

# ПОДХОД К ПЛАНИРОВАНИЮ ОПЕРАТИВНОГО ПЕРЕРАЗМЕЩЕНИЯ ПОДПРОГРАММ, В МУЛЬТИКОМПЬЮТЕРАХ, С УЧЕТОМ ОТКАЗОВ МЕЖПРОЦЕССОРНЫХ СВЯЗЕЙ

**Соколова Юлия Васильевна**

К.т.н, ведущий специалист, Научно-производственное объединение им. С.А. Лавочкина, Химки  
sokolovaiuv@laspace.ru

## APPROACH TO PLANNING THE OPERATIONAL REPLACEMENT OF SUBPROGRAMS IN MULTICOMPUTERS, TAKING INTO ACCOUNT THE FAILURES OF INTERPROCESSOR COMMUNICATIONS

**Yu. Sokolova**

*Summary.* The use of computing systems to manage critical objects operating in real time is of particular relevance. The problem of failures in multiprocessor systems is considered. The issue of operational restoration of the system functioning in case of failure of one of the processors and/or processor connections by means of its reconfiguration is touched upon. An approach to scheduling the load of processors in critical multiprocessor systems is proposed, which provides an increase in the performance of multiprocessor systems and an increase in their availability factor when bypassing a faulty processor connection.

*Keywords:* multiprocessor system, scheduling, loading, high availability, assignment, method, algorithm.

*Аннотация.* Использование вычислительных систем для управления ответственными объектами, работающими в режиме реального времени, приобретает особую актуальность. Рассмотрена проблема отказов в мультипроцессорных системах. Затрагивается вопрос оперативного восстановления функционирования системы при отказе одного из процессоров и/или процессорных связей путём её реконфигурации. Предложен подход к планированию загрузки процессоров в мультипроцессорных системах критического назначения, обеспечивающего повышение производительности мультипроцессорных систем и увеличения их коэффициента готовности при обходе неисправной процессорной связи.

*Ключевые слова:* мультипроцессорная система, планирование, загрузка, высокая готовность, назначение, метод, алгоритм.

**П**овышение готовности, т.е. снижение времени простоя системы, предполагает подавление, в определенных пределах, влияния отказов и сбоев на работу системы с помощью средств контроля и коррекции ошибок, а также средств автоматического восстановления вычислительного процесса после проявления неисправности, включая аппаратную и программную избыточность, на основе которой реализуются различные варианты отказоустойчивых архитектур.

Особую актуальность приобретает использование вычислительных систем для управления ответственными объектами, работающими в режиме реального времени.

При отказе одного из процессоров и/или межпроцессорных связей, необходимо оперативно восстановить функционирование системы, т.е. её реконфигурация (отключение неисправного процессора или обход неисправной межпроцессорной связи).

Введём множество подпрограмм, которое описывается математической абстракцией — графом задач  $G = \langle X, E \rangle$ , где  $X$  — множество вершин, которые соответствуют отдельным задачам, а  $E$  — множество ребер графа или, иными словами, множество связей между ними. Множество вершин  $X$  запишем в массив в соответствии с топологией мультипроцессора. Множество связей между вершинами графа  $G$  заняты значениями, определяющими объём данных (в байтах), передаваемых между соответствующими задачами. Граф  $G$  задаётся матрицей смежности  $M = \|m_{ij}\|_{N \times N}$ , где  $N = n^2 = |X|$ ,  $m_{ij}$  — объём данных, которые передаются между  $i$ -м и  $j$ -м процессорным модулем.

Мультикомпьютер описывается графом  $H = \langle P1, V \rangle$ , где множество вершин  $P1$ , соответствуют процессорным модулям, а множество ребер  $V$  соответствуют межпроцессорным связям. Разделим множество  $P1$  на два непересекающихся подмножества:  $P1 = P \cup L$ , где  $P$  — множество основных процессоров, а  $L$  —

множество запасных процессоров. Идентификаторы процессоров множества  $P$  представим в виде матрицы  $P = \|p_{ij}\|_{n \times n}$ . Множество запасных процессоров  $L$  представим в виде матрицы  $L = \|l_{ij}\|_{N \times N}$ .

Размещение пакета подпрограмм, описываемых графом  $G$ , в мультимпьютере может быть описано отображением  $\beta_s = X_s \rightarrow P$ , где  $s$  — номер варианта размещения подпрограмм  $\{x_{qk}\}$  по процессорным модулям  $\{p_{qv}\}$  ( $s = \overline{1, N!}$ ). Мощность множества отображений  $\Psi = \{\beta_s\}$  равна числу перестановок номеров задач  $\{x_{qv}\}$  в массиве  $X$ :  $|\Psi| = N!$ . Для описания множества длин  $d_{ij}$  кратчайших маршрутов передачи данных введем массив  $D = \|d_{ij}\|_{N \times N}$ , который строится, исходя из матрицы смежности, соответствующей графу  $G$ .

Пусть  $\Psi$  — множество всевозможных отображений вида  $\beta_s$ . Тогда задачу размещения [3] можно сформулировать как поиск такого отображения  $\beta^* \in \Psi$ , что

$$T_{\beta^*} = \min_{\Psi} \left\{ \max_{\beta \in \Psi} \left\{ T_{\beta}(p_{a,b}, p_{x,y}) \right\} \right\}, \quad (1)$$

где  $T_{\beta}(p_{a,b}, p_{x,y})$  — задержка при передаче данных, определяемая временем обработки принятых и/или подготовки пакета передаваемых данных между процессорными модулями  $p_{a,b}$  и  $p_{x,y}$  соответствующих отображению  $\beta$ , и вычисляемая по формуле

$$T_{\beta}(p_{a,b}, p_{x,y}) = d_{ij} \cdot m_{ij}. \quad (2)$$

Пусть массив  $Z = \|z_{ij}\|_{N \times N}$  содержит флаги, показывающие исправность процессоров  $p_{q,v} \in P1$ :

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

где  $\alpha = \overline{1, n}, \beta = \overline{1, n}$ , тогда массив  $\Theta = \|\Theta\|_{N \times N}$  содержит флаги, показывающие исправность запасных процессоров  $l_{i,j}, i = \overline{1, n}, j = \overline{1, n}$ . Учитывая

$$\Theta_{\alpha,\beta} = \begin{cases} 1, & \text{если } \Theta_{q,v} \text{ неисправен;} \\ 0, & \text{если } \Theta_{q,v} \text{ исправен,} \end{cases}$$

где  $\alpha = \overline{1, n}, \beta = \overline{1, n}$ . Пусть массив  $K = \|k_{ij}\|_{N \times N}$  содержит флаги, показывающие исправность связей межпроцессорных связей  $p_{q,v} \in P1$ :

$$k_{\alpha,\beta} = \begin{cases} 1, & \text{если } p_{q,v} \text{ неисправна;} \\ 0, & \text{если } p_{q,v} \text{ исправна,} \end{cases}$$

где  $\alpha = \overline{1, n}, \beta = \overline{1, n}$ . При условии, что в  $P = \|p_{ij}\|_{n \times n}$  отказал процессор  $p_{ij}(i = \overline{1, n}, j = \overline{1, n})$ , симво-

лом  $\Leftrightarrow$  обозначим поиск в  $\Theta$   $l_{s,t}$  ( $s = \overline{1, n}, t = \overline{1, n}$ ):  $(|i-s| \rightarrow \min) \& (|j-t| \rightarrow \min)$ . Процедуру выполнения реконфигурации подпрограмм в соответствии с [3] определим символом *falloc*. При отказе межпроцессорной связи по [4] применим алгоритм Дейкстры и обозначим это символом *dic*.

Предлагаемый метод отказоустойчивого перераспределения подпрограмм является продолжением идей, представленных в [2–4] и состоит из следующих этапов.

1. введём массивы: смежности, расстояний, исправности основных и запасных процессоров;
2. введем массивы: обход 1 и обход 2;
3. при отказе основного процессора в массиве запасных процессоров ищется ближайший свободный процессор;
4. если резервный процессор найден, то необходимо произвести реконфигурацию системы путём переназначения неисправного процессора на соответствующий запасной, иначе п. 2;
5. выполняем поиск свободного запасного процессора и п. 2, иначе необходима полная замена массива процессоров;
6. выполнить реконфигурацию;
7. если отказала межпроцессорная связь, то начальной точкой обхода примем процессор  $p_{q,v}$ , а конечной точкой —  $p_{q+1,v+1}$  ( $q = \overline{1, n}; v = \overline{1, n}$ );
8. применяем алгоритм Дейкстры для начальной и конечной точек обхода, и п. 9;
9. введём переменные  $x, y$ , имеющие первоначально одинаковые значения, в которых хранится объём передаваемой информации;
10. введём два динамических массива  $a[i], b[i]$ ;
11. массивам  $k[1, i] := 0; l[1, j] := 0$ ;
12. начальной точкой возьмём  $i$ -й процессор и установим ему флаг 0;
13. суммируя объём передаваемой информации, перебором ищем всевозможные пути до конечной точки обхода;
14. в массиве  $b[i]$  запоминаем найденный путь;
15.  $a := \min\{x, y\}; a[i] := b[i]$ ;
16. повторяем п. 14–16 для всех остальных путей;
17. по найденному маршруту в массив смежности прибавляется необходимое количество байтов.

С учетом вышеизложенного предложена система с учётом отказа одного из процессоров и/или межпроцессорных связей путём её реконфигурации (рис. 1а). Для системы, представленной на рис. 1а, массивы размещения, смежности, расстояний, путей и весов выглядят, как показано, соответственно, на рис. 1б-д.

Предлагаемая программная система позволяет выполнять следующие функции: задать размерность мультимпьютерной системы, частоту процессора, частоту



отказов. Предполагаются следующие возможности варианты приоритета отказов: только межпроцессорные связи, только процессоры, межпроцессорные связи отказывают чаще процессоров, процессоры отказывают чаще межпроцессорных связей и одинаковый приоритет как процессоров, так и межпроцессорных связей. Программная система позволяет задавать матрицу  $W$ , оценивает время, необходимое для реконфигурации системы и показывает суммарную длину межпроцессорных связей.

Например, для мультипроцессорной системы размером  $7 \times 7$  были получены следующие результаты: с учетом необходимости пяти межпроцессорных тактов на одну на перестройку внутренних связей при тактовой частоте работы процессора 500 МГц.

В случае выбора режима отказа только межпроцессорных связей полный отказ системы наступил после 331 отказа. При этом время наработки на отказ составляет  $10^{-9}$  сек.

В режиме работы отказов только процессоров получены следующие результаты: фатальный отказ системы наступает после 40 перестановок, время наработки на отказ при этом составило  $420 \times 10^{-9}$  сек при затрате на одну замену резервным процессором в среднем  $20 \times 10^{-9}$  сек.

В режиме работы «межпроцессорные связи отказывают чаще процессоров» получены следующие результаты: полный отказ системы наступает после 140 перестроек системы, через  $840 \times 10^{-9}$  со средним временем одной перестройки  $40 \times 10^{-9}$  сек.

Режим работы «процессоры отказывают чаще межпроцессорных связей» дал следующие результаты: полный отказ системы наступает после 90-й перестройки. При этом фатальный отказ происходит через  $780 \times 10^{-9}$  со средним временем на одну перестройку  $30 \times 10^{-9}$  сек.

Вариант «процессоры и межпроцессорные связи отказывают с одинаковой частотой» дал следующие результаты: фатальный отказ наступает после 99 перестройки за время  $660 \times 10^{-9}$  сек.

Анализ полученных результатов позволяет сделать вывод о том, что предложенный метод реконфигурации системы может быть применен к многопроцессорным системам. В случае высокой доступности (отслеживание, наблюдение, полеты и т.д.) полученное время реакции на отказ может оказаться недостаточным и необходимо применение дополнительных аппаратных средств, что является предметом дальнейших исследований.

#### ЛИТЕРАТУРА

1. Борзов Д.Б., Басов Р.Г., Титов В.С., Соколова Ю.В. Устройство планирования загрузки процессоров в мультипроцессорных системах критического назначения // Труды МАИ, 2020. Выпуск № 115.
2. Борзов Д.Б., Соколова Ю.В. Методика перераспределения подпрограмм в отказоустойчивых мультикомпьютерах. Сборник трудов XVIII Международной научно-технической конференции «Машиностроение и техносфера XXI века. Т1». Донецк, 2011. С. 90–93.
3. Борзов Д.Б., Соколова Ю.В. Перераспределение подпрограмм в отказоустойчивых мультикомпьютерах при отказе связей. Сборник материалов X Международной конференции «Опτικο-электронные приборы и устройства в системах распознавания образов, обработки изображений и символической информации». Курск, 2010. С. 238–240.
4. Борзов Д.Б. Метод снижения коммуникационной задержки путем субоптимального размещения задач в матричных базовых блоках кластера // Телекоммуникации, 2008. № 4.
5. Левитин А.В. Глава 9. Жадные методы: Алгоритм Дейкстры // Алгоритмы: введение в разработку и анализ. М.: «Вильямс», 2006. С. 189–195.

© Соколова Юлия Васильевна (sokolovaiuv@laspace.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»