

E-mail: aleksunin@gmail.com
344090, Ростов-на-Дону, ул. Мельчакова, 10
Тел.: +7(8632)696991

Aleksunin Eugene Sergeevich
Federal State-Owned Educational Establishment of Higher Vocational Education
«Southern Federal University»
E-mail: aleksunin@gmail.com
10, Melchikova street, Rostov-on-Don, 344090
Phone: +7(8632)696991

УДК 681.78

В.В.Коробкин, Е.И.Чернов

**ОДИН ИЗ ПОДХОДОВ К РАСПРЕДЕЛЕНИЮ ЗАДАЧ ПО
РЕСУРСАМ УПРАВЛЯЮЩЕЙ СИСТЕМЫ МЕХАТРОННОГО
КОМПЛЕКСА**

Предлагаемый метод распределения задач в компьютерной сети позволяет сократить накладные расходы на решение программы оптимизации и повысить надежность работы системы в условиях отказа (деградации) ресурсов.

Компьютерная сеть; метод коллективного принятия решений; перераспределение задач; итерационная оптимизация.

V.V.Korobkin, E.I.Chernov

**ONE OF APPROACHES TO DISTRIBUTION OF PROBLEMS
ON RESOURCES CONTROL SYSTEM OF MECHATRONICS
COMPLEX**

The offered method of distribution of problems in a computer network allows to reduce an overhead charge for the decision of the program of optimisation and to raise reliability of work of system in the conditions of refusal (degradation) of resources.

Computer network; method of collective decision-making; redistribution of problems; iterative optimisation.

Система управления сложного мехатронного комплекса представляет многоуровневую иерархию средств промышленной автоматизации, основу которых составляют промышленные компьютеры, промышленные контроллеры, датчики, исполнительные механизмы, пульта, мониторы, сенсорные панели, принтеры и др. Обмен информацией между ресурсами осуществляется с использованием промышленных шин типа Modbus, CAN, Profibus и др. Использование избыточной (резервной) аппаратуры для обеспечения надежности и безопасности функционирования мехатронного комплекса, большого набора задач и функций (управление, диагностика, мониторинг, прогноз, моделирование, регистрация событий, документооборот) порождает проблему динамического распределения задач по ресурсам системы управления (СУ).

В централизованных иерархических сетях задача распределения решается, как правило, с помощью специально выделенного компьютера (host – машины). Недостаток такого подхода – низкая живучесть системы при выходе из строя центрального компьютера.

Возникает актуальная проблема разработки методов распределения потока задач в промышленных сетях, с помощью которых можно было бы оптимизировать это распределение с точки зрения некоторого критерия, а также обеспечить надежность работы системы в условиях отказа (деградации) ресурсов [1].

Данная задача разбивается на два этапа:

- при начальном пуске системы, при смене технологической операции с верхних уровней УС поступает поток задач, которые необходимо загрузить в компьютеры УС. Если произошел отказ одного или нескольких активных модулей, УС не должна останавливать процесс управления, а производит определение отказавших модулей, выключение их из процесса и реконфигурацию системы. Задачи, которые должны были выполняться в отказавших модулях, оптимально распределяются по рабочим ресурсам системы;

- перераспределение задач, которые произвольным (случайным) образом распределены по компьютерам сети.

Проблему распределения задач в компьютерной сети формально можно представить следующим образом.

Часть 1. Для выполнения совокупности несвязанных между собой задач имеется компьютерная сеть (рис. 1), содержащая n контроллеров $C_r (r = 1 \dots n)$ разной производительности V_r , причем последовательность и время поступления задач определяется заранее известной циклограммой выполнения глобальной задачи, например задачи управления технологическим производством (процессом), моделирования в реальном времени системы управления подвижными объектами и т.п. и/или по команде извне (от компьютера более высокого уровня иерархии, оператора системы и др.) [2, 3]. Аналогично загрузка ресурсов сети производится в случае выхода из строя любого контроллера. Задачи, которые должны были решаться в нем, распределяются по «живой» системе. Это возможно в том случае, если копии этих задач хранятся в памяти каждого компьютера.



Рис. 1. Структура многоконтроллерной системы (компьютерной сети)

Поток задач Z в текущий момент времени содержит m задач $z_i \in Z$ ($i=1 \dots m$). Задача z_i имеет вычислительную сложность H_i . Время выполнения контроллером $C_r (r = 1 \dots n)$ задачи z_i ($i=1 \dots m$) оценивается следующим выражением:

$$T_{ri} = h(V_r, H_i),$$

где T_{ri} – время выполнения r -м компьютером i -й задачи; h – некоторая функция.

При поступлении на вход системы нового потока задач Z проводится процедура отыскания компьютера C_r с минимальной загрузкой (Tr_{min}) и очередная

задача z_i из множества Z поступает в компьютер Cr и время $Tr_{\min i}$ решения задачи суммируется с Tr_{\min} и получаем новое значение Tr .

Задача z_i исключается из подмножества Z , получаем новое подмножество

$$Z_{m-i} = Z_m \setminus z_i.$$

Далее опять отыскивается компьютер Cr с минимальным временем решения задач своего подмножества (Tr_{\min}) и очередная задача $z_i^{k+1} \in Z^{k+1}$ поступает в этот компьютер. Здесь $k=i$ – шаг распределения задач. Процедура распределения повторяется до тех пор, пока все задачи потока Z не получат место в соответствующих контроллерах.

Это напоминает игру «тетрис», когда падающую фигуру укладывают таким образом, чтобы не было пустот и произошло выравнивание на дне колодца.

Часть 2. Перераспределение задач по компьютерам.

Пусть в данной системе на выполнении находится группа несвязанных между собой задач Z , содержащая в текущий момент времени m задач $z_g \in Z(g = 1, m)$, которые произвольным (случайным) образом распределены по компьютерам.

В качестве критерия оптимизации распределения задач из множества Z по компьютерам $T_r (r = 1, n)$ целесообразно принять критерий минимума времени T выполнения всех задач из множества Z . В свою очередь, значение времени T_z определяется следующим выражением:

$$T_z = \max\{T_1, T_2, \dots, T_n\}, \quad (1)$$

где $Tr (r = 1, n)$ – время выполнения компьютером Cr , подмножества $Z_g \subseteq Z$ возложенных на него задач из множества Z . Это время определяется как

$$T_r = T_{ra} + T_{rb} + \dots + T_{rc},$$

где a, b, \dots, c – номера задач из множества Z , включенных в подмножества Z_j .

Необходимо найти такое разбиение множества задач Z на подмножества $Z_j (j = 1, n)$ (т.е. распределение задач потока Z , по компьютерам $C_j (j = 1, n)$, чтобы значение времени $T(1)$ выполнения всех задач множества Z было минимальным.

Метод полного перебора предполагает $C = (2^n - 1)^m$ вариантов [1], где n – число компьютеров в системе, а m – число задач в потоке.

Понятно, что осуществить такой перебор при изменении множества задач в потоке, особенно в реальном времени, практически невозможно.

Число анализируемых вариантов можно резко сократить ($C = 100 * m^2 (n-1)$), если применить для решения данной задачи метод коллективного планирования действий, предложенный в [1]. При этом существенное сокращение вариантов перераспределения задач по ресурсам системы может быть достигнуто в отличие от методики [1], если не перебирать в каждом итерационном шаге все задачи множества Z , а выбрать пару наиболее и наименее загруженных компьютеров ($T_{i \max}^k, T_{j \min}^k$), где ($i = 1, n$), ($j = 1, n$), затем из подмножества задач Z_i компьютера Pi выделяется некоторая задача $z_n \in Z_i^k$ и включается в подмножество Z_j^k . В результате получается новое подмножество

$$Z_j^{k+1} = Z_j^k \cup z_n,$$

где $K=0$ – начальный цикл итерации, а $K+1$ – следующий.

При этом подмножество задач в компьютере C_i уменьшается на эту задачу и соответственно уменьшится $T_i^{k+1} = T_{i \max}^k - T_{i \min}^{k+1}$, а подмножество задач в компьютере C_j увеличится, следовательно $T_j^{k+1} = T_{j \min}^k + T_{j \max}^{k+1}$. Другими словами, одна из задач наиболее нагруженного компьютера перемещается в наименее нагруженный компьютер. Назовем компьютер C_i инициатором перераспределения и этим номером будем индексировать переменные. Далее для $K+1$ цикла найдем

$$T_z^{k+1} = \max (T_1^{k+1}, T_2^{k+1}, \dots, T_n^{k+1}).$$

Затем определим

$$\Delta T_3^{k+1} = T_3^{k+1} - T_3^k.$$

Если ΔT_3^{k+1} оказывается меньше нуля, то новое распределение задач лучше, чем предыдущее, следовательно задача z_n включается в подмножество задач Z_j^{k+1} .

Если $\Delta T_i^{k+1} = 0$, ситуация не улучшилась и это может быть по двум причинам:

а) в системе существует по меньшей мере еще один компьютер C_w ($w = 1, n$) и ($w \neq j \neq i$), загруженный так же, как и i , т.е. время выполнения задач в котором $T_w^k = T_{imax}^k$. Следовательно, перераспределение задач прошло успешно и задача z_n включается в подмножество задач Z_j^{k+1} . И хотя время решения задач T_3^{k+1} в системе не изменилось, уменьшилась нагрузка на компьютер C_i ;

б) нагрузка компьютера C_j на $K+1$ шаге будет равна C_i на K -м шаге ($T_j^{k+1} = T_{imax}^k$), следовательно, новое распределение задач не лучше предыдущего и z_n не включается в подмножество задач Z_j^{k+1} .

Если ΔT_i^{k+1} больше нуля, то это также означает, что перераспределение задач не улучшает ситуацию, и поэтому подмножества Z_j^k и Z_i^k не изменяются. Далее компьютер j^k анализирует следующую задачу из подмножества Z_i^k и включает в свое множество задач Z_j^k ту, которая даст отрицательное значение величины ΔT_j^{k+1} .

В результате получаем новое перераспределение задач, т.е. новые подмножества, причем время T_3 всех задач системы будет меньше, чем при предыдущем распределении. В том случае, если перебор всех задач из множества Z_i^k не приведет к улучшению ситуации, т.е. $\Delta T_j^{k+1} \geq 0$, это говорит о том, что минимально нагруженный j^k -й компьютер (T_{jmin}) по быстродействию уступает i^k -му. Поэтому на этом шаге компьютер P_j игнорируется и опять отыскивается минимально нагруженный компьютер P_f (T_{fmin}), где ($f = 1, n-2$). Далее процедура повторяется: компьютер P_f анализирует задачи из подмножества Z_i^k и включает в свое множество задач Z_f^k ту, которая даст отрицательное значение величины ΔT_i^{k+1} . В результате получаем новое распределение задач, представляемое новыми подмножествами $Z_1^{k+1}, Z_2^{k+1}, \dots, Z_n^{k+1}$, которое дает меньшее по сравнению с предыдущим общее время T_3 решения всех задач множества Z . Следует заметить, что подмножества, которые не участвовали в перераспределении, не изменяются.

Затем ищется новая пара компьютеров (i, j) в результате получаем новое ($k+2$)-е перераспределение.

В каждом итерационном цикле один компьютер (самый загруженный) последовательно может анализировать все задачи, входящие в подмножество Z_i^k .

Аналогичным образом процедура итерационной оптимизации повторяется до тех пор, пока в очередном цикле не окажется, что никакое перераспределение задач не приводит к уменьшению времени решения T_3 , т.е.

$$T_3^{k+1} = T_3^k,$$

где T_3^{k+1} – время решения всех задач, получаемое после ($k+1$)-го итерационного цикла, T_3^k – время решения всех задач, получаемое после k -го итерационного цикла.

Максимальное число задач в подмножестве Z_i^k – m . Это может быть тогда, когда один компьютер загружен всеми задачами, а остальные имеют нулевую нагрузку.

Максимальное количество итерационных циклов $C = m - m/n$.

Следовательно, предложенный метод по трудоемкости лучше предыдущих.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Каляев И.А., Гайдук А.Р., Капустян С.Г. Распределенные систем планирования действий коллектива роботов. – М.: Янус-К, 2002. – С.292.

2. Информационная технология: Рекомендации Р50-34.119-90. Архитектура локальных вычислительных сетей в системах промышленной автоматизации. –М.: Госстандарт, 1991
3. *Кругляк К.В.* Промышленные сети: цели и средства. // Современные технологии автоматизации, 2002. № 4.

Коробкин Владимир Владимирович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г.Таганроге

E-mail: yvk@mvs.tsure.ru

347928, Таганрог, ул. Чехова, 2

Тел.: +7(8634)311865

Чернов Евгений Иванович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г.Таганроге

E-mail: kafmps@ttpark.ru.

Korobkin Vladimir Vladimirovich

Taganrog Institute of Technological – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

E-mail: yvk@mvs.tsure.ru

2, Chehova street, Taganrog, 347928, Russia

Phone: +7(8634)371723

Chernov Eugeny Ivanovich

Taganrog Institute of Technological – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

E-mail: kafmps@ttpark.ru

УДК 658.51.011.5.001.2(075.5)

Т.А. Пьявченко, В.М. Карась

АЛГОРИТМЫ ЦИФРОВОГО УПРАВЛЕНИЯ ПРОЦЕССОМ ГОРЕНИЯ В ТОПКЕ ПАРОГЕНЕРАТОРА

В работе рассматриваются особенности цифровой реализации алгоритмов управления процессом горения в топке парогенератора. По результатам работы сделано заключение о компенсации нагрузки парогенератора благодаря использованию корректирующего регулятора.

Алгоритмы управления; погрешности вычислений; шаг дискретности; компенсирующий регулятор.

T.A. Pyavchenko, V.M. Karas

ALGORITHMS OF DIGITAL CONTROL COMBUSTION ACTION IN A STEAM GENERATOR FIREBOX

In work are considered particularities to digital realization of algorithms of control combustion action in a steam generator firebox. As a result of work is made a conclusion about compensation of the load a steam generator, due to use correcting regulator.