

## МЕТОДИКА ПЛАНИРОВАНИЯ РАЗМЕЩЕНИЯ ПРОГРАММ В ТОРОИДАЛЬНЫХ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ

Борзов Д.Б., Дюбрюкс С.А., Минайлов В.В. (ЮЗГУ, г. Курск, Россия)  
Тел. +7(4712)22-26-75; E-mail: [borzovdb@kursknet.ru](mailto:borzovdb@kursknet.ru)

**Abstract:** Describes a problem in planning programs placing toroidal multiprocessor systems when applied to mission-critical systems (system monitoring, tracking, targeting, and etc.). An approach to solving this problem.

**Keywords:** planning, programs, toroid, multiprocessor systems, mission-critical systems.

Мультимикропроцессоры (МП) – один из широко распространенных классов высокопроизводительных вычислительных систем [1]. Современные МП – системы с высокой степенью параллелизма, включающие десятки и сотни процессорных модулей, взаимодействующих через коммуникационную сеть. Перспективным направлением развития МП в части дальнейшего повышения их производительности, снижения энергопотребления и улучшения массо–габаритных характеристик является переход к масштабируемой реализации мультимикропроцессоров (ММ) [2]. Они позволяют получить высокую эффективность межпроцессорных взаимодействий, снижая долю последовательной составляющей, упрощая программирование мультимикропроцессора, содержащего несколько десятков процессоров при поиске варианта размещения.

Вариантом применения ММ-систем могут быть тороидальные системы (ТС), топологически представляющие собой множество матрично-ориентированных процессорных модулей [1]. Такие системы имеют «особое» значение при использовании в системах критического назначения (системы слежения, наблюдения, прицеливания, атомные системы и т.д.) [3].

В случае отказа внутреннего процессорного модуля ТС в случае использования критических систем, возникает необходимость оперативного перераспределения отказавших процессорных модулей для поддержания оперативной коммуникационной задержки и сохранения ее производительности. В этом случае использование программных средств неприемлемо из-за относительно высокого времени решения [3]. В этом случае целесообразно применение соответствующих аппаратных средств планирования размещения. Данная работа является продолжением работы [4].

Исходная программа представляется графом взаимодействия задач  $G=\langle X,E\rangle$ , где  $X$  - множество вершин графа  $G$ , вершины  $x_i \in X$  которого соответствуют подпрограммам, а взвешенные дуги  $e_{ij} \in E$  описывают связи между ними и задают объемы данных  $m_{ij}$  (в байтах), передаваемых при обмене между подпрограммами. Граф  $G$  представляется матрицей смежности:  $W = \parallel m_{ij} \parallel$ , где  $m_{ij}$  – вес дуги  $e_{ij}$ .

Тороидальная система представляется графом  $H=\langle P,V\rangle$ , где

$$P = \left\{ \begin{array}{cccc} P_{1,1} & P_{2,1} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,n} \\ \dots & & & \\ P_{n,1} & P_{n,2} & \dots & P_{n,n} \end{array} \right\} - \quad (1)$$

множество идентификаторов процессорных модулей, топологически представляющие матрицу  $|P|_{x \times y}$ , где мощность  $|P| = N = n^2$  - число процессорных модулей;  $V$  – множе-

ство межмодульных связей, задаваемых матрицей смежности  $\|W\|_{n \times n}$  размером  $n^2 \times n^2$ . Так как тороидальная система фактически представляет собой куб, состоящий из множества слоев, где каждый слой является матричной ММ-системой, то для задания множества слоев, введем вектор:  $Sl = \{P_1, P_2, \dots, P_t, \dots, P_n\}$ . Здесь  $t = \overline{1, N}$ .

Схематичный вид тороидальной системы показан на рисунке 1.

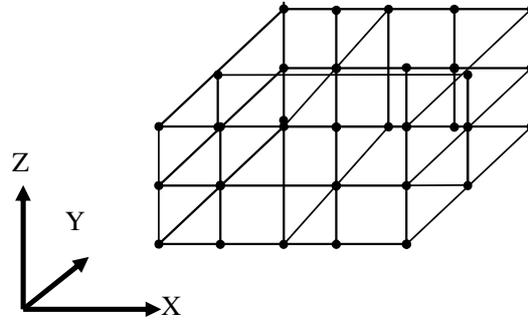


Рис. 1. Топологический вид тороидальной системы

Тороидальной системе, представленной на рисунке 1 назначим соответствующие трехмерные координаты X, Y и Z. В дальнейшем каждый уровень тороида будем представлять независимой матричной мультипроцессорной системой. Например, для тороидальной системы, состоящей из трех слоев, вид нижнего слоя представлен на рисунке 2.

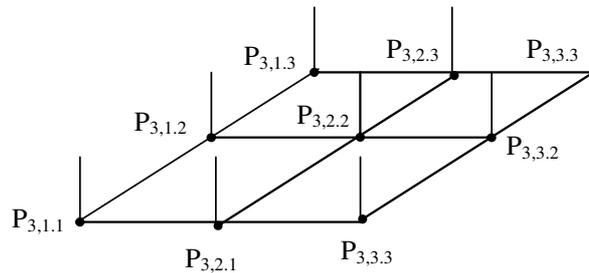


Рис. 2. Вид слоя тороидальной системы.

Здесь на рисунке 2 точками обозначены процессорные модули ТС нижнего слоя, в номер при индексах P задают соответствующие трехмерные координаты. Например, P\_{3,2,2} означает процессор, расположенный в третьем слое с координатами X=2, Y=2.

Таким образом, как видно из рисунка 3, в случае возникновения отказа одного из процессорных модулей ТС, необходима оперативная реакция мультипроцессорной системы. Например, в случае отказа процессорного модуля P\_{3,2,2}, происходит изменение топологической организации мультипроцессорной системы так, как показано на рисунке 3.

Для возможности учета отказов ситуаций введем матрицу расстояний  $D = \|d_{ij}\|_{N \times N}$ ,  $N = n^2 = |H|$ , которая строится по матрице смежности  $\|W\|_{N \times N}$  графа H.

Размещение пакета программ (комплекса задач), описываемых графом G может быть аналитически описано отображением

$$\beta_s = x_{S_{q,k}} \rightarrow p_{q,k}, \quad (2)$$

где  $s = \overline{1, N!}$ ,  $k = \overline{1, n}$ ,  $q = \overline{1, n}$ . Здесь  $s$  – это номер очередной перестановки, соответствующий  $s$ -му варианту размещения. Мощность множества  $\Psi = \{\beta_s\}$  всевозможных отображений 2 равна числу всевозможных перестановок задач  $\{x_{qk}\}$  в матрице  $X$ :  $|\Psi| = N!$ .

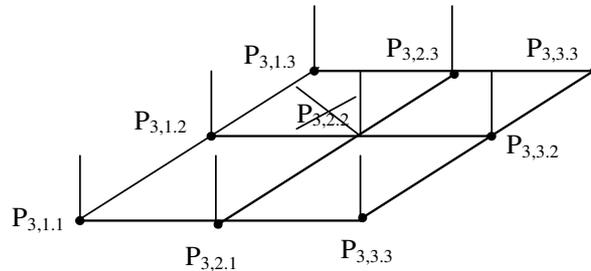


Рис. 3. Отказ процессорного модуля ТС

Пусть  $\Psi$  – множество всевозможных отображений вида (2). Тогда задачу размещения можно сформулировать как поиск отображения  $\beta^* \in \Psi$ , такого, что

$$T_{\beta^*} = \min_{\Psi} \{ \max_{\beta_s \in \Psi} \{ T_{\beta_s}(p_{a,b}, p_{x,y}) \} \}, \quad (3)$$

где  $T_{\beta_s}(p_{a,b}, p_{x,y})$  – коммутационная задержка при передаче данных между модулями  $p_{a,b}$  и  $p_{x,y}$ , соответствующая отображению  $\beta_s$ , которая рассчитывается как произведение

$$T_{\beta_s}(p_{a,b}, p_{x,y}) = d_{ij} \cdot m_{ij}, \quad (4)$$

где  $i = (a-1) \cdot n + b$  и  $j = (x-1) \cdot n + y$ .

В (3) выражение  $\max_{\beta_s \in \Psi} \{ T_{\beta_s}(p_{a,b}, p_{x,y}) \}$  означает поиск максимальной коммуникационной задержки в ТС, а  $\min(\max)$  – дальнейшая ее минимизация.

В ходе поиска контролируется степень уменьшения величины образующихся коммуникационных задержек (4) и принимается решение о целесообразности продолжения поисковых перестановок путем анализа величины отношения достигнутого минимального значения (3) к гипотетической минимально возможной ее величине  $T_{\inf}$ . При вычислении нижней оценки будем назначать дуги графа  $G$  с наибольшим весом на самые короткие маршруты в графе  $H$ , не обращая внимания на ограничения, накладываемые фактическими связями между задачами в графе  $G$ .

На содержательном уровне поиск нижней оценки состоит из следующих шагов:

- 1) выстроить в линейный векторный массив элементы матрицы расстояний по возрастанию;
- 2) выстроить в линейный векторный массив элементы матрицы смежности по убыванию;
- 3) найти частные произведения одноименных компонентов полученных векторов;
- 4) фиксировать максимальное частное произведение, как максимальную коммуникационную задержку, получаемую при гипотетически наилучшем размещении.

Тогда процедуру поиска минимальной нижней оценки можно представить в виде алгоритма:

1) Переписать элементы  $d_{kl} \neq 0$  матрицы  $D$  в вектор-строку  $D' = \|d_{kl}^z\|$  так, что  $d_{k'l'}^{z_1} \leq d_{k'l'}^{z_2} \Leftrightarrow z_1 > z_2$ , где  $z_1$  и  $z_2$  – порядковые номера элементов в  $D'$ .

2) Переписать элементы  $m_{ij} \neq 0$  матрицы  $M$  в вектор-строку  $M' = \|m_{ij}^z\|$  так, что  $m_{i'j'}^{z_1} \geq m_{i'j'}^{z_2} \Leftrightarrow z_1 > z_2$ , где  $z_1$  и  $z_2$  – порядковые номера элементов в  $M'$ .

$$T_{\text{inf}} = \max\{m^z d^z\}, \quad (5)$$

где  $z = \overline{1, |E|}$ ,  $m^z$ ,  $d^z$  – одноименные элементы векторов  $M'$  и  $D'$  с одинаковыми порядковыми номерами.

Предлагается методика планирования размещения программ в ТС, состоящая из следующих шагов: 1) Производится первичное произвольное размещение вершин графа  $G$ , представляющего исходный комплекс программ, описанный графом  $H$  путем наложения матрицы  $M = \|m_{ij}\|_{N \times N}$  на матрицу  $D = \|d_{ij}\|_{N \times N}$ . По формуле (5) находим максимальную коммутационную задержку  $T_H$ , соответствующую данному начальному варианту размещения; 2) Для сопоставления вариантов размещения по критерию (3) осуществляется поиск нижней оценки  $T_{\text{inf}}$ . Затем вычисляется степень близости максимальной коммутационной задержки  $T_H$ , соответствующей первичному варианту размещения, к нижней оценке  $T_{\text{inf}}$  в виде:

$$\eta_H = \frac{T_H}{T_{\text{inf}}}; \quad (6)$$

3) Начиная с элемента  $m_{ij}$ , которому соответствует  $\max\{m_{i,j} \cdot d_{ij}\}$ , пытаемся переместить столбец, его содержащий, на место другого столбца так, чтобы после перестановки и расчета  $T$  по формуле (5) оценка:

$$\eta = \frac{T}{T_{\text{inf}}} \quad (7)$$

снижалась по сравнению с  $\eta_H$  (6) и оценками  $\eta$  по предыдущим вариантам размещения; 4) Анализ достигнутой величины  $\eta$  (6) и оценка степени улучшения размещения выполняется по следующей формуле  $\sigma = \frac{\eta_H}{\eta} = \frac{T_H}{T}$ .

В данной работе была выполнена постановка задачи и методика планирования размещения программ в тороидальных системах. В дальнейших исследованиях планируется разработка соответствующего алгоритма размещения.

**Список литературы:** 1. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем: учебник для вузов / СПб.: Питер, 2004. 668 с. 2. Данильченко Н.В., Макеев С.Н. Масштабируемый мультипроцессор для цифровой обработки сигналов / МЭС-2012. Россия. Москва. октябрь 2012. 3. Борзов Д.Б., Гуляев К.А., Титов В.С. Методы реконфигурации критических систем / XII Всероссийское совещание по проблемам управления Вспу-2014. / Москва 16-19 июня 2014. – С. 1-12. 4. Borzov D.B., Jamil Al Azzeh, Igor V. Zotov and Dmitriy E. Skopin and Dr. Mazin Al Hadidi. An Approach to Achieving Increased Fault-Tolerance and Availability of Multiprocessor-Based Computer Systems / Australian Journal of Basic and Applied Sciences, 8(6) April 2014, Pages: 512-522.