УДК 681.324

СИМНЕНКО В.П.,
СИМОНЕНКО А.В.

ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ РАБОТ ПО РЕСУРСАМ НЕОДНОРОДНОЙ СИСТЕМЕ С ОГРАНИЧЕНИЯМИ РЕАЛЬНОГО ВРЕМЕНИ

В работе предложен алгоритм динамического распределения задач в неоднородной системе в реальном времени. Представлен метод преобразования исходной информации, позволяющая решать задачу планирования известными методами.

This paper presents the dynamic tasks scheduling algorithm for heterogeneous systems in real time. The method for initial information transformation is proposed which allows solving scheduling problem by known methods.

Введение

Планирование в *реальном* времени характеризуется решением следующей задачи – определением плана решения совокупности задач с заданным временем исполнения и ограничениями по времени выхода задач из системы. Система планирования должна обеспечить выполнение требований по минимизации суммарного времени отклонения реального выхода задач из системы (выполнения сроков решения) от сходных временных ограничений при полном соблюдении порядка следования работ. В общем случае эта задача относится к классу NP-полных [5,6].

В большинстве случаях разработчики систем планирования реального времени используют статические алгоритмы и заранее определяют максимальный список заданий, допустив наихудший случай для получения статической управляющей таблицы (плана). Этот план фиксируется и используется для безусловного исполнения в динамическом режиме со слідующими допущениями:

- все временные ограничения остаются неизменными на время выполнения плана;
- все задачи вкладываются в свое критическое время.

В других случаях при помощи приемов стати-ческого 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 соглашений статического планирования (где все известно априори) недопустимо. В этом случае выбирается один из возможных алгоритмов составления расписания и тщательно анализируется на применимость его в ожидаемом динамическом окружении. Как правило, применяются алгоритмы, использующие планирование по спискам и приоритетное обслуживание [1-4]. В данной работе предлагается использовать модифицированный Венгерский алгоритм для решения задачи динамического распределения заявок по ресурсам неоднородной GRID системы с ограничениями реального времени, имеющий меньшую временную сложность чем известные. В работе предлагается метод преобразования исходной информации позволивший применить математический аппарат поиска максимального паросочетания для задачи составления плана распределения задач по ресурсам с ограничениями реального времени, выполняемый планировщиком нижнего уровня.

Постановка задачи

Рассмотрим общую постановку задачи в реальном времени.

На вход СРОД поступает множество работ. Каждая работа характеризуется тремя временными параметрами:

$T_{вх}^{i}$ – время поступления i-ой входной работы в ВС

$T_{вых}^{i}$ – время выхода i-ой работы из ВС

$T_{раб}^{i}$ – время выполнения i-ой работы в относительных единицах на вычислительном узле GRID, имеющем максимальную производительность GRID имеет множество

ресурсов. Каждый ресурс характеризуется производительностью R^j. Для каждой i-ой работы можно определить время решения i-ой задачи на j-ом ресурсе в относительных единицах.

Для этого определим R_{max} как $R_{max} = \max\{R^1,..,R^n\}$.

Определим относительную призводительность каждого вычислительного узла R^j по отношению к R_{max}. Для этого вычисляем

$$Z^j = \frac{R^j}{R_{мах}},\ j=1..n.$$

Имея относительную производительность каждого узла, можно определить отношение работа-ресурс с учетом времени выполнения работы и производительности каждого узла GRID системы.

Для этого сформируем матрицу связности (МС), где каждый элемент определяет относительное время выполнения каждой i-ой работы на j-ом ресурсе с учетом его призводительности $MC[i,j] = \frac{T^i_{раб}}{Z^j}$.

Ввиду того, что $1 \geq Z^j > 0$, то относительное время выполнения каждой работы на ресурсах будет определяться значением MC[i,j]. В связи с тем, что задача, выполняемая в реальном времени, должна завершиться до заданного $T^i_{вых}$, то необходимо вычислить $\Delta_t^{i,r}$ из выражения:

$$\Delta_t^{i,j} = T^i_{вых} - T^i_{вх} - MC[i,j].$$

Так как $(T^i_{вых} - T^i_{вх}) > 0$,

а $(T^i_{вых} - T^i_{вх}) > MC[i,j] > (T^i_{вых} - T^i_{вх})$, то $\Delta_t^{i,j}$ может принимать как положительные так и отрицательные значения. Положительные значения $\Delta_t^{i,j}$ означают, что работа i, назначенная на ресурс j, будет завершена на время $\Delta_t^{i,j}$ до назначенного срока, а отрицательные значения означают, что выход заявки из системы произойдет позже назначенного срока на время $\Delta_t^{i,j}$ (рис. 1).

При вычислении $\Delta_t^{i,j}$ следует учитывать, что на момент планирования ($T_{план}$) все заявки имеют одно и то же базовое время начала планирования независимо от времени прихода заявок в систему. Кроме этого следует учитывать и время работы планидовщика, т.е. время планирования ε.

Поэтому можно принять $T^i_{вх} = T_{план} + \varepsilon$.

	РЕСУРСЫ						
		1	2	3	4	5	6
	1	1	3	-2	-3	4	5
А	2	-2	0	-5	-6	-4	-2
Б	3	3	6	1	7	9	1
О	4	1	-3	-4	-5	-1	-1
Т	5	-1	4	-4	1	-5	-1
Ы	6	8	1	2	4	3	2

Рис. 1. Исходная матрица

В соответствии с требованиями режима реального времени значение $\sum_{z=1}^{n} \left|\Delta_t^{i,j}\right|$ 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.

В такой постановке решение задачи назначения i-ой работы на j-ый ресурс сводится к задаче поиска максимального паросочетания в взвешенном двудольном графе и решатся Венгерским алгоритмом или алгоритмом направленного поиска. Следует отметить, что выполнение поиска варианта рас пределения выполняется для неоднородной вычислительной системы (степень неоднородности не ниже 0.7) и поиск решения выполняется в разреженной бинарной матрице связности. При решении задачи поиска максимального паросочетания можно применить адаптивный алгоритм, временная сложность которого зависит от коэффициента заполнения матрицы связности двудольного графа, а при коэффициенте заполнения менее 0.7 вре-

менная сложность не превышает $O(n^{1.5}\log n)$ [41].

Для поставленной задачи требуется модификАция формирования исходных данных и базовой (начальной) области поиска. Рассмотрим процедуры формирования исходных данных и области поиска на примере решения задачи назначения для 6-ти задач на 6-ти процессорах. После обработки исходных временных ограничений и учета производительности каждого процессора получим исходную матрицу $\Delta_t^{i,j}$ (рис. 3.12). Информация в матрице $\Delta_t^{i,j}$ учитывает только объем работы и относительное время ее выполнения с учетом производительности вычислительных узлов. Однако неоднородность СРОД обуславливает не обходимость учета индивидуальных характеристик каждого вычислительного узла (наличия достаточной оперативной памяти, необходимых данных в узле, необходимых программ для выполнения задания и т.д.), связанных с возможностью выполнения задания. С учетом этих критериев необходима оценка принципиальной возможности выполнения каждой работы в каждом узле. Для этого формируется матрица проверки конфликтности назначений каждой i-ой работы в j-ый узел:

$$Q_{i,j} = \prod_{x=1}^{p} C_x^{i,j} \text{ (рис. 2),}$$

где: $C_x^{i,j}$ – степень выполнения x обязательного требования для назначения i-ой заявки на j-ый ресурс;

На следующем этапе формирования исходной информации необходимо выполнить фильтрацию элементов матрицы $\Delta_t^{i,j}$ в соответствии со значениями элементов матрицы $Q_{i,j}$. В результате получаем новую матрицу $\underline{\Delta_t^{i,j}}$, где символом ∞ обозначены назначения, которые не могут рассматриваться в качестве возможных (рис. 3).

	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 $\Delta_t^{i,j}$

	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 $\underline{\Delta_t^{i,j}}$ ***после фильтрации назначений в матрице*** $\Delta_t^{i,j}$

Для формирования плана распределения задач по процессорам, в соответствии с требованиями Венгерского алгоритма и ограничениям накладываемыми нашей постановкой задачи, требуется сформировать начальную (исходную) зону поиска. Для этого выполним следующие действия. Ввиду того, что положительные значения элементов матрицы $\underline{\Delta_t^{i,j}}$ соответствуют назначениям, которые могут быть, безусловно, включены в решение, т.к. любое назначение, соответствующее координате i, j, не противоречит условиям временных ограничений, то всем положительным элементам присваиваем значения 0, а отрицательным элементам поменяем знак на положительный. В результате получаем новую матрицу $\underline{\Delta_t^{i,j}}$ (рис. 4). Дальнейшие действия по формированию исходной области поиска выполняем в точном соответствии с Венгерским алгоритмом. Из каждого элемента столбца матрицы $\underline{\Delta_t^{i,j}}$ вычитается наименьший элемент этого столбца. $\underline{\Delta_t^{i,j}}(1) = \underline{\Delta_t^{i,j}}(0) - \min_i \Delta_t^{i,j}$).

Из каждого элемента строки полученной матрицы $\underline{\Delta_t^{i,j}}$ вычитается наименьший элемент этой строки. $\underline{\Delta_t^{i,j}}(2) = \underline{\Delta_t^{i,j}}(1) - \min_j \Delta_t^{i,j}$;

В результате выполненных действий в каждой строке и каждом столбце имеем нулевой элемент (рис. 5).

	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. Промежуточна матрица

	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

Для поиска решения из исходной матрицы поиска $\underline{\Delta_t^{i,j}}$ (2) выделяем нулевые элементы и формируем исходную матрицу для поиска максимального паросочетания (OPR) (рис.6). Как уже упоминалось, для получения решения используется один из алгоритмов поиска максимального паросочетания в невзвешенном двудольном графе. Для него необходимо инвертировать значения элементов матрицы OPR. Т.е. заменить в ней все 0-е элементы на 1 и наоборот. В результате 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. 6. Матрица области поиска решения (OPR)

	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. 7. Матрица поиска решения (PR)

Для рассматриваемого примера найдено максимальное паросочетание (рис. 8), которое является окончательным планом размещения заявок по процессорам, т.к. мощность полученного распределения (мощность 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 (мощность полученного паросочетания на равна размерности матрицы PR), необходимо выполнить действия для получения новой области поиска.

Дальнейшие действия определены процедурами Венгерского алгоритма, в результате выполнения которых определяются элементы $\underline{\Delta_t^{i,j}}$, которые могут быть дополнительно включены в зону поиска.

Рассмотрим пример, иллюстрирующий формирование новой зоны поиска.

В качестве исходной примем матрицу, представленную на рис. 9. Поиск максимального паросочетания по отмеченным “0” не дал решения.

	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.

Для данного примера имеем предварительное распределение, отмеченное выделенными нулями. Для дальнейшего поиска решения определяется минимальная опора. (Минимальная опора – минимальное множество линий, содержащих все “0” матрицы). Для этого действуем последовательно:

Отметим в каждой строке, в которой есть решение все нули;

Помечаем знаком “+” каждую строку, не содержащую отмеченных нулей.

Помечаем знаком “+” каждый столбец, содержащий отмеченные нули, какой-либо их помеченных “+” строк.

Помечаем знаком “+” каждую строку, содержащую отмеченный нуль в каком-нибудь столбце, помеченном “+”.

Действия повторяются до тех пор, пока возможно помечать “+” новые строки и столбцы.

Выделим минимальную опору. Для этого отметим все непомеченные “+” 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. 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. 10. Выделенная минимальная опора

Затем формируется новая зона поиска. Для этого рассматривается подматрица, образованная плементами, через которую не проходят отмеченные на рис. 9 линии, и возьмем наименьший элемент этой подматрицы. Вычтем это число из элементов всех тех столбцов, через которые не проходят отмеченные линии, и затем прибавим его к элементам всех тех строк, через которые пунктирные линии проходят. В данном примере единица вычитается из элементов столбцов Y_1, Y_3, Y_4, Y_5 и прибавляется затем к єлементам строк X_2, X_3, X_5. В результате выполненных действий изменяется количество 0, определяющих зону поиска решения, и ищется максимальное паросочетание. Вышеописанные действия повторяются до тех пор, пока не будет получено максимальное паросочетание.

Результат этих действий показан на рис. 10.

	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. Окончательное решение

Заключение

Предложенный в статье метод преобразования исходной информации позволил привести решение задачи распределения заявок по вычислительным ресурсам к решению классической задачи коминаторной математики – поиска максимального паросочетания во взвешенном двудольном графе. А использование метода и алгоритма адаптивного планирования [7] позволяет значительно уменьшить временную сложность классического ‘венгерского алгоритма’, используемого для поиска максимального пароосочетания во взвешенном дудольном графе.

Литература

1. Зыль С. Операционная система реального времени QNX: от теории к практике. 2-издание. – СПб.: БХВ-Петербург, 2004. – 192 с.: ил. ISBN 5-94157-486-X
2. Зыль С. QNX Momentics. Основы применения. – СПб.: БХВ-Петербург, 2004. – 256 с.: ил. ISBN 5-94157-430-4
3. Кёртен Р. Введение в QNX/Neutrino 2. – СПб.: Петрополис, 2001. – 512 с. ISBN 5-94656-025-9
4. Ослэндер Д. М., Риджли Дж. Р., Рингенберг Дж. Д. Управляющие программы для механических систем: Объектно-ориентированное проектирование систем реального времени. – М.: Бином. Лаборатория знаний, 2004. – 416 с. ISBN 5-94774-097-4
5. Papadimitry X., Stayglitsh K., Combinatory optimization, algorithm and complexity, Moscow: Mir (1985).
6. Berge C. , Theorie des graphes et ses application,. Dunod, Paris (1958).
7. Симоненко А.В. Выбор стратегии пространственного планирования в параллельных вычислительных системах. Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техника, Київ 2001 р. № 35. – с. 104-108.
8. Симоненко А.В. Сравнительная характеристика методов адаптивного распределения данных в неоднородной вычислительной системе, Вісник НТУУ «КПІ» Інформатика, управління та обчислювальна техника, Київ 2006 р. № 45. – с. 54-60.