

Д. Б. БОРЗОВ

РАЗМЕЩЕНИЕ ПАРАЛЛЕЛЬНЫХ ПОДПРОГРАММ В МУЛЬТИМИКРОКОНТРОЛЛЕРАХ С УЧЕТОМ ОТКАЗОВ

Рассмотрена проблема размещения параллельных подпрограмм в мультимикроконтроллере с учетом возможного отказа его процессорного модуля, сформулированы методика и алгоритм планирования размещения задач.

Ключевые слова: размещение, мультимикроконтроллер, отказ, реконфигурация, параллельная подпрограмма, процессор.

В настоящее время все большее распространение получают отказоустойчивые мультимикроконтроллеры [1], при этом повышаются требования к их быстродействию.

В случае отказа одного из процессоров мультимикроконтроллера необходимо быстро восстановить его функционирование путем реконфигурации структуры с отключением неисправного процессора и заменой его резервным, расположенным обычно вне поля обрабатывающих процессоров. В результате существенно изменяются конфигурации связей между процессорами мультимикроконтроллера и возрастает длина маршрутов передачи данных. Длина маршрута может быть уменьшена путем оперативного перераспределения задач. Процедуры планирования размещения являются комбинаторными, имеют большую вычислительную сложность и поэтому могут

привести к существенному увеличению времени восстановления системы и снижению коэффициента ее готовности. Отказываться из-за этого от перераспределения задач перед рестартом восстановленной системы нецелесообразно, так как возросшие коммуникационные задержки могут привести к потере системной производительности, превышающей ожидаемый выигрыш от применения параллельной многопроцессорной обработки комплекса взаимодействующих подпрограмм [2—4]. В связи с этим для уменьшения времени восстановления мультимикроконтроллера следует многократно снизить затраты времени на планирование размещения задач. Для этого необходимо создать метод сокращения вычислительной сложности процедур планирования размещения задач в процессорных модулях мультимикроконтроллера.

Множество обрабатываемых подпрограмм описывается графом взаимодействия задач [5]: $G = \langle X, E \rangle$, где X — множество вершин графа G , вершины $x_i \in X$ которого соответствуют подпрограммам, а взвешенные дуги $e_{ij} \in E$ определяют объемы данных (в байтах), передаваемых между задачами при следующих соотношениях между индексами: $i = (q-1)n + k$. Граф G может быть описан матрицей обмена информации (МОИ)

$$M = \|m_{ij}\|_{N \times N},$$

где $N = n^2 = |X|$; m_{ij} — объем передаваемых данных между i -м и j -м процессорными модулями [6].

Мультимикроконтроллерную систему будем представлять топологической моделью в виде графа $H = \langle P_1, V \rangle$, где $P_1 = \{P \cup L\}$, здесь P — множество процессорных модулей мультимикроконтроллера, организованных в матрицу $|P|_{n \times n}$, где $|P| = N = n^2$ — число процессорных модулей мультимикроконтроллера, V — множество межмодульных связей, задаваемых матрицей смежности $\|W\|_{N \times N}$ размером $n^2 \times n^2$ [7]; L — дополнительно введенные резервные модули мультимикроконтроллерной системы:

$$\begin{Bmatrix} l_1 \\ l_2 \\ \dots \\ l_n \end{Bmatrix}. \quad (1)$$

Дополнительные модули (1) образуют столбец, записываемый в крайнем правом столбце матрицы P .

В отсутствие отказов мультимикроконтроллерной системы размещение пакета подпрограмм (задач), описываемых графом G , может быть аналитически описано следующим образом:

$$\beta_s = X_s \rightarrow P, \text{ где } s = \overline{1, N!}, k = \overline{1, n}, q = \overline{1, n}.$$

Здесь s — номер очередной перестановки задач $\{x_{qk}\}$ по процессорным модулям $\{P_{qk}\}$, соответствующий s -му варианту размещения. Мощность множества $\psi = \{\beta_s\}$ всевозможных отображений β_s равна числу всевозможных перестановок задач $\{x_{qk}\}$ в матрице $X: |\psi| = N!$. Резервные процессорные модули l_1, l_2, \dots, l_l в случае полной работоспособности всех процессоров мультимикроконтроллера остаются незадействованными. Для описания множества длин d_{ij} кратчайших маршрутов передачи данных в пределах мультимикроконтроллера введем матрицу минимальных расстояний (ММР)

$$D = \|d_{ij}\|_{N \times N}, \quad N = n^2 = |P|,$$

которую можно построить по матрице смежности.

Пусть Ψ — множество всевозможных отображений вида β_s . Тогда задачу размещения можно сформулировать как поиск такого отображения $\beta^* \in \Psi$ [7], что

$$T_{\beta^*} = \min_{\Psi} \left\{ \max_{\beta \in \Psi} \left\{ T_{\beta}(p_{a,b}, p_{x,y}) \right\} \right\}, \quad (2)$$

где $T_{\beta}(p_{a,b}, p_{x,y})$ — коммутационная задержка при передаче данных между процессорными модулями $p_{a,b}$ и $p_{x,y}$, соответствующая отображению β и вычисляемая как произведение

$$T_{\beta}(p_{a,b}, p_{x,y}) = d_{ij} m_{ij}. \quad (3)$$

Процесс планирования размещения задач может быть выполнен, например, по методу, описанному в статье [6].

В случае отказа, например, процессора $p_{r,t}$ ($r = \overline{1, n}, t = \overline{1, n}$) мультимикроконтроллерной системы размещение пакета задач, описываемых графом G , может быть аналитически описано следующим образом:

$$X_s \rightarrow P_1 = \left\{ \begin{array}{cccccc} x_{s_{1,1}} & x_{s_{1,2}} & \dots & x_{s_{1,k}} & x_{s_{1,k+1}} & \dots & x_{s_{1,n}} \\ x_{s_{2,1}} & x_{s_{2,2}} & \dots & x_{s_{2,k}} & x_{s_{2,k+1}} & \dots & x_{s_{2,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{s_{q,1}} & x_{s_{q,1}} & \dots & x_{s_{q,k}} & x_{s_{q,k+1}} & \dots & x_{s_{q,n}} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{s_{n,1}} & x_{s_{n,2}} & \dots & x_{s_{n,k}} & x_{s_{n,k+1}} & \dots & x_{s_{n,n}} \end{array} \right\} \rightarrow \left\{ \begin{array}{cccccc|c} p_{1,1} & p_{1,2} & \dots & p_{1,k} & p_{1,k+1} & \dots & p_{1,n} & l_1 \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} & p_{2,k+1} & \dots & p_{2,n} & l_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_{q,1} & p_{q,2} & \dots & \boxtimes & p_{q,k} & \dots & p_{q,n-1} & p_{q,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ p_{n,1} & p_{n,2} & \dots & p_{n,k} & p_{n,k+1} & \dots & p_{n,n} & l_t \end{array} \right\}, \quad (4)$$

где $s = \overline{1, N!}, k = \overline{1, n}, q = \overline{1, n}, r = \overline{1, n}, t = \overline{1, n}$; символом \boxtimes обозначен возможный отказ процессора $p_{r,t}$.

В данном случае при отказе процессора $p_{r,t}$ необходимо оперативно заменить его на один из ближайших свободных резервных процессоров l_1, l_2, \dots, l_t . Возможен вариант замены [1], при котором все подпрограммы $x_{s_{q,k}}, x_{s_{q,k+1}}, \dots, x_{s_{q,n}}$, назначенные ранее на процессоры $p_{q,k}, p_{q,k+1}, \dots, p_{q,n}$, переназначаются на один из ближайших свободных резервных процессоров, расположенных правее. В этом случае, например, задание $x_{s_{q,n}}$, назначенное ранее процессору $p_{q,n}$, назначается l_q , $x_{s_{q,n-1}}$ — процессору $p_{q,n}$ и т.д. В то же время все задания $x_{s_{q,1}}, x_{s_{q,2}}, \dots, x_{s_{q,k-1}}$ остаются без изменений.

Рассмотрим справедливость приведенного отображения (4) на примере. Пусть имеется гипотетический вариант размещения, заданный моделью в виде матрицы $|P|_{2 \times 2}$, где $|P| = N = n^2 = 2^2 = 4$:

$$P = \left\{ \begin{array}{cc|c} p_1 & p_2 & l_1 \\ p_3 & p_4 & l_2 \end{array} \right\}. \quad (5)$$

Выражению (5) соответствует ММР:

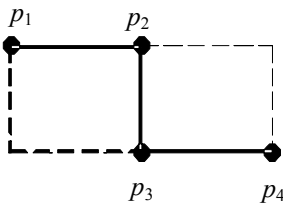
$$D = \begin{vmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} \end{vmatrix}. \quad (6)$$

Пусть также задано множество межмодульных связей V , соответствующее отображению (4), описанное матрицей МОИ: $M = \|m_{ij}\|_{4 \times 4}$, которой ставится в соответствие ММР (6):

$$M = \begin{Bmatrix} 0 & 15 & 0 & 7 \\ 15 & 0 & 10 & 3 \\ 0 & 4 & 0 & 10 \\ 7 & 3 & 4 & 0 \end{Bmatrix} \rightarrow D = \begin{Bmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{Bmatrix}. \quad (7)$$

В выражении (7) левая матрица представляет собой МОИ гипотетического варианта размещения, правая матрица — соответствующая ей ММР (6).

В случае, например, отказа процессорного модуля p_3 , на который назначена подпрограмма, необходимо провести оперативное переназначение подпрограмм с учетом ближайшего резервного процессорного модуля l_2 . В данном случае предлагается вариант замены, когда подпрограмма, назначенная ранее на процессор p_3 , переназначается (фактически — сдвигается) на один модуль правее с учетом резервного модуля l_2 [1]. В результате получаем вариант топологической модели мультимикроконтроллера, представленный на рисунке.



Для топологической модели, представленной на рисунке, получаем модель мультимикроконтроллера, заданную следующей матрицей \tilde{P} :

$$\tilde{P} = \left\{ \begin{array}{ccc} p_1 & p_2 & 0 \\ 0 & p_3 & p_4 \end{array} \right\}. \quad (8)$$

Соответствующая матрица ММР с учетом (8) выглядит следующим образом:

$$D = \begin{Bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{Bmatrix}. \quad (9)$$

Тогда вариант размещения с учетом (8), (9) будет задаваться следующим отображением:

$$M = \begin{Bmatrix} 0 & 15 & 0 & 7 \\ 15 & 0 & 10 & 3 \\ 0 & 4 & 0 & 10 \\ 7 & 3 & 4 & 0 \end{Bmatrix} \rightarrow D = \begin{Bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{Bmatrix}. \quad (10)$$

В результате подпрограмма, ранее назначенная на процессорный модуль p_4 с кратчайшим расстоянием 1, переназначается на резервный процессор с соответствующим расстоянием 3. Следовательно, как видно из выражения (10), после введения резервного процессорного модуля требуется новое оперативное перераспределение задач. Кроме того, в статье [6] было показано, что подпрограммы (подзадачи), которым соответствует наибольший объем передачи данных, предпочтительнее назначать на процессорные модули, которым соответствуют минимальные кратчайшие расстояния D для уменьшения общего времени решения задачи.

Как видно из рассмотренного примера, в результате отказа процессорных модулей мультимикроконтроллера из-за перераспределения подпрограмм на резервные процессоры происходят значительное изменение топологии мультимикроконтроллерной системы и увеличение кратчайших маршрутов, вследствие чего ухудшается общее качество размещения подпрограмм, и как следствие — увеличивается общее время решения задачи. Следовательно, необходимо проводить новое планирование размещения подпрограмм фактически в новой топологической модели мультимикроконтроллера. Эту задачу можно выполнить с помощью метода, представленного в работе [6].

Предложенный способ планирования размещения с учетом отказа мультимикроконтроллерных процессорных модулей позволит уменьшить время восстановления работоспособности мультимикроконтроллера, благодаря чему существенно уменьшается время простоя и повышается производительность системы.

СПИСОК ЛИТЕРАТУРЫ

1. *Зотов И. В.* Организация и синтез микропрограммных мультимикроконтроллеров. Курск: Изд-во „Курск“, 1999. 368 с.
2. *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. СПб: БХВ-Петербург, 2002. 608 с.
3. *Tanenbaum A. S.* Distributed Operation Systems. Prentice-Hall, 1994. 648 p.
4. *Корнеев В. В.* Параллельные вычислительные системы. М.: Нолидж, 1999. 340 с.
5. *Оре О.* Теория графов. М.: Наука, 1968. 352 с.
6. *Борзов Д. Б.* Метод снижения коммуникационной задержки путем субоптимального размещения задач в матричных базовых блоках кластера // Телекоммуникации. 2008. № 4. С. 21—25.
7. *Борзов Д. Б., Аль-Мараят Б. И., Тишкин А. П.* Акселератор планирования размещения задач в кластерных вычислительных системах высокой готовности // Изв. вузов. Приборостроение. 2008. Т. 51, № 2. С. 29—33.

Сведения об авторе

Дмитрий Борисович Борзов

— канд. техн. наук; Курский государственный технический университет, кафедра вычислительной техники; E-mail: borzovdb@kursknet.ru

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
12.09.08 г.