УДК 681.3

Д. Б. Борзов, Ю. В. Соколова, В. В. Минайлов

ПЕРЕРАЗМЕЩЕНИЕ ПОДПРОГРАММ В ОТКАЗОУСТОЙЧИВЫХ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ

Рассмотрена проблема отказа процессора в мультипроцессорных системах, обоснована необходимость переразмещения подпрограмм с учетом отказавших процессоров системы, показана возможность отказа межпроцессорных линков. Предложен алгоритм отказоустойчивого переразмещения при отказе процессора системы.

Ключевые слова: *мультипроцессор, отказ, алгоритм, планирование, размещение.*

В настоящее время все большее распространение получают отказоустойчивые мультикомпьютеры [1]. В случае отказа одного из процессоров и/или процессорных связей правильное функционирование быстро восстанавливается путем реконфигурации структуры с отключением неисправного процессора и обхода неисправной процессорной связи. Как показано в статье [2], значительно изменяется топология многопроцессорной системы и требуется переразмещение назначенных ранее подпрограмм. В настоящей работе в продолжение исследований [3, 4] предлагается вариант решения данной задачи.

Множество реализуемых в мультикомпьютере подпрограмм описывается графом взаимодействия задач $G=\left(X,E\right)$, где

$$X=\begin{Bmatrix} x_{1.1} & x_{1.2} & \dots & x_{1.\upsilon} & \dots & x_{1.n} \\ x_{2.1} & x_{2.2} & \dots & x_{2.\upsilon} & \dots & x_{2.n} \\ \dots & & & & & \\ x_{q.1} & x_{q.1} & \dots & x_{q.\upsilon} & \dots & x_{q.n} \\ \dots & & & & & \\ x_{n.1} & x_{n.2} & \dots & x_{n.\upsilon} & \dots & x_{n.n} \end{Bmatrix}$$

— множество вершин, соответствующих отдельным подпрограммам, а $e_{ij}\in E$ — множество дуг, или связей, между ними при $i,j=\left(q-1\right)n+k$, которые определяются объемами данных (в байтах) m_{ij} и передаются между подзадачами, описываемыми матрицей смежности

$M = \|m_{ij}\|_{N\times|E|}$, где $N = |X|$. Множество вершин X упорядочим в виде матрицы, согласно топологической структуре мультикомпьютера.

Мультикомпьютер описывается топологической моделью в виде графа $H = (P_1, V)$ ($\{p_1\} \in P_1$ соответствуют процессорным модулям), а $\{\upsilon\} \in V$ — множеством ребер, соответствующих межмодульным связям. Разобьем множество P на два непересекающихся подмножества: $P_1 = P \cup L$, где $\{p_1\}$ — множество основных, а $\{l\} \in L$ — множество резервных процессоров. Идентификаторы процессоров множества P упорядочим в виде матрицы $P = \|p_{ij}\|_{n\times n}$, где $n = |P|$. Множество резерва L представим в виде матрицы $L = \|l_{ij}\|_{n\times n}$.

С учетом введенного представления множество P_1 в общем случае будет иметь следующий вид (рис. 1):

$$\begin{vmatrix} p_{1.1} & l_{1.1} & p_{1.2} & l_{1.2} & \dots & p_{1.\upsilon} & l_{1.\upsilon} & \dots & p_{1.n} & l_{1.n} \\ p_{2.1} & l_{2.1} & p_{2.2} & p_{2.2} & \dots & p_{2.\upsilon} & l_{2.\upsilon} & \dots & p_{2.n} & l_{2.n} \\ \dots & & & & & & & & & \\ p_{q.1} & l_{q.1} & p_{q.2} & l_{q.2} & \dots & p_{q.\upsilon} & l_{q.\upsilon} & \dots & p_{q.n} & l_{q.n} \\ \dots & & & & & & & & & \\ p_{n.1} & l_{n.1} & p_{n.2} & l_{n.2} & \dots & p_{n.\upsilon} & l_{n.\upsilon} & \dots & p_{n.n} & l_{n.n} \end{vmatrix}, \quad (1)$$

где $\upsilon = \overline{1,n}$, $q = \overline{1,n}$.

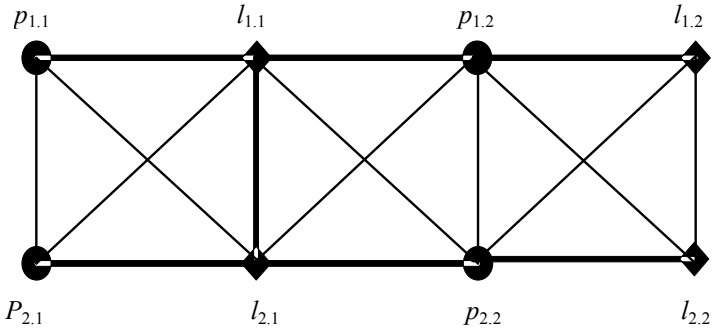

Рис. 1

Для описания множества значений длины d_{ij} кратчайших маршрутов передачи данных введем матрицу минимальных расстояний (ММР) $D = \| d_{ij} \|_{N\times N}$, $N = n^2 = |P|$, которую можно построить по матрице смежности:

$$D = \begin{vmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \dots & & & \\ d_{n,1} & d_{n,2} & \dots & d_{n,n} \end{vmatrix}. \quad (2)$$

Тогда размещение в мультикомпьютере пакета подпрограмм, описываемых графом G, может быть аналитически описано отображением:

$$X_s \to P_1 = \begin{Bmatrix} x_{s_{1.1}} & x_{s_{1.2}} & \dots & x_{s_{1.\upsilon}} & \dots & x_{s_{1.n}} \\ x_{s_{2.1}} & x_{s_{2.2}} & \dots & x_{s_{2.\upsilon}} & \dots & x_{s_{2.n}} \\ \dots & & & & & \\ x_{s_{q.1}} & x_{s_{q.1}} & \dots & x_{s_{q.\upsilon}} & \dots & x_{s_{q.n}} \\ \dots & & & & & \\ x_{s_{n.1}} & x_{s_{n.2}} & \dots & x_{s_{n.\upsilon}} & \dots & x_{s_{n.n}} \end{Bmatrix} \to$$

$$\to \begin{vmatrix} p_{1.1} & l_{1.1} & p_{1.2} & l_{1.2} & \dots & p_{1.\upsilon} & l_{1.\upsilon} & \dots & p_{1.n} & l_{1.n} \\ p_{2.1} & l_{2.1} & p_{2.2} & l_{2.2} & \dots & p_{2.\upsilon} & l_{2.\upsilon} & \dots & p_{2.n} & l_{2.n} \\ \dots & & & & & & & & & \\ p_{q.1} & l_{q.1} & p_{q.2} & l_{q.2} & \dots & p_{q.\upsilon} & l_{q.\upsilon} & \dots & p_{q.n} & l_{q.n} \\ \dots & & & & & & & & & \\ p_{n.1} & l_{n.1} & p_{n.2} & l_{n.2} & \dots & p_{n.\upsilon} & l_{n.\upsilon} & \dots & p_{n.n} & l_{n.n} \end{vmatrix}, \qquad (3)$$

где s — номер варианта размещения задач $\{x_{qk}\}$ по процессорным модулям $\{P_{q\upsilon}\}$, $s = \overline{1, N!}$, символ „→“ означает отображение одной из вершин графа G на один из процессоров P. Мощность множества всевозможных отображений $\Psi = \{\beta_s\}$ равна числу перестановок задач $\{x_{q\upsilon}\}$ в матрице X: $|\Psi| = N!$.

В случае отказа, например, процессора $p_{\alpha,\beta}$ ($\alpha = \overline{1,n}$, $\beta = \overline{1,n}$) размещение задач, описываемых графом G, может быть описано отображением

$$X_s \to P_1 = \begin{Bmatrix} x_{s_{1.1}} & x_{s_{1.2}} & \dots & x_{s_{1.\upsilon}} & \dots & x_{s_{1.n}} \\ x_{s_{2.1}} & x_{s_{2.2}} & \dots & x_{s_{2.\upsilon}} & \dots & x_{s_{2.n}} \\ \dots & & & & & \\ x_{s_{q.1}} & x_{s_{q.1}} & \dots & x_{s_{q.\upsilon}} & \dots & x_{s_{q.n}} \\ \dots & & & & & \\ x_{s_{n.1}} & x_{s_{n.2}} & \dots & x_{s_{n.\upsilon}} & \dots & x_{s_{n.n}} \end{Bmatrix} \to$$

$$\to \begin{vmatrix} p_{1.1} & l_{1.1} & \boxtimes & l_{1.2} & \dots & p_{1.\upsilon} & l_{1.\upsilon} & \dots & p_{1.n} & l_{1.n} \\ p_{2.1} & l_{2.1} & p_{2.2} & l_{2.2} & \dots & p_{2.\upsilon} & l_{2.\upsilon} & \dots & p_{2.n} & l_{2.n} \\ \dots & & & & & & & & & \\ p_{q.1} & l_{q.1} & p_{q.2} & l_{q.2} & \dots & p_{q.\upsilon} & l_{q.\upsilon} & \dots & p_{q.n} & l_{q.n} \\ \dots & & & & & & & & & \\ p_{n.1} & l_{n.1} & p_{n.2} & l_{n.2} & \dots & p_{n.\upsilon} & l_{n.\upsilon} & \dots & p_{n.n} & l_{n.n} \end{vmatrix}. \qquad (4)$$

В данном случае при отказе процессорного модуля $p_{1,2}$ необходимо оперативно его заместить резервным процессором $l_{1,2}$. Такая замена ведет к изменению значений в ММР. При этом изменяется матричная организация мультикомпьютера (рис. 2), а матрица ММР принимает вид:

$$D = \begin{vmatrix} 0 & 2 & 1 & 4 \\ 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 4 & 2 & 3 & 0 \end{vmatrix}.$$

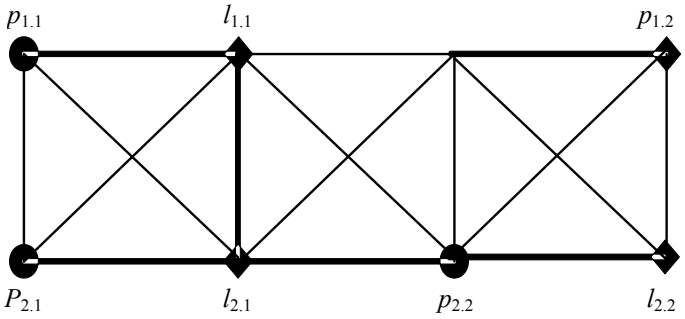

Рис. 2

В этом случае размещение задач с учетом замены основного процессора резервным может быть описано отображением:

$$X_s \to P_1 = \begin{Bmatrix} x_{s_{1.1}} & x_{s_{1.2}} & \dots & x_{s_{1.\upsilon}} & \dots & x_{s_{1.n}} \\ x_{s_{2.1}} & x_{s_{2.2}} & \dots & x_{s_{2.\upsilon}} & \dots & x_{s_{2.n}} \\ \dots & & & & & \\ x_{s_{q.1}} & x_{s_{q.1}} & \dots & x_{s_{q.\upsilon}} & \dots & x_{s_{q.n}} \\ \dots & & & & & \\ x_{s_{n.1}} & x_{s_{n.2}} & \dots & x_{s_{n.\upsilon}} & \dots & x_{s_{n.n}} \end{Bmatrix} \to$$

$$\to \begin{vmatrix} p_{1.1} & l_{1.1} & \boxtimes & p_{1.2} & \cdots & p_{1.\upsilon} & l_{1.\upsilon} & \cdots & p_{1.n} & l_{1.n} \\ p_{2.1} & l_{2.1} & p_{2.2} & l_{2.2} & \cdots & p_{2.\upsilon} & l_{2.\upsilon} & \cdots & p_{2.n} & l_{2.n} \\ \dots & & & & & & & & & \\ p_{q.1} & l_{q.1} & p_{q.2} & l_{q.2} & \cdots & p_{q.\upsilon} & l_{q.\upsilon} & \cdots & p_{q.n} & l_{q.n} \\ \dots & & & & & & & & & \\ p_{n.1} & l_{n.1} & p_{n.2} & l_{n.2} & \cdots & p_{n.\upsilon} & l_{n.\upsilon} & \cdots & p_{n.n} & l_{n.n} \end{vmatrix}. \qquad (5)$$

Пусть матрица

$$Z = \begin{Bmatrix} z_{1.1} & z_{1.2} & \dots & z_{1.\upsilon} & \dots & z_{1.n} \\ z_{2.1} & z_{2.2} & \dots & z_{2.\upsilon} & \dots & z_{2.n} \\ \dots & & & & & \\ z_{q.1} & z_{q.1} & \dots & z_{q.\upsilon} & \dots & z_{q.n} \\ \dots & & & & & \\ z_{n.1} & z_{n.2} & \dots & z_{n.\upsilon} & \dots & z_{n.n} \end{Bmatrix}$$

объединяет тэги, отражающие состояние процессоров $P_{q,\upsilon} \in P_1$:

$$z_{\alpha,\beta} = \begin{cases} 1, \text{ если } p_{q,\upsilon} \text{ неисправен;} \\ 0, \text{ если } p_{q,\upsilon} \text{ исправен,} \end{cases}$$

где $\alpha = \overline{1,n}$; $\beta = \overline{1,n}$.

Пусть матрица

$$\Theta = \begin{Bmatrix} \Theta_{1,1} & \Theta_{1,2} & \cdots & \Theta_{1,n} \\ \Theta_{2,1} & \Theta_{2,2} & \cdots & \Theta_{2,n} \\ \cdots & & & \\ \Theta_{n,1} & \Theta_{n,2} & \cdots & \Theta_{n,n} \end{Bmatrix}$$

содержит тэги, показывающие работоспособность и занятость резервных процессоров $l_{i,j}$ ($i = \overline{1,n}$; $j = \overline{1,n}$). При этом

$$\Theta_{\alpha,\beta} = \begin{cases} 1, \text{ если } \Theta_{q,\upsilon} \text{ неисправен и/или занят;} \\ 0, \text{ если } \Theta_{q,\upsilon} \text{ исправен,} \end{cases}$$

где $\alpha = \overline{1,n}$; $\beta = \overline{1,n}$.

В случае отказа линка $p_{l_{q,\upsilon}}$ ($q = \overline{1,n}$; $\upsilon = \overline{1,n}$; $l = \overline{1,n}$) между процессорами $p_{q.\upsilon}$ и $\upsilon = \overline{1,n}$ ($q = \overline{1,n}$; $\upsilon = \overline{1,n}$) нарушаются маршруты передач данных и необходимо найти кратчайший путь обхода. Для этого можно воспользоваться алгоритмом Дейкстры, применив его локально.

Вышеизложенное позволяет составить обобщенный алгоритм замены отказавшего процессора резервным и найти кратчайший маршрут при отказе межпроцессорной связи.

1. Ввести матрицу смежности.
2. Ввести матрицу расстояний.
3. Ввести матрицу исправности основных процессоров.
4. Ввести матрицу исправности резервных процессоров.
5. Ввести матрицу обхода 1.
6. Ввести матрицу обхода 2.
7. Если отказал основной процессор, то в матрице резервных процессоров искать ближайший свободный резервный процессор.
8. Если резервный процессор найден, то переназначить неисправный процессор на соответствующий резервный, иначе п. 7.
9. Выполнить поиск свободного резервного процессора и ввести матрицу обхода 2, иначе необходима полная замена матрицы процессоров.
10. Выполнить переразмещение подпрограмм по [2].
11. Если отказал линк, то начальной точкой обхода принять процессор $p_{q.\upsilon}$, а конечной — $p_{q+1.\upsilon+1}$ ($q = \overline{1,n}$; $\upsilon = \overline{1,n}$).
12. Применить алгоритм Дейкстры для начальной и конечной точек обхода.
13. Ввести переменные *a*, *b*, в которых хранится объем передаваемой информации.
14. Ввести два динамических массива *c*[*i*], *d*[*i*]; первоначально переменные имеют одинаковые значения.
15. Ввести массивы *k*[1,*i*] := 0; l[1,*j*] := 0.
16. В качестве исходной точки взять *i*-й процессор и установить ему метку 0.
17. Методом перебора искать всевозможные пути до конечной точки обхода, суммируя объем передаваемой информации.
18. Запомнить найденный путь в массиве *d*[*i*]; *a* := min{*a*, *b*}; *c*[*i*] := *d*[*i*].
19. Повторить п. 18 для всех остальных путей.
20. По найденному маршруту в матрицу смежности добавить необходимое количество байтов.

В дальнейших исследованиях планируется формализация предложенной методики и алгоритма, а также разработка аппаратной схемы устройства переразмещения для систем высокой готовности.

Работа выполнена в рамках программы „Научные и научно-педагогические кадры инновационной России на 2009—2013 годы“ (проект 14.B37.21.0598).

СПИСОК ЛИТЕРАТУРЫ

1. *Зотов И. В.* Организация и синтез микропрограммных мультимикроконтроллеров. Курск: Изд-во „Курск“, 1999. 368 с.

2. *Борзов Д. Б.* Метод оперативного переразмещения задач в отказоустойчивых логических мультиконтроллерах // Нейрокомпьютеры: разработка, применение. 2010. № 1. С. 29—33.

3. *Борзов Д. Б., Соколова Ю. В.* Методика переразмещения подпрограмм в отказоустойчивых мультикомпьютерах // Сб. тр. XVIII Междунар. науч.-техн. конф. „Машиностроение и техносфера XXI века“. Донецк, 2011. Т. 1. С. 90—93.

4. *Борзов Д. Б., Соколова Ю. В.* Переразмещение подпрограмм в отказоустойчивых мультикомпьютерах при отказе связей // Сб. матер. X Междунар. науч.-техн. конф. „Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации. Распознавание 2012“. Курск, 2012. С. 238—240.

Сведения об авторах

Дмитрий Борисович Борзов — канд. техн. наук, доцент; Юго-Западный государственный университет, кафедра вычислительной техники, Курск; E-mail: borzovdb@kursknet.ru

Юлия Васильевна Соколова — Юго-Западный государственный университет, кафедра вычислительной техники, Курск; преподаватель; E-mail: jv.sokolova@mail.ru

Виктор Викторович Минайлов — Юго-Западный государственный университет, кафедра вычислительной техники, Курск; преподаватель; E-mail: gkptuip2@mail.ru

Рекомендована Юго-Западным государственным университетом

Поступила в редакцию 18.02.13 г.