22.073

Фуругян М.Г.

Распределение памяти в многопроцессорной системе реального времени с нефиксированными параметрами

Аннотация: Рассматривается задача планирования вычислений в многопроцессорной системе реального времени для случая, когда длительности выполнения прикладных модулей линейно зависят от выделенной им дополнительной памяти. Определяется минимальный объем дополнительной памяти, при котором для заданного директивного срока существует допустимое расписание, а также минимальный директивный срок, для которого существует допустимое расписание при заданном объеме дополнительной памяти. Получены аналитические формулы для указанных величин.

Ключевые слова: многопроцессорная система реального времени, оптимальное расписание, нефиксированные параметры

Вычислительные системы реального времени находят применение в тех случаях, когда за короткий промежуток времени требуется выполнить определенные расчеты, иногда достаточно большого объемы. Такие задачи возникают при проектировании, испытаниях и функционировании сложных технических объектов, как например, самолетов и ракет, электростанций, транспортных систем, конвейерных производств, во многих других областях. Вычислительные системы реального времени позволяют производить нужные расчеты к заданным срокам, что позволяет повысить безопасность работы указанных объектов.

Одна из основных задач вычислительных систем реального времени заключается в построении расписаний выполнения прикладных модулей, при котором каждый модуль успевает завершиться к установленному заранее директивному сроку. В работах [1-3] исследованы различные постановки таких задач и предложены методы их решения. В [4-6] задача планирования вычислений для многоядерной вычислительной системы реального времени решается с использованием обобщенных конечных автоматов и построенной на их основе имитационной модели. С помощью этой модели строится временная диаграмма, описывающая функционирование системы и позволяющая осуществить непосредственную проверку того, что каждая работа выполняется в заданном директивном интервале.

В настоящей работе рассматривается задача планирования вычислений в многопроцессорной системе реального времени для случая, когда длительности выполнения прикладных модулей линейно зависят от выделенной им дополнительной памяти. Кроме того, определяется минимальный объем дополнительной памяти, при котором для заданного директивного срока существует допустимое расписание, а также минимальный директивный срок, для которого существует допустимое расписание при заданном объеме дополнительной памяти. Получены аналитические формулы для указанных величин.

Постановка задачи. Требуется выполнить n программных модулей $W = \{1, 2, \ldots, n\}$ на m идентичных процессорах во временном интервале $[0; T]$. Все модули $i \in W$ допускают при их выполнении прерывания и переключения с одного процессора на другой. При этом временными издержками на прерывания и переключения можно пренебречь. Помимо процессоров каждому модулю $i \in W$ может быть выделена дополнительная память объемом v_i, которая закрепляется за этим модулем и другими модулями использоваться не может. Суммарный объем дополнительной памяти равен V. При этом должны выполняться следующие ограничения

$$v_i \in [0; v_i^0], \; i = \overline{1, n}, \tag{1}$$

$$\sum_{i \in W} v_i \le V. \tag{2}$$

Длительность выполнения модуля i оставляет

$$t_i = t_i^0 - a_i v_i, \tag{3}$$

т.е. линейно зависит от объема выделенной этому модулю дополнительной памяти. Здесь v_i^0, t_i^0 и a_i – заданные положительные величины; v_i^0 – это максимальный объем памяти, который может быть выделен модулю i, t_i^0 – это длительность выполнения модуля i в случае, когда дополнительная память ему не выделяется. Предполагается, что $t_i^0 - a_i v_i^0 > 0$, $t_i^0 \le T$, $v_i^0 \le V$ при всех $i \in W$.

Требуется определить распределение дополнительной памяти, удовлетворяющее соотношениям (1), (2), и найти расписание, при котором каждый модуль $i \in W$ успевает полностью выполниться в директивном интервале $[0; T]$ (такое расписание будем называть допустимым), либо показать, что такого расписания не существует. Кроме того, необходимо найти а) минимальный объем дополнительной памяти $V_{\min}$, обеспечивающий для заданного директивного срока T существование допустимого расписания; б) минимальную величину директивного срока $T_{\min}$, при котором для заданного объема дополнительной памяти V допустимое расписание существует.

Решение задачи. В [1] доказано, что в рассматриваемом случае необходимым и достаточным условием существования допустимого расписания является выполнение неравенства

$$\sum_{i \in W} t_i \le mT. \tag{4}$$

В [1] также представлен алгоритм построения допустимого расписания для данного случая (алгоритм упаковки), имеющий линейную вычислительную сложность. Из (3), (4) следует, что в рассматриваемой постановке для существования допустимого расписания необходимо и достаточно существования распределения памяти v_i, $i \in W$, удовлетворяющего соотношениям (1), (2) и неравенству

$$\sum_{i \in W} (t_i^0 - a_i v_i) \le mT. \tag{5}$$

Перепишем (5) в следующем виде

$$\sum_{i \in W} a_i v_i \ge \sum_{i \in W} t_i^0 - mT. \tag{6}$$

Из неравенства (6) видно, что в первую очередь нужно выделять максимально возможный объем памяти модулям $i \in W$ с наибольшими значениями коэффициентов a_i с тем, чтобы минимизировать длительности их выполнения. Упорядочим работы W по не возрастанию величин a_i, $i \in W$, и пусть $a_1 \ge a_2 \ge \ldots \ge a_n$. Распределим память между модулями W по следующему простому алгоритму.

Алгоритм 1 (распределение памяти).

1) Инициализация: $i = 0$.

2) Вычислить: $i = i + 1$, $v_i = \min(v_i^0, V)$, $V = V - v_i$.

3) Если выполнены неравенства $V > 0$ и $i < n$, то перейти на шаг 2.

4) Если $V = 0$ или $i = n$, алгоритм завершает работу.

В результате работы этого алгоритма будут определены объемы памяти v_i и длительности t_i всех модулей $i \in W$, вычисленные согласно (3). В случае выполнения неравенства (6) допустимое расписание существует и строится с помощью алгоритма упаковки [1]. Если неравенство (6) нарушено, то это означает, что при данных входных параметрах допустимого расписания не существует. Предложенный алгоритм распределения невозобновляемого ресурса, а также алгоритма упаковки имеют вычислительную сложность $O(n)$.

Для определения минимального объема памяти $V_{\min}$, при котором для заданного директивного интервала $[0; T]$ допустимое расписание существует, предлагается следующий алгоритм (алгоритм 2).

Алгоритм 2.

1) Если в случае, когда при $v_i = v_i^0$, $i = \overline{1,n}$, неравенство (6) не выполнено, то допустимого расписания не существует ни при каком значении V.

2) Если при $v_i = v_i^0$, $i = \overline{1,n}$, неравенство (6) выполнено, то обозначим через i_0 минимальный номер, при котором $\sum_{i=1}^{i_0-1} a_i v_i^0 < A$, $\sum_{i=1}^{i_0} a_i v_i^0 \geq A$, где $A = \sum_{i \in W} t_i^0 - mT$

3) Если $\sum_{i=1}^{i_0} a_i v_i^0 = A$ то $V_{\min} = \sum_{i=1}^{i_0} v_i^0$.

4) Если $\sum_{i=1}^{i_0} a_i v_i^0 > A$, то $V_{\min} = \sum_{i=1}^{i_0-1} v_i^0 + \dfrac{A - \sum_{i=1}^{i_0-1} a_i v_i^0}{a_{i_0}}$.

Определим теперь минимальный директивный срок $T_{\min}$, для которого при заданном значении R существует допустимое расписание. Из (6) следует, что

$$T_{\min} = \frac{1}{m} \min_{r_1,\ldots,r_n} \left(\sum_{i \in W} t_i^0 - \sum_{i \in W} a_i v_i \right) = \frac{1}{m} \left(\sum_{i \in W} t_i^0 - \max_{r_1,\ldots,r_n} \sum_{i \in W} a_i v_i \right) \qquad (7)$$

при выполнении соотношений (1), (2). Как было показано выше, максимум в (7) достигается при значениях $v_1, \ldots, v_n$, вычисляемых с помощью алгоритма 1. Отметим, что в [7] рассмотренные в данном разделе задачи были сведены к задачам линейного программирования с $\theta(n)$ переменными и $\theta(n)$ ограничениями.

Алгоритмы решения подобных задач являются более трудоемкими, чем предложенные в настоящей работе.

Литература:

1. *Танаев В.С., Гордон В.С., Шафранский Я.М.* Теория расписаний. Одностадийные системы. – М.: Наука, 1984. – 384 с.
2. *Brucker P.* Scheduling Algorithms. – Heidelberg, Springer, 2007 – 371 p.
3. *Лазарев А.А.* Теория расписаний. Оценка абсолютной погрешности и схема приближенного решения задач теории расписаний. – М.: МФТИ, 2008. – 222 с.
4. *Глонина А.Б., Балашов В.В.* О корректности моделирования модульных вычислительных систем реального времени с помощью сетей временных автоматов // Моделирование и анализ информационных систем. – 2018. – Т. 25. № 2. – С. 174-192.
5. *Глонина А.Б.* Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем // Вестник ЮУрГУ. Сер. Вычисл. математика и информатика. – 2017. – Т.6. № 4. – С. 43-59.
6. *Глонина А.Б.* Инструментальная система проверки выполнения ограничений реального времени для конфигураций модульных вычислительных систем // Вестник МГУ. Сер. 15. Вычисл. математика и кибернетика. – 2020. – № 3. – С. 16-29.
7. *Косоруков Е.О., Фуругян М.Г.* Некоторые алгоритмы распределения ресурсов в многопроцессорных системах. // Вестник МГУ. Сер. 15. Вычисл. математика и кибернетика. – 2009. – № 4. – С. 34-37.