

ИНФОРМАТИКА

УДК 004.05

ОБ ОПТИМИЗАЦИИ ПРОЦЕДУРЫ РЕКОНФИГУРИРОВАНИЯ В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ

В. В. Гроль, В. А. Романкевич, С. М. Мораведж

Национальный технический университет Украины, Киев
E-mail: romankev@scs.ntu-kpi.kiev.ua

В работе рассматриваются вопросы, связанные с организацией оптимизированной по ряду критериев процедуры настройки (программирования) компонентов многопроцессорной системы с учетом компенсации возможных отказов исправными компонентами. Разработан алгоритм формирования подмножеств процессорных элементов и получены аналитические соотношения, позволяющие оптимизировать временные либо аппаратные затраты в таких системах.

Ключевые слова: многопроцессорные системы, реконфигурация, отказоустойчивость.

On Optimization of Reconfiguration Procedure in Multiprocessor Systems

V. V. Grol, V. A. Romankevich, S. M. Moravej

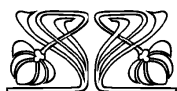
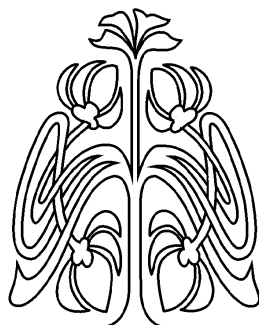
Some questions concerned to organization of programming procedure being optimized in accordance with a number of parameters for multiprocessor fault-tolerant system's components in consideration with compensation by properly functioned components of possible faults are reviewed in the paper. An algorithm to form a subsets of processor elements is developed and an anayitic expressions allow to reduce the time or hardware spendings in such systems are produced.

Key words: multiprocessor systems, reconfiguration, fault-tolerance.

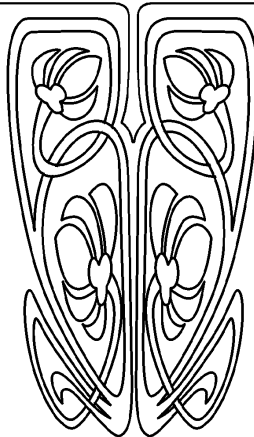
ВВЕДЕНИЕ

Отказоустойчивые реконфигурируемые многопроцессорные системы (ОМС) управления [1], о которых речь идет ниже, обладают рядом особенностей, среди которых взаимное тестирование и реконфигурирование при обнаружении отказов своих компонентов [2] с целью обеспечения выполнения всех функций управления объектом. Бывает так, что множество процессоров, которые могут «подхватить» задачи, решаемые вышедшим из строя процессором, предопределено чисто инженерными соображениями (число связей и источников внешней информации, объем избыточности по производительности и времени и др.). Однако часто в этом плане остается определенный выбор, что порождает свои задачи оптимизации.

Если обозначить число компонентов ОМС как N , то каждому из элементов можно поставить в соответствие некоторое подмножество из k элементов, $|k| = 1, 2, \dots, N - 1$. При этом возникает задача нахождения соответствующего подмножества для каждого элемента, т. е. задача разбиения N элементов на N подмножеств с учетом возможности вхождения каждого из элементов в несколько подмножеств. Достаточно легко можно показать, что с точки зрения оптимального использования памяти программ процессорных элементов предпочтительным вариантом представляется ситуация, когда каждый элемент входит в k подмножеств, т. е. каждый процессор может участвовать в компенсации отказов одного из k процессоров.



НАУЧНЫЙ
ОТДЕЛ





1. АЛГОРИТМ РАСПРЕДЕЛЕНИЯ ПРОЦЕССОРОВ ПО ПОДМНОЖЕСТВАМ

В основу алгоритмов решения поставленной задачи может быть положен, например, учет топологии связей компонентов системы. Так, использование алгоритма распределения для систем со слабой связностью компонентов (топология «кольцо», «последовательная цепочка» и др.), будет оптимальным, если в каждое подмножество включать k «соседних» (ближайших по цепи) элементов. Пусть i — номер некоторого модуля (процессорного элемента), $i = 1, 2, \dots, N$. Предположим, что при этом справедливы условия:

$$\begin{cases} \forall i = 1, 2, \dots, N \quad \exists K_i, i \in K_i \mid K_i = k + 1 \\ \forall i = 1, 2, \dots, N \quad K_i = K_{im} \cup K_{ip}, K_{im} \cap K_{ip} = \emptyset \\ \forall i = 1, 2, \dots, N \quad [0 \leq |K_{im}| \leq (k + 1), 0 \leq |K_{ip}| \leq (k + 1)]. \end{cases} \quad (1)$$

В последних соотношениях K_i , K_{im} и K_{ip} — некоторые подмножества элементов, причем K_{im} и K_{ip} составляют искомое подмножество K_i . Эти два подмножества (K_{im} и K_{ip}) включают в совокупности $k + 1$ имен «соседних» (по кольцу) элементов, т. е.:

$$\begin{cases} K_{im} = \{(i - |K_{im}|), (i - |K_{im}| + 1), \dots, (i - 2), (i - 1)\} \\ K_{ip} = \{(i + 1), (i + 2), \dots, (i + |K_{ip}| - 1), (i + |K_{ip}|)\}. \end{cases} \quad (2)$$

Учитывая кольцевую топологию связей в ОМС, операции суммирования и вычитания в соотношениях (2) выполняются по модулю N . Алгоритм распределения достаточно прост и в обобщенном виде представлен ниже.

Очевидным представляется условие, чтобы числа $|K_{im}|$ и $|K_{ip}|$ отличались не больше, чем на единицу (при нечетном значении величины $k + 1$ наилучшим вариантом будет $|K_{im}| = |K_{ip}|$). Понятно также, что среди процессоров множества K_i не должно быть неисправных, для чего после выявления любого отказа нумерацию процессоров следует изменить.

Для альтернативного варианта топологии с большей степенью связности компонентов ОМС (например, связи типа «каждый с каждым») условие формирования подмножеств элементов, находящихся достаточно близко друг к другу, снимается. Таким образом, появляется возможность произвольного (случайного) выбора элементов ОМС при образовании подмножеств для выполнения процедур реконфигурации системы в случае возникновения неисправностей.

С учетом вышеизложенных соображений обобщенный алгоритм разбиения компонентов ОМС по подмножествам для ОМС с большой степенью связности элементов можно представить в виде следующей последовательности действий.

1. Для заданных значений N и k выделить N счетчиков $T_i, i = \overline{1, N}$, разрядность каждого из счетчиков равна $\lceil \log_2 k \rceil$ бит, и, кроме того, сформировать N одномерных массивов $A_i, i = \overline{1, N}$, для хранения k идентификаторов выбранных компонентов ОМС в каждом из этих массивов.

2. Выполнить начальную установку счетчиков $\forall i = 1, 2, \dots, N \quad (\langle T_i \rangle = 0)$.

3. Установить счетчик компонентов $i = 1$.

4. Присвоить значение счетчику компонентов текущего подмножества $\langle K \rangle = k$.

5. Выбрать случайно величину $j, j \in \{1, 2, \dots, N\}, j \neq i$ учитывая, что при этом $\langle T_j \rangle$ имеет минимальное значение.

6. Проверить условие: $\langle T_j \rangle = k$? Да: перейти к п. 5. Нет: перейти к п. 7.

7. Присвоить: $\langle K \rangle = \langle K \rangle - 1$.

8. Записать значение j в массив A_i .

9. Проверить условие: $\langle K \rangle = 0$? Да: перейти к п. 10. Нет: перейти к п. 5.

10. Присвоить: $i = i + 1$.

11. Проверить условие $i > N$? Да: перейти к п. 12. Нет: перейти к п. 4.

12. Конец алгоритма.



Отметим, что кажущиеся нарушения требования п. 5 алгоритма о минимальности состояния, соответствующего выбранного счетчика T_i , связано с тем, что минимальное состояние $T_4 = 1$ не может являться основанием для включения компонента «четыре» в подмножество идентификаторов, хранящееся в массиве A_4 вследствие равенства индексов T_4 и A_4 . Очевидно, что некий компонент не может входить в подмножество других компонентов ОМС, выполняющих рабочие функции именно этого процессорного элемента системы в случае его (элемента) неисправности. В п.5 алгоритма указанное соображение отражено в виде условия $j \neq i$.

2. ОПТИМИЗАЦИЯ ВЫБОРА ПАРАМЕТРА k

Применение одного из вариантов распределения компонентов реконфигурируемой ОМС на подмножества может быть использовано для формирования данных заполнения памяти программ процессорных элементов ОМС. Тот факт, что в случае отказа одного из элементов системы k других элементов начинают решать помимо собственной основной задачи еще и часть задачи отказавшего элемента, приводит к появлению возможности оптимального выбора параметра k в соответствие с некоторыми критериями. Одним из таких критериев может оказаться общее время выполнения любым из k компонентов как своей задачи, так и задачи по реконфигурированию системы. Допустим, что цикл решения собственной задачи любого процессорного элемента включает в себя два временных интервала, а именно интервал самотестирования t_{st} и интервал собственно решения задачи, т. е. выполнения основной системной функции t_{tk} . При вхождении системы в режим реконфигурирования к этим двум интервалам (тактам) добавляется еще три временных составляющих: такт t_{sb} решения части задачи отказавшего процессорного элемента, такт t_{cl} получения результатов выполнения тактов t_{sb} остальных $k-1$ «резервных» процессорных элементов и обработки этой информации, а также такт t_{out} — время выдачи управляющих сигналов (выходной информации) взамен сигналов отказавшего компонента ОМС.

Обозначим через α_{sb} некоторый коэффициент, определяющий степень ускорения (замедления) решения задачи отказавшего процессора k «резервными» элементами. Тогда можно записать, что $t_{sb} = (\alpha_{sb} \cdot t_{tk})/k$.

Для интервала сбора (объединения) результатов решения подзадач: $t_{cl} = \alpha_{cl} \cdot k \cdot t_{op}$, где: t_{op} — время обработки результатов решения дополнительной задачи одним из «резервных» процессорных элементов, α_{cl} — некоторый коэффициент, учитывающий эффективность (производительность) алгоритма сборки, $0 < \alpha_{cl} \leq 1$.

Если обозначить через α_{out} долю операций вывода при решении основной задачи, тогда $t_{out} = \alpha_{out} \cdot t_{tk}$, причем очевидно, что $0 < \alpha_{out} < 1$. Следовательно, время t_{cm} выполнения функций реконфигурирования каждым из k «резервных» компонентов ОМС можно найти следующим образом:

$$t_{cm} = t_{sb} + t_{cl} + t_{out} = \alpha_{sb} \cdot (t_{tk}/k) + \alpha_{cl} \cdot k \cdot t_{op} + \alpha_{out} \cdot t_{tk}.$$

Для определения оптимального значения k , при котором время решения задачи реконфигурации будет минимальным, найдем частную производную:

$$\frac{\delta t_{cm}}{\delta k} = -\alpha_{sb} \cdot \frac{t_{tk}}{k^2} + \alpha_{cl} \cdot t_{op},$$

приравняв к 0 которую, получаем:

$$k_{opt}(t) = \sqrt{\frac{\alpha_{sb} \cdot t_{tk}}{\alpha_{cl} \cdot t_{op}}}.$$

Другим критерием оптимального разбиения может быть объем памяти для хранения специализированного программного обеспечения в каждом из процессоров ОМС. Если обозначить память процессора как M_p , то распределение этой памяти на отдельные зоны можно записать в виде соотношения:

$$M_p = m_t \circ m_{tk} \circ M_{p1} \circ M_{p2} \circ \dots \circ M_{pk},$$



где \circ — символ операции конкатенации областей памяти, m_t — зона хранения программы самотестирования процессора, m_{tk} — зона памяти для хранения рабочей (системной) программы процессора, M_{pi} ($i = 1, 2, \dots, k$) — области хранения информации, используемой в режиме компенсации последствий отказов при появлении неисправного компонента в ОМС. При этом

$$\forall i = 1, 2, \dots, k \quad (M_{pi} = m_{st} \circ m_{col} \circ m_{out}),$$

причем m_{st} хранит программу решения соответствующей подзадачи отказавшего процессора, m_{col} — зона памяти для хранения подпрограммы сбора результатов выполнения k подзадач, m_{out} предназначена для хранения подпрограммы выдачи информации вместо отказавшего процессорного элемента системы.

Допустим, что $m_{st} = f(k) \cdot (m_{tk}/k)$, где $f(k)$ — некоторая нелинейная функция, определяемая особенностями процедуры разделения задачи, которую решает неисправный процессор, на k подзадач. Предположим, что $f(k) = \beta_{st}/k$, где β_{st} — некий коэффициент, определяемый эффективностью процедуры распараллеливания. Так как $0 < f(k) \leq 1$, то $0 < \beta_{st} \leq k$. Следовательно, $m_{st} = (\beta_{st} \cdot m_{tk})/k$. Обозначая через β_{col} коэффициент, определяемый эффективностью процедуры сборки результатов решения k задач, получаем

$$m_{col} = \beta_{col} \cdot k \cdot m_{st} = (\beta_{col} \cdot k \cdot \beta_{st} \cdot m_{st})/k^2 = (\beta_{col} \cdot \beta_{st} \cdot m_{tk})/k.$$

Тогда объем памяти программ m_{cm} каждого процессора ОМС (без учета объема памяти для хранения процедур самотестирования и решения собственных системных задач) определяется, следующим образом:

$$m_{cm} = k(m_{st} + m_{col} + m_{out}) = [(\beta_{st} \cdot m_{tk})/k] + \beta_{col} + \beta_{st} + m_{tk} + m_{out}.$$

Находим частную производную:

$$\frac{\delta m_{cm}}{\delta k} = -\frac{\beta_{st} \cdot m_{tk}}{k^2} + m_{out}$$

и, приравнявая ее к 0, получаем (не учитывая отрицательный корень):

$$k_{opt}(m) = \sqrt{\frac{\beta_{st} \cdot m_{tk}}{m_{out}}}.$$

ЗАКЛЮЧЕНИЕ

Предложенные в данной работе алгоритмы распределения процессорных элементов ОМС на подмножества, а также полученные аналитические соотношения для нахождения настроечных параметров, позволяют на этапе проектирования системы оптимизировать программно-алгоритмические средства, обеспечивающие эффективное выполнение процедур реконфигурирования ОМС в случае появления отказовых ситуаций. В частности, инженер-проектировщик системы получает возможность оптимизировать специализированные программные средства ОМС в соответствии с альтернативными критериями, а именно временными затратами на выполнение функций реконфигурирования либо объемом блоков памяти процессорных элементов для хранения рабочего и служебного программного обеспечения ОМС.

Библиографический список

1. Каляев И. А., Левин И. И., Семерников Е. А., Шмойлов В. И. Реконфигурируемые мультимедийные вычислительные структуры / под общ. ред. И. А. Каляева. Ростов н/Д: Изд-во ЮНЦ РАН, 2008. 320 с
2. Белявский В. Е., Валуйский В. Н., Романкевич А. М., Романкевич В. А. Самодиагностируемые многомодульные системы : некоторые оценки тестирования // Автоматика и телемеханика. 1999. № 8. С. 148–153.