Фуругян М.Г.

Алгоритмы оптимизации контроля в вычислительных системах реального времени

Аннотация: Рассматривается задача оптимального расположения модулей контроля в вычислительной системе реального времени. Данная задача формулируется в виде минимаксной задачи. Исследуются следующие структуры графа частичного порядка выполнения прикладных модулей: последовательная цепочка, несколько независимых последовательных цепочек. Разработаны алгоритмы построения оптимальной расстановки модулей контроля.

Ключевые слова: система реального времени, оптимальная расстановка модулей контроля

Введение

При разработке и функционировании вычислительных систем реального времени необходимо учитывать возможность возникновения сбоев и ошибок, возникающих при выполнении прикладных модулей. Для контроля правильности вычислений, обнаружения сбоев и повышения безопасности функционирования сложных технических объектов в систему помимо прикладных модулей вводят специальные программы – модули контроля. Эти модули помимо отслеживания сбоев и ошибок в системе сохраняют промежуточную информацию, что позволяет при рестарте системы воспользоваться сохраненными в них данными, а не выполнять все вычисления заново. Таким образом, наличие модулей контроля может существенно улучшить надежность работы системы и уменьшить временные издержки, вызванные возникновением сбоев и ошибок, что крайне важно для систем жесткого реального времени.

Задачи оптимизации расположения модулей контроля исследовалась ранее рядом авторов и при их решении разрабатывались, в основном, вероятностные модели [1-3]. В [4] предполагается, что модули контроля уже расставлены некоторым образом. Для этого случая разработан алгоритм построения зоны рестарта, т.е. минимальной совокупности прикладных модулей, подлежащей перезапуску после обнаружения ошибки некоторым модулем контроля. В [5] разработан алгоритм, позволяющий определить такое расположение модулей контроля, при котором математическое ожидание суммарного времени, затраченное на выполнение всех модулей, минимально.

В настоящей работе предполагается, что программное обеспечение вычислительной системы реального времени состоит из прикладных модулей, на множестве которых задано отношение частичного порядка в виде независимых ориентированных цепочек. Количество модулей контроля фиксировано. Каждый модуль контроля располагается непосредственно после некоторого прикладного модуля. При обнаружении ошибки каким-либо модулем контроля определяется цепочка повторного выполнения прикладных модулей. Требуется таким образом расположить модули контроля, чтобы максимально возможная длина такой цепочки была минимальной.

1. Постановка задачи

Задан набор прикладных модулей $W = \{w_1, w_2, \cdots, w_N\}$, подлежащих выполнению на вычислительной системе. Каждый модуль w_i выполняется без прерываний и переключений с одного процессора на другой, а длительность его выполнения равна $t_i = t(w_i)$, $i = \overline{1, N}$. На множестве W задано отношение частичного порядка выполнения в виде ориентированного графа $G = (W, A)$, состоящего из нескольких последовательных цепочек, где W – множество узлов, A – множество дуг. Если $(w_i, w_j) \in A$, то модуль w_j может выполняться только после завершения модуля w_i. В этом случае будем также использовать обозначение $w_i \to w_j$. Если

$$w_{i_1} \to w_{i_2} \to \ldots \to w_{i_{n-1}} \to w_{i_n}, \tag{1}$$

то модули w_{i_j}, $j = \overline{1, n-1}$, являются предшественниками модуля w_{i_n}. Введем обозначения:

$$P(w_i) = \{w_j \in W : w_j \to w_i\},\ Q(w_i) = \{w_j \in W : w_i \to w_j\}, \tag{2}$$

$$P_0 = \{w_i \in W : P(w_i) = \varnothing\}, \quad Q_0 = \{w_i \in W : Q(w_i) = \varnothing\}, \tag{3}$$

т.е. $P(w_i)$ и $Q(w_i)$ – это соответственно непосредственный предшественник и непосредственный последователь прикладного модуля w_i, а P_0 и Q_0 – это прикладные модули, не имеющие соответственно непосредственного предшественника и непосредственного последователя.

Помимо прикладных модулей в системе имеется K модулей контроля, $K \le N$. Каждый модуль контроля связан с некоторым прикладным модулем и выполняется сразу же после завершения этого прикладного модуля. С каждым прикладным модулем связан не более чем один модуль контроля. Предполагается, что время выполнения модуля контроля существенно меньше времени выполнения прикладных модулей и им можно пренебречь. Пусть $\overline{W} = \{w_{i_1}, w_{i_2}, \ldots, w_{i_K}\}$ – множество всех прикладных модулей, с которыми связаны модули контроля. Будем называть его расстановкой модулей контроля (или, просто, расстановкой).

Модуль контроля после своего выполнения сигнализирует об отсутствии или наличии ошибки. В первом случае вычисления продолжаются по ранее построенному расписанию. В случае обнаружения модулем контроля ошибки необходимо повторить выполнение некоторых прикладных модулей следующим образом.

Пусть модулем контроля, связанным с некоторым прикладным модулем $w_{i_n} \in \overline{W}$, была обнаружена ошибка. В графе G рассматривается цепочка π вида (1), в которой $w_{i_j} \notin \overline{W}$, $j = \overline{2, n-1}$, и, кроме того, либо $w_{i_1} \in P_0$, либо существует прикладной модуль $w_{i_1 - 1} \in \overline{W}$, такой, что $w_{i_1 - 1} \to w_{i_1}$. Иными словами, указанная цепочка начинается с прикладного модуля, который либо не имеет непосредственного предшественника, либо имеет непосредственного предшественника, с которым связан модуль контроля, а все остальные прикладные модули, кроме w_{i_n}, не являются таковыми. Будем называть такие цепочки π цепочками рестарта. В случае обнаружения ошибки модулем контроля, связанным с прикладным модулем w_{i_n}, необходимо для цепочки вида (1) повторить выполнение всех ее прикладных модулей. Длиной цепочки рестарта π вида (1) будем называть величину $t(\pi) = \sum_{j=1}^{n} t_{i_j}$.

Пусть $\Pi(w_{i_n})$ – множество всех цепочек π вида (1) и пусть $t(\Pi(w_{i_n}) = \max\limits_{\pi \in \Pi(w_{i_n})} t(\pi)$, т.е. $t(\Pi(w_{i_n})$ – это максимальная длина таких цепочек. Допустимой расстановкой будем называть такую расстановку $\overline{W}$, при которой каждый прикладной модуль из Q_0 связан с модулем контроля, т.е.

$$Q_0 \subseteq \overline{W} . \tag{4}$$

Если расстановка не является допустимой, то контроль вычислений не может быть выполнен полностью. Всюду в дальнейшем будем предполагать, что $K \geq |Q_0|$ и что к каждому прикладному модулю из Q_0 прикреплен модуль контроля, поскольку при $K < |Q_0|$ допустимой расстановки не существует.

Пусть Ω – множество всех допустимых расстановок. Задача заключается в выборе допустимой расстановки $\overline{W}$, для которой величина $\max\limits_{w \in \overline{W}} t(\Pi(w))$ минимальна. Иными словами, требуется определить величину

$$T_{opt} = \min_{\overline{W} \in \Omega} \max_{w \in \overline{W}} t(\Pi(w)) \tag{5}$$

и допустимую расстановку $\overline{W}$, на которой реализуется указанный минимакс. Такую расстановку будем называть оптимальной. Таким образом, при оптимальной расстановке минимизируется максимальная длина цепочки рестарта.

Рассмотрим случай, когда граф G является последовательной цепочкой $w_1 \to w_2 \to \ldots \to w_N$. В этом случае $Q_0 = \{w_N\}$ и в силу (4) любая допустимая расстановка имеет вид $\overline{W} = \{w_{i_1}, w_{i_2}, \ldots, w_{i_{K-1}}, w_{i_K} = w_N\}$. Без ограничения общности можно считать, что в этом случае $\Omega = \{\overline{W} : 1 \leq i_1 < i_2 < \cdots < i_{K-1} < i_K = N\}$. Введем обозначения:

$I_k = \{i_{k-1}+1, \cdots, i_k\}$, $T_k = \sum_{i \in I_k} t_i$, где $k = \overline{1, K}$, $i_0 = 0$. В соответствии с (5) задача заключается в поиске величины

$$T_{opt} = \min_{\overline{W} \in \Omega} \max_{k=\overline{1, K}} T_k \tag{6}$$

и расстановки $\overline{W}$, реализующей указанный минимакс. Разработаны два алгоритма – точный (алгоритм 1) и приближенный (алгоритм 2). Алгоритм 1 основан на переборе всех допустимых расстановок, число которых составляет C_{N-1}^{K-1}. Его вычислительная сложность при фиксированном K составляет $O(N^{K-1})$, т.е. алгоритм 1 является полиномиальным. Введем обозначения:

$$T_1^* = \left(\sum_{i=1}^{N} t_i\right) / K \text{; } T_2^* = \max_{i=\overline{1, N}} t_i; \;\; T^* = \max(T_1^*, T_2^*). \tag{7}$$

Каждая из величин T_1^*, T_2^* и T^* является нижней оценкой величины T_{opt}, т.е. $T_{opt} \geq T^*$.

Алгоритм 2 минимизирует максимальное отклонение величин T_k, $k = \overline{1, K}$, от T^*, т.е. находит $\min_{\overline{W} \in \Omega} \max_{k=\overline{1, K}} \left|T_k - T^*\right|$ и расстановку $\overline{W}$, реализующую этот минимакс. Его вычислительная сложность составляет $O(KN)$. Для определения отклонения величины $T = \max_{k=\overline{1, K}} T_k$ от величины T_{opt} применима оценка $T - T_{opt} \leq T - T^*$, которая может быть вычислена после работы алгоритма 2.

Рассмотрим случай, когда граф G состоит из p независимых последовательных цепочек, длины которых равны N_i, $i = \overline{1, p}$. Для этого случая разработан алгоритм 3, основанный на специальной процедуре включения модуля контроля в цепочку рестарта. Согласно (5), этот модуль контроля должен разбивать указанную цепочку на две цепочки так, чтобы длина максимальной из них была бы минимально возможной. Вычислительная сложность алгоритма 3 составляет $O((\sum_{i=1}^{p} N_i)(K-p))$, где p – число цепочек.

Литература:

1. *Grassi V., Donatiello L., Tucci S.* On the Optimal Checkpointing of Critical Tasks and Transaction-Oriented Systems // IEEE Trans. Software Eng. – Jan. 1992. – Vol. 18. № 1. – P. 72-77.

2. *Coffman E., Gilbert E.* Optimal Strategies for Scheduling Checkpoints and Preventive Maintenance // IEEE Trans. Reliability. – Apr. 1990. – Vol. 39. № 1. – P. 9-18.

3. *Bruno J.L., Coffman E.G.* Optimal Fault-Tolerant Computing on Multiprocessor Systems // Acta Informatica. – 1997. – Vol. 34. – P. 881-904.

4. *Белый Д.В., Сушков Б.Г.* Модель организации рестартов в системах реального времени. – М.: ВЦ РАН, 1996. – 32 с.

5. *Гречук Б.В., Фуругян М.Г.* Алгоритмы организации рестартов в системах реального времени с произвольным графом связей. – М.: ВЦ РАН, 2004. – 32 с.

Сташенко В.И., Скворцов О.Б., Троицкий О.А.

Особенности оценки вибрационных воздействий в электромеханических системах с импульсным управлением

Аннотация: Рассмотрены вопросы генерации вибрации в проводниковых элементах мощного энергетического оборудования, работающих в условиях прохождения электрических импульсов. Проводниковые элементы в условиях действия токов высокой плотности испытывают не только значительные температурные воздействия, но и большие механические динамические нагрузки. Действие механических вибраций влияет на механические свойства металла и может приводить к снижению усталостной прочности, но может также быть использовано при решении задач диагностики состояния обмоток мощных электрогенераторов и двигателей.

Ключевые слова: электрогенератор, импульсный инвертор, электрический импульс, вибрация, циклическая