Proceedings of IAM, V.6, N.1, 2017, pp.54-73

DESIGN OF GRID SERVICES FOR RUNNING VIRTUAL MACHINES AS COMPUTING TASK BASED ON TRANSITION SYSTEMS

S.L. Kryvyi[1], YU.V. Boyko[2], S.D. Pogorilyy[2], O.F. Boretskyi[2], M.M. Glibovets[3]

[1]Faculty of Computer Science and Cybernetics of Taras Shevchenko National University of Kyiv, Ukraine
[2]Faculty of Radiophysics, Electronics and Computer Systems of Taras Shevchenko National University of Kyiv, Ukraine
[3] Faculty Of Computer Sciences of National University of Kyiv-Mohyla Academy, Ukraine
e-mail: grid@grid.org.ua

Abstract. Proposed to apply method of grid system design on grid services for running virtual machines as regular computing task. The resulting global transition system is translated into a Petri net (PN). With the help of the PN is checked correctness of design, in particular the absence of deadlocks, dead transitions etc.

Keywords: grid, grid service, virtual machine, transition systems, Petri network, Diophantine equation

AMS Subject Classification: 11D04

1. Introduction

An important problem of mathematical correct software and related equipment that will function properly and safely considered in [6], which proposed a number of methods of designing IT infrastructure. The methods in this monograph is primarily concerned parallel, distributed and cloud computing and only partially considered distributed computing. Grid system today is one of the most popular computational structures that have arisen in connection with the need to perform a large number of high computational complexity geographically distributed high-performance systems[9]. Architecture computing grid [3] has a number of features which lead to the use of the application software only compatible with Linux operating system, the ability to use only the version that is installed on a particular cluster, grid computing system, and lack of access to interactive tasks.

Bypass restrictions listed were invited by running virtual machines(VM) as grid computing tasks [1] and even build PaaS service based on it [10]. Because end users are not experts in IT are developed for specialized virtual laboratory [5] consisting of grid services which work with the grid. These grid services developed as separate and independent components to be integrated into other virtual laboratory. To add a new grid service that would carry out technical tasks to run as a virtual machine grid task raises the question of constructing a mathematical model of this grid service to its correct design.

In previous work were asked to simulation grid structure with transition systems of simultaneous product and Petri Nets [4]. The result of the simulation showed the use of the simulation as a method of designing a correct mathematical software in the grid. This paper proposes a method to use the design grid of service launch virtual machines as usual computing tasks using transition systems of simultaneous product and Petri Nets.

2. Transition Systems and Petri Net

Let's introduce the necessary definitions and concepts [2].

Definition 2.1. Transition system (TS) is 5-tuple $A = (S, T, \alpha, \beta, s_0)$, where

S - finite or infinite set of states,
T - finite or infinite set of transitions,
$\alpha : T \to S$ associates with each transition its source state,
$\beta : T \to S$ associates with each transition its target state,
$s_0 \in S$ - initial state of the TS.

TS $A = (S, T, \alpha, \beta, s_0)$ called finite if the sets S and T finite.

Only the final TS will be considered further. The final TS shown as oriented graph, whose vertices are the states of TS, and arcs - transitions.
Given transition t is represented by the triple $(\alpha(t), t, \beta(t)) = (s, t, s')$, a step of TS A.
Path in TS $A = (S, T, \alpha, \beta, s_0)$ is a finite or infinite sequence of transitions $t_1, t_2, \ldots, t_n, \ldots$ such that for any $i = 1, 2...$ there is such equality $\beta(t_i) = \alpha(t_{i+1})$. If path $t_1, t_2, \ldots, t_n$ is finite, then number n is considered its length.

Definition 2.2. Let X is some alphabet. Marked TS (MTS) is 6-tuple $A = (S, T, \alpha, \beta, s_0, h)$, where

$(S, T, \alpha, \beta, s_0)$ represents a TS,
$h : T \to X$ associates a mark $h(t) \in X$ with each transition t.

Intuitively, the transition symbol indicates an action or event that triggers the transition. MTS can be non-deterministic. The predetermined state is the same effect can induce two different transition in two different states, i.e. from

$$\alpha(t) = \alpha(t') \text{ and } h(t) = h(t')$$

it does not imply that $t = t'$.

Finite-state *X*-machines and TS. Finite-state *X*-machine (FSM) represents TS. Really, let $A = (A, X, f, a_0)$ - Finite-state *X*-machine. Then is represents a TS

$A_T = (A, T, \alpha, \beta, a_0, h)$, where $t \in T$, $\alpha(t) = a, \beta(t) = a'$, corresponds to A if $h(t) = x$ and $f(a, x) = a'$ is the transition in A.

Composition of TS. Let $A_1, \ldots, A_n$ - transition systems, where $A_i = (S_i, T_i, \alpha_i, \beta_i, a_0^i)$, $i = 1, \ldots, n$. *A* synchronization constraint **T** is a subset of a set

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

where ε is a special symbol, which means the absence of any action in corresponding TS. If $t = (t_1, \ldots, t_n) \in$ and $t_i \neq \varepsilon$ then we say that TS A_i participates in t and if the transition $t_i = \varepsilon$, then TS A_i does not participate in t.

Elements of **T** are called global transitions. Tuple $A = (A_1, \ldots, A_n, T)$ is called the composition of $A_1, \ldots, A_n$ under **T**, and TS $A_1, \ldots, A_n$ are called components of *A*. Intuitively global transition $t = (t_1, \ldots, t_n)$ models the possible transitions in $A_1, \ldots, A_n$.

Global state $A = (A_1, A_2, \ldots, A_n, T)$ called n-of $(s_1, s_2, \ldots, s_n)$, where $s_i \in S_i$, and state $(s_0^1, s_0^2, \ldots, s_0^n)$ - TS initial state of A.

Step of computation A is a triple (s, t, s'), where $s = (s_1, s_2, \ldots, s_n)$ and $s' = (s_1', s_2', \ldots, s_n')$ - global states, and $t = (t_1, t_2, \ldots \ldots, t_n)$ - global transition, which satisfies the following conditions: $\forall i \in \{1, 2, \ldots, n\}$

- if $t_i \neq \varepsilon$, then $s_i = \alpha(t_i)$ and $s_i' = \beta(t_i)$;
- if $t_i = \varepsilon$, then $s_i' = s_i$.

Global transition t called to be admissible in the global state s, if there is a global state s' such that (s, t, s') is step of computation.

The sequence of global transitions $t_1, t_2, \ldots, t_k, \ldots$ called the global computation, if there is a sequence of global states s_0, $s_1, \ldots, s_k$ ..., such that (s_{i-1}, t_i, s_i) - step of computation for each $i \in \{1, 2, \ldots, k\}$.

Many properties of the TS transition can be explored through its simulation by Petri nets (PN).

PN (P, T, F, M_0) models composition $A = (A_1, A_2, \ldots A_n, T)$ of TS $A_i = (S_i, T_i, \alpha_i, \beta_i, s_0^i)$, where $A_i \cap A_j = \varnothing$, $i \neq j$ $i, j = 1, 2, \ldots, n$, if

- $P = S_1 \cup S_2 \cup \ldots \cup S_n$, $T = \boldsymbol{T}$, $F = \{(s,t) \mid t_i \neq \varepsilon$ and $s = \alpha_i(t_i)\} \cup \{(t,s) \mid t_i \neq \varepsilon$ and $s = \beta_i(t_i)\}$ for some $i \in \{1,2,\ldots,n\}$, where t_i means i -th component of $t \in \mathbf{T}$,
- $M_0 = (s_0^1, s_0^2, \ldots, s_0^n)$.

It is easily to see the TS composition semantics and semantics corresponding PN, are compatible in the sense that the sequence of global transitions $t_1, t_2 \ldots t_k$ is global history of TS composition A.

Representation of TS composition in form of PN allows to apply methods for analyzing PN properties to analysis of TS composition properties.

X-machines composition. Modeling of *X*-machines composition with help of TS composition is carried out in two stages. At the first stage finite-state machines are modeled by the relevant TS, and on second - composition of received TS. And then such TS is modeled by corresponding PN.

Let us illustrate this with an example.

Example 2.1. Let there be given X-machines:

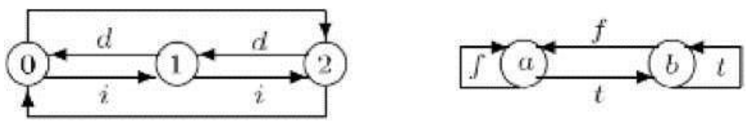

Figure 1. X-machines A1 and A2

Corresponding TS T1 and T2 for machines A1 and A2 have the form:

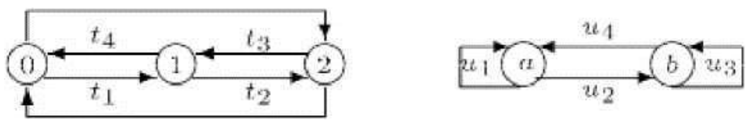

Figure 2. TS corresponding X-machines A1 and A2

In this TS set of transition is represented like this:

$$t_1 = (0,i,1), t_2 = (1,i,2), t_3 = (2,d,1), t_4 = (1,d,0), t_5 = (0,d,2), t_6 = (2,i,0),$$
$$u_1 = (a,f,a), u_2 = (a,t,b), u_3 = (b,t,b), u_4 = (b,f,a).$$

Let the designer has defined these synchronization constraints:

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

Then composition TS will be the $A = (T_1, T_2, \boldsymbol{T})$, and corresponding PN looks like:

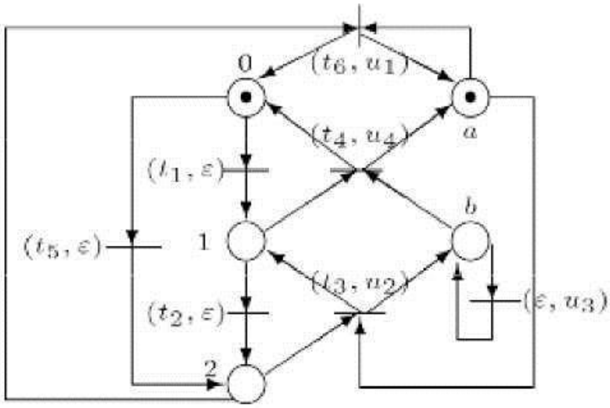

Figure 3. PN modeling synchronous composition of A

Let the initial markup of PN puts one token in places 0 a. It is not clear that in this marking PN has a deadlock which results in a sequence of global transitions: $(t_5, \varepsilon), (t_3, u_2), (t_2, \varepsilon)$.

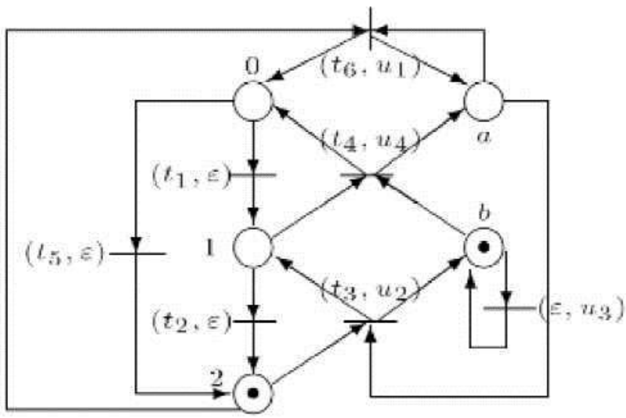

Figure 4. Markup, which is deadlock in PN

Indeed, this sequence of operations leads to the transitions marking PN, which is shown in Fig. 4, in which the PN can not have any transition. Consequently, the machines fall into a state of $(2, b)$, from which T there are no transitions. Presence of deadlock PN is caused by a human mistake and it is very important to detect this type of error in the design phase.

These types of situations are identified in the PN representation of TS, which explains its usage.

3. Tools to analyze the property of Petri Net

One of the main tools for analyzing the PN is to solve the equation of state, i.e., solution of linear homogeneous Diophantine equations of the form *Ax=0*, where *A* - incidence matrix of the PN, and *x* - transitions response vector (Parik vector). Solutions of such a system are called invariants of transitions and their existence for the PN has the specificity, which consists in the fact that you need to find is not the basis of set of solutions, and generating a minimum set of solutions. This minimal set is the set of invariants of PN transition. For the construction of this set in this paper we apply TSS-algorithm [10], which find this set.

3.1. Brief description of TSS-algorithm.

Consider the set of vectors of the canonical basis $M_0' = \{e_1, e_2, ..., e_q\}$ and the first equation $L_1(x) = a_{11}x_1 + a_{12}x_2 + ... + a_{1q}x_q = 0$ of system S. With the function $L_1(x)$ divide the elements of the set M_0' on such groups: $M_1^0 = \{e \mid L_1(e) = 0\}$, $M_1^+ = \{e \mid L_1(e) > 0\}$ and $M_1^- = \{e \mid L_1(e) < 0\}$. It is easy to note that when one of the sets - $M_1^0 \cup M_1^+$ or $M_1^0 \cup M_1^-$ - empty then equation $L_1(x) = 0$ it has no nontrivial solutions in the set of natural numbers. Suppose that at least two sets of M_1^0, M_1^+, M_1^- - non-empty, then consider the set

$$M_0' = M_1^0 \cup \{y_{ij} \mid y_{ij} = -L_1(e_i)e_j + L_1(e_j)e_i, e_j \in M_1^+, e_i \in M_1^-\}$$

Using the feature $L_2(x)$, divide the elements of the set M_1' analogous to the preceding three groups $M_2^0 = \{e \mid L_2(e) = 0\}$, $M_2^+ = \{e \mid L_2(e) > 0\}$ and $M_2^- = \{e \mid L_2(e) < 0\}$.

Let assume that, at least two of these sets is not empty, then we construct a set

$$M_2' = M_2^0 \cup \{y_{ij} \mid y_{ij} = -L_2(e_i)e_j + L_2(e_j)e_i, e_j \in M_2^+, e_i \in M_2^-\}.$$

Let thus construct the set M_j' from set of $M_j^0 = \{e_r^0 \mid L_j(e_r^0) = 0\}$, $M_j^+ = \{e_i^+ \mid L_j(e_i^+) > 0\}$ and $M_j^- = \{e_s^- \mid L_j(e_s^-) < 0\}$ using the $L_j(x)$ and this set is not empty.

Theorem 3.1. The elements of M_j' is a solution of system $L_1(x) = 0 \wedge L_2(x) = 0 \wedge ... \wedge L_j(x) = 0$.

The proof obviously it follows from the construction of elements of the set M_j', $j = 1, 2, ..., p$.

Definition 3.2. Set M'_j which is constructed above is called a minimal generating set of decisions or TSS(Truncated Set of Solutions) of system $S' = L_1(x) = 0 \wedge L_2(x) = 0 \wedge ... \wedge L_j(x) = 0$

Let $M'_j = \{e'_1, ..., e'_q\}$ - TSS system S', and M_j - the set of all it‘s solutions. Then we have this statement.

Theorem 3.3. An arbitrary vector $x \in M_j \setminus M'_j$ it can be represented as a non-negative linear combination

$$tx = b_1 e'_1 + ... + b_l e'_l$$

where $t, b_1 \in N, t > 1, e'_i \in M'_j, i = 1, ..., l$.

The proof is based on the following lemma.

Lemma 3.4. An arbitrary non-negative linear combination of the form $y = ce_i^+ + ... + de_i^-$ — can be represented as a linear combination of non-negative $ky = ue_i^+ + ... + ve_{is}^0$ or as a non-negative linear combination of $ky = ue_s^- + ... + ve_{is}^0$, where k, u, v - integers, $e_{is}^0 \in M'_j, e_i^+ \in M_j^+, e_s^- \in M_j^-$.

The proof. Let $y = ce_i^+ + ... + de_s^-$ - arbitrary linear combination. Then the set M'_j there is a vector $e_{is}^0 = ae_i^+ + ... + be_s^-$, where $e_i^+ \in M_j^+, e_s^- \in M_j^-$. compare the numbers c/a and d/b. There are possible such cases.

Case 1. $c/a > d/b$. Assuming that $d/b = \lambda$ and $c/a = \lambda + \mu$, where λ and μ - positive rational numbers whose denominators are equal, respectively, λ_1, μ_1. Then we have

$$y = (\lambda a + \mu a)e_i^+ + be_s^- = (ae_i^+ + be_s^-) + \mu ae_i^+ = \lambda e_{is}^0 + \mu ae_i^+$$

Let k the least common multiple of the numbers λ_1 and μ_1. Multiplying the vector y on k, get

$$ky = (k\mu)ae_i^+ + (k\lambda)e_{is}^0 = ue_i^+ + ve_{is}^0.$$

Case 2. $c/a < d/b$. Assuming that $c/a = \lambda$ and $d/b = \lambda + \mu$, where λ and μ - positive rational numbers whose denominators are equal, respectively, λ_1, μ_1 Then, doing the same as above, we get

$$ky = (k\mu)be_s^- + (k\lambda)e_{is}^0 = ue_i^+ + ve_{is}^0.$$

Case 3. $c/a = d/b$. Assuming that $c/a = \lambda = d/b$, where λ and μ - positive rational numbers whose denominators are equal, respectively, λ_1. Then, doing the same as above, we obtain

$$\lambda_1 y = (\lambda_1 \lambda)e_{is}^0.$$

The proof of the theorem for the number of induction p the system of equations S. $p=1$ (induction basis). Based on the fact that the set of vectors M_0' is the basis of the entire set N^q, any element x, that is a solution of $L_1(x)=0$, can be submitted via the vector $e_i^0 \in M_1^0, i=1,2,...,k$, $e_j^+ \in M_1^+, j=1,...,m$, $e_t^- \in M_1^-, t=1,...,n$, $b_i, c_j, d_t \in N$ as a linear combination

$$x = b_1e_1^0 + ... + b_ke_k^0 + c_1e_1^+ + ... + c_me_m^+ + d_1e_1^- + ... + d_ne_n^-$$

If all the numbers c_j u d_t are 0, nothing to prove. Assume that there is a strictly positive number among these. Note that when some of the numbers c_j is a strictly positive, then among the numbers d_t they should also be, because otherwise we get a contradiction

$$L_1(x) = 0 = c_1L(e_1^+) + ... + c_mL(e_m^+) > 0$$

Suppose, for definition $c_1 > 0$ u $d_1 > 0$. Using Lemma 3.4, replace the sum $c_1e_1^+ + d_1e_1^-$ - nonnegative linear combination x' of vectors e_1^+ and e_{11}^0 or vectors e_1^- and e_{11}^0. After this substitution, we obtain the vector

$$t'x = b_1e_1^0 + ... + b_ke_k^0 + c_2e_2^+ + ... + c_me_m^+ + d_2e_2^- + ... + d_ne_n^- + x',$$

in which the total number of non-zero coefficients of the e_j^+ and e_t^- will be equal to zero, that is, until we get this expansion:

$$tx = b_1e_1^0 + ... + b_ke_k^0 + a_{11}e_{11}^0 + ... + a_{mn}e_{mn}^0,$$

where $e_i^0 \in M_1^0$, $i=1,...,k$, $e_{is}^0 = -L(e_s^-)e_j^+ + L(e_j^-)e_s^+$, $e_j^+ \in M_1^+$, $e_s^- \in M_1^-$, $j=1,...,m$, $s=1,...,n$, $a_{js} \in N$.

This induction basis is justified.

The induction step. Let us assume that the theorem is correct for all $j < p$. Let us show that it holds for $j = p$. Let the vector $x \in M_p / M_p'$, i.e. x - set of solutions S. Then, since x - solution with any of its subsystems, which consists of the first $p-1$ systems of equations S, This vector has the expansion of the assumption $t'x = b_1e_1^0 + ... + b_ke_k^0 + c_1e_1^+ + ... + c_me_m^+ + d_1e_1^- + ... + d_ne_n^-$, where $e_i^0 \in M_p^0, i=1,...,k, e_j^+ \in M_p^+, j=1,...,m, e_t^- \in M_p^-, t=1,...,n, b_i, c_j, d_t \in N$.

Applying to vector $z = t'x$ Lemma 3.4 and computations analogous to those described above in justifying the induction basis we obtain for the vector z decomposition

$$t''z = t't''x = b_1e_1^0 + ... + b_ke_k^0 + a_{11}e_{11}^0 + ... + a_{mn}e_{mn}^0,$$

$$e_i^0 \in M_p^0, i = 1,...,k, e_{js}^0 = -L(e_s^-)e_j^+ + L(e_j^+)e_s^-, e_j^+ \in M_p^+, e_s^- \in M_p^0,$$
$$j = 1,...,m, s = 1,...,n, a_{js} \in N$$

We can now formulate a criterion of combining inspection SBLDE(systems of bounded linear Diophantine equations).

Theorem 3.5. HSLDE $S = L_1(x) = 0 \wedge L_2(x) = 0 \wedge ... \wedge L_p(x) = 0$ compatible then and only then if it TSS $M'_p \neq \varnothing$, then, according to Theorem 2.1 arbitrary vector M'_p is a solution of S. Consequently, S – compatible.

Contrariwise, let the system $L_1(x) = 0 \wedge L_2(x) = 0 \wedge ... \wedge L_{p-1}(x) = 0 \wedge L_p(x) = 0$ compatible and y - it solution. If $y \in M'_p$ then everything is proved. Let us assume that $M'_p \neq \varnothing$. Then, for the building of TSS-set for some $1 \leq j \leq p-1$ all vectors $s_1, s_2, ..., s_r$ from M'_j or $L_{j+1}(s_k) < 0$ or $L_{j+1}(s_k) > 0$ where $k = 1, 2, ..., r$. Because the y It is the solution HSLDEE, then y and its subsystems is a solution $L_1(x) = 0 \wedge L_2(x) = 0 \wedge ... \wedge L_{j+1}(x) = 0$ based on Theorem 3.3 exist $t > 1, t \in N$ such that $ty = b_1 s_1 + b_2 s_2 ... + b_r s_r$. But then

$$0 = L_{j+1}(ty) = b_1 L_{j+1}(s_1) + b_2 L_{j+1}(s_2) + ... + b_r L_{j+1}(s_r) > 0$$

(or < 0). But this does not contradict the fact that y It is the solution of HSLDE.

TSS can be divided into LCD (lowest common divisor) its coordinates if the LCD nonzero. This makes it possible to reduce the size and the coordinates of these vectors more efficiently perform computations.

4. Development of grid Service

Launching virtual machine in grid by Rainbow framework [1] mostly similar to procedure that user performs on regular computing grid task. But there are differences which are covered next.

Administrator of grid service or virtual laboratory sends a request to the national level Certification Authority of the grid for registration in the grid system, receives a digital certificate for host on which service is set up, register that service in the virtual organization (VO) and after that service itself prepares special tasks and executes them to run virtual machines. Preparation tasks include creating a job description file which has proper requirements for the VM, preparing input data, creating a proxy certificate. After these preparations, the job is submitted to the grid and depending on the workload of resources and availability of Rainbow framework it gets executed or queued for execution [7, 11].

The primary aim of development of the grid service must be its correct functioning, control of the correctness of the implementation of the work conditions in the system, successful VM running and return of the results of VM work to the end user in the case of a successful outcome. To satisfy these requirements in the grid, it is necessary to resort to formal methods of verification of their implementation, and formal methods, in turn, require the construction of an appropriate mathematical model on which such tests can be performed. Create detail model of grid-service with all features is considered almost impossible because of high complexity, so creation of model is performed at some level of abstraction.

Thus, the work of grid service in the grid infrastructure begins with the actions of two protocols: the registering of service in the grid and preparation of grid task with VM inside protocol. Implementation of actions under these protocols is disposable. This means that if the protocols of registration and preparation of the grid task are successful then the are no more actions to do. If the registration for whatever reason had not succeed, then administrator of such service should be informed and he should repeat the registration attempt again. Let's present the actions of both protocols as the transition systems [6,8], which are fairly obvious because of its simplicity.

4.1. Protocol of registration of grid service. Actions of this protocol are as follows. Before grid service can access the computing resources there is a need to register it in the system and obtain the host certificate. To do this administrator sends a request signed by his certificate for registration to CA and if digital

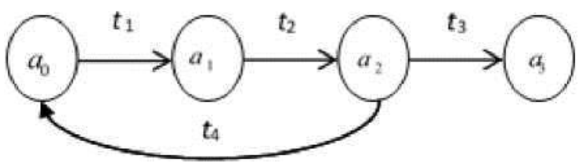

Figure 5. TS of registration in grid system

signature is valid, the host certificate of grid service signed by a trusted national certificate authority and then registered in the grid system. These actions are modeled by a simple transition system (TS)

$A_1 = (\{a_0, a_1, a_2, a_3\}, \{t_1, t_2, t_3, t_4\}, \alpha_1, \beta_1, a_0)$:

Interpretation of places and transitions in this TS is:

a_0 - query generation (generation of host certificates),

t_1 - registration request (administrator of grid-service sending signed certificate to the trusted certificate authority), where $\alpha_1(t_1) = a_0, \beta_1(t_1) = a_1$,

a_1 - query processing (verification of service administrator digital signature),

t_2 - the process of registration (certificate signing), where $\alpha_1(t_2) = a_1, \beta_1(t_2) = a_2$,

a_2 - analysis of the registration process,

t_3 - registration process, where $\alpha_1(t_3) = a_3, \beta_1(t_3) = a_4$,

a_3 - user registration (obtaining a signed certificate by the user)

t_4 - registration failed (failure in digital signature verification), where $\alpha_1(t_4) = a_2, \beta_1(t_4) = a_0$.

If the registration is successful (state a_3 in TS), then in state a_3 registration system informs the service administrator and middleware about new legitimate service as grid users. If the registration is for some reason not made in the system, then in state a_2 registration system performs an action t_4, which informs the administrator of service that it has not been registered in the grid system.

The second stage of work in the grid system is to become ma ember of one or more of the available virtual organizations (VO). In this paper we consider case of membership by default in one of the VOs.

4.2. Protocol of preparing grid task with virtual machine.

Successfully registered and received a digital certificate to access the grid structure, service proceeds to the protocol of preparation of proper grid task. This protocol provides the following sequence of actions to be performed by the grid service: a) create a job description file with VM parameters; b) prepare the VM image and input data; c) determine the location of the data and VM image (loading from local storage of grid service or from the grid storage); g) generate a proxy certificate with the appropriate access attributes.

This protocol is modeled by A_2=({1,2,3,4,5,6,7}, $\{r_1, r_2, r_3, r_4, r_5\}, \alpha_2, \beta_2$, 1):

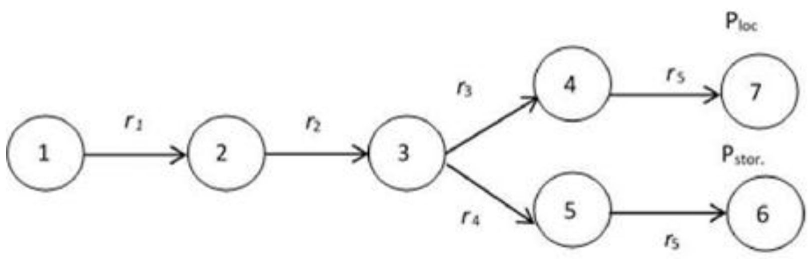

Figure 6. TS of preparing grid task

Interpretation of places and transitions in this TS is:

1- grid task preparation initialization,

r_1- creation of job description file with VM parameters,

2- data and VM image preparation start, file description created,

r_2 - input data and VM image preparation,

3- analysis of data and VM image location,

r_3 - preparation of data and VM image from local storage,

4- data and VM image from user local storage are ready,

r_4 - preparation of data and VM image from grid-storage,
5- data an VM image from grid-storage ready,
r_5 - creating a proxy certificate with appropriate access attributes.
6- ready state of data and VM image from grid-storage.
7- ready state of data VM image from user local storage.

In the transition to state 6 TS must inform the grid service and grid middleware about task readiness. For this purpose, additional states are entered in the model, which is a synchronous product of the TS, in the form of Petri nets (PN).

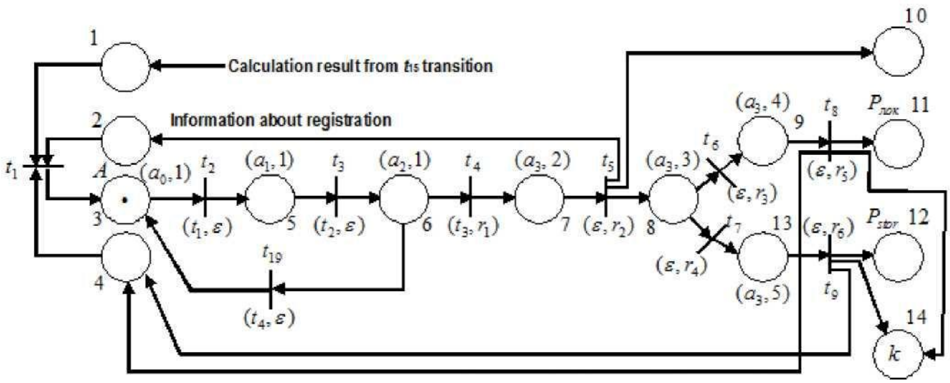

Figure 7. TS modeling synchronous product of TS

5. The network model of registration protocol and task preparation

Now we can build a model of registration and preparation of the product of two simultaneous TI A1 * A2 with global transitions:
T = {(t1,e), (t2, e), (t3, r1), (t4, e), (e, r2), (e, r3), (e, r4), (e, r5), (e, r6)}

Let build Petri network that simulates the operation of the two systems By this TS:

In this PN beside places of synchronous product of above mentioned TS appear places obtained by performing of product of synchronous data in TS and TS that simulates the computing environment (by 10 places 11,12, and 14 connected to both networks). TS of computing environment and synchronous product of the corresponding TS not shown because of the cumbersome, and the details of such process is shown on above example. The final result of the product in the form of a Petri net, description of the places and transitions which are given below:

Position 3 corresponds to the grid service A, which initializes (puts a flag in this place) to start work with grid.

Place 4 informs the grid service administrator about obtaining the certificate and readiness to execute grid task.

Places 2 and 10 inform the grid service administrator and grid middleware of successful completion of the registration process and obtaining a host certificate for grid service.

Place 11 (P_{loc}) means using data and VM image from local storage, and the place 12 (P_{stor}) - input data and VM image on the grid-storage.

Position 14 indicates that the proxy certificate is created with parameter k.

5.1. Network model of the grid computing element(CE) running virtual machine a grid task.

In constructing the model of the CE of grid system we will assume that all computing resources are ranked by numbers k_1, k_2, ... k_m, where m - number of calculators, k - parameters of calculator i.

Places 15, 16, ... - data storage’s.

Place 17 - found computing element with the necessary parameters and Rainbow framework support (places 22, 23) (calculator with the parameters kj (place 24) CE does not have the necessary parameters ($k_1 < k$).

Places 20, 21, 22, 23 n 24, 25 refer to the computing element and its load.

Place 18 - data and VM image is loaded, place 19 - running VM as grid task.

Place 27 - state of failure, and the place 26 - state of recovery working capacity of computing cluster as computing element in grid.

Transitions:

t_{10} - initialize of the required data and VM image on the media,

t_{11} - search computing element with proper parameters and Rainbow framework support,

t_{12} - initialize and load the local stored data and VM image,

t_{13} - loading data and VM image from the storage,

t_{14} - start of running VM,

t_{15} - end of running VM,

t_{16} - initialization of the emergency state and resume the task parameters, data and VM image with finding another computing element with the necessary parameters and Rainbow framework support, and restart of virtual machine.

t_{17} - initialization of recovery working capacity of computing cluster as grid computing element.

t_{17} - computing cluster working capacity recovery.

In the above mentioned PN similar transition to the transition t_{12}, triggered on condition of working capacity of computing element (no flag in place 27), its load (presence of flag at location 22), it respond VM required parameters and support of the Rainbow framework (in place of 23 is $k < k_2$ flags). Computing element is chosen according these requirements. In the case of high load or malfunction of the computing element grid service will look for another free and serviceable computing element with the necessary parameters and Rainbow framework support, on which grid service can run VM as grid task (in this example of PN transition t'_{11} does not work, because the $k_1 < k$). If the computing element with the necessary parameters and Rainbow support isn't available in the moment, grid service is waiting for submit VM as grid when it will be available.

Then the network model of grid computing element running virtual machine as a grid task becomes as on Figure 8.

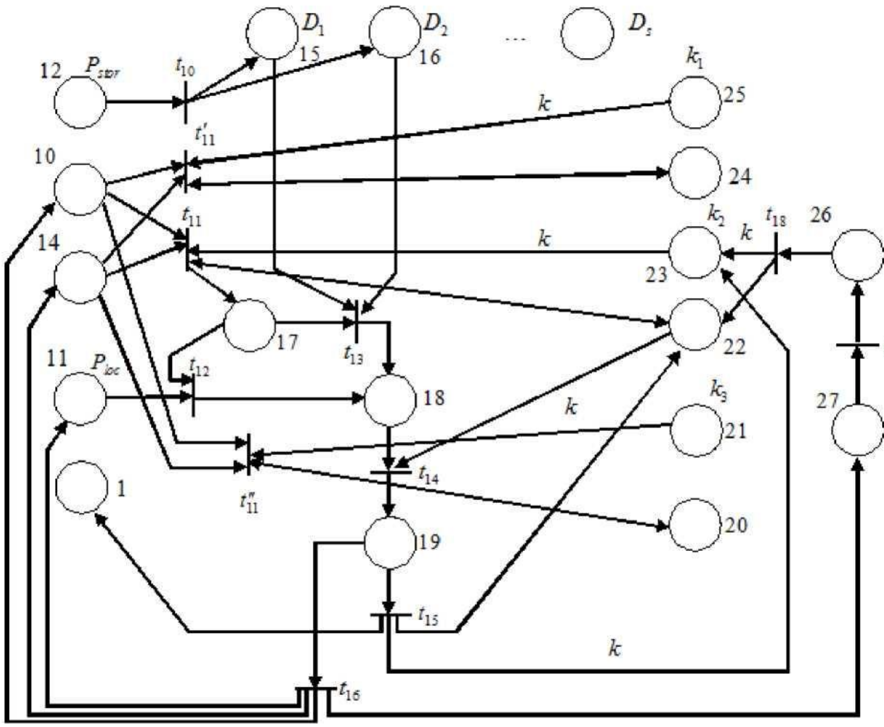

Figure 8. The network model of grid system of three clusters with grid computing elements that support Rainbow framework

6. Justification of network model of grid computing element running virtual machine as a grid task

Constructing Petri network that models the grid computing element and its interaction with grid service of running virtual machine as a grid task, we get this project done by the specification of a system that is at the same time and the mathematical model of the projected grid computing element and grid service. Now the task is to validate the constructed model. Thus the first task is to ensure to ensure the working capacity of the system. The persistence of the system means that the constructed model, all transitions will take part in the process of its operation (all live transitions). If some transitions in the PN does not work, their presence suggests that something in the system project done incorrectly or data transitions redundant. The same applies to the PN places. If some places, due to the presence of the dead transitions are unattainable in the operation of the PN, it is also indicative of the project defect.

PN survivability test is performed in two ways: by constructing a reachable marking graph of PN and by analyzing the transition invariants. Construction of the set of invariants of transitions is reduced to the solution of the equation of state in the set of natural numbers of the form Ax = 0, where A - the incidence matrix of the PN, and x - vector coordinate values are equal to the number of actuations of the corresponding transitions. If some coordinate all solution vector is 0, it means that the transition had not triggered during operation the PN (go dead) regardless of initial system layout.

For PN, where it is assumed that three computing elements (n = 3), the equation of state is a matrix of dimension 19 to 27, as in the PN has 19 conversions and 27 seats:

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

↑ numbers of transitions number of places in PN.

Applying TSS-algorithm [7] for the solution of the equation of state with the above matrix, we find it such decisions, which are invariants of transitions:

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

As can be seen from the above solutions transitions 16,17,18 are dead because never take part in the operation of the PN. These transitions are associated with emergency state of the interaction of grid service and computing element, and it means that in case of failure of computing element VM will not be running. Analysis of causes by which these transitions do not work shows that grid service must to restart grid task, download again data and VM image with the recover of the computing element in order to restart virtual machine on it. In other words in the project of interaction of grid service and computing element described by this mathematical model, action is needed to restore a custom task settings, and to restore its data.

PN correction associated with the transmission of information about the data and VM image read renewal (which means sending in place 12 or 11 in place of flags, depending on the location of data and VM image). This correction leads to the equation of state with the matrix A:

A:

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

In this case, this matrix corresponds to the recovery data and VM image from local storage, as testified by the decision equation of state with the reduced matrix:

x1=	(0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1),
x2=	(1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	0),
x3=	(1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	0	0),
x4=	(0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	0).

As seen from this set of invariants, all transitions in a live network, indicating the entire PN liveliness in general. The first invariant describes a situation where registration of grid service in the grid infrastructure is not executed (triggered transitions 2, 3 and 19). The second invariant describes the successful registration of grid service of the situation, obtaining a certificate, using data and VM image from local storage and the successful VM running (and trigger transitions 2,3,4.5,6,8,11,12,14,15 1). Third invariant describes the same situation, but with data and VM images from grid storage's (2,3,4,5,7,9,10,11,13,14,15 transitions triggered and 1). Fourth invariant describes an emergency situation and indicates that the registration of grid service and obtaining of the host certificate is not required. There is need to upload the data and VM image, recover grid task with VM parameter and submit grid task once more (triggered transitions 11,12,14,16,17,18). If data and VM image recovered grid storage, then the invariants have the form

y1=	(0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1),
y2=	(1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	0),
y3=	(1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	0	0),
y4=	(0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	0).

Settings of computing element can be restored and data and VM image reloaded from grid storage. Thus, the first property - the property of liveliness - in the model implemented. From this property it follows that in this model the property of fairness is performed, which consists in the fact that if the grid task with VM is in the grid, it will be performed sooner or later. It should be apparent from the fact that the corresponding transitions (leading to a free computing elements) will be performed in the same way as in the above example.

These solutions of equation of state make it possible to formulate such claim.

Theorem 6.1. PN modeling interaction of grid service and computing element for running VM as grid task is alive, and it satisfies the property of justice.

7. Properties of boundedness and justice

Consider the feasibility of the PN boundedness property, which in the case of its feasibility, confirms the feasibility of the justice property. Property of boundedness means that the in the PN does not exist places where the flags may accumulate in unlimited quantities. This situation corresponds to the case when in the computing element, all the time some grid tasks are running, but no VM running in the system.

To study the properties of this PN is necessary to solve a system of linear homogeneous Diophantine equations in the set of natural numbers of the form A = 0, where AT - matrix transpose of the incidence matrix A. Solutions of such system are called invariants of places and if all the places are covered in the PN by positive values, the PN bounded. In our example transpose matrix has the form

–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	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	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	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	0	0	0	0	–1	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	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	0	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	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	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	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
0	0	0	0	0	0	0	0	0	–1	0	0	0	–3	0	0	1	0	0	0	0	0	–3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	–1	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	0	0	–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	0	0	–1	1	0	0	–1	0	0	0	0	0
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	0	0	0	0	0	0	0	0	1	1	0	0	3	0	0	0	0	–1	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	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	1	3	0	0	–1	0
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

Solutions of system Ay = 0 have the form:

	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
p1=	(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),
p2=	(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),
p3=	(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),
p4=	(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	0	0),
p5=	(0	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	1),
p6=	(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),
p7=	(1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0),
p8=	(1	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0),
p9=	(1	0	1	0	1	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0),
p10=	(3	0	3	0	3	3	3	3	3	0	0	0	3	1	0	0	3	3	3	0	0	0	0	0	0	0	0),
p11=	(0	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	1),
p12=	(0	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),

From these decisions obviously get the boundedness of PN, since all of its places covered with positive values. Consequently, there is a

Theorem 7.1. SP modeling interaction of grid service and computing element for running VM as grid task is bounded, and it satisfies the property justice.

8. Conclusions

Development of new grid services is an important element of expansion of grid virtual laboratories functions, expanding the application of grid computing and development of collaboration of researches requires tools for formal design.

Adequate model proposal for grid service that run virtual machine grid task and its interaction with grid computing element as Petri net that allow to strictly prove boundedness, live and validity of the designed system.

Approach can be used to model other grid services, their interactions with each other and with another elements of the grid infrastructure.

References

1. Boretskyi O., Salnikov A., Shusar I., Sudakov O., Boyko Y. Rambow framework: Runnrng vntual machmes on demand as a grid jobs. In Intelhgent Data Acquisition and Advanced Computmg Systems: Technology and Apphcarions (IDAACS), (2015), IEEE 8th Internarional Conference on Vol. 2, 2015, pp. 972-976.
2. Esparza J., Heljanko K., Unfoldmgs A., Parrial-Order Approach to Model Checkmg, Springer Verlag, 2008, 170 p.
3. Foster Ian, Carl Kesselman, eds., The Grid 2: Blueprint for a new computmg mfrastructure. Elsevier, 2003.
4. Kryvyp S.L., Boyko Y.V., Pogorilyy S.D., Boretskyp O.F., Glybovets M.M., Design of Grid Structures on the Basis of Transition Systems whh the Substanriarion of the Correctness of Then Operarion, Cybernerics and Systems Analysis, Vol.53, No.1, 2017, pp.105-114.
5. Salnikov A.O., Sliusar I.A., Sudakov O.O., Savytskyp O.V., Kornelyuk A.I., MolDynGrid vhtual laboratory as a part of Ukrainian Academk Grid mfrastructure. In Intelhgent Data Acquisition and Advanced Computmg Systems: Technology and Apphcarions, (2009), IDAACS 2009. IEEE Internarional Workshop on, pp. 237-240.
6. 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. Монографія за редакцією чл.-кор. НАН України, д-ра фіз.-мат. наук, проф. А. В. Анісімова. ВПЦ «Київський університет», 2016р. (в процесі друкування).
7. Бойко, Ю.В., Зинов'єв, М.Г., Свістунов, С.Я., Судаков, О.О., Український академічний грід: досвід створення і перші результати експлуатації. Математические машины и системы, № 1, 2008, с.67-84

8. Кривий С.Л., Лінійні діофантові обмеження та їх застосування, Букрек: Чернівці, 2015, 224 с.
9. Погорілий С.Д., Бойко Ю.В., Левченко Р.Т., Мар'яновський В.А., Методи кластерних обчислень, За редакцією д-ра техн. наук., проф. С. Д. Погорілого, К:, ВПЦ "Київський університет", 2013, 416 с.
10. Сальников А.А., Вишневский В.В., Борецкий А. Ф., Платформа как сервис» в грид для интерактивного анализа медицинских данных, Математические машины и системы, 1, 2015, ISO 690.
11. Слюсар Е.А., Автоматизированная служба репликации файлов для организации высокой доступности данных в грид-инфраструктуре, УСиМ. - Киев: Академ—периодика, № 4, 2012, с.63-74.

Keçid sistemlərinə əsaslanan hesablama vəzifəsi kimi virtual maşınların işlənməsi üçün şəbəkə xidmətlərinin dizaynı

S.L. Kryvyi, YU.V. Boyko, S.D. Pogorilyy,
O.F. Boretskyi, M.M. Glibovets

XÜLASƏ

Virtual maşınların müntəzəm hesablama vəzifəsi kimi işlənməsi üçün şəbəkə xidmətlərinə dair şəbəkə sistemi dizaynı metodunu tətbiq etmək təklif edilmişdir. Alınmış qlobal keçid sistemi Petri şəbəkəsinə (PŞ) çevrilmişdir. PŞ-nin köməyi ilə dizaynın düzgünlüyü, xüsusilə kilidlərin, ölü keçidlərin olmaması və s. yoxlanılmışdır.

Açar sözlər: şəbəkə, şəbəkə xidməti, virtual maşın, keçid sistemi, petri şəbəkəsi, Diofant tənliyi.

Проектирование грид-службы запуска виртуальных машин как грид-заданий на основе транзационных систем

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

РЕЗЮМЕ

Предлагается применить метод проектирования для грид-службы для запуска виртуальных машин в качестве обычной вычислительной задачи. Полученная глобальная транзиционная система переведена в сеть Петри(СП). С помощью СП проверяется правильность конструкции, в частности отсутствие взаимоблокировок, мертвых переходов и т. д.

Ключевые слова грид, грид-служба, виртуальная машина, транзационная система, сеть Петри, Диофантовое уравнение