

УДК 681.324

АНАЛИЗ ПАРАЛЛЕЛЬНЫХ ДИНАМИЧЕСКИХ АЛГОРИТМОВ
ДЕЦЕНТРАЛИЗОВАННОГО РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ В
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

О.Г. Монахов

Введение

Проблема распределения заданий между процессорами (элементарными машинами - ЭМ) вычислительной системы (ВС) возникает одной из первых при организации функционирования ВС с программируемой структурой и распределенным управлением [1]. Разработка и реализация децентрализованных методов распределения заданий позволяет в условиях неполной информации о состоянии системы и априорной неопределенности характеристик поступающих заданий динамически перераспределять вычислительную нагрузку ЭМ с целью оптимизировать функционирование ВС. Важным свойством децентрализованных методов распределения заданий является их работоспособность в случае отказов отдельных ЭМ.

В статье приводятся параллельные динамические алгоритмы децентрализованного распределения заданий в вычислительных системах, основанные на локальных асинхронных вычислениях, и делается анализ результатов имитационного моделирования ВС с программируемой структурой, функционирующей на основе данных алгоритмов.

§1. Постановка задачи

Пусть задания, поступающие на исполнение в ВС, образуют множество (набор) заданий, характеристики которых (такие как время исполнения) априорно неизвестны.

Пусть Q - множество заданий, поступающих на исполнение в ВС. Каждое задание $q_i \in Q$, $i = \overline{1, z}$, может поступить в любую из ЭМ системы, представляет из себя параллельную программу и требует для

своего исполнения одновременно ресурсов $q_i(n)$, связанных ЭМ, $1 \leq q_i(n) \leq N$, где N - количество ЭМ в ВС, т.е. подсистему ранга $q_i(n)$, $z = |Q|$.

Представим структуру системы (или ее сети межмашинных связей) графом $\Gamma(P, E)$, вершинам P которого соответствуют элементарные машины, а ребрам E - линии связи между ними. Примем, что задан некоторый функционал $W = W(\Gamma, Q, X)$ качества распределения заданий в системе, который зависит от графа Γ , множества Q поступающих заданий и вектора X распределения заданий.

Таким образом, цель динамического децентрализованного алгоритма распределения заданий - обеспечить экстремум функционала W в условиях априорной неопределенности характеристик поступающих заданий из множества Q и состояния системы (вектора X) при заданной структуре системы Γ . Примем, что процесс выполнения задания q_i состоит из трех этапов [2]. На первом этапе происходит перераспределение задания между ЭМ системы с целью минимизации заданного функционала качества распределения заданий и определение корневой ЭМ, т.е. ЭМ, относительно которой на втором этапе выполнения задания будет формироваться подсистема ранга $q_i(n)$. Процесс формирования подсистемы составляет второй этап выполнения задания и представляет собой процесс образования связанного подмножества ЭМ относительно корневой ЭМ. После того, как подсистема требуемого ранга для задания q_i сформирована, наступает третий этап - загрузка ЭМ подсистемы и исполнения данного задания. Каждый из этапов реализуется своим децентрализованным алгоритмом. Два первых этапа исполняются на своих, включающих все исправные машины ВС, служебных подсистемах, которые обозначим Q_P и Q_L соответственно для первого и второго этапов.

Рассмотрим целевые функции, используемые при распределении заданий и характеризующие неравномерность загрузки машин системы. Обозначим через $L_\rho(i)$ окрестность ЭМ _{i} с радиусом ρ , $0 \leq \rho \leq d$, d - диаметр графа межмашинных связей ВС. Окрестность $L_\rho(i)$ назовем множество номеров всех машин, которые расположены на расстоянии, не превышающем ρ от ЭМ _{i} . Под нагрузкой машины ЭМ _{i} , $1 \leq i \leq N$, понимается x_i - число заданий, ожидающих исполнения и исполняемых в этой ЭМ. Введем также понятие средней нагрузки по системе: $Mx = (\sum_{i=1}^N x_i) / N$. Показатель неравномерности нагрузки машин системы имеет вид:

$$W_1 = \sum_{i=1}^N |x_i - Mx|, \quad W_2 = \sum_{i=1}^N (x_i - Mx)^2. \quad (1)$$

В качестве оценки средней нагрузки по системе в ЭМ_i можно использовать величину средней нагрузки ЭМ в окрестности некоторого радиуса ρ относительно ЭМ_i:

$$Mx_i = \left(\sum_{k \in L_\rho(i)} x_k \right) / |L_\rho(i)|.$$

В этом случае функционал (I) примет вид: $W_1 = \sum_{i=1}^N |x_i - Mx_i|$ при условии, что число заданий в системе равно R, а нагрузка каждой ЭМ ограничена числом M: $\sum_{i=1}^N x_i = R$, $0 \leq x_i \leq M$.

Если для каждого поступившего в БС задания $q_i \in Q$ известно время его исполнения t_i , то в качестве обобщений нагрузки ЭМ_k для задания q_i может быть взята величина: $h_{ik} = \sum_{j=1}^{x_k} t_j + t_i x_{ik}$, которая равна сумме времен исполнения всех заданий, находящихся в ЭМ_k, и задания q_i , где $x_{ik} = 1$, если задание q_i назначено на выполнение в ЭМ_k, и $x_{ik} = 0$ - в противном случае, $1 \leq k \leq N$, $1 \leq i \leq |Q|$. Показатель неравномерности обобщенной нагрузки при распределении задания q_i примет следующий вид:

$$W_i = \sum_{k=1}^N |h_{ik}(x_{ik}) - Mh_k|, \quad (2)$$

а для всего множества Q имеем следующий показатель: $W = \sum_{i=1}^Z W_i^2$, где

$$Mh_k = \left(\sum_{j \in L_\rho(k)} \sum_{n=1}^{x_j} t_n \right) / |L_\rho(k)|, \quad \sum_{k=1}^N x_{ik} = q_i(n).$$

Отметим, что в качестве показателей неравномерности нагрузки могут быть использованы функционалы, аналогичные (I) и (2), но характеризующие не сумму, а максимальное отклонение нагрузки; например, $W = \max_{1 \leq i \leq N} |x_i - Mx_i|$ [3].

Задача этапа формирования подсистемы имеет следующий вид.

Пусть задание $q_i \in Q$ требует для своего исполнения подсистему ранга $q_i(n)$. Построение такой подсистемы сводится к выделению

нию $q_i(n)$ связанных машин системы, находящихся на минимальном расстоянии от корневой ЭМ и имеющих минимальную обобщенную нагрузку. Под обобщенной нагрузкой ЭМ_k в этом случае понимается величина $h_{ik} = x_k + r_{k0}$, где r_{k0} - расстояние ЭМ_k от корневой ЭМ. Задача формирования подсистемы сводится к минимизации функционала $W_i = \sum_{k=1}^N h_{ik} \cdot x_{ik}$, а для всего множества заданий Q - к минимизации следующей целевой функции: $W = \sum_{i=1}^Z W_i$ или $W = \sum_{i=1}^Z W_i^2$ при следующих ограничениях:

$$\sum_{k=1}^N x_{ik} = q_i(n), \quad \sum_{i=1}^Z x_{ik} \leq M,$$

где $x_{ik} = 1$, если ЭМ_k входит в подсистему q_i , и $x_{ik} = 0$ - в противном случае. Если $q_i(n) > 2$, то должно выполняться условие связности: $\forall (k, j \in P: x_{ij} > 0, x_{ik} > 0) \exists (p_{kj} \in E^d: (\forall (m, l) \in p_{kj}: x_{im} > 0, x_{il} > 0))$, где p_{kj} - путь из ЭМ_k в ЭМ_j, проходящий через ЭМ, включенные в подсистему. Таким образом, задача минимизации функционала (1) или (2) при данных ограничениях сводится к задаче нелинейного целочисленного программирования.

Выбор функционала (1) или (2) в качестве критерия эффективности алгоритма распределения заданий продиктован, во-первых, стремлением равномерно загрузить ресурсы системы (как ЭМ, 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. Равномерное распределение нагрузки между ЭМ системы имеет важное значение с точки зрения организации отказоустойчивого функционирования ВС. Так как затраты на реконфигурацию системы зависят от величины нагрузки отказавших ЭМ, то перегрузка отдельных ЭМ, в случае их отказа, может сделать время реконфигурации системы неприемлемо большим. Неприемлемо большим может быть и время решения задач в этом случае. Кроме того, вероятность отказа перегруженных ЭМ увеличивается, так как уменьшается время для исполнения процедур самодиагностики.

§2. Параллельные динамические алгоритмы децентрализованного распределения заданий

Алгоритмы динамического децентрализованного распределения заданий подробно изложены в работах [2-4] и в общем виде сводятся к выполнению в каждой ЭМ системы следующей процедуры:

1. На основе сбора информации из окрестности $L_\rho(i)$, в каждой машине ЭМ_i происходит проверка условий оптимальности текущего вектора распределения заданий X .

2. В случае невыполнения условий оптимальности происходит коррекция вектора распределения заданий X , а именно: происходит перемещение заданий в окрестности $L_\rho(i)$ с целью минимизировать функционал W .

Данная процедура повторяется каждый раз при изменении вектора распределения заданий, постоянно корректируя его с целью оптимизации функционирования ЕС.

В простейшем случае эвристический алгоритм равномерного распределения заданий осуществляет перераспределение заданий между машинами ЕС с целью выравнивания их нагрузки с помощью следующей децентрализованной процедуры.

1. Для заданного радиуса $\rho \geq 1$, на основе сбора информации о нагрузках ЭМ из окрестности $L_\rho(i)$, происходит определение средней нагрузки машин по данной окрестности и номера машины, имеющей минимальную нагрузку:

$$Mx_i = \left(\frac{\sum_{k \in L_\rho(i)} x_k}{|L_\rho(i)|} \right), m = \operatorname{argmin}_{k \in L_\rho(i)} x_k.$$

2. Проверка условия $(x_i - Mx_i) < \delta$, где $\delta > 0$ - заданная величина. Если данное условие не выполняется, то происходит перенос одного задания q_j из числа распределяемых заданий подсистемы Q_T из машины ЭМ_i в машину с номером m : $x_i := x_i - 1$, $x_m := x_m + 1$.

3. Если проверяемое условие выполнено и число загружаемых заданий подсистемы Q_L в ЭМ меньше некоторой заданной константы $M > 0$, то происходит передача одного задания q_j из подсистемы Q_T в подсистему загружаемых заданий Q_L в машине ЭМ_i. При этом ЭМ_i становится корневой машиной формируемой подсистемы Q_j , предназначенной для реализации задания q_j . Формирование подсистемы Q_j составляет второй этап процесса исполнения задания q_j .

Таким образом, происходит перераспределение поступивших заданий с целью выравнивания нагрузки машин системы.

Алгоритм формирования виртуальных подсистем [2] формирует связанную подсистему Q_j с заданным рангом $r \leq N$. Подсистема Q_j служит для выполнения задания $q_j \in Q$, представленного параллельной программой, состоящей из $r = q_j(n)$ ветвей. Алгоритм формирования подсистем реализует второй этап в процессе исполнения заданий – формирования подсистемы и представляет собой совокупность асинхронных параллельных процессов, исполняемых в каждой ЭМ служебной подсистемы Q_T . Исходными данными для алгоритма служат число машин формируемой подсистемы и корневая машина, определенная в результате работы алгоритма распределения заданий. Корневая ЭМ является первой машиной строящейся подсистемы. Формирование подсистемы происходит путем включения в подсистему наименее нагруженных ЭМ, непосредственно связанных каналами с машинами, уже включенными в подсистему. На этом же этапе задаются адресация ЭМ подсистемы и коммутационные функции, для чего используются децентрализованные алгоритмы, описанные в [5].

Отметим свойства приведенных алгоритмов, позволяющие использовать их для организации отказоустойчивого функционирования ВС и в процессе адаптации системы к следующим изменениям: 1) к изменению индивидуальной потребности пользователей в ресурсах; 2) к изменению общего количества ресурса и 3) к изменению числа пользователей в системе. Во-первых, при реализации данных алгоритмов не требуется в явном виде знания общего количества ресурса, находящегося в системе. Для каждой вершины $i \in P$ необходимо знать информацию о состоянии только локальной области системы $L_p(i)$. Следовательно, при достаточно медленном изменении общего количества ресурсов в системе, с помощью приведенных алгоритмов осуществляется перераспределение ресурсов между пользователями путем локальных изменений в процессе функционирования системы, минимизируя тем самым заданную целевую функцию и обеспечивая адаптацию системы к изменению ресурсов.

Во-вторых, децентрализованный и итеративный характер данных алгоритмов позволяет перераспределять ресурсы, вызывая только локальные перестройки и в случае изменения общего числа пользователей в системе или при изменении потребностей пользователей в ресурсах, так как при реализации метода не требуется знание этих величин для всей системы. Таким образом, и в этом случае приведенные алгоритмы обеспечивают адаптацию системы к произошедшим изменениям, не требуя распределения ресурсов на всей системе заново

для каждого нового состояния, осуществляя только локальные изменения в процессе функционирования системы.

§3. Анализ децентрализованных алгоритмов распределения заданий и формирования подсистем

Для анализа и экспериментальной проверки работоспособности децентрализованных распределения заданий и формирования подсистем, для исследования влияния параметров данных алгоритмов и характеристик поступающих заданий на характеристики функционирования ВС была разработана имитационная модель ВС с программируемой структурой [6]. Моделирование проводилось для децентрализованного эвристического алгоритма распределения заданий, описанного выше, и децентрализованного алгоритма формирования подсистем [2].

Задания поступают в систему через ЭМ, называемые точками поступления заданий. Каждая ЭМ может быть выбрана в качестве точки поступления задания, и, следовательно, число таких точек ограничено только числом ЭМ в системе. Потоки заданий, поступающих в данную ЭМ, моделируются как стационарные пуассоновские с заданной интенсивностью λ_i , $i \in \{1, 2, \dots, N\}$. Каждое задание определяется следующими параметрами: r - число ЭМ, необходимое для выполнения задания, h - время передачи текста задания между двумя соседними ЭМ (под соседними понимаются две ЭМ, непосредственно соединенные линией связи), g - время, необходимое для выполнения любой параллельной ветви задания на одной ЭМ. Параметры задания имеют равномерное распределение в следующих интервалах: $r \in [1, R]$, $h \in [1, H]$, $g \in [1, G]$. В качестве единицы времени в модели принято время обмена служебным сообщением между двумя соседними ЭМ.

Задания, находящиеся в каждой ЭМ, распределены по очередям T, L, C нераспределенных, загружаемых и исполняемых заданий. Следовательно, внутреннее состояние $ЭМ_i$, $i \in \{1, 2, \dots, N\}$, характеризуется состоянием очередей T_i, L_i и C_i . Нагрузку машины ЭМ будут характеризовать следующие две суммы длин очередей: $\chi_i = |T_i| + |L_i| + |C_i|$ и $\pi_i = |L_i| + |C_i|$. Пусть на длины очередей и нагрузку наложены следующие ограничения: $T_i \leq M$ и $\pi_i \leq M$, где $M > 0$ - некоторая константа, задающая желаемую степень мультипрограммирования ЭМ. Дисциплина выбора заданий на обслуживание из каждой очереди - "первый пришел-первый обслужен".

Децентрализованный алгоритм распределения заданий осуществляет перераспределение заданий между ЭМ с целью выравнивания на-

грузки машин системы. Данный алгоритм реализуется в каждой ЭМ процессом РАСПРЕДЕЛЕНИЕ_i, $i \in \{1, 2, \dots, N\}$. Задание поступает в систему в ЭМ_i, если $|T_i| \leq M$. Задание, поступившее в ЭМ_i, заносится в очередь нераспределенных заданий T_i . При ненулевой длине T_i , для заданного радиуса ρ , происходит опрос машин из окрестности $L_\rho(i)$ с целью определения средней нагрузки машин по данной окрестности и номера машины, имеющей минимальную нагрузку:

$$Mx_i = \left(\sum_{k \in L_\rho(i)} x_k \right) / |L_\rho(i)|, \quad m = \operatorname{argmin}_{k \in L_\rho(i)} x_k.$$

Если $(x_i - Mx_i) \leq \delta$, где $\delta \geq 0$ — заданная величина, то состояние ЭМ_i остается без изменения. Иначе задание из очереди T_i машины ЭМ_i пересылается в очередь T_m машины ЭМ_m. Аналогично задание обрабатывается процессом РАСПРЕДЕЛЕНИЕ_m и в ЭМ_m. Таким образом, пересылка задания, поступившего в систему, продолжается до тех пор, пока задание не будет помещено в очередь загружаемых заданий какой-либо ЭМ. Переход задания из очереди T_j нераспределенных заданий ЭМ_j в очередь L_j загружаемых заданий происходит, если $(x_j - Mx_j) > \delta$ и $\pi_j < M$, после чего наступает этап формирования связанной подсистемы с требуемым числом машин для исполнения задания. Формирование такой подсистемы начинается с того, что машина ЭМ_j, в которой произошел переход к этапу формирования данной подсистемы, включается во вновь формируемую подсистему как корневая ЭМ. Далее формирование подсистемы происходит по алгоритму, описанному в [2], путем включения в подсистему наименее нагруженных ЭМ, непосредственно связанных с машинами, уже входящими в подсистему.

После завершения этого этапа происходит переход задания из очереди загружаемых заданий в очередь распределяемых заданий во всех ЭМ сформированной подсистемы. Машины выделенной подсистемы приступают к выполнению данного задания. При этом предполагается, что ресурсы каждой ЭМ делятся между различными заданиями, входящими в очередь исполняемых заданий.

Моделирующая программа и моделируемые алгоритмы написаны на языке ЛЯПАС-М и реализованы на ЭВМ БЭСМ-6. Рассматриваемая модель позволяет исследовать вычислительные системы с числом ЭМ $N = 1-20$, степенью графа межмашинных связей $\nu < 8$, степенью мультипрограммирования $m = 2-20$.

В результате моделирования определяются математическое ожидание, дисперсия и стандартное отклонение величин нагрузки x_i и

π_i в каждой ЭМ_i, $i \in P$. Определяются также следующие показатели:

1) коэффициент занятости каждой ЭМ_i: $k_{u1} = T_{c1}/T_m$, где T_{c1} - время, затраченное ЭМ_i на выполнение заданий, T_m - общее время моделирования;

2) средний коэффициент занятости ЭМ: $k_u = \frac{1}{N} \sum_{i=1}^N k_{u1}$;

3) степень неравномерности загрузки ЭМ характеризует стандартное отклонение коэффициента занятости ЭМ:

$$\sigma_u = \sqrt{\frac{N}{\sum_{i=1}^N (k_{u1} - k_u)^2 / (N-1)}};$$

4) производительность (пропускная способность) системы - среднее число заданий, исполняемых системой в единицу времени: $\eta = |Q_m|/T_m$, где $|Q_m|$ - число заданий, исполненных на системе за период моделирования T_m ;

5) среднее время распределения задания - время с момента поступления задания в систему до постановки его в очередь загружаемых заданий T_p ;

6) среднее время формирования подсистем - время с момента постановки задания в очередь загружаемых заданий до постановки его в очередь исполняемых заданий T_f ;

7) среднее время решения задания - время с момента постановки задания в очередь исполняемых заданий до того момента, когда задание покинет систему T_c ;

8) среднее время обслуживания (пробывания) задания в системе $T_{об} = T_p + T_f + T_c$;

9) коэффициент, характеризующий относительный расход времени на распределение задания $k_p = T_p/T_c$;

10) коэффициент, характеризующий относительный расход времени на формирование подсистемы для исполнения задания $k_f = T_f/T_c$;

11) средний коэффициент простоя ЭМ: $k_{п} = 1 - k_u$.

Моделирование проводилось для кольцевых одномерных структур с числом вершин 10 и 20 и для R_1 (10,4,4)-графа с описанием {10; 1; 1,4,6,9} [7], при следующих значениях параметров: $N = 10, 20$; $v = 2, 4$; $M = 3, 5, 6, 7, 10, 20$; $\rho \neq 1-7$; $\lambda = 0, 1-0, 9$; $n = 2, 5, 10$; $R = 1, 4, 5, 10$; $\alpha = G/N = 1, 5, 10, 25, 50, 100$; одна и две точки поступления заданий в ВС.

Анализ результатов моделирования показывает следующее.

I. Наличие явления "насыщения", которое заключается в том, что при увеличении интенсивности поступления заданий λ , начиная с не-

которой точки λ_s , значение характеристик η , k_{II} , $T_{об}$, σ_{II} остается постоянным. Например, на рис. I, а, $\lambda_s = 0,3$. Это явление объясняется наличием конечной входной очереди для поступающих в ВС заданий. Длина данной очереди в каждой ЭМ ограничена величиной M . При интенсивности λ_s входная очередь заполнена, поступающие задания получают отказ, увеличение интенсивности поступления заданий $\lambda > \lambda_s$ не увеличивает числа заданий в системе и, следовательно, не оказывает влияния на перечисленные характеристики системы.

2. Существование оптимальных значений радиуса опрашиваемой окрестности ρ , при которых достигается максимум величин η и k_{II} и минимум величин $T_{об}$ и σ_{II} . Так, при увеличении ρ уменьшается $T_{об}$ (см. рис. I) за счет более равномерного распределения нагрузки, с другой стороны, при увеличении ρ увеличивается T_p за счет увеличения времени просмотра окрестности данного радиуса. Эти две взаимопротивоположные тенденции обуславливают наличие оптимального радиуса ρ^* , при котором производительность системы η имеет максимальную величину. На рис. I величина $\rho^* = 2$.

Существование оптимального радиуса опрашиваемой окрестности ρ , обеспечивающего максимальное значение k_{II} и минимум степени неравномерности загрузки ЭМ σ_{II} , объясняется тем, что, с одной стороны, увеличение ρ позволяет находить наименее нагруженные процессоры и выравнивать их нагрузку, а, с другой стороны, при увеличении ρ происходит увеличение задержки при передаче информации о состоянии системы, старение этой информации, и, следовательно, уменьшение точности управления нагрузкой машин. На рис. I максимальное значение k_{II} и минимум σ_{II} достигаются при $\rho^* = 2$. Таким образом, при $\rho < \rho^*$ эффективность системы уменьшается из-за отсутствия достаточно полной информации о состоянии системы, а при $\rho > \rho^*$ — из-за больших накладных расходов на получение такой информации. Выбор оптимального радиуса ρ основан на разумном компромиссе между этими двумя тенденциями и определяется заданным критерием эффективности.

3. Наличие оптимальных значений степени мультипрограммирования ЭМ и допустимой длины очередей — M^* , при которых достигается максимум величин η и k_{II} и минимум σ_{II} и $T_{об}$. Так, при увеличении $M > M^*$ увеличивается средняя загрузка ЭМ — k_{II} и уменьшается степень неравномерности их загрузки — σ_{II} (рис. I и 2), что объясняется возрастанием числа заданий в системе и возможностью их более равномерного распределения. Как следствие этого, при увеличении M происходит и увеличение производительности системы —

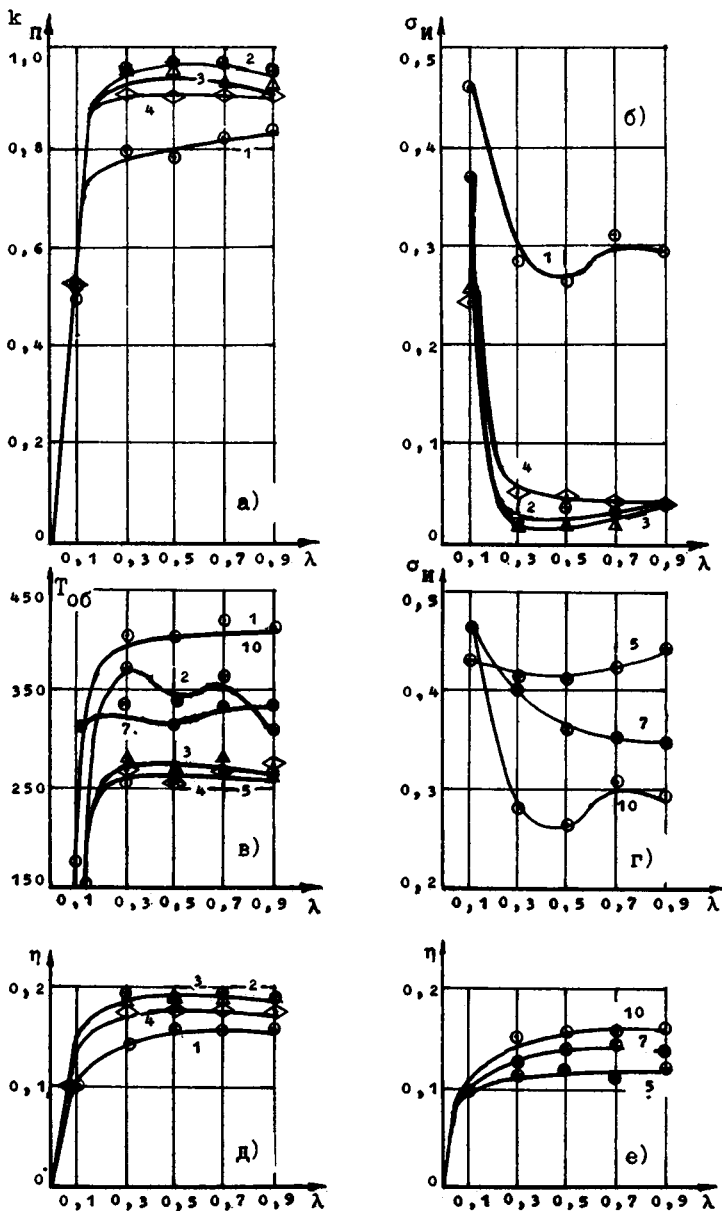


Рис. I. Результаты моделирования для системы с параметрами:
 $N = 10, V = 2, G = 100, \Gamma = 1, H = 10, \Pi = 1, M = 10$.
 Кривая 1 - $\rho = 1, 2 - \rho = 2, 3 - \rho = 3, 4 - \rho = 4,$
 $5 - m = 5, 7 - m = 7, 10 - m = 10,$ при $\rho = 1$.

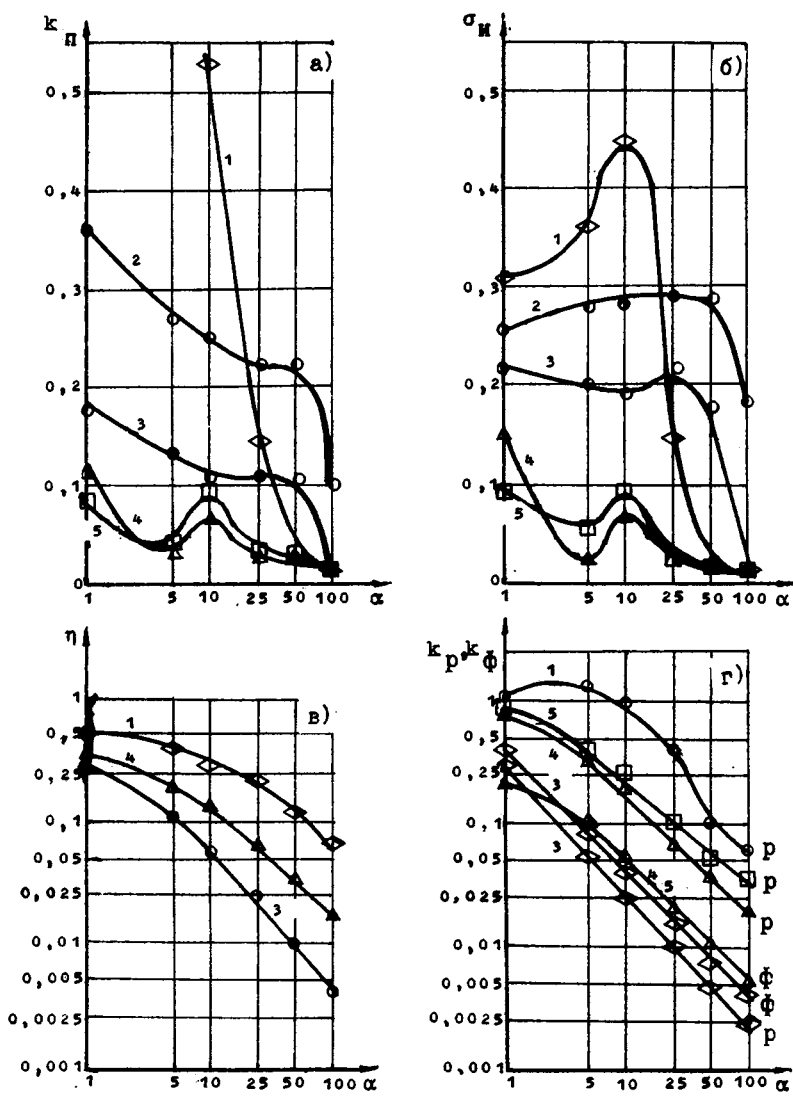


Рис. 2. Результаты моделирования для системы с параметрами: $N = 10, V = 4, \Gamma = 5, H = 5, \text{Ц} = 1$. Кривая 1 - $r = 1, \rho = 2, M = 6$; 2 - $r = 10, \rho = 2, M = 3$; 3 - $r = 10, \rho = 1, M = 6$; 4 - $r = 5, \rho = 1, M = 6$; 5 - $r = 5, \rho = 2, M = 6$.

п (рис.1 и 2). Но, с другой стороны, увеличение M приводит также к увеличению времени ожидания заданий в очередях T , L и C , и, следовательно, к увеличению времени пребывания задания в системе - $T_{об}$ (рис.1). Таким образом, выбор оптимального значения M определяется заданным критерием эффективности и происходит на основе компромисса между стремлением увеличить производительность системы η и уменьшить время обслуживания каждого задания - $T_{об}$.

4. Существование оптимальной величины максимального ранга подсистемы R^* (или максимального числа ЭМ, требуемых для исполнения задания), при которой достигается минимум величин $\sigma_{и}$ и $k_{п} = 1 - k_{и}$. Средний коэффициент простоя ЭМ - $k_{п}$ и степень неравномерности нагрузки ЭМ $\sigma_{и}$ принимают минимальные значения при $R^* = N/2$ (рис.2). При $R < R^*$ происходит недогрузка системы, обусловленная ограниченной пропускной способностью сети связи системы при пересылке заданий, что увеличивает значения $k_{п}$ и $\sigma_{и}$. При $R > R^*$ увеличение значений $k_{п}$ и $\sigma_{и}$ происходит из-за недостаточного числа свободных ЭМ для выполнения заданий, требующих большого числа ЭМ.

При увеличении значения R происходит уменьшение времени распределения задания - T_p и уменьшение коэффициента затрат времени на распределение - k_p , а также уменьшается влияние величины радиуса опрашиваемой окрестности - ρ на характеристики системы ($\sigma_{и}$, $k_{п}$, η) (см.рис.2). Это объясняется увеличением равномерности нагрузки на системе в результате работы алгоритма формирования подсистем.

Уменьшение пропускной способности системы - η с ростом R (рис.2) объясняется тем, что каждое задание в этом случае требует для своего исполнения в среднем большее число ЭМ, т.е. подсистему большего ранга, это обуславливает уменьшение числа подсистем, одновременно размещаемых на ВС, и, следовательно, уменьшение числа заданий, решаемых в каждый момент времени на ВС.

5. Рассмотрим влияние отношения максимального времени счета заданий к максимальному времени передачи заданий - $\alpha = G/N$ на характеристики системы. Величина производительности системы - η и относительные расходы на управление - k_p и $k_{ф}$ обратно пропорциональны либо величине α , либо некоторой степени величины α (см. рис.2). Так, при $R \geq N/2$ относительные расходы на управление определяются степенной зависимостью: $k_{p(\phi)} = c/\alpha^m$, где c и m - постоянные. Например, $c = 0,25; 0,4; 0,5; 1,8$ и $m = 1$ (кривая Зр, Зф, 4ф, 4р на рис.2). Аналогичная зависимость получена при $R \geq N/2$

и $\alpha \geq 5$ для производительности системы: $\eta = c/\alpha^m$, где c и m - постоянные. Например, $c = 1,1$ и $m = 0,9$ (кривая 4 на рис.2), $c = 0,8$ и $m = 1,15$ (кривая 3 на рис.2).

При увеличении α в диапазоне 1-25 в некоторых случаях происходит увеличение значений k_{Π} и σ_{Π} (рис.2). Это объясняется тем, что алгоритмы распределения заданий не успевают достаточно точно реагировать на относительно быстрое изменение состояния системы, и, следовательно, происходит недостаточно полная и равномерная загрузка машин системы. При дальнейшем увеличении $\alpha > 25$ увеличивается время счета заданий, относительная скорость изменения состояния системы уменьшается, точность управления возрастает, что приводит к постоянному уменьшению значений k_{Π} и σ_{Π} (рис.2).

Влияние радиуса опрашиваемой окрестности ρ на характеристики системы (k_{Π} , σ_{Π} , η) при увеличении α в диапазоне 1-10 постоянно уменьшается и при $\alpha \geq 25$ - практически незаметно (рис.2).

6. При увеличении числа точек поступления заданий в систему уменьшается степень неравномерности загрузки машин - σ_{Π} и коэффициент простоя - k_{Π} , кроме того, в этом случае, при больших значениях $\alpha > 50$ уменьшаются относительные расходы на распределение и формирование подсистем (k_p и k_{Φ}).

Таким образом, проведенный анализ результатов моделирования показывает возможность эффективной организации в ВС с программируемой структурой адаптивного децентрализованного распределения заданий, характеристики которых априорно неизвестны. Экспериментально доказано существование оптимальных значений величин радиуса опрашиваемой окрестности ρ^* и допустимого размера очередей M^* , обеспечивающих экстремум заданных критериев эффективности ВС. Увеличение степени централизации управления в этом случае, за счет увеличения радиуса опрашиваемой окрестности $\rho > \rho^*$, приводит к увеличению расходов на управление и снижению эффективности функционирования ВС. Следует отметить, что при увеличении ранга или времени исполнения поступающих заданий оптимальное значение величины радиуса ρ^* уменьшается, а степень децентрализации алгоритма управления распределением заданий - увеличивается. Следовательно, в тех случаях, когда для заданного уровня производительности ВС расходы на централизованное управление превышают расходы на управление с помощью данного децентрализованного алгоритма, применение децентрализованного алгоритма для ВС с рассматриваемыми параметрами является более эффективным. Получены простые расчетные фор-

мулы для определения производительности ВС и расходов на управление распределением заданий для ВС с рассматриваемыми параметрами. Показана зависимость характеристик функционирования ВС от локальных параметров ВС и параметров поступающих заданий.

Автор выражает благодарность Корнееву В.В. за полезные обсуждения данной работы и Авдеенко И.Н. за помощь в проведении машинных экспериментов.

Л и т е р а т у р а

1. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой структурой. - Электронное моделирование, 1979, № 1, с. 42-50.
2. КОРНЕЕВ В.В., МОНАХОВ О.Г. О децентрализованном распределении заданий в вычислительных системах с программируемой структурой. - Электронное моделирование, 1981, № 6, с.15-22.
3. МОНАХОВ О.Г. О распределенном динамическом управлении ресурсами в вычислительных системах с программируемой структурой. - АВТ, 1984, № 3, с. 9-17.
4. МОНАХОВ О.Г. Об одном методе децентрализованного распределения заданий в вычислительных системах с программируемой структурой. - АВТ, 1983, № 1, с. 7-13.
5. МОНАХОВ О.Г., МОНАХОВА Э.А. Организация межмашинных взаимодействий в вычислительной системе с программируемой структурой МИКРОС. - Настоящий сборник, с. 85-104.
6. МОНАХОВ О.Г., АВДЕЕНКО И.Н. Исследование децентрализованного алгоритма распределения заданий в вычислительных системах с программируемой структурой. У I Всесоюзная школа-семинар по вычислительным сетям. Тезисы докладов. Ч.2.М. - Винница, 1981, с.33-37.
7. 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.

Поступила в ред.-изд.отд.
9 октября 1984 года