УДК 004.383
DOI: 10.20535/SRIT.2308-8893.2016.3.04

# ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РАБОТ ПО РЕСУРСАМ В НЕОДНОРОДНОЙ СИСТЕМЕ С ОГРАНИЧЕНИЯМИ РЕАЛЬНОГО ВРЕМЕНИ

**В.П. СИМОНЕНКО, А.М. СЕРГИЕНКО**

**Аннотация.** Предложен метод динамического распределения работ в неоднородной вычислительной системе в реальном времени. Основой метода является предварительная подготовка исходной информации с учетом ограничений на продолжительность планирования, сложности выполняемых работ, а также индивидуальных характеристик ресурсов, таких как производительность, емкость памяти, наличие загруженных исходных данных и математического обеспечения. Алгоритм такой подготовки состоит в формировании матрицы запасов времени выполнения работ на ресурсах и в последовательности преобразований этой матрицы в матрицу стоимостей с применением матрицы проверки конфликтности назначений. После подготовки информации задача планирования решается венгерским алгоритмом поиска максимального паросочетания в графе.

**Ключевые слова:** расписание, двудольный граф, венгерский алгоритм, планировщик.

## ВВЕДЕНИЕ

Планирование вычислений состоит в определении плана решения совокупности работ с определенным временем исполнения и ограничениями по времени их выхода из вычислительной системы (ВС). Система планирования должна обеспечить выполнение требования минимального суммарного временного отклонения моментов выхода работ из ВС от заданных сроков их выполнения при соблюдении порядка следования. Планирование в реальном времени, т.е. динамическое планирование, означает, что такая задача должна быть решена за интервал времени, который оказывается меньшим чем некоторое значение, определяемое допустимым временем реакции ВС на поток заявок на работы. Поэтому такое планирование является довольно трудной задачей. В общем случае задача планирования, даже с одним вычислительным узлом, относится к классу NP-полных [1, 2].

В большинстве случаев разработчики систем планирования в реальном времени используют статические алгоритмы и заранее определяют максимальный список заявок на работы, допустив наихудший случай для получения статической таблицы, управляющей распределением работ, т.е. плана. Этот план фиксируется и используется для безусловного исполнения в динамическом режиме со следующими допущениями:

— все временные ограничения остаются неизменными на время выполнения плана;

— все работы вкладываются в свое критическое время исполнения и выхода работ из системы.

В других случаях при помощи приемов статического планирования создается статический список приоритетов для использования в динамическом режиме во время диспетчеризации самих работ.

Если система реального времени работает только в динамическом режиме, то использование допущений статического планирования, когда все известно априори, недопустимо. В этом случае выбирается один из возможных алгоритмов составления расписания и тщательно анализируется на применимость его в динамическом режиме в конкретной вычислительной системе. Как правило, применяются алгоритмы, использующие планирование по спискам, и приоритетное обслуживание [3–6].

В работе предлагается динамическое планирование выполнять с использованием математического аппарата поиска максимального паросочетания. Для этого разработан метод преобразования исходной информации о множестве работ в исходные данные для составления плана с помощью этого аппарата. Причем распределение заявок на работы по ресурсам реализуется для неоднородной параллельной вычислительной системы (НПВС) с учетом ограничений реального времени и выполняется планировщиком нижнего уровня. В качестве этого аппарата целесообразно использовать венгерский алгоритм, а также его модификацию, описанную авторами в работе [7], которая имеет меньшую временную сложность.

## ПОСТАНОВКА ЗАДАЧИ

Рассмотрим общую постановку задачи планирования с ограничениями реального времени. На вход НПВС поступает множество работ. Из него выбирается порция $n$ работ, которые следует распределить на $n$ имеющихся ресурсов. Каждая работа характеризуется тремя временными параметрами:

$T_{\text{in}}^i$ — момент поступления $i$-й входной работы в ВС;

$T_{\text{out}}^i$ — момент выхода $i$-й работы из ВС;

$T_w^i$ — длительность выполнения $i$-й работы на вычислительном узле НПВС, имеющем максимальную производительность, $i = 1,\ldots,n$, которая равна оценке сложности этой работы.

Распределенная ВС, такая как НПВС, имеет множество ресурсов; $j$-й характеризуется производительностью $R^j$. Для $i$-й работы можно определить время решения ее на $j$-м ресурсе в относительных единицах. Для этого определим $R_{\max} = \max\{R^1,\ldots,R^n\}$.

Определим относительную производительность $j$-го вычислительного узла как $Z^j = R^j / R_{\max}$, $j = 1,\ldots,n$.

Имея значения относительной производительности узлов, можно определить отношение работа–ресурс с учетом времени выполнения работы и производительности каждого узла НПВС. Для этого сформируем матрицу связности $C$, в которой элемент $C[i,j] = T_w^i / Z^j$ определяет относительное время выполнения $i$-й работы на $j$-м ресурсе с учетом его производительности. Причем значение $C[i,j]$ определяет относительное время выполнения

*i*-й работы на *j*-м ресурсе ввиду того, что $0 < Z^j \le 1$. Элемент $C[i, j]$ также учитывает наличие загруженных исходных данных и математического обеспечения. Например, при их отсутствии $C[i, j] = 0$.

В системах с несколькими обслуживающими ресурсами часто используется стратегия минимизации максимальной из продолжительностей использования ресурсов, что обычно связано с повышением быстродействия такой системы при функционировании в интерактивном режиме. В работе используется критерий минимума времени выполнения всех работ, поскольку предполагается, что все заявки на работы являются независимыми и требуется, чтобы не достигалось максимально допустимое время реакции $T^i_{\text{out}}$ на *i*-ю заявку. Обычно $T^i_{\text{out}}$ указывается в спецификации заявки. При этом для выполнения каждой работы обеспечивается требуемое быстродействие системы.

В связи с тем, что работа, выполняемая с ограничениями в реальном времени, должна завершиться до заданного времени реакции $T^i_{\text{out}}$, то необходимо учитывать временной запас загрузки ресурса:

$$\Delta^{i,j}_t = T^i_{\text{out}} - T^i_{i\text{in}} - C[i, j].$$

Запас $\Delta^{i,j}_t$ может принимать как положительные, так и отрицательные значения в зависимости от сложности $T^i_w$ работы и относительной производительности $Z^j$ *j*-го ресурса. Положительные значения $\Delta^{i,j}_t$ означают, что работа *i*, назначенная на ресурс *j*, будет завершена на промежуток времени $\Delta^{i,j}_t$ до назначенного срока, а отрицательные — что выход заявки из системы произойдет позже назначенного срока с задержкой $\Delta^{i,j}_t$. Примерная матрица $\Delta_t$ элементов $\Delta^{i,j}_t$ для некоторых наборов работ и ресурсов показана на рис. 1.

Ресурсы

| Работы | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 3 | –2 | –3 | 4 | 5 |
| 2 | –2 | 0 | –5 | –6 | –4 | –2 |
| 3 | 3 | 6 | 1 | 7 | 9 | 1 |
| 4 | 1 | –3 | –4 | –5 | –1 | –1 |
| 5 | –1 | 4 | –4 | 1 | –5 | –1 |
| 6 | 8 | 1 | 2 | 4 | 3 | 2 |

*Рис. 1.* Исходная матрица $\Delta_t$

При вычислении $\Delta^{i,j}_t$ следует учитывать, что на момент планирования $T_P$ все заявки имеют одно и то же базовое время начала планирования неза-

висимо от времени прихода заявок в систему, т.е. считается, что заявки хранятся в буфере. Кроме этого, следует учитывать время работы планировщика, т.е. длительность планирования $T_\Sigma$. Поэтому можно принять $T_{\text{in}}^i = T_P + T_\Sigma$.

В соответствии с требованиями режима реального времени значение $\sum_{i=1}^{n} \Delta_t^{i,j}$ должно быть больше или равно 0. При невозможности получения варианта распределения с выполнением требования по времени выхода для всех заявок и при включении в распределение значений с $\Delta_t^{i,j} < 0$ их сумма должна быть минимальной.

## ПРИВЕДЕНИЕ УСЛОВИЙ ПЛАНИРОВАНИЯ К ЗАДАЧЕ ПОИСКА МАКСИМАЛЬНОГО ПАРОСОЧЕТАНИЯ

В указанной выше постановке решение задачи назначения можно свести к задаче поиска максимального паросочетания во взвешенном двудольном графе, которая решается венгерским алгоритмом или алгоритмом направленного поиска. Следует отметить, что поиск варианта распределения выполняется для НПВС, для которой решение ищется в разреженной бинарной матрице связности. Эта матрица формируется в соответствии с венгерским алгоритмом и с ее помощью ищется максимальное паросочетание для невзвешенного двудольного графа. В этом случае можно применить адаптивный алгоритм, временная сложность которого зависит от коэффициента заполнения матрицы связности двудольного графа, а при коэффициенте заполнения менее 0,7 временная сложность не превышает $O(n^{1,5} \log n)$ [7].

Для поставленной задачи требуется модификация формирования исходных данных и формирование базовой (начальной) области поиска. Рассмотрим процедуры формирования исходных данных и области поиска на примере решения задачи назначения для шести работ на шести процессорных узлах. После обработки временных ограничений и учета производительности каждого узла получим исходную матрицу $\Delta_t$ (рис. 1).

Информация в матрице $\Delta_t$ указывает только объем работы и относительное время ее выполнения с учетом производительности вычислительных узлов. Однако из-за неоднородности НПВС приходится учитывать индивидуальные характеристики каждого вычислительного узла, связанные с возможностью выполнения работы. Это наличие оперативной памяти достаточного объема, исходных данных в узле, программ, соответствующих заявке на работу и т.п. Для правильного планирования требуется оценка принципиальной возможности выполнения работ в каждом узле, которая учитывает эти критерии. Для этого необходимо сформировать матрицу $Q$ проверки конфликтности назначений *i*-й работы в *j*-й узел:

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

где $C_x^{i,j}$ — степень выполнения *x*-го обязательного требования для назначения *i*-й заявки на *j*-й ресурс. Для матрицы на рис.1 матрица $Q$ показана на рис. 2. Ввиду того, что обязательное требование следует выполнять безусловно, то элементы $Q_{i,j}$ могут принимать лишь значения 1 или 0.

На следующем этапе формирования исходной информации необходимо выполнить фильтрацию элементов матрицы $\Delta_t$ в соответствии со значениями элементов матрицы $Q$. В результате получаем новую матрицу $\Delta_t'$, показанную на рис. 3, где символом ∞ обозначены варианты невозможных назначений:

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 1 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 | 1 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 |

*Рис. 2.* Матрица проверки конфликтных назначений $Q_{i,j}$

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 3 | ∞ | ∞ | 4 | ∞ |
| 2 | –2 | 0 | –5 | ∞ | –4 | –2 |
| 3 | ∞ | 6 | ∞ | 7 | 9 | ∞ |
| 4 | ∞ | –3 | –4 | –5 | –1 | ∞ |
| 5 | –1 | ∞ | ∞ | 1 | –5 | –1 |
| 6 | 8 | 1 | ∞ | 4 | 3 | 2 |

*Рис. 3.* Матрица $\Delta_t'$ после фильтрации конфликтных назначений $\Delta_t$

Для формирования плана распределения работ по процессорным узлам в соответствии с требованиями венгерского алгоритма и ограничениями, накладываемыми постановкой задачи, требуется сформировать начальную или исходную зону поиска. Для этого выполним следующие действия.

Необходимо учесть, что положительные значения элементов матрицы $\Delta_t$ соответствуют назначениям, которые безусловно включаются в решение, так как любое назначение, соответствующее $\Delta_t^{i,j} > 0$, не противоречит условиям временных ограничений. Поэтому всем положительным элементам присваиваем значения 0, а у отрицательных элементов поменяем знак на положительный. В результате получаем новую матрицу $\Delta_t''$ (рис. 4).

## ПОИСК ПЛАНА ПО ВЕНГЕРСКОМУ АЛГОРИТМУ

Дальнейшие действия по формированию исходной области и собственно поиск паросочетания выполняются в соответствии с венгерским алгоритмом. Для этого из каждого элемента столбца матрицы $\Delta_t''$ вычитается наименьший элемент этого столбца:

$$\Delta_t'''^{i,j} = \Delta_t''^{i,j} - \min_i \Delta_t''^{i,j} \; . \; \Delta$$

Из каждого элемента строки полученной матрицы $\Delta_t'''$ вычитается наименьший элемент этой строки:

$$\hat{\Delta}_t^{i,j} = \Delta_t'''^{i,j} - \min_i \Delta_t'''^{i,j}.$$

В результате в каждой строке и каждом столбце матрицы $\hat{\Delta}_t$ имеем нулевой элемент (рис. 5).

Для поиска решения из исходной матрицы поиска $\hat{\Delta}_t$ выделяем нулевые элементы и формируем исходную матрицу для поиска максимального паросочетания (OPR) (рис. 6). Для получения решения используем один из алгоритмов поиска максимального паросочетания в невзвешенном двудольном графе. Для него необходимо инвертировать значения элементов матрицы OPR, т.е. заменить в ней все нулевые элементы на единичные и наоборот. Получаем матрицу поиска решения PR (рис. 7):

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | ∞ | ∞ | 0 | ∞ |
| 2 | 2 | 0 | 5 | ∞ | 4 | 2 |
| 3 | ∞ | 0 | ∞ | 0 | 0 | ∞ |
| 4 | ∞ | 3 | 4 | 5 | 1 | ∞ |
| 5 | 1 | ∞ | ∞ | 0 | 5 | 1 |
| 6 | 0 | 0 | ∞ | 0 | 0 | 0 |

*Рис. 4.* Промежуточная матрица $\Delta_t'''$

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | ∞ | ∞ | 0 | ∞ |
| 2 | 2 | 0 | 2 | ∞ | 4 | 2 |
| 3 | ∞ | 0 | ∞ | 0 | 0 | ∞ |
| 4 | ∞ | 2 | 0 | 4 | 0 | ∞ |
| 5 | 1 | ∞ | ∞ | 0 | 5 | 1 |
| 6 | 0 | 0 | ∞ | 0 | 0 | 0 |

*Рис. 5*. Исходная матрица поиска $\hat{\Delta}_t$

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 2 | 1 | 0 | 1 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 0 | 0 | 1 |
| 4 | 1 | 1 | 0 | 1 | 0 | 1 |
| 5 | 1 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 0 | 1 | 0 | 0 | 0 |

*Рис. 6.* Матрица области поиска решения (OPR)

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 |

*Рис. 7.* Матрица поиска решения (PR)

Для рассматриваемого примера, следуя венгерскому алгоритму, найдено максимальное паросочетание (рис. 8). Оно является окончательным планом размещения заявок по узлам, поскольку мощность полученного паросочетания равна размерности решения задачи, т.е. все заявки распределены.

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | **1** |
| 2 | 0 | **1** | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 1 | **1** | 0 |
| 4 | 0 | 0 | **1** | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | **1** | 0 | 0 |
| 6 | **1** | 1 | 0 | 1 | 1 | 1 |

*Рис. 8.* План распределения

В том случае, если совершенное паросочетание не получено, т.е. когда мощность полученного паросочетания меньше размерности матрицы PR, необходимо найти новую область поиска в соответствии с венгерским алгоритмом. В результате такого выполнения определяются элементы $\Delta_t^{i,j}$, которые могут быть дополнительно включены в зону поиска.

Рассмотрим пример, иллюстрирующий формирование новой зоны поиска, в соответствии с венгерским алгоритмом. В качестве исходной примем матрицу, представленную на рис. 9.

| | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ | $Y_6$ |
|---|---|---|---|---|---|---|
| $X_1$ | 3 | **0** | ∞ | 2 | 0 | **2** |
| $X_2$ | 5 | **2** | 8 | **0** | ∞ | 4 |
| $X_3$ | 2 | ∞ | **0** | 3 | **4** | 3 |
| $X_4$ | 8 | 0 | **7** | ∞ | 3 | 1 |
| $X_5$ | **0** | 1 | 0 | ∞ | 0 | 0 |
| $X_6$ | ∞ | ∞ | 1 | 3 | **0** | 6 |

*Рис. 9.* Исходная матрица, не имеющая прямого решения

Для данного примера имеем предварительное распределение, отмеченное выделенными нулями. Поиск максимального паросочетания по этой матрице не дает решения. Для дальнейшего поиска решения определяется минимальная опора. Минимальной опорой называют минимальное множество строк или столбцов, т.е. линий, содержащих все нулевые элементы. Для этого выполняют алгоритм [8]:

1. Отмечают в каждой строке, в которой есть решение, все нули.
2. Помечают знаком «+» каждую строку, не содержащую отмеченных нулей.
3. Помечают знаком «+» каждый столбец, содержащий отмеченные нули, в какой-либо из строк, помеченных знаком «+».
4. Помечают знаком «+» каждую строку, содержащую отмеченный нуль в каком-либо столбце, помеченном знаком «+».
5. Действия повторяют до тех пор, пока возможно помечать знаком «+» новые строки и столбцы.

Выделим минимальную опору. Для этого отметим все непомеченные знаком «+» строки (в примере — $X_2, X_3, X_5$) и все помеченные столбцы (в примере — $Y_2, Y_5$). В результате, получаем помеченную матрицу (рис. 10).

| | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ | $Y_6$ | |
|---|---|---|---|---|---|---|---|
| $X_1$ | 3 | **0** | ∞ | 2 | 0 | **2** | + |
| $X_2$ | 5 | **2** | 8 | **0** | ∞ | 4 | |
| $X_3$ | 2 | ∞ | **0** | 3 | **4** | 3 | |
| $X_4$ | 8 | 0 | **7** | ∞ | 3 | 1 | + |
| $X_5$ | **0** | 1 | 0 | ∞ | 0 | 0 | |
| $X_6$ | ∞ | ∞ | 1 | 3 | **0** | 6 | + |
| | | + | | | + | | |

*Рис.10.* Матрица с выделенной минимальной опорой

Затем формируется новая зона поиска. Для этого рассматривается подматрица, образованная элементами, через которые не проходят отмеченные на рис. 10 линии и берется наименьший элемент этой подматрицы, в примере — равный 1. Этот элемент вычитается из элементов всех тех столбцов, через которые не проходят отмеченные линии, и затем он прибавляется к элементам всех тех строк, через которые проходят выделенные линии. В данном примере единица вычитается из элементов столбцов $Y_1, Y_3, Y_4, Y_6$ и прибавляется затем к элементам строк $X_2, X_3, X_5$. В результате этих действий изменяется количество нулей, определяющих зону поиска решения.

После этого снова выполняется попытка найти максимальное паросочетание. Вышеописанные действия повторяются до тех пор, пока не будет получено максимальное паросочетание. Результирующая матрица показана на рис. 11.

| | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ | $Y_6$ |
|---|---|---|---|---|---|---|
| $X_1$ | 2 | **0** | ∞ | 1 | 0 | **1** |
| $X_2$ | 5 | **3** | 8 | **0** | ∞ | 4 |
| $X_3$ | 2 | ∞ | **0** | 3 | **5** | 3 |
| $X_4$ | 7 | 0 | **6** | ∞ | 3 | **0** |
| $X_5$ | **0** | 2 | 0 | ∞ | 1 | 0 |
| $X_6$ | ∞ | ∞ | 0 | 3 | **0** | 5 |

*Рис. 11.* Матрица окончательного решения

Таким образом, условия задачи динамического планирования с ограничениями реального времени для НПВС формально преобразуются в матрицу поиска, которая обрабатывается с использованием алгоритма поиска максимального паросочетания с быстрым получением эффективного плана.

## ЗАКЛЮЧЕНИЕ

Предложенный в работе метод преобразования исходной информации приводит решение задачи распределения заявок по вычислительным ресурсам с ограничениями реального времени к классической задаче комбинаторной математики — поиска максимального паросочетания во взвешенном двудольном графе. Быстрое решение такой задачи позволяет выполнять динамическое планирование в современных параллельных ВС. Дальнейшее ускорение составления расписания можно получить, применив алгоритм адаптивного планирования, предложенный в работе [7], который основан на улучшении алгоритма поиска максимального паросочетания во взвешенном двудольном графе.

## ЛИТЕРАТУРА

1. *Papadimitriou C.H.* Combinatory optimization, algorithm and complexity / C.H. Papadimitriou, K. Steiglitz. — New Jersey: Prentice-Hall. —1982. — 496 p.
2. 2. *Berge C.* Theorie des graphes et ses application / C. Berge. — Paris: Dunod. — 1958. — 275 p.
3. *Зыль С.* Операционная система реального времени QNX: от теории к практике. — 2-изд. / С. Зыль. — СПб.: БХВ — Петербург, 2004. — 192 с. ISBN 5-94157-486-X
4. *Зыль С.* QNX Momentics. Основы применения / С. Зыль. — СПб.: БХВ-Петербург, 2004. — 256 с.: ил. ISBN 5-94157-430-4
5. *Кёртен Р.* Введение в QNX/Neutrino 2 / Р. Кёртен . — СПб.: Петрополис. — 2001. — 512 С. ISBN 5-94656-025-9.
6. *Ослэндер Д.М.* Управляющие программы для механических систем: Объектно-ориентированное проектирование систем реального времени / Д.М. Ослэндер, Дж.Р. Риджли, Дж.Д. Рингенберг. — М.: Бином. Лаборатория знаний. — 2004. — 416 с. ISBN 5-94774-097-4.
7. *Симоненко В.П.* Улучшенный алгоритм назначения для планировщиков заданий в неоднородных распределенных вычислительных системах / В.П. Симоненко, А.М. Сергиенко, А.В. Симоненко // Системні дослідження та інформаційні технології. — 2016. — № 2. — С. 20–35.
8. *Kauffman A.* Introduction to a la combinatorque en vue des applications / A. Kauffman. — Paris: Dunod, 1968.

*Поступила 25.05.2016*