ОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
(Вычислительные системы)
1981 год Выпуск 90

УДК 681.322.06+681.3.323

ОБ ОДНОМ ПОДХОДЕ К ДЕЦЕНТРАЛИЗОВАННОМУ РАСПРЕДЕЛЕНИЮ ЗАДАНИЙ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ

О.Г.Монахов

Вычислительной системой с программируемой структурой называется совокупность элементарных машин (ЭМ), объединенных сетью связи с распределенным по машинам, децентрализованным управлением [1,2]. Структура сети связи задана неориентированным графом $\Gamma = (V, H)$, где $H = \{(a,b) \mid a,b \in V\}$ - множество ребер, соответствующих линиям связи, $V = \{v_i \mid i \in \overline{1,N}\}$ - множество вершин, представляющих ЭМ [3,4]. Элементарные машины совмещают функции программно-управляемого коммутатора сети и вычислителя (универсальной ЭВМ).

По проблемам планирования, диспетчирования и распределения ресурсов в вычислительных системах имеется достаточно большое количество публикаций [1,5,6], но в них предполагается априорное знание характеристик поступающих заданий (таких как время исполнения, объем программы, число блоков программы и последовательность их исполнения).

В данной работе делается попытка сформулировать основные положения адаптивного децентрализованного управления распределением заданий, о которых известно только число требуемых для их исполнения ЭМ.

Отметим, что используемый в работе подход к распределению потоков заданий в вычислительной системе аналогичен подходу, предложенному в [7] для расчета маршрутов с минимальным временем передачи сообщений в сетях связи.

§I. Постановка задачи

Пусть в вычислительную систему, структура которой представлена графом $\Gamma = (V, H)$, поступает эргодический стационарный поток заданий с вектором интенсивностей $B = (\lambda_1, \lambda_2, \ldots, \lambda_N)$. Обозначим через Λ_i интенсивность общего потока заданий, поступающих в ЭМ$_i$, $1 \le i \le N$. Общий поток складывается из входного потока с плотностью λ_i и потока заданий, поступивших из соседних с ЭМ$_i$ машин. Обозначим через J_i множество номеров соседних с ЭМ$_i$ машин, введем также $E_i = J_i \cup \{0\}$, $1 \le i \le N$. Пусть φ_{ik} – доля общего потока заданий, поступивших в ЭМ$_i$, которая направляется в машину ЭМ$_k$, $k \in E_i$, при этом φ_{i0} – доля общего потока заданий, остающаяся на выполнение в ЭМ$_i$. Величины φ_{ik}, $1 \le i \le N$, $k \in E_i$, удовлетворяют следующим ограничениям:

$$\text{а)}\quad \varphi_{ik} \ge 0; \qquad \text{б)}\quad \sum_{k \in E_i} \varphi_{ik} = 1. \tag{1}$$

Заметим, что с учетом ограничений (I) величину φ_{ik}, $1 \le i \le N$, $k \in E_i$, можно интерпретировать как вероятность, с которой задание, поступившее в ЭМ$_i$, передается в соседнюю ЭМ$_k$, или при $k = 0$ – вероятность, с которой поступившее задание остается на выполнение в ЭМ$_i$.

Таким образом, выполняются следующие соотношения:

$$\Lambda_i = \sum_{k \in E_i} \Lambda_i \varphi_{ik}, \quad 1 \le i \le N; \tag{2}$$

$$\Lambda_i = \lambda_i + \sum_{k \in J_i} \Lambda_k \varphi_{ki} = \lambda_i + \sum_{k \in J_i} \lambda_{ki}, \quad 1 \le i \le N; \tag{3}$$

где $\lambda_{ki} = \Lambda_k \varphi_{ki}$, $i \in E_k$.

Назовем множество $\Phi = \{\varphi_{ik}\}$, $1 \le i \le N$, $k \in E_i$, планом распределения заданий. Примем, что задан некоторый функционал $W = W(\Gamma, B, \Phi)$ качества распределения заданий в системе, который зависит от графа сети связи системы (Γ), входного потока (B) и плана распределения заданий (Φ). Задача синтеза управления, в данной постановке, сводится к нахождению такого децентрализованного алгоритма распределения заданий в системе, который бы в условиях априорной неопределенности характеристик входного потока заданий (B) осуществлял выбор значений величин $\varphi_i \in \Phi$, $1 \le i \le N$, $k \in E_i$, обеспечивая экстремум функционала W при заданной структуре сети связи (Γ) и ограничениях (I):

$$W^*(\Gamma,B,\Phi^*) = \underset{\Phi}{\mathrm{extr}}\; W(\Gamma,B,\Phi). \qquad (4)$$

В общем случае задача оптимизации плана распределения заданий является задачей нелинейного программирования [8]. Рассмотрим в качестве примера два функционала, значения которых задают соответственно среднее время выполнения задания в системе и неравномерность загрузки машин системы.

ПРИМЕР I. Процесс выполнения задания, которое требует одновременно ресурсов n, $1 \le n \le N$, связанных ЭМ, т.е. подсистему ранга n, состоит из трех этапов. На первом этапе происходит перераспределение задания в соответствии с планом Φ и определение корневой ЭМ, т.е. ЭМ, относительно которой на втором этапе выполнения задания будет формироваться подсистема требуемого ранга. Определение направления, по которому должно передаваться задание из ЭМ$_i$, $1 \le i \le N$, на этапе перераспределения, происходит либо с помощью случайной процедуры по распределению вероятностей $\{\varphi_{ik}\} \subset \Phi$, $k \in E_i$, либо выбирается такое направление j, что $\varphi_{ij} = \max\limits_{k \in E_i} \varphi_{ik}$. При этом, если номер направления передачи равен нулю, то ЭМ$_i$ становится корневой ЭМ для вновь формируемой подсистемы. Процесс образования подсистемы аналогичен процессу, описанному в [9], и отличается тем, что включение новых ЭМ в подсистему происходит не равномерно по всем направлениям, а пропорционально величинам $\varphi_{ik} \in \Phi$. После того, как подсистема требуемого ранга сформирована, образующие ее ЭМ приступают к исполнению задания. Таким образом, время выполнения задания в вычислительной системе складывается из времени передачи задания по сети связи системы от точки поступления задания в систему до корневой машины подсистемы, предназначенной для его исполнения [9], и времени исполнения задания на подсистеме с учетом времени ее образования. Нас интересует T - среднее время, которое проводит задание в системе:

$$T = D + S, \qquad (5)$$

где D - среднее время с момента поступления задания в систему до момента его прихода в ЭМ, которая принимает данное задание на выполнение, S - среднее время с момента принятия решения о выполнении задания до конца его выполнения (до того момента, когда задание покинет систему).

Обозначим $\gamma = \sum\limits_{i=1}^{N} \lambda_i$. Среднее число заданий, ожидающих или использующих канал связи $(i,k) \in H$, равно $\lambda_{ik} D_{ik}$, где D_{ik} -

средняя задержка передачи задания по каналу связи (i, k). С другой стороны, среднее число нераспределенных заданий в сети связи системы равно $\gamma \cdot D$, откуда следует, что

$$\gamma \cdot D = \sum_{(i,k)\in H} \lambda_{ik} D_{ik}. \qquad (6)$$

Аналогично получается следующее равенство для заданий, находящихся на этапе выполнения:

$$\gamma \cdot S = \sum_{i=1}^{N} \lambda_{i0} S_i, \qquad (7)$$

где S_i - среднее время обслуживания (исполнения) задания в ЭМ$_i$. Подставляя (6) и (7) в (5), получаем:

$$T = \frac{1}{\gamma}\Big(\sum_{(i,k)\in H} \lambda_{ik} D_{ik} + \sum_{i=1}^{N} \lambda_{i0} S_i\Big). \qquad (8)$$

Примем, аналогично тому, как это сделано в [7], что величины D_{ik} и S_i зависят только от интенсивностей потоков соответственно λ_{ik} и λ_{i0}. В этом случае (8) перепишется в следующем виде (опуская постоянный коэффициент):

$$T = \sum_{(i,k)\in H} D_{ik}(\lambda_{ik}) + \sum_{i=1}^{N} S_i(\lambda_{i0}). \qquad (9)$$

Таким образом, задача синтеза управления распределением заданий в вычислительной системе при данном функционале качества (9) состоит в нахождении такого децентрализованного алгоритма, который бы выбирал план распределения заданий Φ, минимизирующий среднее время выполнения задания T при ограничениях (I) и заданной структуре сети связи системы Γ.

Использование критерия минимизации среднего времени выполнения задания для алгоритмов распределения влечет за собой использование достаточно сложных процедур оценки изменения задержки в линиях связи и изменения времени исполнения в ЭМ при изменении потоков в системе.

ПРИМЕР 2. Под нагрузкой машины ЭМ$_i$ понимается x_i - число заданий, ожидающих исполнения и исполняемых в этой ЭМ$_i$ [9]. Введем также понятие средней нагрузки по системе $Mx = (\sum_{i=1}^{N} x_i)/N$. Показатель неравномерности нагрузки по системе имеет следующий вид:

$$W = \sum_{i=1}^{N} w_i = \sum_{i=1}^{N} | x_i - Mx | . \tag{10}$$

Примем, что x_i, $1 \le i \le N$, зависит только от интенсивности λ_{i0}, поступающих на выполнение в ЭМ$_i$ заданий. В качестве оценки средней нагрузки по системе в ЭМ$_i$ используется величина средней нагрузки по множеству соседних с ЭМ$_i$ машин:

$$Mx_i = \frac{\sum_{k \in E_i} x_k}{|E_i|} .$$

В этом случае функционал (10) примет следующий вид:

$$W = \sum_{i=1}^{N} w_i(\lambda_{k0},\ k \in J_i \cup i) = \sum_{i=1}^{N} |x_i(\lambda_{i0}) - Mx_i(\lambda_{k0},\ k \in E_i)| . \tag{11}$$

Таким образом, целью алгоритма распределения заданий при данном критерии (11) является минимизация неравномерности нагрузки по системе путем выбора соответствующих значений величин $\varphi_{ik} \in \Phi$, $1 \le i \le N$, $k \in E_i$. Выбор функционала (11) в качестве критерия эффективности алгоритма распределения заданий продиктован, во-первых, стремлением равномерно загрузить ресурсы системы (как элементарные машины, так и сеть связи), во-вторых, стремлением поставить каждую ветвь выполняемого параллельного алгоритма по возможности в одинаковые условия, тем самым сократив время, необходимое на их синхронизацию, и, в-третьих, стремлением создать достаточно простой (в смысле процедур оценки состояния системы) и работоспособный в условиях априорно-неопределенных характеристик поступающих заданий алгоритм.

§2. Условия оптимальности плана распределения заданий

Рассмотрим необходимые условия минимума функционала W в задаче (4), считая его непрерывным вместе со своими первыми частными производными: $\frac{\partial W}{\partial \varphi_{ij}}$, $\frac{\partial W}{\partial \lambda_i}$, $1 \le i \le N$, $j \in E_i$.

УТВЕРЖДЕНИЕ I. Необходимым условием локального минимума функционала качества распределения заданий $W = W(\Gamma, B, \Phi)$ является выполнение в установившемся режиме для всех $\varphi_{ij} \in \Phi$, $1 \le i \le N$, $j \in E_i$, $k \in E_i$ соотношений:

$$\frac{\partial W}{\partial \varphi_{ik}} = \frac{\partial W}{\partial \varphi_{ij}} \quad \text{при} \quad \varphi_{ik} > 0 \ \text{и} \ \varphi_{ij} > 0;$$
$$\frac{\partial W}{\partial \varphi_{ik}} \le \frac{\partial W}{\partial \varphi_{ij}} \quad \text{при} \quad \varphi_{ik} > 0 \ \text{и} \ \varphi_{ij} = 0. \tag{12}$$

ДОКАЗАТЕЛЬСТВО. Задача определения $\min\limits_{\Phi} W(\Gamma,B,\Phi)$ при ограничениях (I) имеет $\sum\limits_{i=1}^{N} |E_i|$ неизвестных (Φ) и $N + \sum\limits_{i=1}^{N} |E_i|$ ограничений, удовлетворяющих в силу линейности условию регулярности [8]. Учитывая это, составим функцию Лагранжа:

$$L = W + \sum_{i=1}^{N} \beta_i (1 - \sum_{k \in E_i} \varphi_{ik}), \tag{13}$$

где β_i - неопределенный множитель. Из (I3), используя теорему Куна-Таккера [8], определим условия, которым должны удовлетворять оптимальные значения φ_{ik} при $i = 1,2,\ldots,N$, и $k \in E_i$:

$$\frac{\partial L}{\partial \varphi_{ik}} = \frac{\partial W}{\partial \varphi_{ik}} - \beta_i \ge 0; \tag{14}$$

$$\beta_i \frac{\partial L}{\partial \beta_i} = \beta_i (1 - \sum_{k \in E_i} \varphi_{ik}) = 0, \ \beta_i \ge 0; \tag{15}$$

$$\varphi_{ik} \frac{\partial L}{\partial \varphi_{ik}} = \varphi_{ik} \left(\frac{\partial W}{\partial \varphi_{ik}} - \beta_i \right) = 0. \tag{16}$$

Для выражения (I6) верно следующее:

$$\text{либо} \ \varphi_{ik} > 0 \quad \text{и} \quad \frac{\partial W}{\partial \varphi_{ik}} = \beta_i ,$$

$$\text{либо} \ \varphi_{ik} = 0 \quad \text{и} \quad \frac{\partial W}{\partial \varphi_{ik}} > \beta_i ,$$

$$\text{либо} \ \varphi_{ik} = 0 \quad \text{и} \quad \frac{\partial W}{\partial \varphi_{ik}} = \beta_i .$$

С учетом этого, из (I4)-(I6) находим соотношения (I2), устанавливающие, что для данного $i \in V$ все направления связи (i, k), $k \in E_i$ с $\varphi_{ik} > 0$, имеют одинаковое значение производной $\frac{\partial W}{\partial \varphi_{ik}}$, которое меньше или равно значению $\frac{\partial W}{\partial \varphi_{ij}}$ по направлениям связи (i, j), $j \in E_i$ с $\varphi_{ij} = 0$. Следует отметить, что так как ограничения (I) – линейные, следовательно, входящие в них функции можно рассматривать и как выпуклые, и как вогнутые, то при выпуклой целевой функции (4) условия (5) являются не только необходимыми, но и достаточными для существования оптимального плана распределения заданий Φ, при котором достигается глобальный минимум функционала W.

В случае критерия минимизации среднего времени выполнения задания применим к функционалу (9) соотношения (I2) и определим, что необходимыми условиями минимизации T являются следующие:

$$\begin{aligned} \frac{\partial T}{\partial \varphi_{ik}} &= \frac{\partial T}{\partial \varphi_{ij}} \quad \text{при} \quad \varphi_{ik} > 0 \quad \text{и} \quad \varphi_{ij} > 0; \\ \frac{\partial T}{\partial \varphi_{ik}} &\le \frac{\partial T}{\partial \varphi_{ij}} \quad \text{при} \quad \varphi_{ik} > 0 \quad \text{и} \quad \varphi_{ij} = 0 \end{aligned} \tag{17}$$

для $1 \le i \le N$, $k \in E_i$, $j \in E_i$.

В самом деле, пусть $\varphi_{ik} \in \Phi'$ не удовлетворяют условию (I2), т.е. существуют такие i, j, k, что: $\varphi_{ik} > 0$, $\frac{\partial W}{\partial \varphi_{ik}} > \frac{\partial W}{\partial \varphi_{ij}}$, тогда достаточно небольшого увеличения φ_{ij} и соответствующего уменьшения φ_{ik} (для выполнения условия (I)), чтобы уменьшить T, следовательно, множество Φ' не минимизирует функционал T.

§3. Рекуррентные формулы для проверки условий оптимальности плана распределения заданий

Для проверки соотношений (I2) необходимо вычислять производные $\frac{\partial W}{\partial \varphi_{ij}}$, где $i, j \in V$. Выведем рекуррентные формулы, по которым в ЭМ$_i$ можно организовать вычисление производных $\frac{\partial W}{\partial \varphi_{ij}}$, зная информацию о состоянии ЭМ$_i$ и соседних с ней ЭМ, $1 \le i \le N$.

ЛЕММА I. Если в вычислительной системе потоки заданий удовлетворяют соотношениям (2) и (3), а план распределения заданий Ф-ограничениям (I), то верно следующее:

$$\frac{\partial\Lambda_m}{\partial\lambda_i} = \delta_{im} + \sum_{k\in J_i} \varphi_{ik} \frac{\partial\Lambda_m}{\partial\lambda_k}, \quad 1 \le m, \ i \le N; \tag{18}$$

где $\delta_{im} = 1$ при $m = i$ при $\delta_{im} = 0$ при $m \ne i$;

$$\frac{\partial\Lambda_j}{\partial\varphi_{ij}} = \Lambda_i(1 - \varphi_{ij}\cdot\varphi_{ji})\frac{\partial\Lambda_j}{\partial\lambda_j}, \quad 1 \le i \le N, \quad j \in E_i; \tag{19}$$

$$\frac{\partial\Lambda_m}{\partial\varphi_{ij}} = \Lambda_i \frac{\partial\Lambda_m}{\partial\lambda_j}; \quad 1 \le m, \ i \le N, \quad m \ne j, \quad j \in E_i. \tag{20}$$

ДОКАЗАТЕЛЬСТВО. Введем фиктивную машину с номером 0 и примем, что если поступившее задание принимается на выполнение в ЭМ$_i$, $1 \le i \le N$, то оно передается в фиктивную машину, которая имеет каналы связи со всеми ЭМ системы. Таким образом,

$$\Lambda_0 = \sum_{m=1}^{N} \Lambda_m \varphi_{m0}. \tag{21}$$

С учетом этого и выражений (I)-(3) можно записать следующую систему линейных уравнений:

$$\Lambda_i = \lambda_i + \sum_{k=0}^{N} \Lambda_k \varphi_{ki}, \quad 0 \le i \le N, \tag{22}$$

где $\lambda_0 = 0$, $\varphi_{0i} = 0$ при $0 \le i \le N$. Пусть $\varphi_{ik} = 0$, $0 \le i \le N$, $k \notin E_i$. Применяя матричную форму, получаем: $\Lambda = B + \Lambda\Phi$, где $\Lambda = \{\Lambda_i\}$ и $B = \{\lambda_i\}$, $0 \le i \le N$,- векторы размерности $1 \times (N+1)$, а $\Phi = \{\varphi_{ij}\}$, $0 \le i, j \le N$, - матрица $(N+1)\times(N+1)$. Как отмечалось ранее, учитывая ограничения (I), матрицу Φ можно интерпретировать как стохастическую переходную матрицу конечной поглощающей цепи Маркова. При этом единственным поглощающим состоянием является нулевое состояние. Решая полученное матричное уравнение, находим:

$$\Lambda = B(I - \Phi)^{-1}, \tag{23}$$

где I - единичная матрица.

Продифференцируем данное выражение:

$$\frac{\partial \Lambda_m}{\partial \lambda_i} = [(I - \Phi)^{-1}]_{im}, \quad 0 \le i, m \le N, \tag{24}$$

здесь $[A]_{im}$ – элемент a_{im} матрицы A. Подставляя (24) в (23), найдем решение системы (22):

$$\Lambda_m = \sum_{i=0}^{N} \lambda_i \frac{\partial \Lambda_m}{\partial \lambda_i}, \quad 0 \le i \le N.$$

Выделим из матрицы Φ подматрицу $Q = \{\varphi_{ij}\}$, $1 \le i, j \le N$, размером $N \times N$, которая описывает поведение процесса в множестве невозвратных состояний до его перехода в поглощающее состояние. Уравнение (23) перепишем в виде:

$$\Lambda = B(I - Q)^{-1}, \tag{25}$$

где $\Lambda = \{\Lambda_i\}$, $B = \{\lambda_i\}$, $1 \le i \le N$. Дифференцируя (25), находим:

$$\frac{\partial \Lambda_m}{\partial \lambda_i} = [(I-Q)^{-1}]_{im}, \quad 1 \le i, m \le N, \tag{26}$$

где $F = (I-Q)^{-1}$ – фундаментальная матрица [10] поглощающей цепи Маркова, для которой верно следующее соотношение: $F = \sum_{k=0}^{\infty} Q^k$. С учетом этого преобразуем правую часть (26):

$$[F]_{im} = [\sum_{k=0}^{\infty} Q^k]_{im} = [I+Q(I+Q(I+Q(\ldots)))]_{im} =$$

$$= \delta_{im} + \sum_{k \in J_i} \varphi_{ik}[F]_{km}, \quad 1 \le i, m \le N.$$

Подставляя в найденное выражение равенство (26), получаем соотношение (18).

Для доказательства выражения (20) продифференцируем систему (22) по φ_{ij}:

$$\frac{\partial \Lambda_m}{\partial \varphi_{ij}} = \sum_{k \in E_m} \frac{\partial \Lambda_k}{\partial \varphi_{ij}} \cdot \varphi_{km} + \Lambda_i \delta_{mj}, \quad 0 \le m, i, j \le N. \tag{27}$$

При фиксированных i и j данная система уравнений относительно производных $\frac{\partial \Lambda_m}{\partial \varphi_{ij}}$ аналогична системе (22). Используя решение (23), находим:

$$\frac{\partial \Lambda_m}{\partial \varphi_{ij}} = \Lambda_i[(I - \Phi')^{-1}]_{jm}, \quad 0 \le m \le N , \tag{28}$$

где матрица Φ' равна матрице Φ с элементом $\varphi_{ij} = 0$. Как и в предыдущем случае, выделим из матрицы Φ' подматрицу Q' для невозвратных состояний. Выражение (28) перепишется в виде:

$$\frac{\partial \Lambda_m}{\partial \varphi_{ij}} = \Lambda_i[(I-Q')^{-1}]_{jm} = \Lambda_i[F']_{jm}, \quad 1 \le m \le N. \tag{29}$$

Учитывая, что фундаментальная матрица $F' = \sum_{k=0}^{\infty} (Q')^k$, где $Q'=Q-P$ и P - матрица $N \times N$ с единственным не отличным от нуля элементом $P_{ij} = \varphi_{ij}$, получаем:

$$[F']_{jm} = [\sum_{k=0}^{\infty} (Q-P)^k]_{jm} = [\sum_{k=0}^{\infty} \sum_{l=0}^{k} (-1)^l C_k^l Q^{k-l} P^l]_{jm} =$$

$$= [\sum_{k=0}^{\infty} Q^k]_{jm} + [\sum_{k=0}^{\infty} \sum_{l=1}^{m} (-1)^l C_k^l Q^{k-l} P^l]_{jm} =$$

$$= [F]_{jm} + [(\sum_{k=0}^{\infty} \sum_{l=1}^{k} (-1)^l C_k^l Q^{k-l} P^{l-1}) P]_{jm} .$$

Подставляя в первое слагаемое равенство (26) и учитывая, что второе слагаемое равно нулю при $m \ne j$, выражение (29) приводим к виду (20). Подставляя (20) в уравнение при $m = j$ в системе (27) и учитывая равенство (3), получаем (I9):

$$\frac{\partial \Lambda_i}{\partial \varphi_{ij}} = \sum_{\substack{k \in E_m \\ k \ne i}} \frac{\partial \Lambda_k}{\partial \varphi_{ij}} \varphi_{km} + \Lambda_i = \Lambda_i \sum_{k \in E_m \setminus i} \frac{\partial \Lambda_k}{\partial \lambda_j} \varphi_{km} + \Lambda_i =$$

$$= \Lambda_i \left(\frac{\partial \Lambda_j}{\partial \lambda_j} - \varphi_{ij} \frac{\partial \Lambda_i}{\partial \lambda_j} \right) = \Lambda_i (1-\varphi_{ij} \ \varphi_{ji}) \frac{\partial \Lambda_j}{\partial \lambda_j} .$$

В заключение приведем еще два соотношения, которые будут использованы в дальнейшем. Умножая обе части равенства (20) на φ_{mi}, получим:

$$\frac{\partial \lambda_{ml}}{\partial \varphi_{ij}} = \Lambda_i \frac{\partial \lambda_{ml}}{\partial \lambda_j}, \quad m \neq j,\ 1 \leq m,\ i \leq N,\ l \in E_m \tag{30}$$

Дифференцируя равенство $\lambda_{ij} = \varphi_{ij}\Lambda_i$ по λ_j и учитывая (3), найдем:

$$\frac{\partial \lambda_{ij}}{\partial \lambda_j} = \varphi_{ij} \frac{\partial \Lambda_i}{\partial \lambda_j} = \varphi_{ij} \cdot \varphi_{ji} \frac{\partial \Lambda_j}{\partial \lambda_j} = \varphi_{ij}\ \varphi_{ji}. \tag{31}$$

Лемма доказана.

УТВЕРЖДЕНИЕ 2. В вычислительной системе, удовлетворяющей соотношениям (I)-(3) при заданном (4) функционале качества распределения заданий W, производные $\frac{\partial W}{\partial \varphi_{ij}}$ определяются по следующим рекуррентным соотношениям:

$$\frac{\partial W}{\partial \varphi_{ij}} = \Lambda_i (1 - \varphi_{ij} \cdot \varphi_{ji}) \left(\frac{\partial W}{\partial \lambda_{ij}} + \frac{\partial W}{\partial \lambda_j} \right), \tag{32}$$

$$\frac{\partial W}{\partial \lambda_i} = \varphi_{i0} \frac{\partial W}{\partial \lambda_{i0}} + \sum_{k \in J_i} \varphi_{ik} \left(\frac{\partial W}{\partial \lambda_{ik}} + \frac{\partial W}{\partial \lambda_k} \right), \tag{33}$$

где $1 \leq i \leq N$, $j \in E_i$.

ДОКАЗАТЕЛЬСТВО. Из зависимости $W = W(B)$ и выражения (3) следует:

$$\frac{\partial W}{\partial \lambda_i} = \sum_{m=1}^{N} \frac{\partial W}{\partial \Lambda_m} \cdot \frac{\partial \Lambda_m}{\partial \lambda_i}. \tag{34}$$

Подставляя в (34) выражение (I8), получаем:

$$\frac{\partial W}{\partial \lambda_i} = \frac{\partial W}{\partial \Lambda_i} + \sum_{k \in J_i} \varphi_{ik} \frac{\partial W}{\partial \lambda_k}. \tag{35}$$

Первое слагаемое, исходя из равенств (2) и (3), определяется следующим образом:

$$\frac{\partial W}{\partial \Lambda_i} = \sum_{k \in E_i} \frac{\partial W}{\partial \lambda_{ik}} \cdot \frac{\partial \lambda_{ik}}{\partial \Lambda_i} = \varphi_{i0} \frac{\partial W}{\partial \lambda_{i0}} + \sum_{k \in J_i} \varphi_{ik} \frac{\partial W}{\partial \lambda_{ik}}.$$

Подставляя найденное выражение в (35), получаем равенство (33).

Аналогично (34) можно записать следующее:

$$\frac{\partial W}{\partial \varphi_{ij}} = \frac{\partial W}{\partial \Lambda_i} \cdot \frac{\partial \Lambda_i}{\partial \varphi_{ij}} + \frac{\partial W}{\partial \Lambda_j} \cdot \frac{\partial \Lambda_j}{\partial \varphi_{ij}} + \sum_{\substack{m=1 \\ m \neq i,j}}^{K} \frac{\partial W}{\partial \Lambda_m} \cdot \frac{\partial \Lambda_m}{\partial \varphi_{ij}} \cdot \qquad (36)$$

Исходя из равенств (2) и (3) и используя (30) и (31), первое слагаемое в (36) определяем следующим образом:

$$\frac{\partial W}{\partial \Lambda_i} \frac{\partial \Lambda_i}{\partial \varphi_{ij}} = \left(\sum_{k \in E_i} \frac{\partial W}{\partial \lambda_{ik}} \cdot \frac{\partial \lambda_{ik}}{\partial \Lambda_i} \right) \frac{\partial \Lambda_i}{\partial \varphi_{ij}} = \sum_{k \in E_i} \frac{\partial W}{\partial \lambda_{ik}} \cdot \frac{\partial \lambda_{ik}}{\partial \varphi_{ij}} =$$

$$= \Lambda_i \left(\frac{\partial W}{\partial \lambda_{ij}} + \sum_{\substack{k \in E_i \\ k \neq j}} \frac{\partial W}{\partial \lambda_{ik}} \cdot \frac{\partial \lambda_{ik}}{\partial \lambda_j} \right) = \Lambda_i \left(\frac{\partial W}{\partial \lambda_{ij}} - \frac{\partial W}{\partial \lambda_{ij}} \cdot \frac{\partial \lambda_{ij}}{\partial \lambda_j} + \right.$$

$$\left. + \frac{\partial W}{\partial \Lambda_i} \cdot \frac{\partial \Lambda_i}{\partial \lambda_j} \right) = \Lambda_i \left[\frac{\partial W}{\partial \lambda_{ij}} (1 - \varphi_{ij} \cdot \varphi_{ji}) + \frac{\partial W}{\partial \Lambda_i} \cdot \frac{\partial \Lambda_i}{\partial \lambda_j} \right].$$

Подставляя найденное выражение и выражения (19) и (20) в (36), получаем формулу (32). Утверждение 2 доказано.

Дадим объяснение полученным соотношениям. Определим в начале эффект, вызванный минимальным приращением (плотности) входного потока $\Delta\lambda_i$, на заданный функционал качества распределения заданий W. Приращение $\Delta\lambda_i$ вызывает, во-первых, приращение потока заданий на выполнение в ЭМ$_i$, т.е. $\Delta\lambda_{io}$, и, следовательно, соответствующее приращение функционала $\Delta W(\Delta\lambda_{io})$. Во-вторых, приращение $\Delta\lambda_i$ вызывает приращение потока заданий по линии связи (i, k) – $\Delta\lambda_{ik}$, где $k \in J_i$ и $\varphi_{ik} > 0$, что также приводит к приращению функционала качества $\Delta W(\Delta\lambda_{ik})$. В-третьих, $\Delta\lambda_i$ вызывает приращение потока заданий в соседнюю машину ЭМ$_k$, которое приводит к такому же приращению функционала W, как и приращение входного потока λ_k в ЭМ$_k$. Суммируя указанные приращения по всем соседним ЭМ, получаем выражение (33).

Формула (32) определяет приращение функционала качества в зависимости от приращения доли потока заданий, направляемых от ЭМ$_i$ в ЭМ$_j$ – $\Delta\varphi_{ij}$. Приращение $\Delta\varphi_{ij}$ вызывает, во-первых, приращение потока заданий по линии (i, j), пропорциональное Λ_i – общему потоку заданий, поступающих в ЭМ$_i$, что приводит к приращению функционала качества $\Delta W(\Delta\lambda_{ij})$, и, во-вторых, вызывает приращение потока заданий в соседнюю машину ЭМ$_j$, которое приводит к такому же приращению функционала W, как и приращение входного потока $\Delta\lambda_j$ в ЭМ$_j$. Кроме того, в обоих случаях из указанных приращений $\Delta W(\Delta\lambda_{ij})$ и $\Delta W(\Delta\lambda_j)$ необходимо вычесть ту часть приращения функционала W, которая вызвана вторичным приращением потоков вследствие их циклического характера и которую можно оценить величиной $\varphi_{ij}\ \varphi_{ji}$.

УТВЕРЖДЕНИЕ 3. При сепарабельном функционале качества распределения заданий вида: $W = \sum_{i=1}^{N} \sum_{k \in E_i} W_{ik}(\lambda_{ik})$ для проверки условий (12) в каждой ЭМ необходимо знание только локальной информации о состоянии системы.

ДОКАЗАТЕЛЬСТВО этого утверждения следует из соотношений (32), (33) и равенства: $\partial W/\partial\lambda_{ik} = \partial W_{ik}/\partial\lambda_{ik}$ при $1 \le i \le N,\ k \in E_i$. Таким образом, создаются предпосылки для организации оптимального децентрализованного управления распределением заданий.

Рассмотрим организацию вычисления производных по формулам (32) и (33) в случае критерия минимизации среднего времени выполнения задания. Так как функционал (9) является сепарабельным, то при его подстановке в (32) и (33) получим:

$$\frac{\partial T}{\partial\varphi_{i0}} = \Lambda_i \frac{\partial S_i}{\partial\lambda_{i0}};$$

$$\frac{\partial T}{\partial\varphi_{ij}} = \Lambda_i(1 - \varphi_{ij}\cdot\varphi_{ji})\left(\frac{\partial D_{ij}}{\partial\lambda_{ij}} + \frac{\partial T}{\partial\lambda_j}\right),\ j \in J_i; \tag{37}$$

$$\frac{\partial T}{\partial\lambda_i} = \varphi_{i0}\frac{\partial S_i}{\partial\lambda_{i0}} + \sum_{k \in J_i}\varphi_{ik}\left(\frac{\partial D_{ik}}{\partial\lambda_{ik}} + \frac{\partial T}{\partial\lambda_k}\right); \tag{38}$$

где $1 \le i \le N$. Из соотношений (37),(38) следует, что для вычисления производной $\frac{\partial T}{\partial\varphi_{ij}}$ в ЭМ$_i$ необходимо только получить от соседних ЭМ$_j$, $j \in J_i$, величины производной $\frac{\partial T}{\partial\lambda_j}$ и переменной φ_{ji}, а также иметь оценку величины изменения средней задержки передачи данных по каналу связи (i,j) в зависимости от изменения потока λ_{ij}. Для вычисления производных $\frac{\partial T}{\partial\varphi_{i\sigma}}$ и $\frac{\partial T}{\partial\lambda_i}$ в ЭМ$_i$, кроме указанных величин, необходимо также иметь оценку величины изменения среднего времени выполнения задания в ЭМ$_i$ в зависимости от изменения потока заданий, поступающих на выполнение в данную машину – λ_{i0}. Таким образом, расчет по соотношениям (37), (38) и проверка условий (17) требуют знаний только локальной информации о состоянии системы.

Остановимся на проблеме получения оценок производных $\frac{\partial D_{ik}}{\partial \lambda_{ik}}$ и $\frac{\partial S_i}{\partial \lambda_{i0}}$ в ЭМ$_i$. Рекуррентная локальная процедура получения этих оценок без действительного изменения потоков λ_{ik} и λ_{i0} приведена в работе [11]. Указанная процедура основана на вычислении величины изменения времени обслуживания заявок при фиктивном удалении какой-либо заявки из системы путем регистрации временных интервалов между прибытием и уходом заявок.

Более простой способ оценки величин производных $\frac{\partial D_{ik}}{\partial \lambda_{ik}}$ дан в работе [12], в которой используется следующая приближенная формула:

$$\frac{\partial D_{ik}}{\partial \lambda_{ik}} = \frac{1}{\mu C_{ik}}(1 + L_{ik})(1 + \frac{L_{ik}}{2}),$$

где μ – параметр экспоненциального закона распределения длин сообщений, C_{ik} – пропускная способность канала (i,k), L_{ik} – длина очереди сообщений в канале (i,k).

В случае использования критерия неравномерности нагрузки функционал (11) также является сепарабельным, и при проверке условий оптимальности (12) соотношения (32) и (33) имеют вид:

$$\frac{\partial W}{\partial \varphi_{ij}} = \Lambda_i(1 - \varphi_{ij}\ \varphi_{ji})\frac{\partial W}{\partial \lambda_j}, \quad j \in J_i; \tag{39}$$

$$\frac{\partial W}{\partial \varphi_{i0}} = \Lambda_i \frac{\partial w_i}{\partial \lambda_{i0}}; \tag{40}$$

$$\frac{\partial W}{\partial \lambda_i} = \varphi_{i0} \frac{\partial w_i}{\partial \lambda_{i0}} + \sum_{k \in J_i} \varphi_{ik} \frac{\partial W}{\partial \lambda_k}; \tag{41}$$

где $1 \leq i \leq N$.

Проверка условий (I2) и вычисление по соотношениям (39)-(41), в случае функционала качества распределения заданий вида (11), требует в каждой ЭМ знания только локальной информации о состоянии системы, что дает возможность организовать децентрализованное управление процессом распределения заданий.

Отметим, что вычисления оценки производной $\frac{\partial w_i}{\partial \lambda_{i0}}$ в ЭМ$_i$ может быть организовано на основании рекуррентной локальной процедуры, аналогичной уже упомянутой процедуре из работы [II], либо, используя подход, указанный в [I2], по приближенной формуле:

$$\frac{\partial w_i}{\partial \lambda_{i0}} \approx (1+x_i)(1+\frac{x_i}{2}),$$

где x_i - нагрузка ЭМ$_i$.

§4. Алгоритм оптимизации плана распределения заданий

Рассмотрим адаптивный децентрализованный алгоритм оптимизации плана распределения заданий, который на каждом шаге итерации так корректирует план распределения Φ, что обеспечивает последовательное приближение значения сепарабельного функционала W к минимуму.

Исходя из необходимых условий оптимальности (I2) и формул (32) и (33), приведем общую схему такого алгоритма, состоящую из двух этапов.

На первом этапе в каждой машине ЭМ$_i$, $1 \le i \le N$, по всем направлениям связи (i,j), $j \in E_i$, вычисляются на основании информации, полученной от соседних ЭМ производные $\frac{\partial W}{\partial \varphi_{ij}}$. На втором этапе по вычисленным значениям производных происходит проверка условий (I2), и если данные условия не выполняются, т.е. существуют направления связи $j_1, j_2, \dots, j_k$, $j_r \in E_i$, $r = 1,2,\dots,k$, $k \le |E_i|$, для которых производные не равны между собой, а переменные $\varphi_{ij_r} > 0$, то следует скорректировать план распределения Φ, а именно увеличить ту переменную φ_{ij}, для которой значение $\frac{\partial W}{\partial \varphi_{ij}}$ минимально, и уменьшить остальные переменные φ_{ij}, $j \in E_i$, соблюдая ограничения (1). Таким образом, алгоритм задает на каждом шаге итерации отображение A текущего плана распределения Φ^1 в новый план

$\Phi^2 = A(\Phi^1)$ и состоит из следующих действий, выполняемых в каждой машине ЭМ$_i$, $1 \le i \le N$.

I. При текущем плане распределения Φ^1 в ЭМ$_i$, $1 \le i \le N$, вычисляются производные $\frac{\partial W}{\partial \varphi^1_{ik}}$, $k \in E^*_i = \{k \in E_i \ \& \ \varphi^1_{ik} > 0\}$, по формулам (32) и (33), на основании оценок производных $\frac{\partial W}{\partial \lambda_{ik}}$, полученных в ЭМ$_i$ и информации, полученной от соседних ЭМ.

2. Проверяются условия (I2), а именно: определяются

$$\frac{\partial W}{\partial \varphi^1_{ik1}} = \min_{k \in E^*_i} \frac{\partial W}{\partial \varphi^1_{ik}}, \quad \frac{\partial W}{\partial \varphi^1_{ik2}} = \max_{k \in E^*_i} \frac{\partial W}{\partial \varphi^1_{ik}}, \tag{42}$$

$$b_i = \frac{\partial W}{\partial \varphi^1_{ik2}} - \frac{\partial W}{\partial \varphi^1_{ik1}}.$$

Если $b_i < \varepsilon$, где $0 < \varepsilon < 1$ - заданная величина, то считаем план распределения в ЭМ$_i$,установившемся и удовлетворяющим условиям оптимальности (I2), $\Phi^2 := \Phi^1$, переход на 4. Если данное условие не выполняется, то переход на 3.

3. Коррекция плана распределения заданий в ЭМ$_i$, $1 \le i \le N$:

$$\begin{aligned} \varphi^2_{ik} &= \varphi^1_{ik} - \Delta\varphi_{ik}, \quad k \in E^*_i \setminus k1; \\ \varphi^2_{ik1} &= \varphi^1_{ik1} + \sum_{k \in E^*_i \setminus k1} \Delta\varphi_{ik}, \end{aligned} \tag{43}$$

где $\Delta\varphi_{ik} = \min[\varphi^1_{ik}, \eta\, a_{ik}/b_i]$, $k \in E^*_i$,

$$a_{ik} = \frac{\partial W}{\partial \varphi^1_{ik}} - \frac{\partial W}{\partial \varphi^1_{ik1}},$$

$0 \le \eta \le 1$ - масштабный параметр, влияющий на скорость сходимости алгоритма. Переход на 4.

4. Работа ЭМ$_i$ в течение времени τ (длительность итерации управления) с планом $\Phi := \Phi^2$, после чего переход на I. Докажем сходимость данного алгоритма.

УТВЕРЖДЕНИЕ 4. А л г о р и т м А, з а д а ю щ и й к о р р е к ц и ю п л а н а р а с п р е д е л е н и я з а д а н и й в к а ж д о й м а ш и н е $ЭМ_i$, $1 \le i \le N$, п о с о о т н о ш е н и я м (43), п р и с о о т в е т с т в у ю щ е м в ы б о р е п а р а м е т р а η $(0 < \eta < 1)$ с х о д и т с я, и в с л у ч а е с т р о г о й в ы п у к л о с т и ф у н к ц и о н а л а W и м е е т м е с т о:

$$\lim_{m \to \infty} W(\Phi^m) = \min_{\Phi} W(\Phi), \tag{44}$$

г д е $\Phi^m = A(\Phi^{m-1})$, $m \ge 1$.

ДОКАЗАТЕЛЬСТВО. На шаге итерации с номером m, для машин, где не выполняются условия (12), в соответствии с (43) находим новый план $\Phi^{m+1} = A(\Phi^m)$. Определим при этом изменения функционала W :

$$\Delta W = W(\Phi^{m+1}) - W(\Phi^m) = \sum_{i=1}^{N} \left(\frac{\partial W}{\partial \varphi_{ik1}} \Delta\varphi_{ik1} - \sum_{k \in E_i^* \setminus k1} \frac{\partial W}{\partial \varphi_{ik}} \Delta\varphi_{ik} \right) +$$

$$+ O\left(\sqrt{\sum_{i,k} (\Delta\varphi_{ik})^2}\right).$$

Учитывая, что $\Delta\varphi_{ik1} = \sum_{k \in E_i^* \setminus k1} \Delta\varphi_{ik}$, получим:

$$\Delta W = \sum_{i=1}^{N} \left(\sum_{k \in E_i^* \setminus k1} \frac{\partial W}{\partial \varphi_{ik1}} \Delta\varphi_{ik} - \sum_{k \in E_i^* \setminus k1} \frac{\partial W}{\partial \varphi_{ik}} \Delta\varphi_{ik} \right) + O\left(\sqrt{\sum_{i,k} (\Delta\varphi_{ik})^2}\right).$$

Поскольку на каждом шаге итерации, как это следует из (42), выполняется $\frac{\partial W}{\partial \varphi_{ik1}} < \frac{\partial W}{\partial \varphi_{ik}}$ $\forall k \in E_i^* \setminus k1$, то всегда $\Delta W < 0$, и, таким образом, алгоритм A обеспечивает последовательное уменьшение W, что доказывает сходимость данного алгоритма, а в случае строгой выпуклости функционала W доказывает его сходимость к точке глобального минимума, т.е. существование предела в выражении (44).

Таким образом, показано, что применение децентрализованного итеративного алгоритма A к задаче (4) обеспечивает ее решение. Структура предложенного алгоритма остается постоянной для довольно широкого класса функционалов качества распределения заданий в вычислительной системе. Следует отметить, что основные трудности практической реализации алгоритма связаны, во-первых, с выбором начального плана распределения заданий Φ^0 , во-вторых, с выбором параметров ε и η, определяющих точность и скорость сходимости данного алгоритма, и, в-третьих, с определением величины периода обновления τ, определяющего оперативность управления. Выбор значений указанных величин может быть осуществлен экспериментальным путем.

§5. Алгоритмы реализации оптимального плана распределения заданий

Пусть с помощью алгоритма оптимизации плана распределения заданий в вычислительной системе получен план Φ, при этом в каждой ЭМ$_i$, $1 \le i \le N$, определено множество величин $\{ \Phi_{ik} \} \subset \Phi$, $k \in E_i$. Рассмотрим способы реализации полученного плана,т.е. алгоритмы, осуществляющие, в соответствии с планом Φ ,выделение подсистем с требуемым числом элементарных машин для исполнения поступающих заданий. Примем, что процесс выделения подсистемы заданного ранга, предназначенной для выполнения поступившего задания, состоит из двух этапов. На первом этапе происходит определение корневой ЭМ, т.е. ЭМ, относительно которой на втором этапе процесса выделения подсистемы будет формироваться подсистема требуемого ранга [9]*). Назовем первый этап - этапом перераспределения задания, а второй - этапом формирования подсистемы.

Рассмотрим два наиболее простых децентрализованных алгоритма перераспределения задания: вероятностный и релаксационный.

Вероятностный алгоритм перераспределения заданий основан на интерпретации плана $\{\Phi_{ik}\} \subset \Phi$, $k \in E_i$, в ЭМ$_i$, $1 \le i \le N$, в качестве распределения вероятности передачи поступившего в ЭМ$_i$ задания по выходному направлению (i,k) , где $k \in E_i$, и состоит в следующем.

При поступлении задания в ЭМ$_i$, $1 \le i \le N$, выполняется процедура случайного выбора направления дальнейшей передачи, данного задания в соответствии с распределением вероятностей $\{ \Phi_{ik} \} \subset \Phi$, $k \in E_i$. При этом, если выбирается направление $(i,0)$, то задание переходит к этапу формирования подсистемы, а ЭМ$_i$ становится корневой ЭМ для вновь формируемой подсистемы.

*) Отметим, что эвристические децентрализованные алгоритмы для частного случая - поиска одной свободной машины в распределенной вычислительной системе рассмотрены в [14].

Недостатки данного алгоритма заключаются в большом времени поиска корневой ЭМ, что приводит к увеличению среднего времени выполнения задания, и в большом числе шагов до прибытия к корневой ЭМ, что увеличивает нагрузку на сеть связи системы. Достоинства приведенного алгоритма состоят в его простоте, точности реализации оптимального плана распределения заданий, и устойчивости к колебаниям внешней нагрузки. При релаксационном алгоритме перераспределения заданий направление (i,j) дальнейшей передачи поступившего в ЭМ$_i$ задания определяется из следующих соотношений: $j \in E_i$ и $\varphi_{ij} = \max\limits_k \varphi_{ik}$. При $j = 0$ ЭМ$_i$ становится корневой ЭМ для формируемой подсистемы.

Данный алгоритм имеет меньшее время поиска корневой ЭМ по сравнению с вероятностным, но вместе с тем обладает и меньшей устойчивостью, создает большую неравномерность нагрузки при распределении заданий. В обоих алгоритмах перераспределения предполагается, что если число одновременно исполняемых заданий в ЭМ$_i$, $1 \le i \le N$, при поступлении нераспределенного задания, равно предельной степени мультипрограммирования, то процедура выбора следующего направления передачи задания происходит без учета направления $(i, 0)$. С другой стороны, если при поступлении задания в ЭМ$_i$ данная машина оказывается свободной, то для поступившего задания ЭМ$_i$ становится корневой ЭМ. На этапе формирования подсистемы происходит построение, в соответствии с планом распределения заданий Φ, подсистемы с требуемым числом элементарных машин $-n$. Указанная подсистема строится алгоритмом ФОРМИРОВАНИЕ. Обозначим: Q - множество ЭМ, входящих в построенную подсистему, R_i - множество ЭМ, соседних с ЭМ$_i$, которые имеют нагрузку, меньшую предельной степени мультипрограммирования и не входят в множество Q. Идея алгоритма ФОРМИРОВАНИЕ состоит в том, что количество ЭМ, которое необходимо включить в подсистему Q по каждому направлению (i,k), $k \in R_i$, выходящему из ЭМ$_i$ $\in Q$, пропорционально величинам $\varphi_{ik} \in \Phi$, $k \in R_i$. Пусть пропорциональное распределение количества ненайденных машин $-n_i$ по направлениям (i,k), $k \in R_i$, которые выходят из ЭМ$_i$, $1 \le i \le N$, осуществляется оператором РАСПРЕДЕЛЕНИЕ (n_i), тогда выполнение алгоритма ФОРМИРОВАНИЕ сводится к следующему.

I. Создание подсистемы Q из одной корневой машины ЭМ$_j$. Определить в ЭМ$_j$ множество R_j. Если $R_j \neq \varnothing$, то для каждого $i \in R_j$ с помощью алгоритма РАСПРЕДЕЛЕНИЕ $(n-1)$ находятся величины $n_i \neq 0$ - количество ненайденных ЭМ, и передаются в ЭМ$_i$. Пусть b_i - направление отхода в ЭМ$_i$, $1 \le i \le N$, тогда $b_j := 0$.

2. Если $ЭМ_i$, $1 \le i \le N$, получив n_i от $ЭМ_j$, не принадлежит подсистеме Q, то $Q:=Q \cup \{ЭМ_i\}$, т.е. $ЭМ_i$ включается в подсистему Q, $b_i:=j$, и находится множество R_i. Если $R_i \neq \emptyset$, то для каждого $k \in R_i$ определяются, с помощью алгоритма РАСПРЕДЕЛЕНИЕ (n_i-1), величины $n_k \neq 0$, которые передаются в $ЭМ_k$. Если $R_i = \emptyset$, то величина $n_i - 1$ передается обратно в $ЭМ_j$, где $j = b_i$. При $n_i - 1 = 0$ $ЭМ_i$ считает процесс ФОРМИРОВАНИЕ оконченным.

3. Если $ЭМ_i$, $1 \le i \le N$, получив n_i от $ЭМ_j$, принадлежит подсистеме Q, то $R_i:=R_i \setminus j$. Если $R_i \neq \emptyset$, то для каждого $k \in R_i$ находятся величины n_k, с помощью алгоритма РАСПРЕДЕЛЕНИЕ (n_i), и передаются в $ЭМ_k$. Если $R_i = \emptyset$, то величина n_i передается в $ЭМ_k$ где $k = b_i$, при этом если $k = 0$, то процесс формирования подсистемы из корневой $ЭМ_i$ получает отказ и происходит возвращение на этап перераспределения задания.

Заметим, что в данном алгоритме предполагается посылка извещений о включении в подсистему от каждой вновь включенной машины в корневую ЭМ. Таким образом, корневая ЭМ считает процесс формирования подсистемы оконченным, если число полученных извещений равно n-1.

Предложенный децентрализованный алгоритм реализует второй этап процесса выделения подсистемы - формирование связного подмножества требуемого числа ЭМ. Третий этап состоит в назначении виртуальных адресов ЭМ, входящих в подсистемы, согласование их с физическими адресами, и загрузке заданий в машины адресной подсистемы, после чего задание поступает на выполнение [9,13].

Заключение

Рассмотренный подход к распределению заданий в вычислительных системах с программируемой структурой позволяет, в случае сепарабельного функционала качества распределения, организовать децентрализованное управление потоками заданий. Итеративный децентрализованный алгоритм коррекции монотонно приближает план распределения заданий к оптимальному, адаптируя вычислительную систему к поступающему потоку заданий, характеристики которого априорно не определены. Все это дает возможность организовать управление распределением заданий, близкое к оптимальному и повысить эффективность использования вычислительной системы.

Общность рассмотренного подхода позволяет надеяться, что основные результаты данной работы остаются верными и в случае их приложения для управления процессом распределения заданий в сетях ЭВМ и в распределенных вычислительных системах.

В заключение автор выражает благодарность Корнееву В.В. за полезные обсуждения данной работы.

Литература

1. ЕВРЕИНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. - Новосибирск: Наука, 1978. - 318 с.

2. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой структурой. - Электронное моделирование, 1979, №1, с. 42-50.

3. КОРНЕЕВ В.В., МОНАХОВ О.Г. Графы межмашинных связей однородных вычислительных систем. - Изв. АН СССР. Тех. кибернетика, 1980, № 2, с. 195-201.

4. МОНАХОВ О.Г. Параметрическое описание структур однородных вычислительных систем. -В кн.: Вопросы теории и построения вычислительных систем (Вычислительные системы, вып. 80). Новосибирск, 1979, с. 3-17.

5. БАРСКИЙ А.Б. Планирование параллельных вычислительных процессов. - М.: Машиностроение, 1980. - 191 с.

6. ЛИПАЕВ В.В. Распределение ресурсов в вычислительных системах. - М.: Статистика, 1979. - 247 с.

7. GALLAGER R.G. A minimum delay routing algorithm using distributed computation.-IEEE Trans.Commun.,1977,v.25,N 1,p.73-85.

8. ХЕДЛИ Дж. Нелинейное и динамическое программирование.-М.: Мир, 1967. - 506 с.

9. КОРНЕЕВ В.В., МОНАХОВ О.Г. О децентрализованном распределении заданий в вычислительных системах с программируемой структурой. - В кн.: Архитектура вычислительных систем с программируемой структурой (Вычислительные системы, вып. 82). Новосибирск, 1980, с. 3-17.

10. КЕМЕНИ Дж., СНЕЛЛ Дж. Конечные цепи Маркова. -М.: Наука, 1970. - 272 с.

11. SEGAL A The modeling of adaptive routing in data-communication networks.-IEEE Trans.Commun.,1977,v.25,N 1,p.85-95.

12. AGNEW C.E. On quadratic adaptive routing algorithms. - Comm.ACM,1976,v.19,N 1,p.18-22.

13. КОРНЕЕВ В.В., МОНАХОВ О.Г.Организация межмашинных взаимодействий в вычислительных системах с программируемой структурой.- Электронное моделирование. 1980, № 5, с. 16-22.

14. ГАВРИЛОВ А.В., ЖИРАТКОВ В.И. Методы поиска свободной ЭВМ в распределенных вычислительных системах. - Программирование, №4, 1981, с. 70-77.

Поступила в ред.-изд.отд.
24 августа 1981 года