



ПРОГРАМНО-ТЕХНІЧНІ КОМПЛЕКСИ

С.Л. КРЫВЫЙ, Ю.В. БОЙКО, С.Д. ПОГОРЕЛЫЙ, А.Ф. БОРЕЦКИЙ, Н.Н. ГЛЫБОВЕЦ

УДК 004.75

ПРОЕКТИРОВАНИЕ ГРИД-СТРУКТУР НА ОСНОВЕ ТРАНЗИЦИОННЫХ СИСТЕМ С ОБОСНОВАНИЕМ ПРАВИЛЬНОСТИ ИХ ФУНКЦИОНИРОВАНИЯ

Аннотация. Рассмотрен метод проектирования грид-системы на основе транзисционных систем и их синхронных произведений. Полученная глобальная транзисционная система транслируется в сеть Петри (СП). С помощью СП проверяется корректность проектных решений, в частности отсутствие тупиков, мертвых переходов и т.д.

Ключевые слова: грид-структура, транзисционные системы, диофантовые уравнения.

ВВЕДЕНИЕ

В связи с тем, что сложность программного обеспечения постоянно возрастает, возникает важная проблема построения корректного математического обеспечения и соответствующего оборудования с правильным и надежным функционированием. Эта проблема рассматривалась в работе [1], где предлагались методы проектирования ИТ-инфраструктур. Изложенные методы относились прежде всего к параллельным, распределенным и облачным вычислениям и только частично касались грид-вычислений. Грид-системы в настоящее время являются наиболее востребованными вычислительными структурами в связи с необходимостью выполнять большое количество вычислений высокой сложности на географически распределенных высокопроизводительных комплексах. Грид-системы позволяют пользователю с его рабочего места запускать задания и получать результаты выполнения таких заданий. При этом пользователь может не знать, на каком из доступных ресурсов выполняется его задание. Задача пользователя состоит в том, чтобы сформулировать описание своего задания в соответствии с правилами, выполнение которых позволит получить доступ к вычислительной среде. Контроль за выполнением таких правил возлагается на специализированное промежуточное программное обеспечение (ППО), получившее название middleware. ППО является промежуточным звеном между пользователем и грид-инфраструктурой.

В настоящей статье представлена модель грид-системы, которая абстрагирована от способов трансмиссии управляющих команд ППО, связанных с особенностями вычислителей в структуре, на основе транзисционных систем, к которым относятся и сети Петри (СП).

Структура работы представлена следующим образом. В начале статьи изложен метод проектирования, базирующийся на использовании транзисционных систем и их синхронном произведении. Далее описывается преобразование полученного синхронного произведения в СП. Выполняется анализ свойств сети Петри, которая моделирует всю систему в целом. В процессе проектирования систем

© С.Л. Крывый, Ю.В. Бойко, С.Д. Погорелый, А.Ф. Борецкий, Н.Н. Глыбовец, 2017

определение синхронного произведения выполняется проектировщиком, а использование СП, как исполняемой спецификации, является средством проверки правильности принятых проектных решений. Метод проектирования вначале иллюстрируется на простом примере, а затем применяется к проектированию грид-структуры и обоснованию ее свойств.

ТРАНЗИЦИОННЫЕ СИСТЕМЫ И СЕТИ ПЕТРИ

Введем необходимые определения и понятия.

Определение 1. Транзиционной системой (ТС) называется четверка $A = (S, T, \alpha, \beta, s_0)$, где S — конечное или бесконечное множество, элементы которого называются состояниями; T — конечное или бесконечное множество, элементы которого называются переходами; α и β — два отображения из T в S , ставящие в соответствие каждому переходу $t \in T$ два состояния: $\alpha(t)$ и $\beta(t)$, которые называются соответственно началом и концом перехода t ; $s_0 \in S$ — начальное состояние ТС.

Транзиционная система $A = (S, T, \alpha, \beta, s_0)$ называется конечной, если множества S и T конечны. Далее будем рассматривать только конечные ТС. Они изображаются в виде орграфа, вершинами которого являются состояния ТС, а дугами — переходы.

Функции α и β будем записывать в виде тройки (s, t, s') , где $s, s' \in S, t \in T, \alpha(t) = s, \beta(t) = s'$.

Путем длины n ($n > 0$) в ТС $A = (S, T, \alpha, \beta, s_0)$ называется последовательность переходов t_1, t_2, \dots, t_n таких, что для любого $i, 1 \leq i \leq n$, имеет место равенство $\beta(t_i) = \alpha(t_{i+1})$. Аналогично бесконечным путем называется бесконечная последовательность переходов $t_1, t_2, \dots, t_n, \dots$ таких, что для любого $i \geq 1$ имеет место равенство $\beta(t_i) = \alpha(t_{i+1})$.

Определение 2. Пусть X — некоторый алфавит. Помеченной ТС (ПТС) называется пятерка $A = (S, T, \alpha, \beta, s_0, h)$, где $(S, T, \alpha, \beta, s_0)$ является ТС, h — отображение из T в X , приписывающее каждому переходу t его отметку $h(t) \in X$.

Интуитивно отметка перехода означает действие или событие, которое провоцирует этот переход. ПТС может быть и недетерминированной. В заданном состоянии одно и то же действие может провоцировать два разных перехода в два разных состояния, т.е. из $\alpha(t) = \alpha(t')$ и $h(t) = h(t')$ не следует, что $t = t'$.

Конечный X -автомат (КА) представляет собой ТС. Действительно, пусть $A = (A, X, f, a_0)$ — конечный X -автомат. Тогда ему соответствует ТС $A_T = (A, T, \alpha, \beta, a_0, h)$, где $t \in T$, при этом если $f(a, x) = a'$, то $t = (a, x, a')$ и $h(t) = x$, где $\alpha(t) = a, \beta(t) = a'$.

Произведение ТС. Пусть A_1, \dots, A_n — транзиционные системы, где $A_i = (S_i, T_i, \alpha_i, \beta_i, a_0^i), i = 1, \dots, n$.

Ограничением синхронизации называется подмножество T множества

$$(T_1 \cup \{\varepsilon\}) \times \dots \times (T_n \cup \{\varepsilon\}) \setminus \{(\varepsilon, \dots, \varepsilon)\},$$

где ε — тождественный переход, который означает отсутствие какого-либо действия в ТС. Элементы из T называются глобальными переходами. Если $t = (t_1, \dots, t_n) \in T$ и $t_i \neq \varepsilon$, то считают, что ТС A_i участвует в переходе t .

Кортеж $A = (A_1, \dots, A_n, T)$ называется произведением ТС, а ТС A_1, \dots, A_n называются компонентами A . Интуитивно глобальный переход $t = (t_1, \dots, t_n)$ моделирует возможные переходы в A_1, \dots, A_n . Если переход $t_i = \varepsilon$, то ТС A_i не участвует в глобальном переходе t .

Глобальным состоянием $A = (A_1, A_2, \dots, A_n, T)$ называется n -ка (s_1, s_2, \dots, s_n) , где $s_i \in S_i$, а состояние $(s_0^1, s_0^2, \dots, s_0^n)$ называется начальным состоянием ТС A .

Шагом вычислений A называется тройка (s, t, s') , где $s = (s_1, s_2, \dots, s_n)$ и $s' = (s'_1, s'_2, \dots, s'_n)$ — глобальные состояния, а $t = (t_1, t_2, \dots, t_n)$ — глобальный переход, который удовлетворяет $\forall i \in \{1, 2, \dots, n\}$ таким условиям:

- если $t_i \neq \varepsilon$, то $s_i = \alpha(t_i)$ и $s'_i = \beta(t_i)$;
- если $t_i = \varepsilon$, то $s'_i = s_i$.

Глобальный переход t называется допустимым в глобальном состоянии s , если существует глобальное состояние s' такое, что (s, t, s') является шагом вычисления. Последовательность глобальных переходов $t_1, t_2, \dots, t_k, \dots$ называется глобальным вычислением, если существует последовательность глобальных состояний $s_0, s_1, \dots, s_k, \dots$ такая, что (s_{i-1}, t_i, s_i) — шаг вычислений для каждого $i \in \{1, 2, \dots, k\}$.

Многие свойства произведения ТС можно исследовать моделированием сетями Петри.

Произведение транзиторных систем и сети Петри. Сеть Петри (P, T, F, M_0) моделирует произведение $A = (A_1, A_2, \dots, A_n, T)$ транзиторных систем $A_i = (S_i, T_i, \alpha_i, \beta_i, s_0^i)$, где $A_i \cap A_j = \emptyset$ при $i \neq j$, $i, j = 1, 2, \dots, n$, если $P = S_1 \cup \dots \cup S_n$; $T = T$, $F = \{(s, t) \mid t_i \neq \varepsilon, s = \alpha_i(t_i)\} \cup \{(t, s) \mid t_i \neq \varepsilon \text{ и } s = \beta_i(t_i)\}$ для некоторого $i \in \{1, 2, \dots, n\}$, где t_i означает i -ю компоненту $t \in T$; $M_0 = (s_0^1, s_0^2, \dots, s_0^n)$.

Нетрудно заметить, что семантика произведения ТС и семантика СП, которая его моделирует, согласованы, т.е. последовательность глобальных переходов $t_1 t_2 \dots t_k$ представляет собой глобальную историю произведения ТС A тогда и только тогда, когда она является допустимой последовательностью срабатываний переходов в СП. Представление произведения ТС в виде СП позволяет применить методы анализа свойств СП для анализа свойств произведения ТС.

Моделирование произведения X -автоматов с помощью произведения ТС выполняется за два этапа. На первом этапе автоматы моделируются соответствующими ТС, а на втором — произведением полученных ТС. Затем такая ТС моделируется соответствующей СП.

Проиллюстрируем сказанное на примере.

Пример 1. Пусть имеем X -автоматы A_1 (рис. 1, а) и A_2 (рис. 1, б) и соответствующие ТС T_1 и T_2 для автоматов A_1 (рис. 2, а) и A_2 (рис. 2, б). В этих ТС имеем множества переходов:

$$t_1 = (0, i, 1), \quad t_2 = (1, i, 2), \quad t_3 = (2, d, 1), \quad t_4 = (1, d, 0), \quad t_5 = (0, d, 2), \quad t_6 = (2, i, 0),$$

$$u_1 = (a, f, a), \quad u_2 = (a, t, b), \quad u_3 = (b, t, b), \quad u_4 = (b, f, a).$$

Предположим, что проектировщик определил ограничения синхронизации

$$T = \{(t_1, \varepsilon), (t_2, \varepsilon), (t_3, u_2), (t_4, u_4), (t_5, \varepsilon), (t_6, u_1), (\varepsilon, u_3)\}.$$

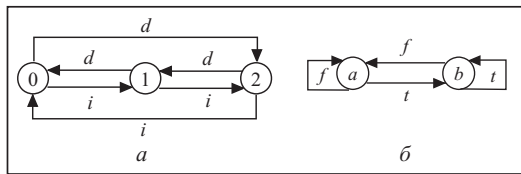


Рис. 1

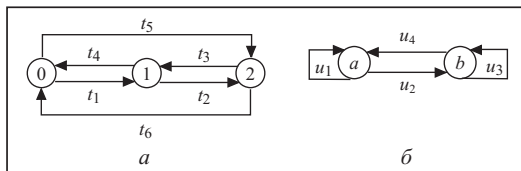


Рис. 2

Тогда произведение ТС определяется как $A = (T_1, T_2, T)$. Соответствующая СП, моделирующая синхронное произведение A , представлена на рис. 3.

Пусть начальная разметка СП ставит в места 0 и а по одной фишке. Не является очевидным то, что при этой разметке СП имеет тупик, к которому приводит последовательность глобальных переходов $(t_5, \varepsilon), (t_3, u_2), (t_2, \varepsilon)$.

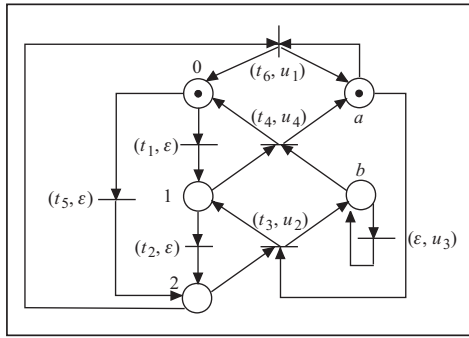


Рис. 3

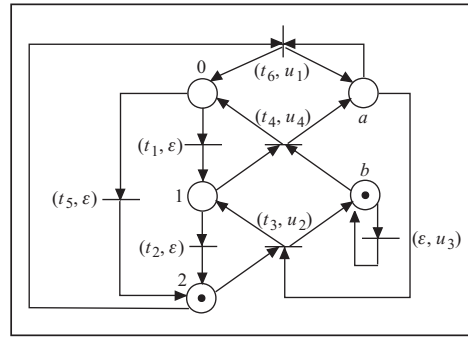


Рис. 4

Действительно, эта последовательность срабатываний переходов приводит к разметке СП, показанной на рис. 4, при которой в СП невозможен ни один переход. Следовательно, автоматы попадают в состояние $(2, b)$, из которого в T не существует переходов. Наличие тупика в СП означает ошибку разработчика, и важным является обнаружение подобного типа ошибок на этапе проектирования.

Ситуации такого типа идентифицируются в СП, чем и обусловлено их использование.

РАЗРАБОТКА ГРИД-СТРУКТУРЫ

Работа пользователя в грид-системе состоит в следующем. Пользователь посылает запрос к ППО на регистрацию в грид-системе, получает персональный цифровой сертификат, регистрируется в виртуальной организации (ВО), подготавливает и запускает свои задания. Подготовка заданий состоит в формировании файлов-описаний заданий, подготовки входных данных, создания прокси сертификата. После этих подготовительных действий задания отправляются в грид и в зависимости от загруженности ресурсов выполняются или ставятся в очередь на выполнение [2, 3, 5, 6].

Основные требования при разработке грид-системы — корректность ее функционирования, контроль за правильностью выполнения условий работы в системе, правильность вычислений и возврат результатов выполнения задания пользователю в случае их успешного завершения. Чтобы в грид-системе эти требования удовлетворялись, необходимо прибегнуть к формальным методам проверки их выполнения, что обуславливает построение соответствующей математической модели, на которой и выполняется эта проверка. Поскольку построить модель грид-системы с учетом всех присущих ей особенностей практически невозможно ввиду ее большой сложности, то такое построение выполняется на некотором уровне абстракции.

Таким образом, работа в грид-системе начинается с выполнения двух протоколов: регистрации в грид-системе и подготовки задания. Выполнение действий, предусмотренных этими протоколами, одноразовые. Это значит, что если протоколы регистрации и подготовки задания выполнены успешно, то больше никакие действия с ними не выполняются. Если же действия по регистрации по каким-либо причинам не выполнены, то об этом сообщается пользователю, и он повторяет попытку регистрации. Представим действия обоих протоколов в виде транзитивных систем [1, 4], которые достаточно очевидны в силу своей простоты.

1. Протокол регистрации в грид-системе. Действия этого протокола состоят в следующем. Прежде чем получить доступ к вычислительным ресурсам, необходимо зарегистрироваться в системе и получить сертификат. С этой целью

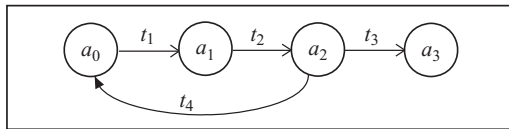


Рис. 5

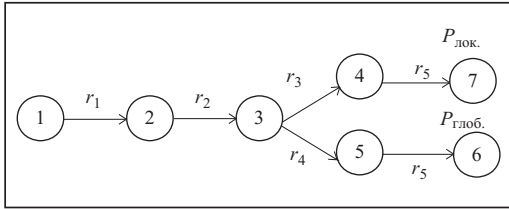


Рис. 6

пользователь со своего рабочего места посылает запрос на регистрацию в гриде, получает подписанный доверенным центром сертификат и тем самым регистрируется в системе. Эти действия моделируются простой ТС $A_1 = (\{a_0, a_1, a_2, a_3\}, \{t_1, t_2, t_3, t_4\}, \alpha_1, \beta_1, a_0)$ (рис. 5).

Интерпретация мест и переходов в данной ТС следующая: a_0 — формирование запроса (генерация сертификата пользователем); t_1 — запрос на регистрацию (отправка сертификата в доверенный центр), где $\alpha_1(t_1) = a_0, \beta_1(t_1) = a_1$; a_1 —

обработка запроса (проверка пользователя); t_2 — процесс регистрации (подпись сертификата), где $\alpha_1(t_2) = a_1, \beta_1(t_2) = a_2$; a_2 — анализ процесса регистрации; t_3 — регистрация выполнена, где $\alpha_1(t_3) = a_3, \beta_1(t_3) = a_4$; a_3 — регистрация пользователя (получение пользователем подписанного сертификата), t_4 — регистрация не выполнена (отказ в подписи), где $\alpha_1(t_4) = a_2, \beta_1(t_4) = a_0$.

Если регистрация выполнена успешно (состояние a_3 в ТС), то система регистрации информирует пользователя и ППО о легальном пользователе. Если регистрация по каким-либо причинам в системе не выполнена, то в состоянии a_2 система регистрации выполняет действие t_4 , которое информирует пользователя о том, что он не прошел регистрации в системе.

Вторым этапом работы в грид-системе есть получение членства в одной или нескольких доступных виртуальных организациях. В рамках данной статьи рассматривается ситуация членства по умолчанию в одной из ВО.

2. Протокол подготовки задания. Пройдя успешно регистрацию и получив цифровой сертификат доступа к грид-структуре, пользователь приступает к выполнению протокола подготовки своего задания. При этом предусматривается следующая последовательность действий: а) создать файл-описание задания; б) подготовить данные; в) определить место расположения данных (загрузка с локальных носителей или из грид-хранилищ); г) создать прокси сертификат с необходимыми атрибутами доступа.

Протокол моделируется транзитивной системой $A_2 = (\{1, 2, 3, 4, 5, 6, 7\}, \{r_1, r_2, r_3, r_4, r_5\}, \alpha_2, \beta_2, 1)$ (рис. 6).

Интерпретация мест и переходов в данной ТС такова: 1 — старт формирования задания, r_1 — создание файла-описания задания, 2 — старт подготовки данных, файл-описания создан, r_2 — подготовка данных, 3 — анализ расположения данных, r_3 — подготовка данных на локальных носителях, 4 — локальные данные готовы, r_4 — подготовка данных из грид-хранилищ, 5 — данные из хранилищ готовы, r_5 — создание прокси сертификата с необходимыми атрибутами доступа, 6 — состояние готовности данных из глобальных грид-хранилищ, 7 — состояние готовности данных с локальных носителей.

При переходе в состояние 6 ТС должна информировать пользователя о готовности задания и ППО об этой готовности. С этой целью вводятся дополнительные состояния в модели, представляющей синхронное произведение приведенных ТС, в виде сети Петри.

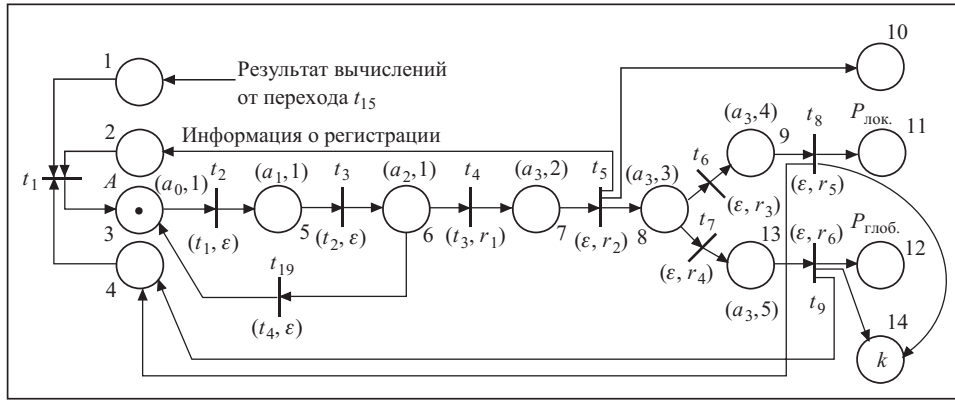


Рис. 7

СЕТЕВАЯ МОДЕЛЬ ПРОТОКОЛОВ РЕГИСТРАЦИИ И ПОДГОТОВКИ ЗАДАНИЯ

Построим модель регистрации и подготовки задания в виде синхронного произведения двух ТС: $A_1 * A_2$ с глобальными переходами:

$$T = \{(t_1, \varepsilon), (t_2, \varepsilon), (t_3, r_1), (t_4, \varepsilon), (\varepsilon, r_2), (\varepsilon, r_3), (\varepsilon, r_4), (\varepsilon, r_5), (\varepsilon, r_6)\}.$$

По этой ТС строим сеть Петри, которая моделирует синхронное произведение ТС обеих систем (рис. 7). В этой СП кроме мест синхронного произведения вышеприведенных ТС появляются места, полученные в результате выполнения синхронного произведения данных ТС, а также ТС, которая моделирует вычислительную среду (по местам 10, 11, 12 и 14 соединяются обе сети). ТС вычислительной среды и синхронное произведение соответствующих ТС ввиду громоздкости не приводятся, а отдельные моменты такого процесса продемонстрированы рассмотренным выше примером. Ниже представлен окончательный результат проектирования в виде сети Петри с описанием мест и переходов в ней:

- место 3 соответствует пользователю A , который инициализирует (ставит фишку в это место) начало работы с грид-системой;
- место 4 информирует пользователя о получении сертификата и готовности задания к выполнению;
- места 2 и 10 информируют пользователя и ППО соответственно об успешном прохождении процесса регистрации и получении сертификата;
- место 11 ($P_{\text{лок.}}$) означает локальность данных, а место 12 ($P_{\text{глоб.}}$) означает, что данные на носителях находятся в хранилищах;
- место 14 означает, что создан прокси сертификат с параметром k .

МОДЕЛЬ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

При построении модели вычислительной среды (ВС) грид-системы будем предполагать, что все вычислительные средства ранжированы числами k_1, k_2, \dots, k_m , где m — количество вычислителей, k_i — параметры вычислителя i . Опишем вновь появившиеся места:

- места 15, 16, ..., s — хранилища данных (D_1, D_2, \dots, D_s);
- место 17 — найден вычислитель с нужными параметрами (места 22, 23) (вычислитель с параметрами k_1 (место 24) не имеет требуемых параметров ($k_1 < k$));
- места 20, 21, 22, 23, 24, 25 соответствуют вычислителю и его занятости;
- место 18 — данные загружены;
- место 19 — вычисления;
- место 27 — состояние аварии;
- место 26 — состояние восстановления работоспособности кластера.

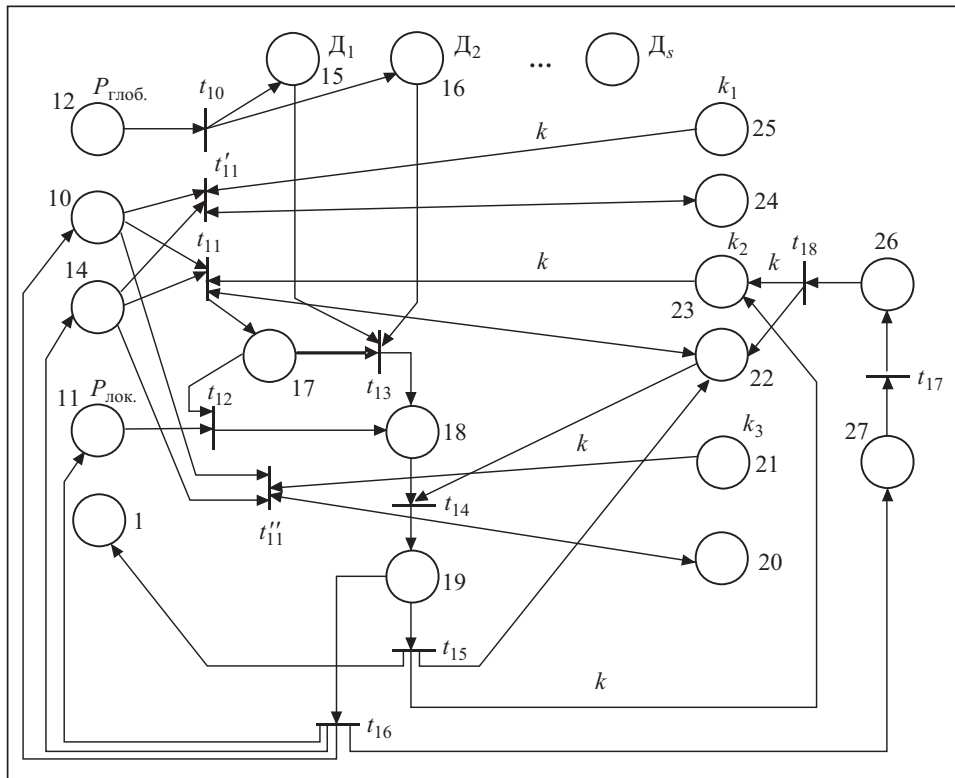


Рис. 8

Переходы: t_{10} — инициализация требуемых данных на носителях; t_{11} — поиск вычислителя; t_{12} — инициализация и загрузка локальных данных; t_{13} — загрузка данных из хранилищ; t_{14} — запуск вычислений; t_{15} — окончание вычислений; t_{16} — инициализация аварийного состояния и возобновление параметров задания и данных в целях поиска другого исправного вычислителя с нужными параметрами и возобновление вычислений; t_{17} — инициализация возобновления работоспособности кластера; t_{18} — восстановление работоспособности кластера.

В приведенной СП переходы, подобные переходу t_{12} , срабатывают при условии исправности вычислителя (отсутствие фишки в месте 27), его незанятости (наличие фишки в месте 22) и наличия требуемых параметров (в месте 23 находится $k < k_2$ фишек). В соответствии с этими условиями выбирается вычислитель с требуемыми параметрами. В случае занятости или неисправности вычислителя ищется другой незанятый и исправный вычислитель с нужными параметрами, на котором и выполняется задание пользователя (в приведенном примере СП переход t'_{11} не срабатывает, поскольку $k_1 < k$). Если вычислитель с требуемыми параметрами в данный момент отсутствует, то задание пользователя ждет своей очереди.

Сетевая модель грид-системы, состоящая из трех кластеров с элементами, представлена на рис. 8.

ОБОСНОВАНИЕ СЕТЕВОЙ МОДЕЛИ ГРИД-СИСТЕМЫ

Построив сеть Петри, которая моделирует грид-систему, получаем выполняемую спецификацию данного проекта такой системы, которая является при этом и математической моделью проектируемой грид-структуры. Теперь задача состоит в том, чтобы проверить корректность построенной модели. Первое задание сводится к тому, чтобы убедиться в живучести такой системы, т.е. что в построенной модели все переходы принимают участие в процессе ее функци-

онирования (все переходы живые). Если некоторые переходы в СП не срабатывают, то это свидетельствует о том, что в проекте системы что-то выполнено неправильно или данные переходы избыточны. Это относится и к местам СП. Если некоторые места ввиду наличия мертвых переходов недостижимы в процессе функционирования СП, то это также свидетельствует о дефектах проекта.

Проверка живучести СП выполняется двумя способами: с помощью построения графа достижимых разметок СП и путем анализа инвариантов переходов. Построение множества инвариантов переходов сводится к решению уравнения состояния в множестве натуральных чисел вида $Ax = 0$, где A — матрица инцидентности СП, а x — вектор, значения координат которого равны количеству срабатываний соответствующих переходов. Если некоторая координата во всех векторах решения равна нулю, то это значит, что данный переход не срабатывает в процессе функционирования СП (переход мертвый), какова бы ни была начальная разметка системы.

Для данной СП, где предполагается наличие трех кластеров-вычислителей ($n = 3$), уравнение состояния имеет матрицу размера 19×27 , поскольку в СП имеется 19 переходов и 27 мест:

		Номер перехода																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Номер места СП	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
	2	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	4	-1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	5	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
	7	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	1	0	0	0	0
	11	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	3	3	0	-3	0	0	0	0	3	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	0	0	0	0	0	0
	19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0
	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	22	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1	0	0
	23	0	0	0	0	0	0	0	0	0	0	-3	0	0	0	3	0	0	3	0	0
	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0

Применяя TSS-алгоритм [4] для решения уравнения состояния с вышеприведенной матрицей, получаем такие его решения, которые являются инвариантами переходов:

		Номер перехода																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	0	0
	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	0	0	0

Как видно из приведенных решений, переходы 16, 17, 18 являются мертвыми, поскольку ни разу не срабатывают в процессе функционирования СП. Эти переходы связаны с аварийным состоянием вычислителя, а это значит, что в случае аварии вычислителя возобновления вычислений происходить не будет. Причина, по которой эти переходы не срабатывают, связана с необходимостью восстановления параметров вычислителя и восстановления данных задания. Иными словами, в проекте грид-структуры, описываемой данной математической моделью, необходимы действия по восстановлению как параметров пользовательской задачи, так и ее данных.

Коррекция СП связана с передачей информации о возобновлении чтения данных (что означает посылку в место 12 или в место 11 фишки в зависимости от места нахождения этих данных). Коррекция приводит к уравнению состояния с такой матрицей A :

		Номер перехода																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Номер места СП	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	2	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	4	-1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
	5	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
	7	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	1	0	0	0
	11	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	1	0	0	0
	12	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	3	3	0	-3	0	0	0	0	3	0	0	0
	15	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	0	0	0	0	0
	19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0
	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	22	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	1	0
	23	0	0	0	0	0	0	0	0	0	0	-3	0	0	0	3	0	0	3	0
	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0

В данной ситуации эта матрица соответствует случаю восстановления локальных данных, о чем свидетельствуют решения уравнения состояния с приведенной матрицей:

$$\begin{aligned}
x_1 &= (0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1), \\
x_2 &= (1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0), \\
x_3 &= (1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0), \\
x_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0).
\end{aligned}$$

Как видим, все переходы в такой сети живые, что свидетельствует о живости всей СП в целом. Первый инвариант описывает ситуацию, когда регистрация в грид-системе не выполнялась (срабатывают переходы 2, 3 и 19). Второй инвариант описывает ситуацию успешной регистрации, получения сертификата, локальность данных и успешное выполнение вычислений (срабатывают переходы 2, 3, 4, 5, 6, 8, 11, 12, 14, 15 и 1). Третий инвариант описывает ту же ситуацию, но с данными на носителях (срабатывают переходы 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15 и 1). Четвертый инвариант описывает аварийную ситуацию и показывает, что повторной регистрации и получения сертификата не требуется, а необходимо только снова восстановить параметры и повторно загрузить локальные данные (срабатывают переходы 11, 12, 14, 16, 17, 18). Если восстанавливаются данные с носителей, то инварианты имеют вид

$$\begin{aligned}
y_1 &= (0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1), \\
y_2 &= (1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0), \\
y_3 &= (1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0), \\
y_4 &= (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0).
\end{aligned}$$

Восстановлению подлежат параметры вычислителя и повторная загрузка данных с носителей. Таким образом, первое свойство — свойство живости — в этой модели выполняется. Отсюда следует, что в данной модели выполняется свойство справедливости, которое состоит в том, что если задание находится в грид-системе, то оно рано или поздно будет выполняться. Это следует очевидным образом из того, что соответствующие переходы, ведущие к свободным вычислителям, будут выполняться точно таким же образом, как и в приведенном примере.

Полученные решения уравнения состояния позволяют сформулировать такое утверждение.

Теорема 1. СП, моделирующая грид-систему, является живой, и в ней выполняется свойство справедливости.

СВОЙСТВО ОГРАНИЧЕННОСТИ И СПРАВЕДЛИВОСТИ

Рассмотрим выполнимость свойства ограниченности СП, которое подтверждает выполнимость свойства справедливости. Свойство ограниченности означает, что в СП не существует мест, в которых фишки могут накапливаться в неограниченном количестве. Такая ситуация соответствует случаю, когда в грид-системе постоянно что-то происходит, но при этом ничего не вычисляется (система циклит).

Для исследования этого свойства сети Петри необходимо решить систему линейных однородных диофантовых уравнений в множестве натуральных чисел вида $A^T y = 0$, где A^T — матрица, транспонированная к матрице инцидентности A . Решения такой системы называются инвариантами мест, и если все места в СП покрываются положительными значениями, то СП ограничена. В данном примере транспонированная матрица имеет следующий вид:

Номер места СП

		Номер места СП																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
Номер перехода	1	-1	-1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	1	0	0	0	0	-1	0	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	1	0	0	0	0	0	0	0	1	-1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	-1	0	0	0	-3	0	0	1	0	0	0	0	0	0	-3	0	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	1	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	-1	0	0	0	0	0	0
	15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	3	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	1	1	0	0	3	0	0	0	-1	0	0	0	0	0	0	0	0	1
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0	-1	0
	19	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Решения системы $A^T y = 0$ имеют вид:

		Номер перехода																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
$p_1 =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0)
$p_2 =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0)
$p_3 =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0)
$p_4 =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0)
$p_5 =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0)
$p_6 =$		(0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0)
$p_7 =$		(1	0	1	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0)
$p_8 =$		(1	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0)
$p_9 =$		(1	0	1	0	1	1	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0)
$p_{10} =$		(3	0	3	0	3	3	3	3	0	0	0	3	1	0	0	3	3	3	0	0	0	0	0	0	0	0	0)
$p_{11} =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0)	
$p_{12} =$		(0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	0	0	0	1	0	0	3	3)	

Из полученных решений очевидным образом имеем ограниченность СП, поскольку все ее места покрываются положительными значениями. Отсюда вытекает следующее утверждение.

Теорема 2. СП, моделирующая грид-систему, ограничена и в ней выполняется свойство справедливости.

ЗАКЛЮЧЕНИЕ

Новые информационно-технологические инфраструктуры, такие как грид-системы, облачные вычисления, позволили с новых позиций рассмотреть различные аспекты коллективной работы научных сообществ, что привело к значительному усложнению суперкомпьютерных вычислений. Для организации корректной коллективной работы необходимо создавать средства формализованного проектирования новых приложений для названных сред.

В статье использован математический аппарат транзиторных систем для формирования совокупности протоколов взаимодействия с грид-системами: регистрации в грид-системе; подготовки задания; регистрации и подготовки задания. Кроме того, разработана адекватная модель суперкомпьютерной вычислительной среды в виде СП, что позволило строго доказать свойства ограниченности, живучести и справедливости проектируемой среды.

СПИСОК ЛІТЕРАТУРИ

1. Бойко Ю.В., Волохов В.М., Глибовець М.М., Єршов С.В., Кривий С.Л., Погорілий С.Д., Ролік О.І., Теленик С.Ф., Ясочка М.В. Методи та новітні підходи до проектування, управління і застосування високопродуктивних ІТ-інфраструктур. За ред. А.В. Анісімова. Київ: Київ. нац. ун-т ім. Тараса Шевченка, 2016. 448 с.
2. Бойко Ю.В., Зинов'єв М.Г., Свістунів С.Я., Судаков О.О. Український академічний грід: досвід створення і перші результати експлуатації. *Математические машины и системы*. 2008. Т. 1, № 1. С. 67–84.
3. Слюсар Е.А. Автоматизированная служба репликации файлов для организации высокой доступности данных в грид-инфраструктуре. *УСиМ*. 2012. № 4 (240). С. 63–74.
4. Кривий С.Л. Лінійні діофантові обмеження та їх застосування. Чернівці: Букрек, 2015. 224 с.
5. Погорілий С.Д., Бойко Ю.В., Левченко Р.І., Мар'яновський В.А. Методи кластерних обчислень. За ред. С.Д. Погорілого. Київ, 2013. 416 с.
6. Судаков О.О., Бойко Ю.В. GRID ресурси інформаційно-обчислювального центру Київського національного університету імені Тараса Шевченка. *Проблеми програмування*. Матеріали V міжнародної науково-практичної конференції з програмування УкрПРОГ'2006 (Київ, 23–25 травня 2006 р.). Київ, 2006. № 2–3. С. 165–169.

Надійшла до редакції 04.04.2016

С.Л. Кривий, Ю.В. Бойко, С.Д. Погорілий, О.Ф. Борецький, М.М. Глибовець ПРОЕКТУВАННЯ ГРІД-СТРУКТУР НА ОСНОВІ ТРАНЗИЦІЙНИХ СИСТЕМ З ОБГРУНТУВАННЯМ ПРАВИЛЬНОСТІ ЇХ ФУНКЦІОНУВАННЯ

Анотація. Розглянуто метод проектування грід-системи на основі транзиційних систем та їхніх синхронних добутоків. Отримана глобальна транзиційна система транслюється в мережу Петрі (МП). За допомогою МП перевіряють коректність проектних рішень, зокрема відсутність тупиків, мертвих переходів тощо.

Ключові слова: грід-структура, транзиційні системи, діофантові рівняння.

S.L. Krivyyi, Y.V. Boyko, S.D. Pogorilyy, O.F. Boretskyi, M.M. Glybovets DESIGN OF GRID STRUCTURES BASED ON TRANSITION SYSTEMS WITH JUSTIFICATION OF THE CORRECTNESS OF THEIR FUNCTIONING

Abstract. The authors consider the method of designing grid systems based on transition systems and their products. The resulting global transition system is translated into a Petri net, which is used to verify the correctness of the design decisions.

Keywords: grid-structure, transition systems, Diophantine equations.

Кривый Сергей Лукьянович,

доктор физ.-мат. наук, профессор Киевского национального университета имени Тараса Шевченко, e-mail: sl.krivoi@gmail.com.

Бойко Юрий Владимирович,

кандидат физ.-мат. наук, доцент Киевского национального университета имени Тараса Шевченко, e-mail: boyko@univ.net.ua.

Погорельый Сергей Демьянович,

доктор техн. наук, профессор, заведующий кафедрой Киевского национального университета имени Тараса Шевченко, email: sdr@univ.net.ua.

Борецкий Александр Францевич,

ассистент кафедры Киевского национального университета имени Тараса Шевченко, e-mail: o.boretskyi@knu.ua.

Глибовец Николай Николаевич,

доктор физ.-мат. наук, профессор, декан Национального университета «Киево-Могилянская академия», e-mail: glib@ukma.kiev.ua.