УДК 519.687.5:004.451.44

М.В. Кавалеров, Н.Н. Матушкин

Пермский национальный исследовательский
политехнический университет

ВОЗМОЖНОСТЬ СУЩЕСТВЕННОГО ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПЛАНИРОВАНИЯ НА ОСНОВЕ НЕПОСРЕДСТВЕННОГО ПРИМЕНЕНИЯ ЛИНЕЙНЫХ ИНТЕРВАЛЬНЫХ ОГРАНИЧЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ

Если исходные ограничения задач жесткого реального времени не являются стандартными (не выражаются через период и относительный крайний срок), то тогда обычно их пытаются преобразовать в стандартные ограничения, чтобы использовать хорошо известные методы планирования с фиксированными приоритетами, ориентированные именно на стандартные ограничения. На небольшом примере показано, что такой подход может существенно уступать подходу, основанному на непосредственном применении линейных интервальных ограничений реального времени в процессе планирования с фиксированными приоритетами.

В системах автоматизации и управления для отдельного вычислительного устройства (ВУ) часто необходимо осуществлять планирование задач жесткого реального времени (РВ), что предполагает разделение процессорного времени между этими задачами при условии соблюдения ограничений РВ. Широкое распространение получила модель планирования с фиксированными приоритетами (ПФП) (Fixed Priority Scheduling) [1].

Модель ПФП. Рассмотрим формальное описание модели ПФП. Планирование выполняется для одного процессора. Есть множество из n задач жесткого РВ, обозначаемое $\{\tau_i\}$. Каждая задача τ_i формирует запросы. Каждый j-й запрос, обозначаемый $\tau_{i,j}$, формируется в момент времени $r_{i,j}$ и сразу же помещается в очередь запросов. Пусть $C_{i,j}$ обозначает процессорное время, необходимое для выполнения $\tau_{i,j}$. Для каждой *периодической* задачи τ_i при $\forall j \geq 1$ справедливо соотношение

$$r_{i,j} = r_{i,j-1} + T_i = O_i + (j-1)T_i, \tag{1}$$

где O_i – начальное смещение; T_i – период. Для каждой *спорадической* задачи τ_i при $\forall j \geq 1$ справедливо соотношение

$$r_{i,j} \geq r_{i,j-1} + T_i, \tag{2}$$

где T_i – минимальный интервал формирования запросов; O_i для определенности всегда равно 0; каждое значение $r_{i,j}$ задается временем наступления некоторого внешнего события. Кроме того, имеется поток апериодических запросов, для планирования которых используется время, свободное от выполнения задач $\{\tau_i\}$. Каждая задача τ_i и запросы $\{\tau_{i,j} \mid \forall j \geq 1\}$ имеют фиксированный приоритет π_i в виде целого числа из интервала $[1, n]$. Каждая задача из $\{\tau_i\}$ имеет уникальный приоритет.

Диспетчеризация – это обслуживание очереди запросов, формируемых задачами $\{\tau_i\}$, согласно приоритетам $\{\pi_i\}$. Выполнение запроса может прерываться другим запросом с более высоким приоритетом.

При обслуживании очереди ограничения РВ напрямую не учитываются. Выполнение запросов периодической задачи τ_i зависит только от O_i, T_i, π_i и состояния очереди. Множество $\{O_i, T_i, \pi_i \mid i \in [1, n]\}$ порождает возможные варианты выполнения запросов задач $\{\tau_i\}$. Различия между вариантами определяются: вариациями значений $C_{i,j}$; различными последовательностями внешних событий, влияющих на формирование запросов спорадических задач и апериодических запросов. Если в случае *каждого* из этих вариантов соблюдается ограничение РВ задачи τ_i, то при данном множестве $\{O_i, T_i, \pi_i \mid i \in [1, n]\}$ *гарантируется* соблюдение этого ограничения РВ, и задача τ_i называется *выполнимой*.

Множество $\{\tau_i\}$ называется *выполнимым*, если, во-первых, обеспечивается выполнимость каждой задачи из $\{\tau_i\}$, во-вторых, гарантируется, что размер очереди ограничен сверху конечным значением.

Планированием (в рамках концепции ПФП) будем называть процесс отыскания такого множества $\{O_i, T_i, \pi_i \mid i \in [1, n]\}$, которое обеспечивает выполнимость $\{\tau_i\}$. Планирование выполняется до старта системы, и его результатом являются либо допустимые значения параметров задач (то есть множество $\{O_i, T_i, \pi_i \mid i \in [1,n]\}$), либо информация о невозможности обеспечить выполнимость $\{\tau_i\}$.

Стандартное ограничение РВ. Стандартным ограничением (СО) жесткого РВ будем называть ограничение следующего вида для периодической задачи τ_i:

$$\begin{cases} s_{i,j} \geq O_i^* + (j-1)T_i^*, \\ f_{i,j} \leq O_i^* + (j-1)T_i^* + D_i, \end{cases}, \tag{3}$$

где $s_{i,j}$ – момент начала выполнения запроса $\tau_{i,j}$; $f_{i,j}$ – момент завершения выполнения запроса $\tau_{i,j}$; O_i^*, T_i^* – параметры ограничения; D_i – относительный крайний срок задачи τ_i. Предполагается, что $O_i^* \geq 0$, $T_i^* > 0$, а также, что условие (3) должно соблюдаться при $\forall j \geq 1$. В работе [2] отмечено, что в случае СО (для периодической задачи) предполагается установка параметров задачи O_i, T_i согласно соотношениям $O_i^* = O_i$, $T_i^* = T_i$.

Пожалуй, СО – это самое распространенное ограничение жесткого РВ, и именно на это ограничение ориентируется большинство разработанных методов планирования. Хотя обычно СО описывается другим образом, а именно как следующая совокупность условий для задачи τ_i:

1) каждое значение $r_{i,j}$ ограничено соотношением (1), когда τ_i – это периодическая задача, или соотношением (2), когда τ_i – это спорадическая задача;

2) для каждого запроса $\tau_{i,j}$ выполняется условие:

$$f_{i,j} \leq d_{i,j} = r_{i,j} + D_i, \tag{4}$$

где $d_{i,j}$ обычно называется *крайним сроком* запроса $\tau_{i,j}$ (или абсолютным крайним сроком). При этом значение $r_{i,j}$ из (4) определяется параметрами O_i, T_i согласно (1) или ограничивается параметром T_i согласно (2). Поэтому обычно говорят, что такое ограничение РВ определяется смещением, периодом и относительным крайним сроком. Проблема такого определения в том, что здесь смешиваются параметры O_i, T_i, определяющие формирование запросов, и параметр D_i, ответственный за ограничение РВ. Применение определения на основе (3) позволяет избежать этого, так как в этом случае все три параметра O_i^*, T_i^*, D_i ответственны только за ограничение РВ [2].

Также к задачам, имеющим СО, будем относить спорадические задачи. У таких задач СО выражается условием (4), то есть СО спорадических задач определяется, по сути, только одним параметром D_i.

Получение вторичных СО для решения проблемы планирования. Но далеко не все задачи имеют СО [3]. В качестве примера можно привести нестандартное ограничение задачи контура управления, определяемое условием:

$$\begin{cases} T_i^{xx\,\min} \le s_{i,j} - s_{i,j-1} \le T_i^{xx\,\max}, \\ f_{i,j} - s_{i,j} \le T_i^{xy\,\max}, \end{cases} \tag{5}$$

где $T_i^{xx\,\min}$, $T_i^{xx\,\max}$, $T_i^{xy\,\max}$ – константы; при этом заранее устанавливается значение $s_{i,0}$. Это ограничение похоже на ограничение, определенное в п. 5.2 работы [4], а при условии, что измерение и управляющее воздействие выполняются одной задачей, они становятся аналогичными.

При наличии нестандартных ограничений (НО) *базовым* подходом к планированию в условиях ПФП можно считать преобразование каждого НО в СО. В работе [5] был предложен алгоритм, который позволяет преобразовать любое НО из класса линейных интервальных ограничений (ЛИО) в СО. И полученные СО будем называть *вторичными*. В частности, в работе [6] дан пример того, как для НО, представляющего собой более общий случай НО (5), формируется условие допустимости, на основе которого можно получить вторичное СО. Для частного случая в виде ограничения (5), относящегося к классу ЛИО, это условие допустимости имеет вид:

$$\begin{cases} O_i^* \ge x_{i,0} + T_i^{xx\,\min}, \\ O_i^* + D_i - C_i \le x_{i,0} + T_i^{xx\,\max}, \\ T_i^* \ge D_i - C_i + T_i^{xx\,\min}, \\ T_i^* + D_i - C_i \le T_i^{xx\,\max}, \\ D_i \le T_i^{xy\,\max}, \end{cases} \tag{6}$$

где C_i – время выполнения любого запроса $\tau_{i,j}$ задачи τ_i при отсутствии прерывания его выполнения. В нашем примере для упрощения будем предполагать, что все запросы одной задачи одинаковы по длительности. Условие (6) означает, что если СО (3), к которому преобразуется НО (5), удовлетворяет условию (6), то тогда соблюдение СО (3) гарантирует соблюдение НО (5). То есть соблюдение вторичного СО гарантирует соблюдение *исходного* НО.

Если в системе задач жесткого РВ среди исходных ограничений имеются только СО, а также НО из класса ЛИО, то все НО можно преобразовать во вторичные СО с помощью алгоритма из работы [5] и получить в итоге систему задач жесткого РВ, имеющих только СО. После этого, используя алгоритм из работы [7], легко можно решить проблему планирования.

Но недостаток такого базового подхода состоит в том, что существенным образом снижается эффективность использования ВУ. Поясним это на следующем примере.

Пример. Пусть на одном процессоре данного ВУ выполняются две задачи жесткого РВ, обозначаемые τ_1, τ_2. Имеется шаг квантования по времени, обозначаемый ε, который равен 1. Задача τ_1 является спорадической и имеет $T_1 = 10$. СО для τ_1 определяется параметром $D_1 = 5$ согласно (3). Задача τ_2 является периодической и имеет исходное НО (5) с $T_2^{xx\,\min} = 50$, $T_2^{xx\,\max} = 60$, $T_2^{xy\,\max} = 60$, а также $s_{2,0} = -55$. Естественно, что значение T_2 заранее не определено. Пусть при исходном процессоре оказывается, что $C_1 = 5$, $C_2 = 25$. Очевидно, для соблюдения всех ограничений РВ надо, чтобы τ_1 имела наивысший приоритет, так как $C_1 = D_1$. Получается, что $\pi_1 = 1$, $\pi_2 = 2$, то есть множество $\{\pi_i\}$ определено.

Согласно базовому подходу необходимо получить вторичное СО для τ_2. В ходе преобразования НО (5) в СО согласно условию (6) могут получаться различные сочетания (O_2^*, T_2^*, D_2), каждое из которых определяет допустимое СО. Для данного примера на основе (6) легко получить вторичное СО с максимальным D_2, а именно СО, определяемое значениями: $O_2^* = O_2 = 0$, $T_2^* = T_2 = 55$, $D_2 = 30$.

На рис. 1 представлено выполнение τ_2 в случае этого вторичного СО. При этом окружностью отмечаются моменты $(\tau_{2,j})$ появления запросов $\tau_{2,j}$ в очереди запросов, а прерывание выполнения запроса (более приоритетными запросами) обозначается линией над временной осью.

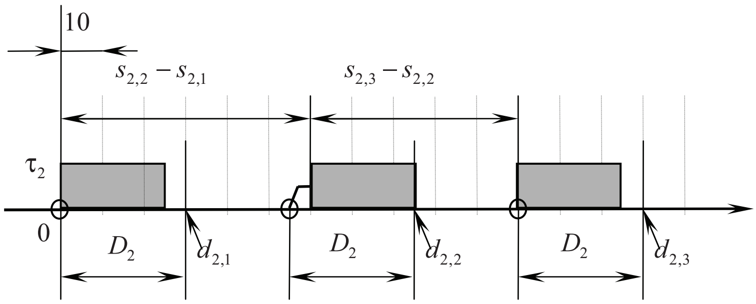

Рис. 1. Пример выполнения задачи τ_2 в случае вторичного СО, определяемого параметрами, $O_2^* = O_2 = 0$, $T_2^* = T_2 = 55$, $D_2 = 30$

На рис. 1 показаны два крайних случая, формирующих максимальное и минимальное значения для выражения $s_{i,j} - s_{i,j-1}$. Эти значения равны 60 и 50 соответственно, что укладывается в исходное НО (5) при $T_2^{xx\min} = 50$, $T_2^{xx\max} = 60$. Из этой диаграммы легко понять, что D_2 является максимальным из возможных. Действительно, крайний срок $d_{2,2}$ предохраняет от нарушения условий $s_{2,2} - s_{2,1} \le 60$ и $s_{2,3} - s_{2,2} \ge 50$, то есть от нарушения части исходного НО (5). Если увеличить D_2, то в наихудшей ситуации момент времени $s_{2,2}$ может сдвинуться правее, что приведет к нарушению ограничений $s_{2,2} - s_{2,1} \le 60$, $s_{2,3} - s_{2,2} \ge 50$ и НО (5) в целом. Если же при этом попробовать уменьшать или увеличивать T_2, то все равно условие $s_{2,2} - s_{2,1} \le 60$ либо условие $s_{2,3} - s_{2,2} \ge 50$ будут нарушаться.

Тут важно подчеркнуть, что такая наихудшая ситуация рассматривается в общем случае, когда могут предполагаться самые разные задачи с более высокими приоритетами, а не только задача τ_1 с указанными параметрами. Дело в том, что получение вторичного СО происходит в отрыве от других задач, так как иначе это уже будет собственно процесс планирования, в котором надо учитывать все имеющиеся задачи РВ. А такой процесс очень быстро усложняется с увеличением количества задач. Конечно, для двух задач он не составляет труда, но это лишь простой пример, используемый здесь, а при большем числе задач все будет гораздо сложнее.

Таким образом, при максимальном из возможных D_2 (равном 30, как было указано выше) получается, что вторичное СО для τ_2 будет нарушаться, то есть многие $d_{2,j}$ могут нарушаться за счет прерывания более приоритетной задачей τ_1 (рис. 2). Очевидно, что вторичные СО с меньшими значениями D_2 будут еще хуже с точки зрения количества нарушаемых крайних сроков. Следовательно, в случае исходных длительностей запросов C_1, C_2 нельзя гарантировать соблюдение вторичного СО для задачи τ_2.

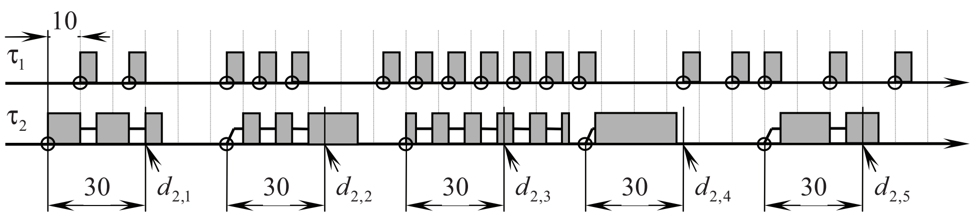

Рис. 2. Выполнение задачи τ_2 с вторичным СО при 100 %-ной длительности запросов (многие крайние сроки нарушаются)

При базовом подходе на этом основании делается вывод о невозможности планирования данных двух задач на данном процессоре, так как на входе алгоритма планирования, например из работы [7], будут только две задачи, имеющие СО. При этом вторичность СО одной из задач на этом этапе уже не принимается в расчет. Тогда одним из решений может быть установка более быстродействующего процессора или в целом ВУ.

Для простоты предположим, что быстродействие ВУ пропорционально уменьшает длительности запросов обеих задач. Нетрудно убедится, что вторичное СО для τ_2 будет гарантированно соблюдаться только лишь при 40 %-ной длительности запросов, когда $C_1 = 2$, $C_2 = 10$, и вторичное СО определяется значениями $O_2^* = O_2 = 0$, $T_2^* = T_2 = 55$, $D_2 = 15$ (рис. 3). Если в этой простой модели считать, что повышение быстродействия в n раз уменьшает длительность запросов тоже в n раз, то тогда данной ситуации соответствует повышение быстродействия ВУ в 2,5 раза. При этом из рис. 2 видно, какой большой процент времени ВУ простаивает.

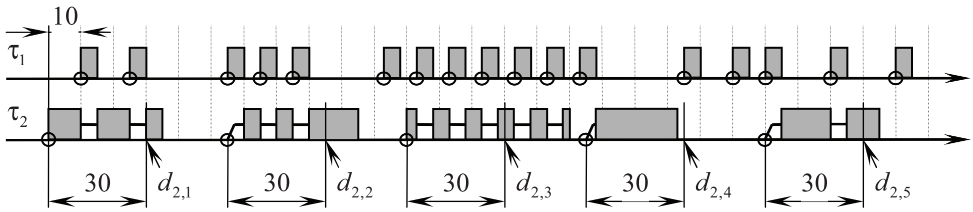

Рис. 3. Выполнение задачи τ_2 с вторичным СО при 40%-ной длительности запросов (все крайние сроки соблюдаются)

На данном примере видно, что базовый подход требует существенного повышения быстродействия ВУ, при этом в итоге значительное время данное ВУ будет простаивать.

Но при внимательном рассмотрении различных вариантов планирования задач τ_1, τ_2 оказывается, что гарантировать соблюдение всех ограничений РВ возможно даже при исходном ВУ в случае выбора $T_2 = 55$. Действительно, учитывая то, что τ_1 может прерывать выполнение τ_2, оказывается, что всегда справедливы условия $50 \le s_{2,j} - s_{2,j-1} \le 60$, $f_{2,j} - s_{2,j} \le 50$. Это нетрудно понять на основе временной диаграммы даже из рис. 2. Но для наглядности на рис. 4 изображен этот же процесс выполнения запросов, при этом с указанием значений $s_{2,j} - s_{2,j-1}$ и без крайних сроков вторичного СО (которое здесь и не требуется).

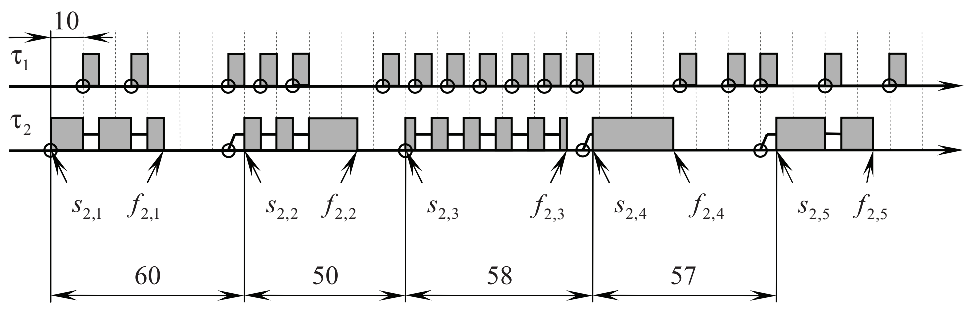

Рис. 4. Выполнение задач в случае, когда τ_2 выполняется как периодическая задача с исходным НО, когда период равен 55, при 100 %-ной длительности всех запросов, то есть при исходном ВУ

На основе рис. 4 нетрудно понять, что для всех запросов выполняются условия $50 \le s_{2,j} - s_{2,j-1} \le 60$, $f_{2,j} - s_{2,j} \le 50$, то есть НО (5) для τ_2 гарантированно соблюдается без всякого уменьшения длительностей запросов.

Но ведь требуется гарантированное соблюдение именно НО (5), а не вторичного СО, используемого согласно базовому подходу. Поэтому очевидно, что непосредственное применение НО является более эффективным подходом.

В рамках реализации этого подхода в работе [8] был предложен ряд алгоритмов, которые позволяют осуществлять планирование задач РВ в условиях ПФП при любых НО из класса ЛИО.

Что на практике дают эти алгоритмы? Дело в том, что НО (5) входит в класс ЛИО, поэтому для этого примера может быть применен любой из алгоритмов, предложенных в работе [8]. При этом будет выдан результат, состоящий в том, что для заданного множества задач (τ_1, τ_2) соблюдение всех ограничений РВ *гарантируется* при условии, что $\pi_1 = 1$, $\pi_2 = 2$, $O_2 = 0$, $T_2 = 55$ при имеющемся ВУ, и это соответствует рис. 4. И не потребуется повышения быстродействия ВУ.

Заключение. На простом примере была продемонстрирована возможность существенного повышения эффективности планирования на основе алгоритмов, представленных авторами в работе [8]. Указанное повышение эффективности выражается в более полном использовании имеющихся вычислительных ресурсов, в частности, не требуется увеличения их быстродействия. Это достигается за счет непосредственного применения исходных ограничений РВ из класса ЛИО в процессе планирования.

Библиографический список

1. Real-Time Scheduling Theory: A Historical Perspective / L. Sha, T. Abdelzaher, K.E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, A.K. Mok // Real-Time Systems 28. – 2004. – P. 101–155.

2. Кавалеров М.В., Матушкин Н.Н. Планирование с фиксированными приоритетами при наличии только стандартных ограничений реального времени // Системы мониторинга и управления: сб. науч. тр. – Пермь: Изд-во Перм. гос. техн. ун-та, 2007. – С. 126–136.

3. Fohler G. Dynamic Timing Constraints – Relaxing Over-constraining Specifications of Real-Time Systems // Proceedings of Work-in-Progress Session: 18th IEEE Real-Time Systems Symposium. – San Francisco, 1997. – P. 27–30.

4. Jitter Compensation for Real-Time Control Systems / P. Marti, G. Fohler, K. Ramamritham, J.M. Fuertes // Proceedings of 22nd IEEE Real-Time Systems Symposium. – London, 2001. – P. 39–48.

5. Кавалеров М.В. Преобразование линейных интервальных ограничений реального времени в стандартные ограничения // Системы управления и информационные технологии. – 2006. – № 4.2(26). – С. 228–233.

6. Кавалеров М.В., Матушкин Н.Н. Получение условий допустимости стандартного ограничения реального времени для примеров линейных интервальных ограничений // Системы мониторинга и управления: сб. науч. тр. – Пермь: Изд-во Перм. гос. техн. ун-та, 2008.

7. Audsley N.C. Optimal Priority Assignment and Feasibility of Static Priority Tasks with Arbitrary Start Times // Technical Report YCS164 / Department of Computer Science; University of York. – UK, 1991. – 31 p.

8. Кавалеров М.В., Матушкин Н.Н. Планирование задач в системах автоматизации и управления при линейных интервальных ограничениях реального времени // Проблемы управления. – 2008. – № 1. – С. 51–61.

Получено 19.09.2011