И. А. ПОГОСЯН

ОБ ОДНОМ АЛГОРИТМЕ ОПТИМАЛЬНОГО ПЛАНИРОВАНИЯ В МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСАХ

Одной из актуальных проблем при проектировании и эксплуатации многомашинных вычислительных комплексов является проблема оптимального планирования или оптимального распределения задач между ЭВМ (процессорами), входящими в состав системы.

В настоящей работе предлагается алгоритм определения оптимального плана распределения задач, который основан на применении стохастических управляемых процессов. Сущность данного алгоритма заключается в следующем. Пусть задан набор задач в ярусно-параллельной форме [1], решаемый в системе из n машин разной производительности за минимальное время. Предполагается, что в любой произвольный момент каждая машина занята выполнением лишь одного оператора. Причем оператор, поступивший на какую-либо машину, обрабатывается до конца. Распределение операторов по машинам производится с учетом функциональных связей, заданных главной матрицей [1], начиная с нулевого яруса. Введем элемент $L_{x\nu}$, соответствующий распределению x-го оператора на ν-ю машину

$$L_{x\nu}=\begin{cases}1, & \text{если } x \text{ оператора решается на } \nu\text{-й машине,}\\ 0, & \text{в противном случае}\end{cases}$$

Распределение операторов рассмотрим как некоторый многоэтапный процесс, при котором каждая из n машин может получить тот или иной оператор или остаться свободной, в зависимости от принятой стратегии $z \in Z$ (или принятого управления на данном этапе).

Иными словами, управление определяет вариант распределения операторов по машинам, что соответствует определенной совокупности $L_{x\nu}$.

Назовем каждую совокупность из n элементов $L_{x\nu}$ состоянием программы и обозначим как S_k. При такой интерпретации появление нового варианта распределения можно рассматривать как переход от одного состояния реализуемой программы к другому.

Следует отметить, что распределение операторов каждого яруса может осуществляться в два и более этапа, если число операторов данного яруса превышает число машин АСОИ, т. е. данный алгоритм предполагает асинхронный процесс планирования, когда переход от одного состояния программы к другому осуществляется по мере освобождения машин от решения полученных операторов. При этом в одних случаях может быть более оптимальным распределение оставшихся операторов, в других — двух и более машин. Следовательно, под управлением на каждом шаге следует понимать управляющее воздействие, вызывающее переход программы от одного состояния к другому в моменты, когда будет решен в общем случае

а) один оператор;

б) два оператора;

. . . .

в) n операторов.

В случае решения n операторов процесс планирования становится синхронным. Как следует из ранее сказанного, выбор каждого последующего состояния зависит от предыдущего состояния и управляющего воздействия. Переход от одного состояния к другому осуществляется в случайные моменты времени с вероятностью, определяемой принятой стратегией. Время пребывания в каждом из состояний в общем случае случайно и определяется наименьшим, в смысле времени выполнения, оператором из входящих в данное распределение, если принята стратегия перехода из одного состояния в другое по мере решения одного оператора, и максимальным из наименьших операторов во всех остальных случаях.

Зависимость между состояниями легко представить в виде граф-дерева, где вершинами служат состояния программы, дуги показывают вероятность перехода от одного состояния к другому в зависимости от стратегии. Такое представление процесса переходов от одного состояния программы к другому указывает на возможность его описания некоторым полумарковским процессом $\Psi^{(z)}(t)$. Полумарковский процесс $\Psi^{(z)}(t)$ определен на множестве состояний программы $S=\{s_k;\ k=1,\ 2\ .\ .\ .\ M\}$ переходными вероятностями $P_{kl}^{(z)}(t)$, зависящими от стратегии $z \in Z$ $(l=1,\ 2\ .\ .\ .\ M)$. После каждого распределения операторов происходит реализация их на машинах. При этом можно выделить следующие фазы, которые проходит любая машина системы в процессе ее функционирования:

а) машина произвела обмен информацией и начала решать полученный оператор;

б) машина решает оператор;

в) машина закончила обработку оператора и готовится к обмену информацией;

г) машина простаивает.

Совокупность фаз, в которых находятся каждая из n машин, назовем состоянием системы. При изменении фазы хотя бы одной машины образуется новое состояние системы. Очевидно, что каждое состояние системы зависит лишь от предыдущего состояния. Действительно, если какая-либо из машин была, например, в фазе (б), то следующей фазой будет (в) независимо от того, какая фаза была перед (б) и т. д. Смена фаз каждой машины происходит в случайные моменты времени, откуда следует, что и переход системы из одного состояния в другое осуществляется также в случайные моменты времени. На основании вышесказанного можно предположить, что смена состояний системы может быть описана, в общем случае, полумарковским процессом $F^{(k)}(t)$, заданном вероятностями перехода $P_{ij}^{(k)}(t)$ на множестве состояний системы $H=\{h_i\}$, где $i, j=1, 2, \ldots, N$. Индекс k указывает на зависимость полумарковского процесса $F^{(z)}(t)$ от k-го состояния полумарковского процесса $\Psi^{(z)}(t)$, поэтому назовем их соответственно управляемым и управляющим полумарковскими процессами. Указанная зависимость обусловлена тем, что каждое новое состояние программы (состояние $\Psi^{(z)}(t)$) характеризуется появлением новых операторов на освободившихся машинах, что приводит в свою очередь как к изменению состояния системы, так и вероятностей переходов системы от одного состояния к другому. В такой интерпретации процесс планирования описывается двумерным процессом $W^{(z)}(t)=[\Psi^{(z)}(t)F^{(k)}(t)]$, компоненты которого являются управляемый и управляющий полумарковские процессы. Двумерный процесс определяется на конечном множестве состояний $\Omega=S\times H$ вектором начальных условий $W^{(z)}(0)=[\Psi^{(z)}(0), F^{(k)}(0)]$ и матрицей переходных вероятностей $P=\|P_{ij}^{kl(z)}\|$ для каждой стратегии $z\in Z$.

$P_{ij}^{kl(z)}(t)$ — вероятность того, что за время, меньшее чем t, управляемый полумарковский процесс (ПМП) $F^{(k)}t$ перейдет из состояния h_i при состоянии S_k управляющего полумарковского процесса $\Psi^{(z)}(t)$ в состояние h_j процесса $F^{(k)}(t)$ при S_l-ом состоянии управляющего ПМП.

Значение $P_{ij}^{kl}(t)$ для случая, когда точки регенерации управляющего полумарковских процессов совпадают, определяется по формуле полной вероятности:

$$P_{ij}^{kl(z)}(t)=\sum_{r\in E}\int_0^t b_{rj}^{kl}\Phi_{ir}^{k(z)}(u)dP_{kl}^{(z)}(u) \tag{1}$$

$$k, l=1, 2 \ldots M \qquad 0\leqslant u\leqslant t$$

$$i, j=1, 2 \ldots N$$

где b_{rj}^{kl} — неотрицательные величины, равные вероятностям перехода управляемого ПМП из состояния h_r, когда управляющий ПМП находится в состоянии $S_k\in S$, состояние h_j управляемого ПМП при S_l-ом состоянии процесса $\Psi^{(z)}(t)$ и удовлетворяющие условию:

$$\sum_{j=1}^{N}\sum_{l=1}^{M} b_{ij}^{kl}=1.$$

$P_{kl}^{(z)}(u)$—вероятность того, что за время, меньшее чем u, управляющий ПМП $\Psi^{(z)}(t)$ перейдет из состояния S_k в состояние S_l в соответствии со стратегией $z \in Z$.

$\Phi_{ir}^{k(z)}(u)$—вероятность того, что в момент u управляемый ПМП $F^{(k)}(t)$ окажется в состоянии h_r, если в начальный момент находится в состоянии h_i при состоянии S_k управляющего ПМП, стратегии $z \in Z$.

Величина $\Phi_{ir}^{k(z)}(t)$ вычисляется согласно [2] из соотношения:

$$\Phi_{ir}^{k(z)}(t)=\delta_{ir}[1-P_i^{k(z)}(t)]+\sum_{l\in H}\int_0^t \Phi_{lr}^{k(z)}(t-\tau)dP_{il}^{k(z)}(\tau) \tag{2}$$

здесь τ—время пребывания управляемого полумарковского процесса $F^{(k)}(t)$ в состоянии h_i до попадания в состояние h_r, когда управляющий процесс находится в состоянии S_k.

$P_{ij}^{kl(z)}(t)$—вероятность того, что управляемый ПМП $F^{(k)}(t)$ проведет в состоянии h_i время, не большее чем t, и перейдет в состояние h_r; управляющий ПМП $\Psi^{(z)}(t)$ находится в k-ом состоянии.

δ_{ir} — символ Кронекера

$$\delta_{ir}=\begin{cases}1, & i=r\\ 0, & i\neq r.\end{cases}$$

Определив таким образом семейство функций распределения $P_{ij}^{kl(z)}(t)$, характеризующих исследуемый процесс для каждой стратегии, можно перейти к определению оптимальной стратегии. Оптимальной стратегии будет соответствовать наибольший „доход" (или наименьший „штраф"), получаемый системой за решения поступающего набора задач. Под доходом в данном случае подразумевается стоимость в условных единицах времени решения набора задач. Стоимость решения складывается из стоимостей машинного времени, времени простоев и обмена. Если назначить за реализованное машинное время положительный доход, за время простоев и обмена — отрицательный, то естественно следует максимизировать положительный доход (или минимизировать отрицательный доход — „штраф"), что будет соответствовать оптимизации производительности системы.

С целью определения оптимальной стратегии запишем функциональное уравнение динамического программирования [3], которое в данном случае для процесса, начинающегося в $\begin{Bmatrix}k\\ i\end{Bmatrix}$ и эволюционирующего при оптимальной стратегии $z^* \in Z$ на каждом шаге, примет вид:

$$Q_i^k(m)=\max_{z\in Z}\left[q_i^{k(z)}+\sum_{l=1}^{M}\sum_{j=1}^{N}P_{ij}^{kl}Q_j^l(m-1)\right], \tag{3}$$

где $Q_i^k(m)$ — полный оптимальный доход, получаемый системой за m шагов, если начальным состоянием процесса $W^{(z)}(t)$ является $\begin{Bmatrix}k\\ i\end{Bmatrix}\in\Omega$;

$q_i^{k(z)}$ — одношаговый доход, получаемый системой в среднем, если процесс $W^{(z)}(t)$ выходит из состояния $\begin{Bmatrix} k \\ i \end{Bmatrix} \in \Omega$ при стратегии $z \in Z$.

$P_{ij}^{kl(z)}$ — вероятность перехода вложенной цепи Маркова двумерного процесса $W^{(z)}(t)$ из состояния $\begin{Bmatrix} k \\ i \end{Bmatrix}$ в состояние $\begin{Bmatrix} l \\ j \end{Bmatrix}$ и стратегии $z \in Z$. Согласно [4],

$$P_{ij}^{kl(z)} = P_{ij}^{kl(z)}(+\infty) \sum_{l=1}^{M} \sum_{j=1}^{N} P_{ij}^{kl(z)} = 1. \tag{4}$$

Величина $q_i^{k(z)}$ вычисляется из выражения, аналогичного соответствующему соотношению из работы [5]

$$q_i^{k(z)} = \sum_{l=1}^{M} \sum_{j=1}^{N} q_{ij}^{kl(z)} P_{ij}^{kl(z)}, \tag{5}$$

где $q_{ij}^{kl(z)}$ — ожидаемый доход за время перехода процесса $W^{(z)}(t)$ из состояния $\begin{Bmatrix} k \\ i \end{Bmatrix}$ в состояние $\begin{Bmatrix} l \\ j \end{Bmatrix}$ при стратегии $z \in Z$:

$$q_{ij}^{kl(z)} = \int_0^{\infty} G_{ij}^{kl(z)}(t) dP_{ij}^{kl(z)}(t), \tag{6}$$

здесь $G_{ij}^{kl(z)}(t)$ — функция дохода, зависящая от $\begin{Bmatrix} k \\ i \end{Bmatrix}$, $\begin{Bmatrix} l \\ j \end{Bmatrix}$ и времени t с момента начала переходного интервала τ_{kl} $(0 \leq t \leq \tau_{kl})$. В частности, для данного случая функция дохода может иметь вид:

$$G_{ij}^{kl(z)}(t) = \begin{cases} G_{ij}^{kl(z)} + \chi_{ij}^{kl(z)} \cdot t, \text{ если } 0 \leq t \leq \tau_{kl}, \\ 0 \end{cases}$$

где $G_{ij}^{kl(z)}$ — фиксированная величина штрафа, выплачиваемая системой за переход процесса $W^{(z)}(t)$ из состояния $\begin{Bmatrix} k \\ i \end{Bmatrix}$ в состояние $\begin{Bmatrix} l \\ j \end{Bmatrix}$, соответствующая условным единицам, начисляемым за время, необходимое системе для обмена информацией.

χ_{ij}^{kl} — средняя интенсивность роста дохода за переход системы из состояния $\begin{Bmatrix} k \\ i \end{Bmatrix}$ в состояние $\begin{Bmatrix} l \\ j \end{Bmatrix}$.

В большинстве практических случаев нас интересует работа системы в стационарном режиме, из чего следует, что рекуррентное уравнение (3) необходимо решать для бесконечного числа шагов. В этом случае, как доказано в работе [6], оптимальные доходы асимптотически приближаются к величине, определяемой выражением:

$$Q_i^{k(z)}(m) \approx R^{(z)} \cdot m + V_i^{k(z)}, \tag{7}$$

где $V_i^{k(z)}$ — разность истинных и средних значений дохода за m шагов, если начальным состоянием процесса $W^{(z)}(t)$ является $\begin{Bmatrix} k \\ i \end{Bmatrix}$. $R^{(z)}$— средний доход, получаемый системой за один шаг в соответствии со стратегией $z \in Z$, значение которого вычисляется по формуле

$$R^{(z)} = \sum_{k=1}^{M} \sum_{i=1}^{N} \pi_{ij}^{kl(z)} q_i^{k(z)}, \tag{8}$$

где $\pi_i^{k(z)}$ — стационарные вероятности вложенной марковской цепи процесса $W^{(z)}(t)$, которые существуют при условии эргодичности цепи Маркова с переходными вероятностями $P_{ij}^{kl(z)}$ и определяются, согласно [7], из системы линейных алгебраических уравнений:

$$\pi_i^{k(z)} = \sum_{l=1}^{M} \sum_{j=1}^{N} \pi_{ij}^{kl(z)} P_{ij}^{kl(z)}$$

$$\sum_{k=1}^{M} \sum_{i=1}^{N} \pi_i^k = 1.$$

Учитывая связность параметров k и i, а также l и j, обозначим каждую пару $\begin{Bmatrix} k \\ i \end{Bmatrix}$ и $\begin{Bmatrix} l \\ j \end{Bmatrix}$ соответственно через c и d $(c, d = 1, 2 \ldots$ $\ldots M \times N)$. Тогда для однородной стратегии можно (3) записать в векторной форме

$$Q^{(z)}(m) = q^{(z)} + P^{(z)} Q^{(z)}(m-1), \tag{10}$$

где $Q^{(z)}(m) = \|Q_c^{(z)}(m)\| = \|Q_i^{k(z)}(m)\|$ — вектор столбец доходов, ожидаемых на m-ом шаге.

$q^{(z)} = \|q_c^{(z)}\| = \|q_i^{k(z)}\|$ — вектор столбец одношаговых доходов

$P^{(z)} = \|P_{cd}^{(z)}\| = \|P_{ij}^{kl(z)}\|$ — матрица переходных вероятностей вложенной цепи Маркова процесса $W^{(z)}(t)$. Из рекуррентного уравнения (10) следует

$$Q(m) - Q(m-1) = P^{(m-1)} q. \tag{11}$$

Предположение эргодичности вложенной цепи Маркова, а следовательно, сходимости матрицы $P^{(m-1)}$ к стационарной матрице

$$\Pi = \|\pi_i^k\|$$

позволит записать:

$$\lim \left[Q(m) - Q(m-1)\right] = \Pi q$$

или:

$$\lim_{m \to \infty} \left[Q_c(m) - Q_c(m-1)\right] = \sum_{c=1}^{MN} \pi_c q_c = \sum_{N}^{M} \sum_{i=1}^{N} \pi_i^k q_i^k.$$

Из выражения (7) следует

$$V(m) = Q(m) - R \cdot m, \tag{13}$$

где $V(m)=\|V_c(m)\|=\|V_i^k(m)\|$ — вектор разностей истинных и средних значений дохода за m шагов.

Учитывая соотношения (8) и (11), выражение (13) запишем в виде:

$$V(m)=\left[J+\sum_{\alpha=1}^{m-1} P^{\alpha}-\Pi\right]q+Q(0), \tag{14}$$

где J—единичная матрица

$Q(0)=\|Q_i^k(0)\|$—начальный доход, получаемый системой.

Из эргодичности вложенной цепи Маркова следует

$$\lim_{m\to\infty}\left[J+\sum_{\alpha=1}^{m-1}(P^{\alpha}-\Pi)\right]=X,$$

где $X=\|x_{cd}\|$ — фундаментальная матрица цепи Маркова с матрицей переходных вероятностей.

Тогда

$$V=\lim_{m\to\infty} V(m)=(X-\Pi)q+Q(0) \tag{15}$$

Подставив в (15) соответствующее значение Πq из (8), получим

$$V=Xq-R+Q(0). \tag{16}$$

Для всех $\begin{Bmatrix}k\\ i\end{Bmatrix}$ выражение (16) примет вид

$$V_i^{k(z)}\approx R^{(z)}+\sum_{l=1}^{M}\sum_{j=1}^{N} x_{ij}^{kl(z)}\cdot q_j^{l(z)}+Q^{k(z)}(0). \tag{17}$$

Подставив (17) в (7), получим формулу полного дохода за m шагов

$$Q_i^{k(z)}(m)=R^{(z)}\cdot m-R^{(z)}+\sum_{l=1}^{M}\sum_{j=1}^{N} x_{ij}^{kl(z)}\,q_j^{l(z)}+Q_i^{k(z)}. \tag{18}$$

Задача определения оптимальной стратегии не может быть решена непосредственным вычислением доходов из (18) для всех i, k и z, так как потребуется достаточно большое количество вычислений. Поэтому, подставив (7) в (3), запишем систему из MN уравнений с $MN+1$ неизвестными:

$$R^{(z)}+V_i^{k(z)}=q_i^{k(z)}+\sum_{l=1}^{M}\sum_{j=1}^{N} P_{ij}^{kl(z)}V_j^{l(z)}. \tag{19}$$

Поскольку нет необходимости вычислять действительные значения неизвестных $V_i^{k(z)}$, приравняем одно из $V_i^{k(z)}$, например, $V_N^{M(z)}$ нулю, тогда количество уравнений станет равным количеству неизвестных. В результате решения преобразованной системы уравнений будут получены относительные значения неизвестных $V_i^{k(z)}$ и $R^{(z)}$, которые обозначим как $V_i^{k(z)}$ и $P^{(z)}$. Выражение (19) примет вид:

$$\rho^{(z)}+v_i^{k(z)}=q_i^{k(z)}+\sum_{l=1}^{M-1}\sum_{j=1}^{N-1} P_{ij}^{kl(z)}v_j^{l(z)}+\sum_{j=1}^{N-1} P_{ij}^{kM}v_j^{M(z)} \tag{20}$$

$$v_N^{M(z)}=0,$$

$$\rho^{(z)} = q_N^{M(z)} + \sum_{l=1}^{M-1} \sum_{j=1}^{N} P_{Nj}^{Mlz} v_j^{l(z)} + \sum_{j=1}^{N-1} P_{Nj}^{MM} v^{M(z)} \quad (2$$

Система (19) позволяет определить оптимальную стратегию, используя алгоритм из работы [5] с соответствующими для данного случая изменениями.

Критерием улучшения служит правая часть уравнения (20). Алгоритм определения оптимальной стратегии дан на блок-схеме (рис. 1). Последний состоит из двух циклов. В зависимости от исходных данных процедура начинается или с первого цикла, или со второго. Однако выбор первоначального состояния или решения, согласно принципу оптимальности [3], не влияет на конечный результат. Процедура заканчивается, если совпадают два последовательных решения второго цикла.

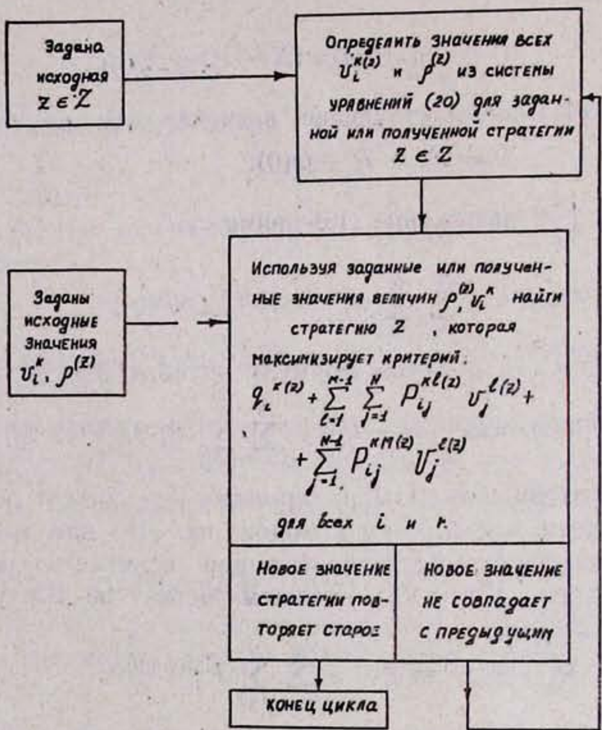

Рис. 1

Определив оптимальную стратегию Z^* и соответствующие ей значения $v_l^{k(z^*)}$, вычисляем величину среднего дохода $R^{(z^*)}$ по формуле (8). Фундаментальную матрицу X вложенной цепи Маркова из [7], а затем значение полного оптимального дохода, получаемого системой при использовании оптимальной стратегии и начальном состоянии $\begin{Bmatrix} k \\ l \end{Bmatrix}$ по формуле (18)

$$Q_i^k(m)=R^{(z)}\cdot m-R^{(z^*)}+\sum_{l=1}^{M}\sum_{j=1}^{N}x_{ij}^{kl(z^*)}q^{l(z^*)}+Q_i^k(0),$$

где $Q_i^k(m)$ — начальный доход.

Полученное значение оптимальной стратегии соответствует оптимальному плану решения заданного набора задач, а оптимальный доход-оптимальному времени их решения.

Ի. Ա. ՊՈՂՈՍՅԱՆ

ԲԱԶՄԱՄԵՔԵՆԱՅԱԿԱՆ ՀԱՇՎՈՂԱԿԱՆ ՀԱՄԱԿԱՐԳԵՐՈՒՄ ՕՊՏԻՄԱԼ ՊԼԱՆԱՎՈՐՄԱՆ ՄԻ ԱԼԳՈՐԻԹՄԻ ՄԱՍԻՆ

Նախագծման ավտոմատացման համակարգերի մշակման և ներդրման ժամանակ այժմեական պրոբլեմներից մեկը օգտագործվող հաշվողական համակարգի պրոցեսորների միջև խնդիրների օպտիմալ բաշխումն է։ Այս աշխատանքում առաջարկված է խնդիրների բաշխման օպտիմալ պլանի որոշման ալգորիթմ, որը հիմնված է ստոխաստիկ կառավարվող պրոցեսների մեթոդների վրա։ Ստացված է խնդիրների տվյալ հավաքածուի լուծման օպտիմալ պլանին համապատասխանող օպտիմալ ստրատեգիայի արժեքը։

ЛИТЕРАТУРА

1. *В. С. Королюк.* Время пребывания полумарковского процесса в фиксированном множестве состояний, УМЖ, 1965, № 3.
2. *В. С. Королюк, А. Ф. Турбин.* Полумарковские процессы и их применение, Киев, 1976.
3. *В. С. Королюк, А. Ф. Турбин.* О некоторых стационарных характеристиках полумарковских процессов, «Кибернетика», 1965, № 5.