УДК 681.324

ПЕРЕРАСПРЕДЕЛЕНИЕ ЗАДАНИЙ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

к.т.н. С.А. Соколов, С.В. Кавун, Д.И. Романюк
(представил д.т.н., проф. Е.И. Бобыр)

В данной статье предлагается алгоритм адаптивной реконфигурации информационно - вычислительной системы за счет перераспределения задач при отказах на основе решения транспортной задачи.

Создание информационно - вычислительных сетей (ИВС) связано с решением ряда научных и прикладных проблем. Одной из таких является повышение эффективной работоспособности или живучести.

В работе рассматривается алгоритм принятия решения в блоке анализа подсистемы операционной системы (ОС) (рис.1), где:

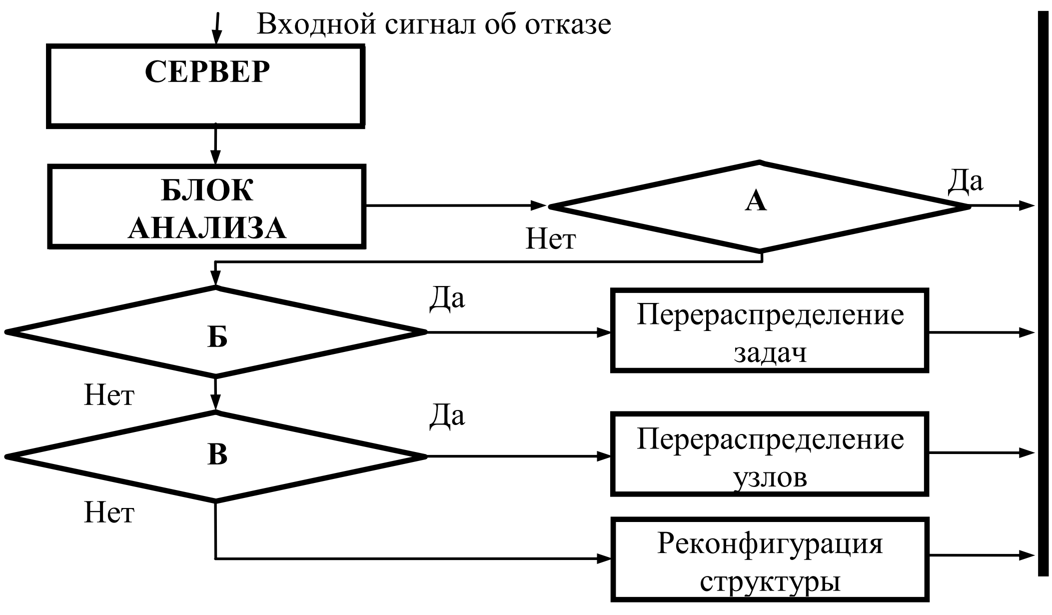

Рисунок 1 - Алгоритм принятия решения в блоке анализа подсистемы ОС

А - определяет, возможно, ли восстановление без перераспределения;

Б - определяет, возможно, ли перераспределение на 1 - м уровне;

В - определяет, возможно, ли перераспределение на 2 - м уровне.

Как только в сети произошел сбой, администратор производит опрос оставшихся узлов на предмет наличия свободного ресурса для принятия задач с отказавшего узла. Таким образом, формируется база данных свободных ресурсов узлов. Запрос производится специально сформированными сигналами, которые на производительность опрашиваемых узлов не оказывают никакого влияния.

После получения сигнала о сбое при невозможности восстановления нормальной работоспособности данного узла, из множества оставшихся узлов находятся такие, которые смогли бы принять выполнение задач с этого узла. Конечно, прием задач осуществляется в объеме свободного ресурса.

Тогда получаем следующие варианты:

1. $\sum_{i=1}^{M} V_{i_s} \le \sum_{j=1}^{K} W_j$ – перераспределение возможно;

2. $\sum_{i=1}^{M} V_{i_s} > \sum_{j=1}^{K} W_j$ – возможна потеря части информации.

<u>Задача А</u>: найти такой подграф $G` \in G$, вершины которого могли бы «принять» все V_s число задач, причем время передачи данных должно быть минимально.

$$T = \sum_{i \in \Omega j} t_{si} \Rightarrow \min, \quad i = \overline{1, n-1}, \;\; j = \overline{1, m}, \tag{1}$$

при условии, что

$$V_s = \sum_{i \in \Omega j} V_i \; . \tag{2}$$

Задачу (1 - 2) можно интерпретировать как транспортную задачу закрытого типа. Тогда математическая модель, после ее интерпретации, примет вид:

$$z = \sum_{i=1}^{M} \sum_{j=1}^{K} C_{ij} X_{ij} \to \min \; ;$$

$$\sum_{i=1}^{M} V_i = \sum_{j=1}^{K} W_j \; ;$$

$$\sum_{i=1}^{M} X_{ij} = v \; ;$$

$$\sum_{j=1}^{K} X_{ij} = w,$$

где X_{ij} - количество перераспределяемых задач;

C_{ij} - затраты на перераспределение задач.

Однако, в некоторых случаях возникает ситуация, когда при перераспределении задач необходимо переслать и исходные данные для их решения на свободных узлах. Тогда задача принимает вид:

$$z = \sum_{i=1}^{M}\sum_{j=1}^{K}(C'_{ij}X'_{ij} + C''_{ij}X''_{ij}) \to \min;$$

$$\sum_{i=1}^{M} V_i = \sum_{j=1}^{K} W_j;$$

$$\sum_{i=1}^{M} X'_{ij} = V, \quad V' \geq 0;$$

$$\sum_{j=1}^{K} X_{ij} = W.$$

В данной методике предлагается новый алгоритм решения задач подобного типа. Его идеология основывается на том факте, что поступление сигналов о сбоях происходит последовательно и, следовательно, при нескольких отказах можно рассматривать задачу о последовательном перераспределении задач с одной исходной вершиной.

<u>Предварительный шаг.</u> Приведение к равенству диафантового уравнения. Если свободные ресурсы не отказавших узлов превышают объем задач V_s, то есть $\sum_{i \in \Omega_j} V_i \neq V_s$, то всем свободным ресурсам на не отказавших узлах присваивается значение V_s.

<u>Шаг 1.</u> Определение подмножества доступных по объему свободных ресурсов вершин. Определяется подмножество $\{\Omega_j\}$ ($j = \overline{1,m}$) вершин, которые могут принять все или часть задач с отказавшего S - го узла. Подмножество определяется путем решения диафонтового уравнения вида:

$$\sum_{i=1}^{n-1} W_i * X_i = V_s, \text{ где}$$

$$X_i = \begin{cases} 1, & \text{если } i \subseteq \Omega_i; \\ 0, & \text{если } i \not\subset \Omega_i. \end{cases}$$

Результатом работы этой подпрограммы будет двоичная матрица одномерных векторов, где номера единиц указывают, что данные вершины или вершина могут принять все задачи с отказавшего узла. Количество таких векторов $d<<n$.

<u>Шаг 2.</u> Нахождение кратчайших путей от исходной вершины ко всем доступным. Для всех $i \in \{\Omega_j\}\ (j = \overline{1,m})$ находим множество кратчайших путей от исходной вершины ко всем доступным.

$$\left\{t^*_{K_j}\right\} = \min_{i \in \Omega_j}\left\{t_{si}\right\}, \text{ где } K_j \in \Omega_j$$

<u>Шаг 3.</u> Нахождение максимального значения из полученного множества. Во множестве Ω_j среди всех полученных на шаге 2 кратчайших путей выбираем максимальный путь, т. е. Находим

$$\left\{t^{**}_j\right\} = \max_{K_j}\left\{t^*_{K_j}\right\}.$$

Так как передача данных с одного узла на несколько других осуществляется практически одновременно, то задержкой в данном случае можно пренебречь, что и позволяет говорить об одновременности передачи информации. То же самое можно сказать и при передаче данных для задач.

$$\lim_{V_\hbar \to \infty} \Delta_{t_{si}} \to 0$$

<u>Шаг 4.</u> Нахождение минимального пути по заданному критерию оптимальности. Из всех полученных самых длинных кратчайших путей выбираем минимальный путь по заданному критерию

$$T = \min_j\left\{t^{**}_j\right\}.$$

И то множество путей Ω_j, в которое будет входить найденный минимальный путь и есть решение нашей задачи.

Решение этой задачи может быть реализовано известными методами [1, 2], но как видно из первоначальной формулировки, задача (1 - 2) относится к классу NP - полных задач. Разработанный же алгоритм имеет полиномиальную сложность.

ЛИТЕРАТУРА

1. Полунин И.Ф. Курс математического программирования. Минск, «Вышэйш. школа», 1970.

2. Системы и модели управления. Обоснование решений методами математического программирования. Методическое пособие. В.И. Вишневецкий, МО СССР, 1982.