5. Chizhenkova R.A., Safroshkina A.A. Effect of low-intensity microwaves on the behavior of cortical neurons. Bioelectrochemistry and Bioenergetics, 1993, v. 30, No. 1, pp. 287-391.

6. Chizhenkova R.A., Safroshkina A.A. Electrical reactions of the brain to microwave irradiation. Electro- and Magnetobiology, 1996, v. 15, No. 3, pp. 253-258.

7. Chizhenkova R.A., Safroshkina A.A., Slashcheva N.A., Chernukhin V.Yu. Bibliometrical analysis of neurophysiological aspects of action of non-ionized radiation. Uspekhi sovremennoy biologii, 2004, v. 124, No. 5, pp. 472-479.

8. Kholodov Yu.A. Influence of electromagnetic fields on central nervous system. Moscow: Nauka, 1966. 283 p.

9. Kholodov Yu.A. Reactions of nervous system on electromagnetic fields. Moscow: Nauka, 1975. 207 p.

УДК 681.63, 519.872

ИМИТАЦИОННАЯ МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ GRID-УЗЛОВ С ОЧЕРЕДЬЮ ДОСТУПА К ОБЩЕЙ ПАМЯТИ[1]

Шелестов А.Ю

Введение

В настоящее время усложнение инфраструктуры распределенных информационных систем привело к необходимости развития подходов к их моделированию. Системный подход к решению этой задачи предложен в [1], а более детально его применение описано в [2]. Одной из наиболее перспективных технологий построения распределенных систем является Grid-технология, которая обеспечивает решение вычислительно сложных задач с использованием распределенных хранилищ данных и вычислительных ресурсов [3]. Для моделирования подобных систем могут использоваться различные подходы, каждый из которых не позволяет провести комплексные исследование. Достаточно полный обзор существующих моделей содержится в [4, 5].

В данной работе для моделирования динамики Grid-систем предлагается использовать сети Петри [6], поскольку такие системы представляют собой набор взаимодействующих друг с другом компонентов (узлов различного типа), которые могут функционировать параллельно и работа которых должна быть синхронизована. Именно при исследовании Grid-систем наблюдения Земли все эти вопросы являются очень актуальными.

Постановка задачи

Основным элементом структурной модели Grid-системы наблюдения Земли [2] является вычислительный узел, выполняющий вычислительные задачи или задачи доступа к данным хранилища (информационный узел) по запросу пользователя или в режиме операционного сервиса. Поскольку узел может содержать несколько различных физических вычислительных устройств (процессоров или многопроцессорных компьютеров), которые используют общую память, то работа отдельных устройств координируется управляющими узлами. Поэтому вычислительные, информационные и управляющие узлы (планировщики) следует рассматривать как базовые элементы распределенной системы доступа и обработки данных. Задача состоит в построении Grid-системы, обеспечивающей прозрач-

[1] Работа выполнена в рамках темы НАНУ «Интеллект», а также при поддержке гранта INTAS-CNES-NSAU “Data Fusion Grid Infrastructure” (Ref. Nr 06-1000024-9154).

 ISBN 7-776-8361-7

ную обработку пользовательских запросов на поиск/обработку данных, синхронизацию доступа, обработку данных и предоставление результатов обработки. Такая система относится к Grid-системам смешанного типа (т.е. является и вычислительной и информационной [3]), поскольку содержит как высокопроизводительные вычислительные узлы, так и обеспечивает доступ к данным распределенных хранилищ [5].

Синхронизация доступа к данным и их корректная обработка предполагает выполнение некоторых базовых свойств, а именно: *взаимное исключение* (mutex) — синхронизация событий при обработке данных, когда два или более вычислительных узла не могут одновременно получить доступ к одним и тем же данным; *справедливость* (fairness) — отсутствие «дискриминации» заданий. Если пользователь сформировал запрос (задание) и отправил его в систему, то это свойство гарантирует, что обязательно наступит момент, когда это задание начнет выполняться и будет выполнено; *отсутствие блокировок* (deadlock free) — в системе не может возникнуть ситуация взаимной блокировки вычислений или доступа к данным (общим или распределенным).

С учетом высокой стоимости компонентов Grid-систем перед построением системы необходимо построить ее модель и исследовать свойства. Построим модель функционирования вычислительного узла с общей памятью и исследуем свойства этой модели. Учитывая специфику работы Grid-системы и необходимость синхронизации доступа к общим сегментам данных при организации параллельных вычислений, в качестве средства моделирования воспользуемся аппаратом сетей Петри.

Поскольку для регламентирования доступа к общей памяти в реальных Grid-системах используется очередь, введем ее в модель и проанализируем основные свойства полученной сети Петри. При этом будем считать, что управление очередью тоже осуществляется управляющим узлом (планировщиком).

Модель взаимодействия вычислительного и управляющего узлов в виде сети Петри

Модель работы вычислительного узла с 4 процессорами, обрабатывающего задания пользователя под управлением планировщика, показана на рис. 1. Эта модель представляет собой одноцветную сеть Петри вида

$$C = (P, T, A), \tag{1}$$

где

$$P = \{ p_1, p_2,...,p_n \}, n = 14 – \tag{2}$$

множество позиций,

$$T = \{ t_1, t_2,...,t_m \}, m = 9 – \tag{3}$$

множество переходов, A – (n×m) матрица инцидентности графа сети, связывающая позиции с переходами сети.

Если элемент $A_{ij}=0$, значит i-я позиция и j-й переход не связаны между собой, если $A_{ij}=k>0$, значит в результате запуска j-й перехода в i-й позиции добавляется k фишек, если $A_{ij}=k<0$, значит в результате запуска j-й перехода из i-й позиции удаляется k фишек.

Начальное состояние СП, представленной на рис. 1, описывается разметкой

$$M_0=(proc, query, want, work, wait, free, busy, 2, 3, 4, 5, 6, block, dop)=$$
$$=(5, 4, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0).$$

Позиции и переходы в данной сети Петри (СП) имеют следующую семантику: proc — свободные в данный момент времени процессоры; query — запрос на выполнение задания от пользователя; want — процессор активен и может приступить к работе; work — процессор получил доступ к общей памяти и выполняет вычисления; wait (1) — активный процессор находится первым в очереди и ожидает доступа к общей памяти; block — при *отсутствии* фишек в этой позиции блокируется доступ к общей памяти, если она занята другим процессором; free — управляющий узел сигнализирует о снятии блокировки общей памяти; busy — управляющий узел сигнализирует о том, что общая память заблокирована для доступа; 2, 3 — вторая и третья позиции в очереди глубины 3, соответственно; 4, 5, 6 — вспомогательные позиции, с помощью которых выполняется перемещение зада-

ний в очереди; dop — вспомогательная позиция, обеспечивающая преобразование полуцикла в цикл для удобства анализа свойств сети.

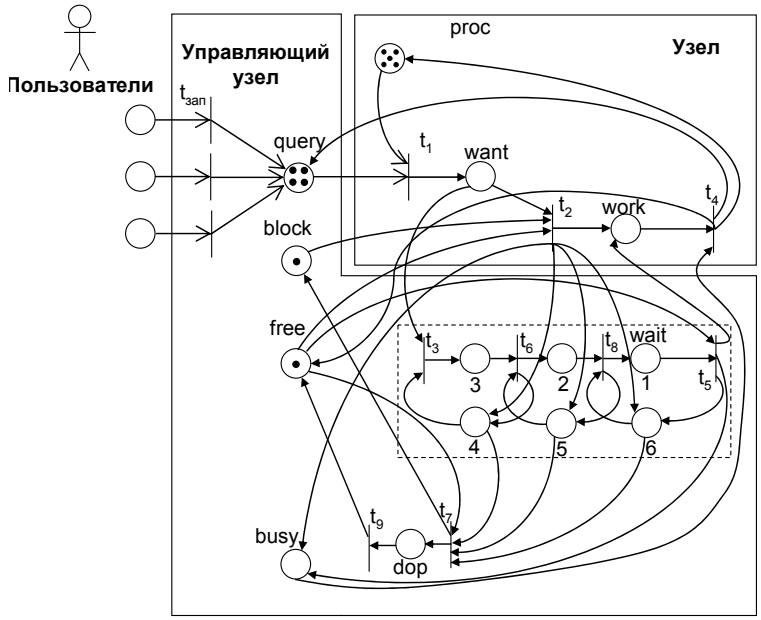

Рис. 1. Модель взаимодействия вычислительного и управляющего узлов при работе над общей памятью с глубиной очереди 3

Из семантики позиций следует следующий смысл переходов:

$t_{зап}$ — наличие запроса на обработку данных; t_1 — один из свободных процессоров готов к обработке запроса; t_2 — процессор получает доступ к общей памяти; t_3 — активный процессор становится в очередь на доступ к общей памяти; t_4 — процессор завершил работу с общей памятью и она разблокирована; t_5 — активный процессор из состояния ожидания переходит в состояние работы с общей памятью, при этом одновременно происходит перемещение элементов очереди; t_6, t_8 — активный процессор перемещается по очереди; t_7 — при отсутствии элементов в очереди к общей памяти разблокируется переход активного процессора в состояние работы. Переход t_9 выполняет вспомогательную роль для модификации структуры сети Петри с целью преобразования полуцикла в цикл. Это сделано во избежание неоднозначностей в процессе последующего анализа сети.

При появлении запроса пользователя на выполнение задания срабатывает сначала один из переходов $t_{зап}$, а затем, при наличии свободных процессоров, переход t_1. В результате этого один из свободных процессоров переходит в состояние активности (готовности к работе), т.е. фишка переходит в позицию want. Если одновременно поступает несколько запросов, то при отсутствии маркера в позиции block первый из них выполняется (фишка помещается в позицию work), а последующие запросы будут перенаправлены в очередь при наличии в ней свободных мест. Выполнение запросов, не попавших в очередь, будет задержано отсутствием маркера во вспомогательной позиции 4. Дальнейшая работа модели очевидна.

Исследование структурных свойств модели

Утверждение 1. Представленная на рис. 1 сеть Петри, описывающая модель функционирования вычислительного узла Grid-системы с очередью доступа к общей памяти, является ограниченной, живой и не содержит недостижимых позиций.

Доказательство. Для доказательства этого утверждения воспользуемся уравнением состояний

$$Ax=0, \qquad (4)$$

где A — целочисленная n×m матрица инцидентности сети Петри, *n* и *m* — мощности множеств *P* и *T* соответственно, а x — m-мерный вектор Париха [7].

 ISBN 7-776-8361-7

Размерность матрицы инцидентности A в (4) для данной сети Петри составляет 14×9 (14 уравнений, 9 неизвестных), а вектора Париха x — 9.

С помощью уравнения состояния (4) определим S- и T-инварианты сети Петри. Это и даст возможность выявить мертвые переходы, недостижимые позиции и проверить ограниченность сети. Для получения множества S- и T-инвариантов сети Петри используется TSS-алгоритм [8, 9], который позволяет построить минимальную порождающую систему решений однородной системы линейных диофантовых уравнений над множеством натуральных чисел N. В соответствии с TSS-алгоритмом система диофантовых уравнений (4) имеет 2 решения:

$$x_1 = \{1, 0, 1, 1, 1, 1, 0, 1, 0\}^T,\ x_2 = \{1, 1, 0, 1, 0, 0, 1, 0, 1\}^T.$$

На основе решений этой системы можно построить T-инварианты (табл. 1).

Табл. 1

T-инварианты сети Петри, представленной на рис. 1

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
1	0	1	1	1	1	0	1	0
1	1	0	1	0	0	1	0	1

Поскольку в табл. 1 нет ни одного столбца, содержащего только нулевые элементы, значит, все переходы в сети Петри являются живыми при данной начальной разметке, т.е. каждый переход в сети срабатывает хотя бы один раз. Несложно удостовериться, что это свойство выполняется и для других возможных начальных разметок (варьироваться может число фишек в позициях query и proc). Следствием этого факта является *отсутствие тупиков*, когда в системе не может возникнуть ситуации взаимной блокировки вычислений или доступа к данным (общим или распределенным). Действительно, тупик возникает в сети Петри, если нельзя запустить один или несколько переходов. Поскольку все переходы являются живыми (активными), то сеть Петри обладает требуемым свойством — характеризуется отсутствием тупиков.

Сгенерируем S-инварианты данной сети Петри. Для этого найдем решения системы уравнений

$$A^T y = 0, \tag{5}$$

где A — целочисленная n×m матрица инцидентности сети Петри, т — символ транспонирования матрицы, y — n-мерный вектор.

Решая систему диофантовых уравнений (5), содержащую 9 уравнений и 14 неизвестных, получаем S-инварианты сети Петри. Как следует из этого множества инвариантов, сеть *ограничена*, поскольку все позиции покрываются положительными инвариантами. Это означает, что ни в одной позиции сети не может скапливаться бесконечное число фишек. Из множества решений системы (5) также следует, что все позиции в сети являются *достижимыми*, т.е. модель не содержит лишних позиций.

Покажем достижимость в этой сети из начальной разметки

$$\begin{aligned} M_0 &= (proc, query, want, work, wait, free, busy, 2, 3, 4, 5, 6, block, dop)^T = \\ &= (5, 4, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0)^T \end{aligned} \tag{6}$$

разметки $M = (1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0)^T$, определяющей выполнение одного из запросов и ожидание в очереди на доступ к общей памяти остальных процессоров.

Для этого рассмотрим систему линейных неоднородных диофантовых уравнений (СЛНДУ) вида

$$Ax = M - M_0, \tag{7}$$

где A — целочисленная n×m матрица инцидентности сети Петри из (4), n и m — мощности множеств позиций P и переходов T соответственно, а x — m-мерный вектор Париха.

Для данной СП уравнение (7) имеет вид

$$\begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & -1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \mathrm{x} = \begin{bmatrix} -4 \\ -4 \\ 0 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad (8)$$

Среди решений системы (8) будут следующие

$$x_1 = (1, 0, 1, 1, 1, 1, 0, 1, 0, 0)^T,\ x_2 = (4, 1, 3, 0, 0, 2, 0, 1, 0, 1)^T. \quad (9)$$

Соответствие компонентов векторов этих решений переходам сети Петри представлено в табл. 2.

Табл. 2

Соответствие решений системы (9) переходам в СП из рис. 1

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_0
1	0	1	1	1	1	0	1	0	0
4	1	3	0	0	2	0	1	0	1

Последний столбец таблицы 2 соответствует фиктивному переходу t_0, используемому при преобразовании неоднородной системы уравнений в однородную. Соответственно, решение x_1 можно отбросить, поскольку последний компонент этого вектора равна 0 (а не 1). Решение x_2 (последний компонент которого равна 1) показывает, что данная разметка M в сети достижима путем четырехкратного срабатывания перехода t_1, однократного срабатывания перехода t_2, трехкратного срабатывания перехода t_3, двукратного срабатывания перехода t_6 и однократного срабатывания перехода t_8.

Таким образом, модель вычислительного узла в виде сети Петри является живой, ограниченной и все позиции в сети являются достижимыми, включая разметку, определяющую корректную работу процессоров над общей памятью.

Утверждение 1 доказано.

Исследование свойства отсутствия блокировок

Определение. Ловушкой [6] называется такое непустое подмножество Q позиций в СП, которое удовлетворяет условию

$$Q_t \bullet \subseteq \bullet Q_t, \quad (10)$$

т.е. если входная позиция перехода t принадлежит некоторому множеству Q, то и все выходные позиции этого перехода тоже принадлежат множеству Q.

Очевидно, выполняется следующее свойство ловушки: если хотя бы одна позиция ловушки помечена начальной разметкой, то множество позиций ловушки всегда будет содержать ненулевое число фишек. Исходя из этого факта, построим минимальное множество ловушек и рассмотрим разметки их позиций. Для простоты множество входных позиций перехода t_i будем обозначать $\bullet t_i$, а множество выходных позиций этого перехода - $t_i \bullet$.

Процесс нахождения минимального множества ловушек состоит из трех этапов:

1. Построение множеств $\bullet t_i$ и $t_i \bullet$ для всех переходов t в СП.
2. Построение системы логических зависимостей на основе результатов п. 1.

 ISBN 7-776-8361-7

3. Построение системы линейных однородных диофантовых неравенств (СЛОДН), отвечающих системе логических зависимостей из п. 2.

Решение полученной системы СЛОДН позволяет получить минимальное множество ловушек данной СП.

Выполним каждый из этих этапов анализа для СП, представленной на рис. 1. Из рисунка видно, что множества входных и выходных позиций $\bullet t_i$ и $t_i \bullet$, $\forall t_i$, $i = \overline{1,9}$ включают следующие элементы.

$$
\begin{array}{lll}
\bullet t_1 = \{proc, query\} & \bullet t_7 = \{4, 5, 6, \ free\} & t_4 \bullet = \{proc, query, free\} \\
\bullet t_2 = \{want, block, free\} & \bullet t_8 = \{2, 6\} & t_5 \bullet = \{work, busy, 6\} \\
\bullet t_3 = \{want, 4\} & \bullet t_9 = \{dop\} & t_6 \bullet = \{2, 4\} \\
\bullet t_4 = \{wort, busy\} & t_1 \bullet = \{want\} & t_7 \bullet = \{block, dop\} \\
\bullet t_5 = \{wait, \ free\} & t_2 \bullet = \{work, busy, 4, 5, 6\} & t_8 \bullet = \{wait, 5\} \\
\bullet t_6 = \{3, 5\} & t_3 \bullet = \{3\} & t_9 \bullet = \{free\}
\end{array} \qquad (11)
$$

Пользуясь этой информацией, построим систему логических зависимостей между позициями (в соответствии с п. 2) и выполним ее эквивалентное преобразование с учетом соотношения

$$A \to B \Leftrightarrow \neg A \vee B . \qquad (12)$$

Получим

$$
\begin{array}{ll}
\neg proc \vee want , & \neg query \vee want , \\
\neg want \vee work \vee busy \vee 4 \vee 5 \vee 6 , & \neg want \vee 3 , \\
\neg block \vee work \vee busy \vee 4 \vee 5 \vee 6 , & \neg free \vee work \vee busy \vee 4 \vee 5 \vee 6 , \\
\neg free \vee block \vee dop , & \neg work \vee proc \vee query \vee free , \\
\neg busy \vee proc \vee query \vee free , & \neg wait \vee work \vee busy \vee 6 , \\
\neg 3 \vee 2 \vee 4 , & \neg 5 \vee 2 \vee 4 , \\
\neg 5 \vee block \vee dop , & \neg 6 \vee block \vee dop , \\
\neg 4 \vee block \vee dop , & \neg 6 \vee wait \vee 5 , \\
\neg 2 \vee wait \vee 5 , & \neg dop \vee free .
\end{array} \qquad (13)
$$

Переходя к третьему этапу нахождения минимального множества ловушек (в соответствии с п. 3), на основе системы (13) построим матрицу D размерности 18×14, столбцы которой соответствуют позициям в СП, а строки — логическим зависимостям системы (13)

$$
\mathrm{D} = \begin{bmatrix}
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix} \qquad (14)
$$

Столбцы матрицы D в (14) соответствуют позициям СП в следующем порядке:

1 — block	6 — query	11 —4
2 — wait	7 — work	12 — 5
3 — busy	8 — want	13 — 6
4 — free	9 — 2	14 — dop
5 — proc	10 — 3	

(15)

Матрица D в (14) определяет СЛОДН

$$Dx \geq 0, \tag{16}$$

которую путем введения дополнительных неизвестных можно преобразовать к системе линейных однородных диофантовых уравнений (СЛОДУ)

$$By = 0, \tag{17}$$

где матрица B размерности 18×32 имеет вид:

$$B = [D \mid C], \tag{18}$$

где C — диагональная матрица размерности (18×18), все ненулевые элементы которой равны -1.

Решениями СЛОДУ (17) являются 78 векторов, определяющие минимальное множество ловушек. Анализируя это множество решений, несложно удостовериться, что все ловушки содержат хотя бы одну позицию, помеченную начальной разметкой

$$M_0=(\text{block, wait, busy, free, proc, query, work, want, 2, 3, 4, 5, 6, dop})=$$
$$=(1, 0, 0, 1, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0). \tag{19}$$

Поскольку данная СП не является сетью свободного выбора или расширенной СП свободного выбора, то это является необходимым (но не достаточным) условием отсутствия блокировок в сети. Следовательно, в данной СП *выполняется необходимое условие отсутствия блокировок*.

Для проверки выполнения достаточного условия отсутствия блокировок нужно построить и исследовать достижимость состояний в транзиционной системе [10], соответствующей данной СП, что и будет сделано ниже при проверке выполнения свойства справедливости.

Исследование свойства взаимного исключения

Перейдем к исследованию выполнения свойства взаимного исключения, т.е. удостоверимся, что в данной СП невозможна ситуация одновременной работы двух или более процессоров с общей памятью.

Для этого рассмотрим систему линейных неоднородных диофантовых уравнений (СЛНДУ) вида

$$Dx = M - M_0, \tag{20}$$

где D — целочисленная матрица инцидентности сети Петри размерности n×m, n и m — мощности множеств позиций P и переходов T соответственно (для данной модели n=14 и m=9), а x — m-мерный вектор Парихa; вектор $M_0 = (block, wait, busy, free, proc, query, work, want, 2, 3, 4, 5, 6, dop)^T$ $= (1, 0, 0, 1, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0)^{\mathrm{T}}$ определяет начальную разметку СП, а вектор $M = (0, 0, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 1, 0)^{\mathrm{T}}$ соответствует гипотетической разметке сети в случае двух одновременно работающих процессоров (седьмой элемент вектора M принимает значение 2). Рассмотрим соответствующую расширенную однородную систему. Она имеет единственное решение

$$\tilde{x} = (1, 1, 0, 1, 0, 0, 1, 0, 1, 0)^{\mathrm{T}}. \tag{22}$$

Поскольку последний компонент вектора решения $\tilde{x}$ в (22) равен 0 (а не 1), то исходная система неоднородных уравнений не имеет решений. Следовательно, разметка M в данной СП недостижима, и два процессора не могут одновременно работать над общей памятью. Этот же результат можно получить путем анализа транзиционной системы или

 ISBN 7-776-8361-7

графа достижимости, который будет построен в следующем разделе. Аналогично можно убедиться в выполнении свойства взаимного исключения для других разметок M.

Исследование свойства справедливости

Для исследования свойства справедливости построим транзиционную систему, соответствующую СП на рис. 1. Анализ достижимых разметок свидетельствует о выполнении свойства справедливости: если некоторая задача попала в очередь, то она обязательно получит доступ к общей памяти. При появлении активного процессора (1 в позиции 3) в разметке μ из этой разметки достижима разметка μ`, при которой фишка перемещается в позицию wait, а затем и в позицию work. А это означает, что процессор рано или поздно получит доступ к соответствующему ресурсу (памяти, данным и т.п.).

При этом единственным способом получения доступа к общей памяти после начала выполнения первого задания является прохождение через очередь (это условие обеспечивается позицией block в СП).

Выводы

В данной статье представлена имитационная модель функционирования узла Grid-системы при работе с общей памятью с учетом очереди на использование общих ресурсов. В качестве средства моделирования применены сети Петри. Выбор математического аппарата для моделирования объясняется необходимостью обеспечения синхронизация доступа к общим ресурсам (данным) Grid-системы при выполнении параллельных вычислений. Анализ предлагаемой модели выполнен на основе уравнений состояний сети, для решения которых применяется TSS-алгоритм решения систем линейных диофантовых уравнений над множеством натуральных чисел, а также построения транзиционной системы или графа достижимости. Анализ построенной модели позволяет утверждать, что она обладает необходимыми свойствами сетей Петри, т.е. является живой, ограниченной и все ее позиции являются достижимыми. Доказано выполнение базовых свойств, необходимых для обеспечения синхронизации доступа к данным и их корректной обработки на узлах Grid-системы. В частности, доказано выполнение необходимых и достаточных условий отсутствия блокировок, с помощью графа достижимости проверено выполнение свойств справедливости и взаимного исключения. Таким образом, все необходимые структурные и поведенческие свойства модели выполняются, что гарантирует корректную работу моделируемой Grid-системы в реальной среде.

In this paper the model of the Grid nodes cooperation with common memory access queue was constructed using Petri network approach. The structural properties of this model were investigated. In particular, was shown, that constructed network is bounded, living and haven't the inaccessible places. For constructed model was also performed the feasibility analysis of the mutex and fairness properties.

1. Згуровский М.З., Панкратова Н.Д. Системный анализ: проблемы, методология, приложения. — К.: Наукова думка, 2005. — 744 с.

2. Шелестов А.Ю. Структурно-функциональный анализ компонентов Grid-систем// Проблемы управления и информатики.— 2007.— № 5. — С.

3. Krauter K., Buyya R., Maheswaran M. A Taxonomy and Survey of GRID Resource Management Systems and Distributed Computing // Software-Practice and Experience, John Wiley & Sons, Ltd. — 2001. — P. 1–10.

4. Менаске Д., Алмейда В. Производительность Web-служб. Анализ, оценка и планирование. — ДиаСофт, 2003. — 480 с.

5. Куссуль Н.Н., Шелестов А.Ю., Лобунец А.Г. Применение методов операционного анализа для оценки производительности GRID-систем// Кибернетика и вычислительная техника. – 2004. – Выпуск 144. – С. 3-20.

6. Дж. Питерсон Теория сетей Петри и моделирование систем. — М.: Мир, 1984. — 264 с.

7. Parikh R. On Context-Free Languages// J. of the ACM, 13, #4, 1966. — PP. 570-581.

8. Кривой С.Л. Критерий совместности систем линейных диофантовых уравнений над множеством натуральных чисел// Доповіді НАНУ, 1999. — № 5. — С. 107-112.

9. Krivoi S. A criteria of Compatibility Systems of Linear Diophantine Constraints // Lecture Notes in Comp. Science. — 2002, № 2328. — С. 264-271.

10. Летичевський О.А. Сучасні проблеми кібернетики. Нормативний курс. Навчальна електронна бібліотека факультету кібернетики Київського національного університету ім. Тараса Шевченка// http://www.unicyb.kiev.ua/Library/.

УДК 621.313.333

АНАЛИЗ РАБОТЫ АСИНХРОННОГО ДВИГАТЕЛЯ ПО ДАННЫМ КАТАЛОГА

Китаев А.И., Глухова В.И.

Разработка системы автоматического управления или электропривода связана с выбором оптимального приводного электрического двигателя. Эта задача решается на основе рассмотрения ряда вариантов построения системы. Причем на этапе проектирования нужных двигателей в наличии может и не быть. Поэтому оценка их функциональных возможностей на основе лабораторных испытаний исключается. И тогда перед разработчиками стоит проблема получения необходимой информации по параметрам схемы замещения и характеристикам двигателя расчетным путем, располагая лишь данными каталога.

Ниже приведено решение такой задачи применительно к асинхронным двигателям (АД), для которых в каталоге оговариваются следующие показатели: номинальное напряжение; номинальная мощность на валу - P_{2H}; перегрузочная способность λ_M или отношение максимального момента M_M к номинальному M_H; коэффициент кратности пускового момента $M_П$ к номинальному с обозначением $\lambda_П$; коэффициент кратности пускового тока $I_{1П}$ к номинальному I_{1H} с обозначением K_I; коэффициент мощности $\cos\varphi_{1H}$ и к.п.д. в номинальном режиме работы η_H.

Попытки решения этой задачи предлагались. Например, в [1] дана методика построения моментной и механической характеристики АД по каталожным данным на основе упрощенной формулы Клосса. Однако автор, предложив свой прием, не сделал акцент на большое расхождение между расчетными и єкспериментальными (каталожными) данными. Например, если критическое скольжение $s_к$ выбрано по условию обеспечения $\lambda_м$, то значения $\lambda_п$ будут отличаться между собой примерно в два раза.

В настоящей работе раскрываются причины этих расхождений и дается методика более развернутого исследования АД с определением параметров схемы замещения, векторной диаграммы и их использованием для расчета рабочих характеристик

Решение построено на использовании метода эквивалентного генератора (МЭГ) и отвечает на основные, сформулированные выше вопросы.

Определение моментной и механической характеристики или зависимостей момента М от скольжения s и частоты вращения ротора n_2.

Как известно [1-3], для расчета момента АД обращаются к исходной формуле:

$$M = \frac{m_1 I_2'^2 R_2'}{\omega_1 s}, \qquad (1)$$

где m_1 - число фаз; I_2' - приведенное значение тока роторной цепи; R_2' - приведенное значение активного сопротивления в цепи ротора; ω_1 - угловая скорость вращения магнитного поля статора.

 ISBN 7-776-8361-7