УДК 681.324

В. П. Симоненко, д-р техн. наук
Национальный технический университет Украины
«Киевский политехнический ин-т»
(Украина,03056, Киев, пр-т Победы, 37,
тел. (044) 4549044, E-mail: svp@comsys.ntu-kpi.kiev.ua)

Теоретические основы проектирования динамических пространственных планировщиков неоднородных GRID систем

Рассмотрена общая математическая модель динамического планирования в распределенной неоднородной GRID системе. Показано, что задача, ориентированная на вычислительный ресурс, сводится к проблеме поиска максимального паросочетания в двудольном графе.

Розглянуто загальну математичну модель динамічного планування в розподіленій неоднорідній GRID системі. Показано, що задача, орієнтована на обчислювальний ресурс, зводиться до проблеми пошуку максимального паросполучення у дводольному графі.

Ключевые слова: динамическое планирование, GRID системы, венгерский алгоритм.

Распределение задач по ресурсам в неоднородной системе распределенной обработки данных (GRID) [1] — одна из наиболее сложных задач организации распределенных вычислений. Сложность задачи распределения или динамического планирования обусловлена неоднородностью как объекта распределения, так и распределяемых задач. Наиболее известные планировщики или диспетчеры задач (заданий) для GRID систем Platform LSF, Windows HPC Server 2008, PBS, Condor, SGE , LoadLever, MOSIX и внешний планировщик MAUI [2—11] предназначены для оптимизации распределения потока задач (заданий) на ресурсы системы. Следует заметить, что при учете свойства неоднородности GRID системы такое распределение не всегда приводит к равномерной загрузке ресурсов [12] и требует применения нового класса пространственных планировщиков, учитывающих и приоритетность задач и неоднородность вычислительной системы.

Постановка задачи. В системе GRID, состоящей из N ресурсов, на момент времени распределения τ имеются N_τ свободных и M независимых ресурсов, готовых к выполнению заданий [1].

Система ресурсов задана графом системы $G_R = (V_R, E_R, W_{VR}, W_{ER})$, где

$V_R = \{R_1, R_2, ..., R_N\}$ — множество вершин, каждый элемент которого представляет собой один из N ресурсов системы, и $R_i \in N$, $i = 1, ..., N$, — множество натуральных чисел;

$E_R = \{E_1, E_2, ..., E_d\}$ — множество дуг, каждый элемент которого определяют связи между двумя ресурсами $E_i = \{R_i, R_j\}$, где $R_i, R_j \in V_R$ и $0 \le d \le N^2$;

$W_{VR} = \{WVR_1, WVR_2, ..., WVR_N\}$ — множество весов вершин, где $WVR_i = \{RE_i, RT_i\}$; для $\forall\, i = 1, ..., N$, $RE_i \in \Re+$ (множество положительных действительных чисел) есть характеристика состояния ресурсов R_i, $RT_i \in \{0$ и $\Re+\}$;

$W_{ER} = \{WER_1, WER_2, ..., WER_p\}$ — множество весов дуг, которое можно представить в виде некоторой матрицы $RC = RC\,[i, j] \in \Re+$, где $i = 1, ..., N$ и $j = 1, ..., N$.

Поток М заданий задан множеством $V_J = \{Job_1, Job_2, ..., Job_M\}$, каждый элемент которого представляет одно из M заданий и $Job_i = \{JN_i, JE_i, JL_i, JM_i, JP_i\}$, $\forall\, i = 1, ..., N$:

$JN_i \in N$ — номер задания;

$JE_i \in \Re^+$ — объем работы задания i;

$JL_i = \{(R^1, \varphi_1), ..., (R^q, \varphi_q)\}$, где $R^1 \in V_R$ — ресурс, с которым данное задание требует обмена данными, $\varphi_1 \in \Re^+$ — объем передачи, $l = 1, ..., q$, $q \in N$;

$JM_i = \{0$ или $R^i\}$ — маска задания, где $R^i \in V_R$ — номер ресурса, на котором возможно или желательно выполнять данное задание;

$JP_i \in \Re+$ — приоритет данного задания.

Определение 1. Отображение Г есть отображение множества заданий $V_J = \{Job_1, Job_2, ..., Job_M\}$ на множество ресурсов $V_R = \{R_1, R_2, ..., R_N\}$ графа системы $G_R = (V_R, E_R, W_{VR}, W_{ER})$, если результат отображения $\Gamma(V_J, V_R)$ есть некоторое множество A: $A = \{a_1, a_2, ..., a_n\}$, где $a_i = (R^i, J^i)$, $R^i \in V_R$, $J^i \in V_J$, $i = 1, ..., n$, $n \in N$.

Обозначим $AR = \{R^1, R^2, ..., R^n\}$, $AJ = \{J^1, J^2, ..., J^n\}$. Следовательно, $|A| = |AR| \cap |AJ|$, $AR \subseteq V_R$, $AJ \subseteq V_J$.

Определение 2. Отображение Г есть распределение заданий V_J на ресурсы V_R, если его результат $\Gamma(V_J, V_R) = A$, где $A = \{(R^1, J^1), (R^2, J^2), ..., (R^n, J^n)\}$, удовлетворяет следующему условию: для $\forall i = 1, ..., n$, $R^i \notin AR \setminus R^i$, $J^i \notin AJ \setminus J^i$, где $AR = \{R^1, R^2, ..., R^n\}$, $AJ = \{J^1, J^2, ..., J^n\}$. Размером данного распределения $\Gamma(V_J, V_R)$ является число n. Тогда $\Gamma(V_J, V_R) \to A$, $n = |A|$.

Определение 3. Результат распределения заданий на ресурсы $A = \Gamma(V_J, V_R)$ называем расписанием для данного распределения Г. Пара

$a_i = (R^i, J^i)$, $i = 1, ..., n$, называется назначением задания $J^i \in V_J$ на ресурс $R^i \in V_R$.

Определение 4. Пусть $X = \{A^1, A^2, ..., A^z\}$, $z \in N$ — множество результатов всех возможных распределений для множества заданий V_J и для множества ресурсов V_R. Тогда $\Gamma(V_J, V_R) \equiv X$. Распределение заданий на ресурсы $\Gamma(V_J, V_R) \to A^*$ есть максимальное распределение для данных множества заданий V_J и множества ресурсов V_R если:

1) $n^* = |A^*|$;

2) $n^* = \max\{|A^1|, |A^2|, ..., |A^z|\}$.

Определение 5. Пусть Δ есть некоторая функция от назначения $a_s = (R^s, J^s)$, т.е. назначения задания J^s на ресурс R^s, $R^s \in V_R$ и $J^s \in V_J$. Тогда $\Delta(a_s) = \Phi$, или $\Phi = \Delta(R^s, J^s)$ и $\Phi_i = \Delta(a_i) = \Delta(R^i, J^i)$, где $i = 1, ..., n$, назовем весом назначения $a_i = (R^i, J^i)$ по Δ.

Определение 6. Сумму весов всех назначений $\{a_1, a_2, ..., a_n\}$ назовем весом $D(A)$ расписания A:

$$D(A) = \sum_{i=1}^{n} \Delta(a_i).$$

Определение 7. Пусть $X_m = \{A_1, A_2, ..., A_m\}$, $m \in N$, есть множество всех максимальных распределений для множества заданий V_J и для множества ресурсов V_R. Тогда $A^* = \Gamma(V_J, V_R)$ — оптимальное расписание распределения (заданий V_J на ресурсы V_R) Γ по измерению Δ, если A^* удовлетворяет следующим условиям:

1) $A^* = \{(R^1, J^1), (R^2, J^2), ..., (R^n, J^n)\}$ является результатом максимального распределения для данных множества заданий V_J и множества ресурсов V_R, т.е. $|A^*| = \max\{|A^1|, |A^2|, ..., |A^z|\}$ (см. определение 5);

2) вес расписания $A^* = \{a_1, a_2, ..., a_n\}$ максимален из $X_m = \{A_1, A_2, ...$ $..., A_m\}$, т.е.

$$D(A^*) = \sum_{i=1}^{n} \Delta(a_i^*) = \max\{D(A_1), D(A_2), ..., D(A_m)\} = \max_{j=1}^{m} \{D(A_1)\}.$$

Требуется найти оптимальное (максимальное по весу заданной функции Δ) расписание $A = \{(R^1, J^1), (R^2, J^2), ..., (R^n, J^n)\}$, $n \in N$, максимального распределения Γ (по определению 7) для N_τ свободных ресурсов V_R и M, готовых к выполнению заданий V_J.

Общая схема решения. Определим модель решения задачи оптимизации и распределения, математическая постановка которой представлена в [2, 3], на основе модели оптимизации и распределения, описанной в [4, 5].

Решение данной задачи для N_τ ресурсов $V_R = \{R_1, R_2, ..., R_N\}$ и M заданий $V_J = \{J_1, J_2, ..., J_M\}$ состоит из следующих этапов:

1. Определение функции Δ весов назначения. Определяются веса $\delta_{i,j}$ ($i = 1, ..., N_\tau$, $j = 1, ..., M$) всех возможных назначений по функции Δ.

2. Поиск оптимального расписания $A = \{a_1, a_2, ..., a_n\}$, где $a_i = (R_i, J_i)$, $R_i \in V_R$, $J^i \in V_J$, $i = 1, ..., n$, $n \in N$, которое удовлетворяет условиям определения 7 и весовым значениям, определенным на первом этапе.

Определение функции измерения качества решения. При оптимизации и распределении функцию Δ для измерения веса назначения задания J_j на ресурс R_i ($R_i \in V_R$ и $J_j \in V_J$) определяем так:

$$\Delta(R_i, J_j) = \delta_{i,j} = \prod_{k=1}^{K} P_k^{i,j} \prod_{x=1}^{H} C_x^{i,j} \prod_{y=1}^{G} L_y O_y^{i,j}. \tag{1}$$

Здесь $\prod_{k=1}^{K} P_k^{i,j}$ — величина приоритета назначения (R_i, J_j), вычисляемая умножением величин всех K приоритетов $P_k^{i,j} \in \Re+$ не только заданий, но и ресурсов (в приоритете учитываются различные факторы: время ожидания заданий, работоспособность ресурсов и др.); $\prod_{x=1}^{H} C_x^{i,j}$ — результат анализа H обязательных требований, где $C_x^{i,j}$ определяет степень выполнения обязательного требования x для назначения задания J_j на ресурс R_i, $C_x^{i,j} \in \{0,1\}$, например требования наличия каналов передачи, объема требуемой памяти, наличия программ, данных и др.; $C_x^{i,j} = 1$, если ресурс R_i полностью удовлетворяет требованиям задания J_j, и $C_x^{i,j} = 0$ в противном случае; $\prod_{y=1}^{G} L_y O_y^{i,j}$ — результат анализа G оптимизирующих требований, где $O_y^{i,j} \in \Re+$ и $0 \le O_y^{i,j} \le 1$ — степень выполнения оптимизирующего требования y назначения задания J_j на ресурс R_i; $L_y \in \Re+$, $L^d \le L_y \le L^u$, — весовой коэффициент оптимизирующего требования y.

В предложенной системе представлений исходной информации $\prod_{k=1}^{K} P_k^{i,j}$ вычисляется с помощью приоритета JP_j задания J_j из выражения

$$\prod_{k=1}^{K} P_k^{i,j} = \mu_i \, \rho_j,$$

где $\rho_j = JP_j = 1/Tw_j$ (Tw_j — время ожидания задания J_j в системе); μ_i — маска задания,

$$\mu_i = \begin{cases} M^0, \text{ если } R_i = J, M_j = R^*, \\ 1, \quad \text{если } R_i = J, M_j \in \{0, R^*\}; \end{cases}$$

$\prod_{x=1}^{H} C_x^{i,j}$ вычисляется с помощью сравнения требований к коммуникационным затратам для задания $JL_j = \{(R^1, \varphi_1), ..., (R^q, \varphi_q)\}$ с множеством дуг графа системы ресурсов $E_R = \{E_1, E_2, ..., E_d\}$: при $\forall\, i = 1, ..., q$ $CC_l^{i,j} = 1$, если $(R_i, R^l) \in E_R$; $CC_l^{i,j} = 0$, если $(R_i, R^l) \in E_R$, т.е.

$$\prod_{x=1}^{H} C_x^{i,j} = C^{i,j} = \prod_{l=1}^{q} CC_l^{i,j};$$

$\sum_{y=1}^{G} L_y O_y^{i,j}$ вычисляется как сумма обратных величин времени выполнения $T_{e_{i,j}}$ и времени, затрачиваемого на коммуникации $T_{c_{i,j}}$.

Коэффициент производительности ресурса $RE_i = k_i$ определяется из WVR_i, объем работы задания $JE_j = \varepsilon_j$ — из матрицы весов дуг графа системы ресурсов $RC[k, l] = \beta_{k,l}$, где $k = 1, ..., N$ и $l = 1, ..., N$ — объемы требований заданий по коммутациям из $JL_i = \{(R^1, \varphi_1), ..., (R^q, \varphi_q)\}$. Тогда $T_{e_{i,j}} = \varepsilon_j\, k_i$; $T_{c_{i,j}} = \sum_{l=1}^{q} (\varphi_l\, \beta_{i,l})$. Следовательно,

$$\sum_{y=1}^{G} L_y O_y^{i,j} = 1/T_{e_{i,j}} + 1/T_{c_{i,j}} = 1/(\varepsilon_j\, k_i) + 1/\sum_{l=1}^{q} (\varphi_l\, \beta_{i,l}).$$

Из (1) находим

$$\Delta(R_i, J_j) = \delta_{i,j} = (\mu_i\, \rho_j)\, C^{i,j} (1/\varepsilon_j\, k_i) + 1/\sum_{l=1}^{q} (\varphi_l\, \beta_{i,l})).$$

Очевидно, что $\delta_{i,j} \geq 0$ для $\forall i = 1, ..., N_\tau$, $j = 1, ..., M_\tau$. Поэтому $\inf(\Delta(R_i, J_j)) = 0$. В случае отсутствия связи между ресурсами i и l значение $\beta_{i,l}$, получаемое для RC [i, l], обусловливает неравенство

$$T_{c_{i,j}} = \sum_{l=p}^{q} (\varphi_l\, \beta_{i,l}) > \lambda_0,$$

Задания

№	1	2	3	4	5	6
1	3,5	0,2	7,1	0,6	6,1	0,1
2	0	4,1	2,1	5,3	8,7	3,3
3	2,1	5,9	0,3	0	7,2	0,7
4	4,5	3,6	5,1	0,8	3,1	0
5	8,1	7,1	0	0	4,9	0,4
6	0,9	0	2,1	6,8	0	8,1

Ресурсы

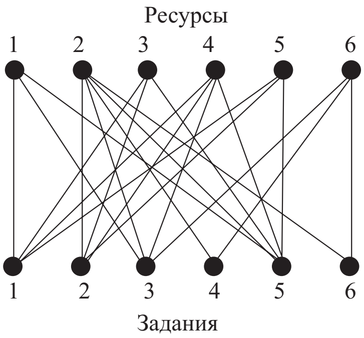

Рис. 1. Форма представления исходной информации для шести ресурсов и шести заданий

где λ_0 — некоторое заданное число, являющееся порогом для определения существования связи между двумя ресурсами. Время выполнения $T_{e_{i,j}}$ имеет некоторую нижнюю границу T^0. Верхняя граница диапазона изменения $\delta_{i,j}$ определяется из выражения $\sup(\Delta(R_i, J_j) = (\mu_{0j}\rho_{0j})[1/T^0 + +1/\lambda_0] = \delta_{\max}$. Найденные значения $\delta_{i,j}$, $i = 1,..., N_\tau$, $j = 1,..., M_\tau$, хранятся в матрице $JR[1,..., N_\tau, 1,..., M_\tau]$.

Определение оптимального распределения. Множество N_τ ресурсов $V_R = \{R_1, R_2, ..., R_{N_\tau}\}$ и M заданий $V_J = \{J_1, J_2, ..., J_M\}$ можно представить как множество вершин некоторого графа G. Тогда множество неориентированных дуг $E = \{E_1, E_2, ..., E_d\}$ между вершинами графа G соответствует множеству возможных назначений заданий J^* на ресурсы R^*. Исходная информация при такой постановке представляется в виде матрицы связности (МС) или двудольного графа (рис. 1). Дуга $E_k = \{R_i, J_j\}$, где $R_i \in V_R$ и $J_j \in V_J$, $k = 1,..., d$, $0 \le d \le N_\tau M$, между вершинами J_j и R_i отсутствует только тогда, когда назначение задания J_j на ресурсе R_i невозможно, т.е. когда $\delta_{i,j} \le \delta_0$, где δ_0 — некоторое заданное число (в данном примере $\delta_0 = 1$).

Второй этап распределения представляет собой задачу назначения. Существует несколько методов решения задачи назначения для взвешенного двудольного графа $G = (V_R, V_J, E, WE)$. Здесь $V_R = \{R_1, R_2, ..., R_{N_\tau}\}$; $V_J = \{J_1, J_2, ..., J_M\}$; $E = \{E_1, E_2, ..., E_d\}$; $E_k = \{R^*, J^*\}$, где $R^* \in V_R$ и $J^* \in V_J$, $k = 1,..., d$, $0 \le d \le N_\tau M$; $WE = \{WE_1, WE_2, ..., WE_d\}$, $WE_k = \Delta(E_k)$, где $k = 1,..., d$, $0 \le d \le N_\tau M$.

Решение задачи назначения для графа размером $N_\tau \times M$, где $N_\tau \ne M$, приводится к решению задачи назначения для графа размером $N \times N$, где $N = \max\{N_\tau; M\}$. В случае планирования для однородной GRID решение задачи назначения для взвешенного графа G приводится к решению задачи назначения для невзвешенного графа G', полученного из графа G снятием весов всех дуг. Задача назначения в такой постановке решена во многих работах [4, 5, 8, 10, 11]. На выбор метода и алгоритма решения влияет временная сложность, так как время решения задач планирования, особенно при динамическом планировании, является основным критерием.

Рассмотрим два наиболее часто используемых подхода к решению данной задачи: поиск максимального потока в сети [6, 7] и поиск максимального паросочетания методом увеличивающего чередующегося пути [7, 9].

При динамическом планировании в GRID системах задача планирования сводится к поиску максимального паросочетания во взвешенном двудольном графе. При этом основным этапом является поиск максимального паросочетания в невзвешенном двудольном графе. Поэтому задача поиска максимального паросочетания в невзвешенном двудольном графе — ключевая и требует специального изучения, так как наиболее известные алгоритмы имеют большую временную сложность, что ограничивает их практическое использование.

Основная идея рассматриваемого подхода заключается в разделении процесса распределения или составления расписания на предварительный анализ исходной информации, т.е. определение стратегии поиска решения и поиска варианта решения с использованием результатов анализа. Этап предварительного анализа исходной информации существенно уменьшает общее время решения по сравнению с классическими методами.

В общем виде требования заявок на захват ресурсов вычислительной системы можно разделить на обязательные C_x, $x=1,\ldots,p$, и оптимизирующие O_y, $y=1,\ldots,k$. С помощью обязательных $\forall\, C_x^{i,j} \in \{0,1\}$ требований анализируется принципиальная возможность предоставления i-й заявки j-го ресурса. Оптимизирующие требования $\forall\, O_y^{i,j} \in [0,1]$ определяют приоритет j-го ресурса для назначения i-й заявки по $O_y^{i,j}$ требованиям. Определение приоритета j-го ресурса для назначения i-й заявки по всем требованиям выполняем по формуле

$$Q_{i,j} = \prod_{x=1}^{p} C_x^{i,j} \prod_{y=1}^{k} R_y O_y^{i,j}, \tag{2}$$

где $C_x^{i,j}$ — степень выполнения x-го обязательного требования для назначения i-й заявки на j-й ресурс; $O_y^{i,j}$ — степень выполнения оптимизирующего требования y для назначения i-й заявки на j-й ресурс; R_y — весовой коэффициент оптимизирующего требования y.

Если система планирования учитывает обязательные и оптимизирующие требования, то значение коэффициента приоритета в матрице связности вычисляется из выражения (2) и находится в диапазоне $Q_{i,\,j} \in [0,\ 1]$, а при диспетчеризации, выполняемой с учетом только обязательных требований,

$$Q_{i,\,j} = \prod_{x=1}^{p} C_x^{i,\,j},\ Q_{i,\,j} \in \{0,1\}.$$

В этом случае МС, отображающая приоритет заявок на ресурсы, примет булевый вид. Единица означает, что ресурс принципиально подходит для размещения заявки, т.е. имеется достаточный объем памяти, процессор имеет необходимые характеристики, вычислительный узел имеет необходимые программы и данные.

Для системы из N ресурсов в какой-то момент времени имеется M работ ($M = N$). Требования работ на захват ресурсов представлены булевой матрицей связности $MC\ [i,\ j]$, $i = 1,..., N$, $j = 1, ..., N$. Необходимо определить отношение работа—ресурс $A = \{(V_i, W_j)\}$, $V_i \in V = \{1, 2, ..., N\}$, $W_j \in W = \{1, 2, ..., N\}$ так, чтобы $\forall V_i \notin V \backslash V_i\ \ \forall\ W_j \notin W \backslash W_j$: $\{MC\ [V_i, W_j] = 1\ V_i \neq W_j\}$.

Введем дополнительные условия:

- ресурс может обслужить только одну заявку;
- процесс обслуживания заявки не может быть прерван;
- каждая работа имеет индивидуальные характеристики и может претендовать на захват только некоторых системных ресурсов;
- нет очереди к ресурсам (отсутствие очередей определяется тем, что на данном уровне планирования планировщик более высокого уровня из общего потока заявок выбрал число заявок, соответствующее числу свободных ресурсов);
- одна работа может быть обслужена только одним ресурсом. Единица связности МС соответствует паре Работа (i) и Ресурс (j) и соответствует выполнению всех K обязательных требований, предъявляемых к системе обработки соответствующей заявки на этом ресурсе. Нуль свидетельствует о невозможности обслуживания.

При такой постановке решение задачи распределения заявок по ресурсам сводится к задаче поиска максимального паросочетания в невзвешеном двудольном графе.

Элементы теории ускорения распределения работ по ресурсам в неоднородных GRID системах. В основу наиболее известных алгоритмов поиска [2—4] максимального паросочетания в произвольном графе положены два подхода: сведение задачи к поиску максимального потока в сети [6] и поиск увеличивающего пути от свободных вершин [8]. В основу поиска увеличивающего чередующегося пути положена схема Диница [7] и разработанный на ее основе алгоритм Хопкрофта—Карпа [9].

ISSN 0204–3572. Electronic Modeling. 2011. V. 33. № 5

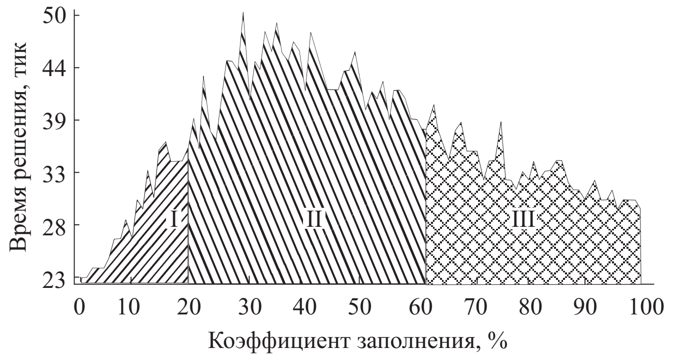

Рис. 2. График зависимости времени решения задачи поиска максимального паросочетания от коэффициента заполнения МС при $N = 10$ (1тик соответствует нескольким тактам работы процессора)

Наилучшие известные алгоритмы, реализующие этот подход и алгоритм Хопкрофта—Карпа, или его модификации, имеют полиномиальную теоретическую временную сложность. Однако эти алгоритмы и программы, их реализующие, имеют сложную структуру или предназначены для частных случаев [2, 3, 5, 10] и не обеспечивают приемлемых временных показателей, что существенно ограничивает применение их в системах оперативного диспетчирования в GRID системах. Для уменьшения временной сложности алгоритмов поиска максимального паросочетания в некоторых случаях предлагаются распараллеленные алгоритмы.

Для алгоритмов, в которых используется поиск увеличивающего чередующегося пути, правильность выбора начального варианта паросочетания в значительной степени влияет на число шагов при поиске увеличивающего пути. В работах [1—3] подчеркивается, что уникальные свойства двудольного графа позволяют уменьшить временную сложность алгоритмов. Для выявления особенностей двудольного графа, влияющих на временную сложность, выполнено статистическое исследование программной модели базового алгоритма Хопкрофта—Карпа с временной сложностью $O(N^{2,5})$.

Результаты моделирования для графов размерности 10 приведены на рис. 2, на котором выделены три зоны. В результате исследований установлена зависимость размеров этих зон от размерности решения задачи. Значительные различия временных затрат в выделенных зонах обусловили необходимость исследования в них свойств двудольного графа.

В большинстве алгоритмов время решения задачи зависит от правильности выбора исходного (базового) варианта решения. Наибольшие трудности при поиске решения в выделенных зонах, влияющие на число шагов, а следовательно, и на время поиска максимального паросочетания, возникают в двудольных невзвешенных графах, в которых перманент МС близок к единице или равен нулю. Эти трудности вызваны тем, что поиск максимального паросочетания основан на центральной теореме Кенига—Холла о существовании паросочетания [6, 7] и теореме Бержа [8], согласно которой паросочетание M в графе G максимально тогда и только тогда, когда в G не существует увеличивающего пути относительно M. Поэтому все известные алгоритмы предусматривают выполнение поиска увеличивающего пути от свободных вершин после генерирования базового варианта даже в том случае, если этого пути не существует, что значительно увеличивает время поиска.

Элементы технологии предварительного анализа структуры двудольного графа. Для упрощения решения задачи поиска максимального паросочетания целесообразно разделить его на несколько этапов, т.е. собственно решению предшествует быстрый анализ исходной информации и выработка рекомендаций для ее дальнейшего использования. Добавление дополнительных шагов, имеющих значительно меньшую временную сложность, чем сам алгоритм, не влияет на теоретическую оценку временной сложности алгоритма в целом, однако позволяет уменьшить размерность решения задачи посредством выделения назначений, которые обязательно нужно сделать, и назначений, которые делать нельзя. При этом можно избежать лишней проверки на возможность включения их в решение.

Кроме того, на этапе подготовки исходной информации возможно вычисление мощности максимального паросочетания. Зная численное значение мощности, можно избежать поиска увеличивающего пути от свободных вершин при достижении расчетной мощности паросочетания на очередном шаге поиска решения.

Для решения задачи назначения требуется определить условия ее решения, т.е. возможность полного распределения всех заявок по ресурсам. Необходимые условия существования полного решения имеют следующий вид:

$$\sum_{j=1}^{N} MC\,[i,\ j] \neq 0,\ i = 1,\ ...,\ M\ , \tag{3}$$

$$\sum_{i=1}^{M} MC\,[i,\ j] \neq 0, j = 1,\ ...,\ N\ . \tag{4}$$

Следует заметить, что выполнение условий (3), (4) является необходимым, но недостаточным для получения полного варианта размещения.

Введем следующие определения.

Определение 8. Задан невзвешенный двудольный граф $G = (V, E)$, где $V=\{V_R, V_J\}$, $V_R = \{R_1, R_2, ..., R_N\}$, $V_J = \{J_1, J_2, ..., J_M\}$ — вершины графа; $E = \{E_1, E_2, ..., E_d\}$ — дуги графа; $E_k = \{R^*, J^*\}$, где $R^* \in V_R$ и $J^* \in V_J$, $k = 1, ..., d$, $0 \le d \le N^2$. Булева матрица RJ [1, ..., N, 1,..., N] называется матрицей связности графа G, если для $\forall\ i = 1, ..., N$, $j = 1, ..., N$

$$RJ\,[i, j] = \begin{cases} 1, \text{ если } (R_i, Job_j) \in E, \\ 0, \text{ если } (R_i, Job_j) \notin E. \end{cases}$$

Определение 9. Подмножество $A = \{a_1, a_2, ..., a_n\}$, где $a_k = (R^k, J^k)$, $R^k \in V_R$, $J^k \in V_J$, $k = 1, ..., n$, $n \in N$, называется паросочетанием графа G или его матрицы RJ, если $A = \{(R^1, J^1), (R^2, J^2), ..., (R^n, J^n)\}$ удовлетворяет следующим условиям: для $\forall\ k = 1, ..., n$, $R^k \notin AR \setminus R^k$, $J^k \notin AJ \setminus J^k$, где $AR =$ $= \{R^1, R^2, ..., R^n\}$, $AJ = \{J^1, J^2, ..., J^n\}$.

Следовательно, $AR \subseteq V_R$, $AJ \subseteq V_J$, или подмножество A ребер графа $G = (V, E)$ называется паросочетанием, если никакие два ребра из A не имеют общей вершины.

Определение 10. Пара $a_k = (R^k, J^k)$, $k = 1, ..., n$, называется назначением для ресурса $R^k \in V_R$ и для задания $J^k \in V_J$.

Определение 11. Пусть $X = \{A^1, A^2, ..., A^z\}$, $z \in N$, есть множество результатов всех возможных паросочетаний для графа $G = (V, E)$ или его матрицы RJ. Паросочетание A^* называется максимальным паросочетанием или решением для данного графа G, если $|A^*| = n^*$ и $n^* = \max\{|A^1|, |A^2|, ..., |A^z|\}$.

Теоретическое обоснование выделения обязательных назначений. Анализ свойств невзвешенного двудольного графа при решении задачи поиска максимального паросочетания, а также анализ известных алгоритмов позволяют сформулировать следующую теорему.

Теорема 1. Если в матрице $RJ\,[i, j]$, $i = 1, ..., N$, $j = 1, ..., N$, графа $G = (V_R$, $V_J, E)$, где $V_R = \{1, 2, ..., N\}$, $V_J = \{1, 2, ..., N\}$, существует решение A мощностью $n = N$ и существуют такие вершины (p, q), что $RJ\,[p, q] = 1$, $RJ\,[p, j] = 0\ \forall\ j \in \{1, ..., N\} \setminus q$ и (или) $RJ\,[p, q] = 0\ \forall\ i \in \{1, ..., N\} \setminus p$, то пара (p, q) всегда участвует в решении A, $(p, q) \in A$.

Доказательство. Из условия теоремы 1 и определений 9, 11 находим решение для графа $2N$ вершин $A = \{a_1, a_2, ..., a_N\}$, где $a_k = (R^k, J^k)$, $R^k \in V_R$, $J^k \in V_J$, $k = 1, ..., N$. Если выполняется условие (3), то проверяем назначение $a = (R, J)$ для $R = p$ в этом решении. Пусть $J = x$. Тогда $RJ\,[p, x] = 1$ и очевидно, что $x \in \{1, ..., N\}$. Согласно (3) $x = q$, тогда $(p, q) \in A$. Аналогично, если выполняется условие (4), то проверяем назначение $a^* = (R^*, J^*)$ для

$J^* = q$ в этом решении. Получаем $RJ\,[x,q]=1,\ x=p,\ (p,q)\in A.$ Отсюда следует, что пара (p, q) всегда принадлежит A.

Определение 8. Назначения (p, q) согласно теореме 1 называются обязательными.

Теорема 2. Любая из вершин невзвешенного двудольного графа, имеющая степень, равную единице, всегда участвует в одном из вариантов максимального паросочетания.

Теорема 2 справедлива как для вершин-заявок, так и для вершин-ресурсов. В случае, если вершины, имеющие степень 1, образуют веер, то Теорема 2 справедлива для любой вершины, входящей в веер, и каждая из них может быть включена в паросочетание, а проверку остальных вершин на возможность получения увеличивающего пути выполнять не следует.

Определение 9. Для матрицы RJ [1, ..., N, 1,..., N] формируем два вектора: KR [1, ..., N] и KJ [1, ..., N]. При i = 1, ..., N

$$KR\,[i]=\sum_{k=1}^{N} RJ\,[i,k],$$

при j = 1, ..., N

$$KJ\,[j]=\sum_{k=1}^{N} RJ\,[k,j];$$

$KR\,[i]$ — степень вершины $R_i \in V_R=\{R_1, R_2, ..., R_N\}$; $KJ\,[j]$ — степень вершины $J_j \in V_J=\{J_1, J_2, ..., J_N\}$.

Следствие 1. Если существует решение A^*, то задачу назначения можно разделить на две части: в первой части участвуют только обязательные назначения (p, q), во второй — оставшиеся назначения из новой МС*, которая может быть получена после удаления строк и столбцов соответствующих обязательным назначениям, определенным по теореме 1.

Доказательство. Если существует решение $A=\{(R^1, J^1), (R^2, J^2), ...$ $..., (R^n, J^n)\}$ от $n=N$ пар, то $A=\{(R^1, J^1), B\}$, где $B=\{(R^2, J^2), ..., (R^n, J^n)\}$ от $N-1$ пар вершин для двудольного графа $G=(V, E)$ от $2N$ вершин. Тогда B может быть найдено в подграфе $G'=(V', E')$ от $2N-2$ вершин после удаления (p,q) и всех дуг, инцидентных этим (p,q) вершинам. Удаление (p,q) вершин в графе отображается в RJ удалением соответствующих строк и столбцов.

Теорема 3. Если в матрице $RJ\,[i, j],\ i=1,...,N,\ j=1,...,N,\ \exists\, FA$ присутствует веер:

$$FA=\{(R^1, q), (R^2, q), ..., (R^f, q)\},\ \ R^k\in\{1, ..., N\},\ \ 2\le f\le N, \tag{5}$$

при $RJ\,[R^k,\ q]=1$ для $\forall\ k=1,...,f$ и $RJ\,[R^k,\ J^k]=0$ для $\forall\ J^k=\{1,...,N\}\setminus q$,
или

$$FA=\{(p,J^1),(p,J^2),...,(p,J^f)\},\ J^k\in\{1,...,N\},\ 2\le f\le N, \qquad (6)$$

при $RJ\,[p,\ J^k]=1$ для $\forall\ k=1,...,f$ и $RJ\,[R^k,\ J^k]=0$ для $\forall\ R^k\in\{1,...,N\}\setminus p$, то любая из вершин FA входит в один из вариантов максимального паросочетания. Задача назначения не имеет полного решения и мощность максимального паросочетания определяется из выражения $M < N-f+1$.

Д о к а з а т е л ь с т в о. Предположим, что имеется одно полное решение $A=\{(R^1,\ J^1),(R^2,\ J^2),\ ...,(R^n,\ J^n)\}$. Если выполняется условие (5), то для вершин $\{R^1, R^2, ..., R^f\}$ имеются f соответствующих пар-назначений $\{(R^1,\ J^1),\ (R^2,\ J^2),\ ...,\ (R^f,\ J^f)\}$ таких, что $RJ\,[R^k,\ J^k]=1,\ k=1,...,f$. Но из (5) получаем $RJ\,[R^k,\ J^k]=0,\ \forall\ J^k\in\{1,...,N\}\setminus q$, а так как $f\ge 2$, то возникает противоречие. Поэтому предположение о существовании полного решения A неверно.

При выполнении (6) приведенные рассуждения также справедливы. Это доказывает, что полного решения не существует. Следует заметить, что возможно существование нескольких вееров одновременно, и эта теорема справедлива для каждого из них.

Следствие 2. Размерность решения задачи поиска максимального паросочетания можно уменьшить, уменьшив число пар вершин, определенных по теоремам 1 и 2. Тогда поиск паросочетания следует вести из нового суграфа.

Следствие 3. Размерность решения задачи поиска максимального паросочетания необходимо уменьшить на число вершин, входящих в веер, а поиск паросочетания вести из нового суграфа.

Следствие 4. Смежные ребра, инцидентные вершинам, определенным по теореме 1, следует удалить из дальнейшего рассмотрения, а исходный граф редуцировать и преобразовать в новый суграф.

Следствие 5. Смежные ребра, инцидентные вершинам, определенным по теореме 3, следует удалить из дальнейшего рассмотрения, а исходный граф редуцировать и преобразовать в новый суграф.

Временная сложность алгоритма поиска максимального паросочетания на основе предложенного подхода для двудольных графов со степенью заполнения матрицы смежности, равной не более 30 %, для взвешенного двудольного графа составляет $O\,(n^{1,5}\log n)$.

Выводы

Анализ наиболее известных подходов к решению задач назначения свидетельствует о том, что они не могут быть приняты в качестве базовых для динамического планировщика. Например, для метода отжига временные параметры процесса поиска решения не удовлетворяют ограничениям по времени планирования, а для решения задачи с помощью оценочных функций при приемлемом времени планирования качество планирования не соответствует желаемому. Наиболее приемлемым является метод направленного поиска, позволяющий осуществить поиск решения наилучшим способом в зависимости от исходной информации.

Венгерский алгоритм поиска максимального паросочетания во взвешенном двудольном графе удовлетворяет предъявляемым требованиям, так как обеспечивает направленный поиск решения и отношение работа—ресурс в предлагаемой модели системы пространственного планирования. Классический Венгерский алгоритм в каждой итерации выполняет поиск максимального паросочетания для невзвешенного двудольного графа с использованием алгоритма Карпа—Хопкрофта и имеет временную сложность $O(n^3)$, которая определяется алгоритмом Карпа—Хопкрофта с временной сложностью $O(n^{2,5})$. Следует заметить, что в неоднородной GRID системе после преобразования исходной информации степень заполнения матрицы связности не превышает 25 % и поиск максимального паросочетания выполняется в зоне I (см. рис. 2).

Замена алгоритма Карпа—Хопкрофта другим алгоритмом с меньшей временной сложностью позволит снизить общую временную сложность Венгерского алгоритма. Элементы теории построения адаптивного поиска максимального паросочетания являются основой этого решения, так как поиск максимального паросочетания для невзвешенного двудольного графа выполняется при степени заполнения МС не более чем на 30 % (для неоднородной ВС с коэффициентом неоднородности не менее 0,7), что позволяет применить предложенные алгоритмы выполнения обязательных назначений и принцип исключающего планирования.

Использование принципа исключающего планирования способствует значительному повышению эффективности системы динамического планирования с сохранением качества решения в заданных пределах.

Предложенные процедуры подготовки исходных данных для планирования в реальном времени позволяют использовать модифицированный Венгерский алгоритм в качестве базового для пространственного планировщика нижнего уровня систем реального времени.

This paper presents a general mathematical model of dynamic scheduling in distributed heterogeneous GRID system. It is shown that searching of computational resource for a task can be solved as a maximum matching problem for a bipartite graph.

1. *Метод* опережающего планирования для грид / В. Н. Коваленко, Е. И. Коваленко, Д. А. Корягин, Э. З. Любимский // Препринт ИПМ.2005. — № 112. — http://www.keldysh.ru/papers/2005/prep112/prep2005_112.html.
2. *5. Platform* LSF 7 Update 6. An Overview of New Features for Platform LSF Administrators // Официальный сайт компании Platform Computing Corporation — 2009. — http://www.platform.com/workload-management/whatsnew_lsf7u6.pdf.
3. *Microsoft* Windows Compute Cluster Server 2003 [Электронный ресурс] // Руководство пользователя. — 2006. — http://msdb.ru/Downloads/ WindowsServer2003/CCS/CCS2003 Guide_Rus.pdf.
4. *TORQUE* Resource Manager Guide // Официальный сайт компании Cluster Resources Inc. — 2009. — http://www.clusterresources.com/products/torque-resource-manager.php.
5. *PBS* Works // Официальный сайт компании Altair Engineering, Inc. — 2006. — http://www.pbsworks.com/.
6. *Commercial-grade* HPC workload and resource management // Официальный сайт компании Altair Engineering, Inc. —2008. — http:/ ww.pbsgridworks. com/Product.aspx? id=1.
7. *What* is Condor? // Официальный сайт продукта Condor. — 2006. — http://www.cs.wisc.edu/condor/description.html.
8. *Sun* Grid Engine 6.2 Update 5 //Официальный сайт компании Oracle Corp. —2009. — http://www.sun.com/software/sge/index.xml.
9. *IBM* Tivoli Workload Scheduler LoadLeveler // Официальный сайт компании «Интерфейс» —2007. — http://www.interface.ru/home.asp?artId=6283. Maui Scheduler Administrator's Guide // Официальный сайт компании Cluster ResourcesInc. — 2008. — http://www. clusterresources. com/products /maui/ docs/index.shtml. — Загл. с экрана.
10. *Moab* Workload Manager // Официальный сайт компании Cluster Resources Inc. — 2008. — http://www.clusterresources.com/products/moab-cluster-suite/workload-manager. php.
11. *Симоненко А. В.* Выбор стратегии пространственного планирования в параллельных вычислительных системах //Вiсник НТУУ «КПI». Iнформатика, управлiння та обчислювальна техника. — 2001. — № 35. — С. 104—108.
12. *Kaufmann A.* Introduction a la combinatorique en vue des applications. — Paris : Dunod, 1968.
13. *Papadimitry X., Stayglitsh K.* Combinatory optimization, algorithm and complexity. — Moscow: Mir , 1985.
14. *Berge C.* Theorie des graphes et ses application. — Paris : Dunod, 1958.

Поступила 21.02.11

СИМОНЕНКО Валерий Павлович, д-р техн. наук, профессор Национального технического университета Украины «Киевский политехнический ин-т», который окончил в 1965 г. Область научных исследований — организация вычислительных процессов в вычислительных системах.