

**Председатель  
редакционного совета**  
Г. Г. Себряков, чл.-корр. РАН

**Главный редактор**  
И. А. Каляев, чл.-корр. РАН

**Заместители председателя  
редакционного совета:**  
С. Ю. Желтов, чл.-корр. РАН  
М. Н. Красильщиков, д.т.н.

**Редакционный совет:**  
В. И. Аверченков, д.т.н.  
(зам. гл. редактора)  
А. И. Башмаков, к.т.н.  
С. Н. Васильев, акад. РАН  
Ю. В. Визильтер, д.ф.-м.н.  
А. И. Кибзун, д.ф.-м.н.  
П. Е. Клейзер (зам. гл. редактора)  
Ю. Н. Кофанов, д.т.н.  
В. В. Лебедев, чл.-корр. РАН  
Е. А. Микрин, акад. РАН  
В. В. Попов, д.т.н.  
А. В. Рыбаков, к.т.н.  
Р. Л. Смелянский, чл.-корр. РАН

**Региональные редсоветы:**

**Белгород**  
И. С. Константинов,  
д.т.н.

**Орел**  
В. Т. Еременко, д.т.н.  
А. В. Косыкин, д.т.н.

**Волгоград**  
В. А. Камаев, д.т.н.

**Переславль-Залесский**  
С. М. Абрамов,  
чл.-корр. РАН

**Иркутск**  
И. В. Бычков,  
акад. РАН

**Ростов-на-Дону**  
В. И. Минкин, акад. РАН  
А. В. Солдатов, д.ф.-м.н.

**Курск**  
О. И. Атакищев, д.т.н.

**Санкт-Петербург**  
Ю. А. Гатчин, д.т.н.

**Минск (Республика  
Беларусь)**  
С. В. Абламейко,  
акад. НАНБ

**Ставрополь**  
П. А. Аверичкин, д.т.н.

**Нижний Новгород**  
Р. Я. Вакулenco, д.э.н.  
С. И. Ротков, д.т.н.

**Таганрог**  
В. В. Курейчик, д.т.н.  
**Тэгу (Южная Корея)**  
Р. А. Зиятдинов, к.ф.-м.н.

**Редакция:**  
О. В. Мочалина  
Н. В. Пантина

Журнал зарегистрирован  
в Министерстве Российской Федера-  
ции по делам печати, телерадио-  
вещания и средств массовых  
коммуникаций.

Свидетельство о регистрации  
ПИ № ФС77-46365 от 26.08.2011 г.

Тел. редакции: (495) 589 56 41,  
514 76 50

Адрес редакции: 119048, г. Москва,  
ул. Усачева, д. 35, стр. 1

E-mail: vkit@idspekt.ru;  
vkitpost@rambler.ru  
Http://www.vkit.ru

## СОДЕРЖАНИЕ

### ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ КОМПЛЕКСЫ ПОДВИЖНЫХ ОБЪЕКТОВ

**Кикин И. С.** Анализ методов формирования функционала терминального управления дальностью полета азробаллистического летательного аппарата ..... 3

### КОМПЬЮТЕРНОЕ ЗРЕНИЕ. ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ

**Бобков А. Е., Леонов А. В.** Процедурная реконструкция территорий на виртуальном глобусе..... 10

**Мужичек С. М., Обросов К. В., Ким В. Я., Лисицын В. М.** Адекватность использования натуральных регистраций изображений при моделировании тепловизионных систем ..... 18

### ИНФОРМАЦИОННАЯ ПОДДЕРЖКА ЖИЗНЕННОГО ЦИКЛА ТЕХНИЧЕСКИХ СИСТЕМ

**Топоров Н. Б.** Использование высокопроизводительных вычислительных систем для формирования программ развития авиационной техники ..... 25

### РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ И ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ. СЕТЕВЫЕ ТЕХНОЛОГИИ

**Каляев А. И., Каляев И. А., Коровин Я. С.** Метод мультиагентного диспетчирования ресурсов в гетерогенной облачной среде при выполнении потока задач ..... 31

**Павский В. А., Павский К. В.** Математическое моделирование функционирования распределенных вычислительных систем с отказами и полным восстановлением ..... 41

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ И ОБРАЗОВАНИИ

**Клячкин В. Н., Шунина Ю. С.** Система оценки кредитоспособности заемщиков и прогнозирования возврата кредитов ..... 45

### ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

**Щеглов К. А., Щеглов А. Ю.** Реализация контроля и разграничения прав доступа к статичным объектам ..... 52

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении, или непосредственно в издательстве.

Индексы по каталогам: «Роспечать» – 84197, «Пресса России» – 39244, «Почта России» – 60263

Журнал входит в перечень изданий, утвержденных ВАК РФ, для публикации т) соискателей ученых степеней

Перепечатка, все виды копирования и воспроизведения материалов, публикуемых в «Вестник компьютерных и информационных технологий», допускаются со ссылкой на информацию и только с разрешения редакции

**Chairman of Editorial Board**

G. G. Sebryakov

**Editor-in-Chief**

I. A. Kaliaev

**Deputy Chairman  
of Editorial Board:**S. Yu. Zheltov,  
M. N. Krasilshchikov**Editorial Board:**V. I. Averchenkov  
(Deputy Editor-in-Chief)  
A. I. Bashmakov  
S. N. Vasilev  
Yu. V. Vizilter  
A. I. Kibzun  
P. E. Kleyzer (Deputy Editor-in-Chief)  
Yu. N. Kofanov  
V. V. Lebedev  
E. A. Mikrin  
V. V. Popov  
A. V. Rybakov  
R. L. Smelyansky**Regional Editorial Board:**

<b>Belgorod</b> I. S. Konstantinov	<b>Orel</b> V. T. Eremenko A. V. Koskin
<b>Daegu (South Korea)</b> R. A. Ziatdinnov	<b>Pereslavl-Zaleskiy</b> S. M. Abramov
<b>Irkutsk</b> I. V. Bychkov	<b>Rostov-on-Don</b> V. I. Minkin A. V. Soldatov
<b>Kursk</b> O. I. Atakishchev	<b>Saint-Petersburg</b> Yu. A. Gatchin
<b>Minsk (The Republic of Belarus)</b> S. V. Ablamejko	<b>Stavropol</b> P. A. Averichkin
<b>Nizhny Novgorod</b> R. Ya. Vakulenko S. I. Rotkov	<b>Taganrog</b> V. V. Kureichik
	<b>Volgograd</b> V. A. Kamaev

**Editorial Staff:**O. V. Mochalina  
N. V. PantinaThe journal is registered by RF  
Ministry of Press, Tele-and-Broad-  
casting and Mass Communications  
Media. Registration certificate  
ПИ № ФС77-46365 от 26.08.2011 г.

Tel.: (495) 589 56 41, 514 76 50

Editorial address: Bul. 1, Usacheva  
St. 35, Moscow, Russia, 119048E-mail: vkitt@idspektr.ru;  
vkittpost@rambler.ru  
Http://www.vkitt.ru**CONTENTS****INFORMATING-CONTROLLING COMPLEXES  
OF MOBILE OBJECTS****Kikin I. S.** Analysis of Methods to Create a Functional of Termination Control of Aeroballistic Vehicle's Range ..... 3**COMPUTER VISION. VIRTUAL REALITY****Bobkov A. E., Leonov A. V.** Procedural Terrain Reconstruction on a Virtual Globe ..... 10**Muzhichek S. M., Obrosof K. V., Kim V. Ya., Lisitsyn V. M.** The Adequacy of the Use of Full-Scale Image Registration in the Simulation of Infrared Imaging Systems ..... 18**INFOTAINMENT FOR LIFE CYCLE OF ENGINEERING SYSTEMS****Toporov N. B.** The Usage of High Performance Computing Systems for the Aviation Technology Development Programs Formation ..... 25**DISTRIBUTED COMPUTING AND INFORMATING-CONTROLLING SYSTEMS.  
NETWORK ENGINEERING****Kalyaev A. I., Kalyaev I. A., Korovin Ya. S.** Method of Multiagent Dispatching Resources in Heterogeneous Cloud Environments while Performing Flow of Incoming Tasks ..... 31**Pavsky V. A., Pavsky K. V.** Mathematical Modeling of Distributed Computer Systems Functioning with Failures and Full Recovery ..... 41**INFORMATION TECHNOLOGIES IN SOCIAL,  
ECONOMIC SYSTEMS AND EDUCATION****Klyachkin V. N., Shunina Yu. S.** System for Borrowers' Creditworthiness Assessment and Repayment of Loans Forecasting ..... 45**INFORMATION SECURITY****Shcheglov K. A., Shcheglov A. Yu.** Static Objects Control and Access Rights Implementation ... 52The journal is being distributed according to a subscription, which is available in any post office or at the publishing house directly. Indexes in the catalogue:  
"Rosspechat" – 84197, "Pressa Rossii" – 39244, "Pochta Rossii" – 60263**Journal is included into the list of editions certified by RF Supreme Attestation Committee for publication of competitors works for scientific degrees**

Reprinting of materials from "Journal of Computer &amp; Information Technology" is possible with writing permission of editorial staff. Reference to "Journal of Computer &amp; Information Technology" at reprint is obligatory



**А. И. Каляев**, канд. техн. наук, **И. А. Каляев**, чл.-корр. РАН, **Я. С. Коровин**, канд. техн. наук  
(Научно-исследовательский институт многопроцессорных вычислительных  
и управляющих систем, Таганрог);  
e-mail: kaliaev@mvs.sfedu.ru

## МЕТОД МУЛЬТИАГЕНТНОГО ДИСПЕТЧИРОВАНИЯ РЕСУРСОВ В ГЕТЕРОГЕННОЙ ОБЛАЧНОЙ СРЕДЕ ПРИ ВЫПОЛНЕНИИ ПОТОКА ЗАДАЧ\*

*Рассмотрена проблема адаптивного диспетчирования (распределения) ресурсов в облачной вычислительной среде (ОВС), включающей в свой состав разнотипные (гетерогенные) вычислительные ресурсы, при решении потока крупномасштабных научных задач, поступающих в произвольные моменты времени и состоящих из множества информационно взаимосвязанных подзадач. Сформулирована задача диспетчирования ОВС для различных вариантов исходной постановки. Предложены принципы и методы мультиагентного диспетчирования ресурсов гетерогенной ОВС при решении потока крупномасштабных задач, обеспечивающие адаптивное распределение всех свободных в текущий момент времени вычислительных ресурсов ОВС с учетом их реальной производительности на различных подзадачах. Приведены алгоритмы работы программного агента мультиагентного диспетчера для различных вариантов организации ОВС, а также результаты исследования их эффективности с помощью программной модели гетерогенной ОВС.*

**Ключевые слова:** облачная вычислительная среда; децентрализованный диспетчер; мультиагентное диспетчирование; поток крупномасштабных задач; адаптивное распределение ресурсов.

**A. I. Kalyaev, I. A. Kalyaev, Ya. S. Korovin** (Scientific Research Institute  
of Multiprocessing Computing and Control Systems, Taganrog)

## METHOD OF MULTIAGENT DISPATCHING RESOURCES IN HETEROGENEOUS CLOUD ENVIRONMENTS WHILE PERFORMING FLOW OF INCOMING TASKS

*The work is devoted to solving the fundamental problem of the adaptive allocation of resources in a CCE (Cloud Computing Environment), which consists of heterogeneous computing resources for solving incoming flow of large-scale scientific tasks coming at random moments and consisting of interconnected subtasks. It is assumed that, in general, all the computing resources that are part of the CCE have different specializations (can solve different sets of sub-tasks), and the different performance when solving similar tasks. The article formulates four options for setting the problem of allocation of heterogeneous resources in CCE depending on the specific conditions of its organization. It is shown that the solution of these problems by centralized controller is extremely difficult. Therefore, this article offers guidelines how to use heterogeneous multi-agent allocation of resources in CCE while solving large-scale flow of tasks, providing adaptive distribution of all available at the current time computing resources with regard to their specialization and the actual performance on various subtasks. Authors describe algorithms of multi-agent software agent dispatching for all four original formulations of the problem. Authors researched developed multi-agent allocation algorithms using the program model of heterogeneous CCE, and results demonstrated their efficiency. Implementing the proposed approach to the problem of scheduling heterogeneous CCE allows you to:*

– Provide a quasi-optimal automatic allocation of resources CCE in resolving problems based on their expertise and real performance on a given task;

– Ensure high payload of resources of CCE;

– Allow unlimited scalability of heterogeneous computing resources in CCE;

– Provide increased fault tolerance in CCE, since it lacks components, failure of which leads to disastrous consequences for the whole system.

**Keywords:** Cloud computing environment; Decentralized controller; Multi-agent dispatching; Flow of large-scale tasks; Adaptive resource allocation.

### Введение

В настоящее время уровень развития телекоммуникационных технологий открывает новые возможности организации распределенных вычислений

на основе множества пространственно удаленных вычислительных ресурсов с помощью сервис-ориентированной инфраструктуры, обеспечивающей «вычисления по требованию». Такие возмож-

\* Работа выполнена при поддержке РФФИ, проект № 15-29-07928.

ности порождают новую парадигму облачных вычислений, т.е. крупномасштабных распределенных вычислений на основе пула абстрактных, виртуализованных, динамически перераспределяемых вычислительных ресурсов, предоставляемых по запросу внешним пользователям через Интернет [1]. В настоящее время *облачные вычислительные среды (ОВС)* находят все более широкое применение для решения крупномасштабных научных и технических задач различных предметных областей, например, физика высоких энергий; науки о Земле; химия и биология; космология и астрофизика; экологическая и техногенная безопасность; промышленность; фармакология и фармацевтика; материаловедение; нефтегазодобыча; медицина и т.п. Особенность таких крупномасштабных научных задач – сложная внутренняя структура, состоящая, как правило, из ряда информационно взаимосвязанных подзадач, причем, зачастую, эффективность их решения в большей степени зависит от типа используемого вычислительного ресурса.

Одним из основных преимуществ концепции облачных вычислений является возможность использования в составе ОВС разнотипных (гетерогенных) вычислительных ресурсов, которые имеют различную реальную производительность при решении тех или иных задач. Данное обстоятельство, с одной стороны, открывает возможность повышения эффективности (т.е. сокращения времени) выполнения крупномасштабных пользовательских задач за счет использования вычислительных ресурсов ОВС, обеспечивающих максимальную реальную производительность при их решении (или их подзадач); с другой стороны, порождает проблему оптимального распределения (диспетчирования) решаемых задач (или их подзадач) по разнотипным вычислительным ресурсам, входящим в состав ОВС. Рассматриваемая проблема многократно усложняется в случае, если ОВС должна обеспечивать решение не одиночной задачи, а некоторого множества (потока) разнородных пользовательских задач, поступающих в произвольные моменты времени [2], что требует разработки новых методов и алгоритмов адаптивного диспетчирования ресурсов гетерогенных ОВС при обработке потока крупномасштабных задач.

**Постановка задачи**

Представим формально проблему диспетчирования ОВС. Предположим, что ОВС должна решать некоторое множество (поток) различных крупномасштабных задач  $Z = \langle Z_1, Z_2, \dots, Z_M \rangle$ , которые поступают от пользователей в случайные моменты

времени. При этом под крупномасштабной задачей будем понимать задачу, состоящую из некоторого множества информационно зависимых (взаимосвязанных) подзадач, каждая из которых имеет значительную вычислительную трудоемкость. В общем виде каждую такую крупномасштабную задачу  $Z_l \in Z$  представим в виде некоторого графа  $G_l(Q_l, X_l)$  (рис. 1), вершинам  $q_j \in Q_l$  которого приписаны некоторые подзадачи  $A_j$ , принадлежащие множеству подзадач  $A = \langle A_1, A_2, \dots, A_c \rangle$ , а дуги  $x(q_j, q_{j+1}) \in X_l$  определяют информационные взаимосвязи между подзадачами. Если две вершины  $q_j$  и  $q_{j+1}$  соединены дугой  $x(q_j, q_{j+1})$ , то данные, полученные в результате решения подзадачи  $A_j$ , приписанной вершине  $q_j$ , необходимы для выполнения подзадачи  $A_{j+1}$ , приписанной вершине  $q_{j+1}$ . Каждой вершине  $q_j \in Q_l$  приписан тип решаемой подзадачи  $A_j \in A$ , а также ее вычислительная трудоемкость  $V_j$ , оцениваемая числом элементарных вычислительных операций, выполняемых при ее решении. Кроме того, положим, что пользователь устанавливает требуемый момент времени  $T_{max}^l$ , к которому он желает получить результат решения своей задачи  $Z_l$ , а также число баллов (бонусов)  $O$ , которое он готов заплатить за это.

Пусть в состав ОВС входит множество вычислительных ресурсов  $R = \langle R_1, R_2, \dots, R_N \rangle$ , каждый из которых может решать некоторый набор (подмножество) типов подзадач  $A_i = \langle A_1^i, A_2^i, \dots, A_L^i \rangle \subseteq A$  ( $i = 1, 2, \dots, N$ ), причем реальная производительность ресурса  $R_i$  при решении подзадачи  $A_j^i \in A_i$  составляет  $S_j^i = F(A_j^i)$  ( $j = 1, 2, \dots, L$ ).

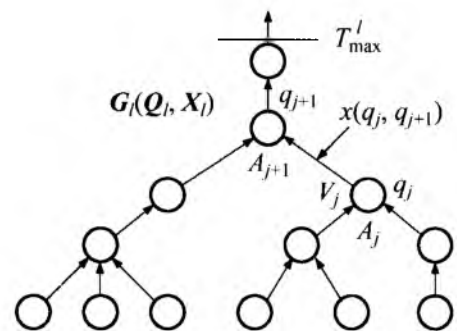


Рис. 1. Граф  $G_l(Q_l, X_l)$  задачи  $Z_l$

Цель работы диспетчера ОВС – такое распределение подзадач потока задач  $Z$  по вычислительным ресурсам  $R_1, R_2, \dots, R_N$ , при котором все задачи  $Z_i \in Z$  будут выполнены к требуемым моментам времени  $T_{\max}^i$ .

В зависимости от конкретных условий организации ОВС сформулированная выше задача диспетчирования может иметь *четыре* различных варианта исходной постановки.

**Вариант 1.** В простейшем варианте все вычислительные ресурсы  $R_1, R_2, \dots, R_N$  ОВС одинаковы и могут решать наборы подзадач одинакового типа, т.е.  $A_i = A_j = A$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ); причем производительность всех ресурсов  $R_1, R_2, \dots, R_N$  при решении подзадач идентичных типов  $A_d \in A$  также одинакова, т.е.  $S_d^i = S_d^j$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ), где  $d$  – тип подзадачи. Данная постановка соответствует гомогенной ОВС, включающей в свой состав полностью однотипные вычислительные ресурсы.

**Вариант 2.** В более сложном варианте наборы типов подзадач, решаемых всеми вычислительными ресурсами ОВС, одинаковы, т.е.  $A_i = A_j = A$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ), однако их производительность при решении подзадач идентичных типов  $A_d \in A$  различна, т.е.  $S_d^i \neq S_d^j$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ). Данная постановка соответствует гетерогенной ОВС, включающей в свой состав однотипные вычислительные ресурсы, однако обладающие различной производительностью.

**Вариант 3.** Здесь типы подзадач, решаемые различными ресурсами ОВС, также различны, т.е.  $A_i \neq A_j$ , причем  $A_i \subseteq A$  и  $A_j \subseteq A$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ), однако они могут и пересекаться, т.е.  $A_i \cap A_j \neq \emptyset$ . В то же время производительность различных ресурсов  $R_i \in R$  и  $R_j \in R$  при решении подзадач идентичных типов  $A_d \in A_i \cap A_j$  одинакова, т.е.  $S_d^i = S_d^j$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ). Данная постановка соответствует гетерогенной ОВС, состоящей из разнотипных по специализации вычислительных ресурсов, однако обладающих одинаковой производительностью при решении идентичных задач.

**Вариант 4.** В самом сложном варианте вычислительные ресурсы  $R_1, R_2, \dots, R_N$ , входящие в ОВС, ориентированы на решение подзадач различных

типов, т.е.  $A_i \neq A_j$  и  $A_i \cap A_j \neq \emptyset$  ( $A_i \subseteq A$ ,  $A_j \subseteq A$ ), причем их производительность при решении подзадач идентичных типов  $A_d \in A_i \cap A_j$  также различна, т.е.  $S_d^i \neq S_d^j$  ( $i=1, 2, \dots, N; j=1, 2, \dots, i-1, i+1, \dots, N$ ). Данная постановка соответствует гетерогенной ОВС, состоящей из полностью разнотипных и по специализации, и по производительности вычислительных ресурсов.

### Принципы мультиагентного диспетчирования облачной вычислительной среды

До недавнего времени проблема диспетчирования ОВС решалась, как правило, посредством специально выделенных серверных узлов, функции которых заключались в приеме пользовательских задач и распределении их по вычислительным ресурсам ОВС (например проект «Облако СО РАН») [3, 4]. Однако такая централизованная организация диспетчера имеет ряд недостатков. *Во-первых*, при большом числе разнородных ресурсов в ОВС их оперативное распределение по задачам (тем более по отдельным информационно связанным подзадачам) с помощью одного центрального диспетчера (дерева диспетчеров) крайне проблематично, причем данная проблема многократно усложняется в случае, если ОВС должна решать не одиночную задачу, а поток пользовательских задач, поступающих в заранее не известные моменты времени. *Во-вторых*, существенно затрудняется масштабирование ОВС (т.е. добавление в ее состав новых вычислительных ресурсов), что требует внесения изменений в алгоритмы работы диспетчера. *В-третьих*, ОВС с централизованным диспетчером имеет низкую отказоустойчивость, поскольку выход из строя служебного серверного узла, реализующего функции диспетчера, приводит к катастрофическим последствиям для всей ОВС в целом.

Указанных недостатков можно избежать, если использовать принципы децентрализованного мультиагентного диспетчирования ОВС [5]. При этом в качестве основных активных элементов ОВС предлагается использование программных агентов, физически реализуемых на каждом вычислительном ресурсе  $R_i \in R$  ( $i=1, 2, \dots, N$ ), входящем в состав ОВС и представляющем его «интересы» в процессе диспетчирования. Взаимодействие пользователей и агентов может осуществляться посредством некоторого множества специально выделенных пассивных узлов, составляющих служебный уровень облачной инфраструктуры и выполняющих функции *досок объявлений* (ДО). На рис. 2 показана структура ОВС с мультиагентным диспетчером.

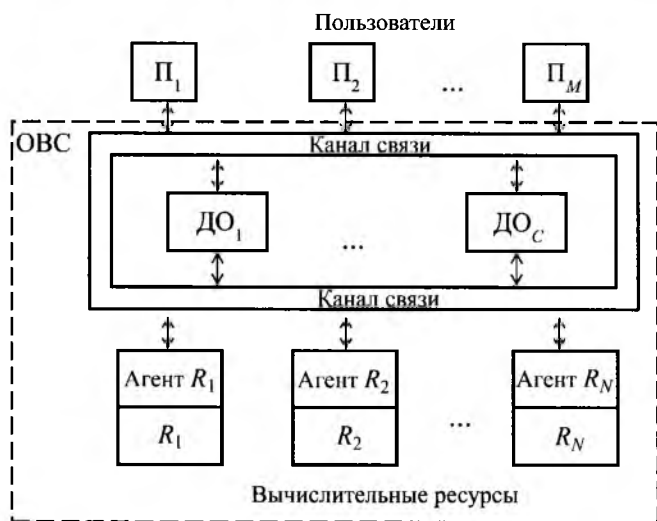


Рис. 2. Структура облачной вычислительной среды с мультиагентным диспетчером

Агенты периодически опрашивают ДО в целях загрузки «своего» ресурса решением пользовательских задач из потока  $Z$ . Если задача  $Z_l \in Z$  не может быть решена с помощью одного ресурса  $R_i$ , то агенты должны сформировать некоторую виртуальную организацию – сообщество, состоящее из подмножества ресурсов  $R_l \subseteq R$ , общая цель которых – решение пользовательской задачи к требуемому моменту времени [6]. При этом, для того чтобы каждый агент был заинтересован в максимальной загрузке «своего» ресурса  $R_i$ , необходимо предусмотреть некоторый механизм вознаграждения за его предоставление для решения задачи  $Z_l$ , величина которого зависит от числа премиальных баллов, установленного пользователем за успешное решение задачи, а также вклада ресурса  $R_i$  в ее решение.

В укрупненном виде работа гетерогенной ОВС с мультиагентным диспетчером описывается следующей последовательностью действий.

1. Пользователь формирует свою задачу  $Z_l \in Z$  в виде графа  $G_l(Q_l, X_l)$  и определяет требуемый момент времени  $T_{\max}^l$ , к которому он желает получить ее решение, а также число бонусов (премиальных баллов)  $O$ , которое он готов заплатить за ее решение. Дескриптор представленной таким образом задачи  $Z_l \in Z$  размещается на одной из ДО (см. рис. 2).

2. Агент ресурса  $R_i$ , не задействованного при решении других задач, опрашивает ДО в поисках работы для «своего» ресурса. В случае обнаруже-

ния на ДО пользовательской задачи  $Z_l \in Z$  агент оценивает целесообразность участия в ее решении. Для этого агент определяет относительную стоимость вычислений (соотношение числа премиальных баллов, предлагаемых пользователем за решение данной задачи, к ее суммарной вычислительной трудоемкости) и если данное значение оказывается ниже некоторого установленного порога, то агент отказывается от участия в решении задачи.

3. Если относительная стоимость решения удовлетворяет агента, то он делает попытку войти в состав сообщества  $R_l$  по решению данной задачи. Для этого агент  $R_i$  выделяет фрагмент задачи, состоящий из последовательности подзадач, не закрепленных за другими агентами, который он может решить к заданному моменту времени с помощью «своего» ресурса. Если такой фрагмент обнаруживается, то агент вступает в сообщество  $R_l$  по решению данной задачи и приступает к его решению.

4. Агент  $R_i$  контролирует ход решения принятого к исполнению фрагмента задачи, периодически оценивая время получения конечного результата. В случае, если агент не успевает обеспечить решение принятого фрагмента к заданному моменту времени, то он сообщает об этом на ДО и выходит из состава сообщества.

5. После успешного завершения решения задачи  $Z_l$  всем агентам сообщества  $R_l$ , принимавшим участие в ее решении, начисляются премиальные баллы, пропорциональные отношению объема выполненных ими вычислений к общей трудоемкости задачи  $Z_l$ .

Основное преимущество предлагаемого мультиагентного подхода к диспетчированию ресурсов в ОВС – адаптация вычислительного процесса к вычислительным возможностям ресурсов, задействованных в составе ОВС. Кроме того, по сравнению с классической централизованной организацией диспетчера облачной среды, в данном случае упрощаются требования к служебным серверам, что позволяет существенно сократить накладные расходы на организацию облачной среды и, как следствие, снизить стоимость облачных вычислений, а также упростить процесс масштабирования облачной среды [7].

Рассмотрим алгоритмы работы агента вычислительного ресурса  $R_p$  ( $p = 1, 2, \dots, N$ ), входящего в состав ОВС, для *четырёх* вариантов исходной постановки задачи диспетчирования.

**Гомогенная облачная вычислительная среда (вариант 1)**

В данном варианте все ресурсы  $R_p \in R$  ( $p = 1, 2, \dots, N$ ) совершенно идентичны, т.е. могут выполнять одинаковый набор подзадач  $A = \langle A_1, A_2, \dots, A_L \rangle$ , причем их производительность при решении подзадач идентичных типов также одинакова. Прежде чем приступить к разработке алгоритма работы агента, введем понятие «нити». Под *нитью* понимаем некоторую последовательность вершин  $H_f = \langle q_1^f, q_2^f, \dots, q_k^f \rangle$  графа  $G_l(Q_l, X_l)$  задачи  $Z_l \in Z$ , в которой вершины  $q_j^f$  и  $q_{j+1}^f$  ( $j = 1, 2, \dots, k-1$ ) соединены дугой  $x(q_j^f, q_{j+1}^f)$ . На рис. 3 показан пример выделения нити  $H_f$  в графе  $G_l(Q_l, X_l)$  задачи  $Z_l$ . Иными словами, нить определяет некоторую последовательность подзадач задачи  $Z_l$ , в которой каждая последующая подзадача использует в качестве исходных данных результат выполнения предыдущей подзадачи. Очевидно, что подзадачи, приписанные вершинам нити  $H_f$ , могут выполняться только последовательно. При этом под длиной  $t_f^p$  нити  $H_f$  будем понимать суммарное время, затрачиваемое на ее решение с помощью ресурса  $R_p$ ,

$$t_f^p = \sum_{i=1}^k \frac{V_i^f}{S_i^p} + t_{\Pi}^p, \quad (1)$$

где  $V_i^f$  – вычислительная трудоемкость подзадачи  $A_i$ , приписанной вершине  $q_i^f$  ( $i = 1, 2, \dots, k$ ) нити  $H_f$ ;

$S_i^p$  – производительность ресурса  $R_p$  при решении подзадачи  $A_i$ ;

$t_{\Pi}^p$  – время пересылки результата решения нити  $H_f$  потребителю (другому ресурсу ОВС или пользователю). Для простоты дальнейших рассуждений считаем, что время  $t_{\Pi}^p$  не зависит от конкретного вычислительного ресурса  $R_p$  и равно  $t_{\Pi}$ .

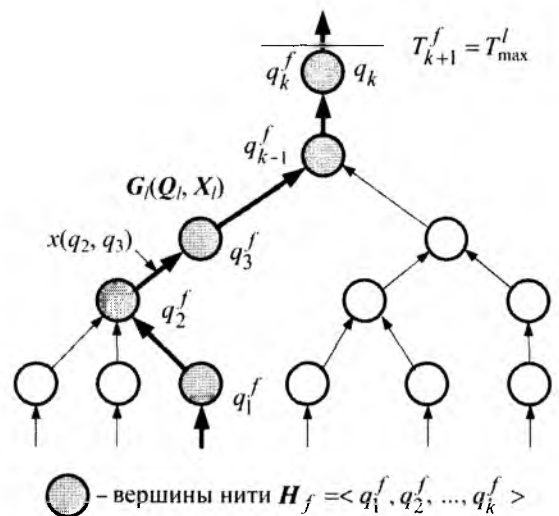
При этом, если требуемый момент времени  $T_{k+1}^f$  исполнения последней вершины  $q_k^f$  нити  $H_f$  известен (см. рис. 3), то определим требуемые мо-

менты времени начала выполнения всех подзадач  $A_d$ , приписанных вершинам  $q_d^f \in H_f$  ( $d = 1, 2, \dots, k$ ) как

$$T_d^f = T_{k+1}^f - \left( \sum_{i=d}^k \frac{V_i^f}{S_i^p} + t_{\Pi} \right), \quad d = 1, 2, \dots, k-1. \quad (2)$$

При размещении задачи  $Z_l$  на ДО требуемый момент времени исполнения, определяемый установленным потребителем временем  $T_{\max}^l$  выполнения всей задачи  $Z_l$ , приписан только конечной вершине  $q_k \in Q_l$  графа  $G_l(Q_l, X_l)$  (см. рис. 3).

Исходя из данных соображений, предложим следующую процедуру диспетчирования ресурсов  $R_1, R_2, \dots, R_N$ , входящих в ОВС, при выполнении потока задач  $Z$ . Агенты, ресурсы которых не задействованы в решении каких-либо задач, обращаются к ДО в поисках работы. Если агент свободного ресурса  $R_p \in R$  обнаруживает на ДО дескриптор задачи  $Z_l$ , то он оценивает целесообразность своего участия в сообществе  $R_l$  по ее решению. Для этого он определяет величину относительной стоимости вычислений как  $D = O/V$ , где  $O$  – число премиальных баллов, предлагаемое пользователем за ее решение;  $V = \sum_{i=1}^M V_i$  – суммарная вычислительная трудоемкость задачи  $Z_l$ , причем  $M = |Q_l|$ ,  $V_i$  – трудоемкость подзадачи  $A_i$ , приписанной вершине  $q_i \in Q_l$ .



**Рис. 3. Пример выделения нити  $H_f$  в графе  $G_l(Q_l, X_l)$  задачи  $Z_l$**

Если величина  $D$  ниже некоторого порога  $D_p$ , установленного для ресурса  $R_p \in R$ , то участие агента  $R_p$  в решении данной задачи считается экономически невыгодным, и он продолжает опрос ДО в поисках других задач. Иначе агент делает попытку войти в сообщество  $R_l$  по решению задачи  $Z_l$ .

Для этого агент  $R_p$  выделяет в графе  $G_l(Q_l, X_l)$ , хранящемся в дескрипторе задачи  $Z_l$  на ДО, наиболее длинную нить  $H_f = \langle q_1^f, q_2^f, \dots, q_k^f \rangle$  согласно (1), конечной вершине которой приписан момент времени ее исполнения  $T_{k+1}^f$  (см. рис. 3), что осуществляется с помощью одного из известных алгоритмов поиска максимального пути на графе [8]. Изначально, в момент размещения дескриптора задачи  $Z_l$  на ДО, таковой нитью является только нить  $H_1 = \langle q_1^1, q_2^1, \dots, q_k^1 \rangle$ , которая начинается на одной из входных, а заканчивается на выходной вершине графа  $G_l(Q_l, X_l)$ , т.е.  $q_k^1 = q_k$ . Очевидно, что выполнение данной нити должно быть завершено к заданному пользователем моменту  $T_{\max}^l$  решения всей задачи  $Z_l$  в целом, который также содержится в ее дескрипторе, т.е.  $T_{k+1}^1 = T_{\max}^l$ .

Далее агент  $R_p$  определяет согласно (2) момент времени  $T_1^1$ , когда ему необходимо приступить к выполнению первой подзадачи нити  $H_1$ , т.е. подзадачи  $A_1^1$ , приписанной вершине  $q_1^1 \in H_1$ , чтобы успеть завершить исполнение всей нити  $H_1$  к заданному моменту времени  $T_{k+1}^1$ :

$$T_1^1 = T_{k+1}^1 - \left( \sum_{i=1}^k \frac{V_i^1}{S_i^p} + t_{\Pi} \right). \quad (3)$$

Если  $T_1^1 < T_{\text{тек}}$ , где  $T_{\text{тек}}$  – текущий момент времени, то агент  $R_p$  не может обеспечить выполнение всей последовательности подзадач нити  $H_1 = \langle q_1^1, q_2^1, \dots, q_k^1 \rangle$  к заданному моменту времени  $T_{k+1}^1$ . Поскольку в данном варианте постановки задачи считается, что все ресурсы ОВС одинаковы и имеют одинаковую производительность при решении однотипных подзадач, то никакой другой ресурс также не сможет выполнить данную нить под-

задач к заданному моменту времени  $T_{k+1}^1$ . Поэтому агент  $R_p$  направляет на ДО сообщение о невозможности решения задачи  $Z_l$  к заданному пользователем моменту времени. В результате, задача  $Z_l$  снимается с ДО, а потребителю посылается сообщение о невозможности решения его задачи за отведенное время. Затем агент ресурса  $R_p$  вновь переходит к опросу ДО в поисках работы.

Если же условие  $T_1^1 \geq T_{\text{тек}}$  выполняется, то агент  $R_p$  принимает на себя исполнение последовательности операций, приписанных вершинам нити  $H_1$ . При этом агент  $R_p$  осуществляет модификацию дескриптора задачи  $Z_l$  на ДО, а именно:

- его идентификатор записывается в список членов сообщества  $R_l$  по решению задачи  $Z_l$ ;
- вершины, входящие в нить  $H_1$ , исключаются из графа  $G_l(Q_l, X_l)$  задачи  $Z_l$ , в результате формируется новый граф  $G_l^1(Q_l^1, X_l^1) = G_l(Q_l, X_l) / H_1$ . На рис. 4 представлен граф  $G_l^1(Q_l^1, X_l^1)$  задачи  $Z_l$ , модифицированный агентом  $R_p$ ;

– всем вершинам графа  $G_l^1(Q_l^1, X_l^1)$ , инцидентным вершинам нити  $H_1$ , приписываются требуемые моменты времени их исполнения. Допустим, что некоторая вершина  $q_f^2$  графа  $G_l^1(Q_l^1, X_l^1)$  инцидентна вершине  $q_b^1$ , принадлежащей нити  $H_1$  (см. рис. 4). Для выполнения подзадачи, приписанной вершине  $q_b^1$ , необходимы результаты выполнения подзадачи, приписанной вершине  $q_f^2$  графа  $G_l^1(Q_l^1, X_l^1)$ . Поэтому, очевидно, что результаты выполнения подзадачи, приписанной вершине  $q_f^2$  должны быть получены и переданы агенту  $R_p$ , выполняющему операции нити  $H_1$ , не позже чем к требуемому моменту времени начала выполнения подзадачи вершины  $q_b^1$ , определяемому согласно выражению (2) по формуле

$$T_b^1 = T_{k+1}^1 - \left( \sum_{i=b}^k \frac{V_i^1}{S_i^p} + t_{\Pi} \right). \quad (4)$$



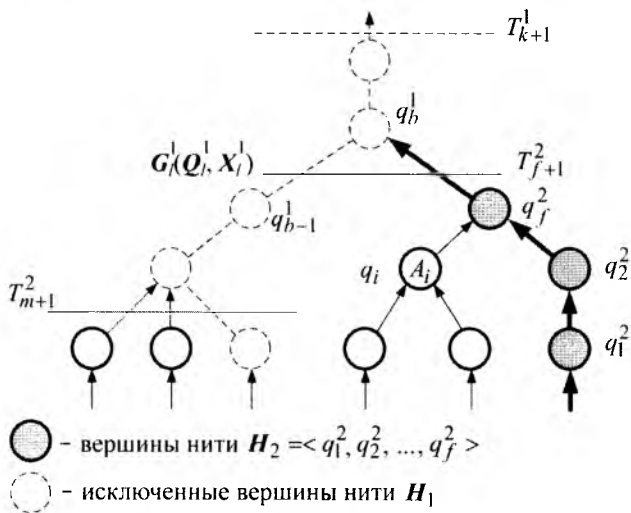


Рис. 4. Граф  $G_l^1(Q_l^1, X_l^1)$  задачи  $Z_l$ , модифицированный агентом  $R_p$

В противном случае, ресурс  $R_p$  не успеет закончить исполнение взятой на себя нити  $H_1$  к требуемому моменту времени  $T_{k+1}^1 = T_{\max}^1$ . Поэтому вершине  $q_f^2$  графа  $G_l^1(Q_l^1, X_l^1)$  приписывается требуемое время ее исполнения  $T_{f+1}^2 = T_b^1$ , а также идентификатор агента  $R_p$ , которому результаты исполнения подзадачи, приписанной вершине  $q_f^2$ , должны быть переданы. Аналогичным образом определяются требуемые моменты исполнения всех остальных вершин графа  $G_l^1(Q_l^1, X_l^1)$ , инцидентных вершинам нити  $H_1$  (см. рис. 4).

Если после модификации новый граф  $G_l^1(Q_l^1, X_l^1)$  задачи  $Z_l$  еще не пустой, т.е.  $G_l^1(Q_l^1, X_l^1) \neq \emptyset$ , то процесс создания сообщества  $R_l$  для выполнения задачи  $Z_l$  продолжается далее. Допустим, что через какое-то время другой свободный агент  $R_c$  ( $c$  – номер агента) обнаруживает на ДО дескриптор задачи  $Z_l$ . При этом агент  $R_c$  первым делом оценивает экономическую целесообразность своего участия в решении задачи  $Z_l$  – определяет значение относительной стоимости вычислений данной задачи как  $D = O/V$ , и если данное значение его устраивает, то предпринимает попытку войти в состав сообщества  $R_l$  по ее решению. Агент  $R_c$ , согласно выражению (1), выделяет в графе  $G_l^1(Q_l^1, X_l^1)$  (содержащемся в дескрипторе задания  $Z_l$  на ДО после модификации, произведен-

ной объектом  $R_p$ ) наиболее длинную нить  $H_2 = \langle q_1^2, q_2^2, \dots, q_f^2 \rangle$  (см. рис. 4), конечной вершине  $q_f^2$  которой приписано требуемое время исполнения  $T_{f+1}^2$  и определяет момент времени, когда необходимо приступить к ее выполнению,

$$T_1^2 = T_{f+1}^2 - \left( \sum_{i=1}^f \frac{V_i^2}{S_i^c} + t_n \right),$$

где  $V_i^2$  – вычислительная трудоемкость подзадачи  $A_i$ , приписанной вершине  $q_i^2$  ( $i = 1, 2, \dots, f$ ) нити  $H_2$ ;

$S_i^c$  – производительность ресурса  $R_c$  при решении подзадачи  $A_i$ .

Если  $T_1^2 < T_{\text{тек}}$ , то данная нить и, соответственно, вся задача  $Z_l$  в целом не могут быть выполнены к требуемому моменту времени. Поэтому задача  $Z_l$  снимается с ДО, а всем агентам, вошедшим ранее в сообщество  $R_l$  по ее решению (идентификаторы которых хранятся в дескрипторе задачи на ДО), сообщается об остановке данного задания, после чего они снова переходят в режим опроса ДО в поисках работы для «своих» ресурсов.

В случае, если  $T_1^2 \geq T_{\text{тек}}$ , то агент  $R_c$  принимает на себя исполнение операций, приписанных вершинам данной нити  $H_2$ , и осуществляет очередную модификацию дескриптора задания  $Z_l$  на ДО:

- идентификатор агента  $R_c$  записывается в список членов сообщества  $R_l$  по решению задачи  $Z_l$ ;

- вершины, входящие в нить  $H_2$ , исключаются из графа  $G_l^1(Q_l^1, X_l^1)$ , в результате чего образуется новый граф  $G_l^2(Q_l^2, X_l^2)$ . На рис. 5 показан граф  $G_l^2(Q_l^2, X_l^2)$  задачи  $Z_l$  после модификации агентом  $R_c$ ;

- вершинам  $q_p^3$  графа  $G_l^2(Q_l^2, X_l^2)$ , инцидентным вершинам  $q_d^2$  нити  $H_2$  приписывается идентификатор агента  $R_c$ , к которому результаты исполнения данных подзадач должны быть направлены, а также требуемое время их исполнения, определяемое по формуле

$$T_{p+1}^3 = T_{f+1}^2 - \left( \sum_{i=d}^f \frac{V_i^2}{S_i^c} + t_n \right).$$

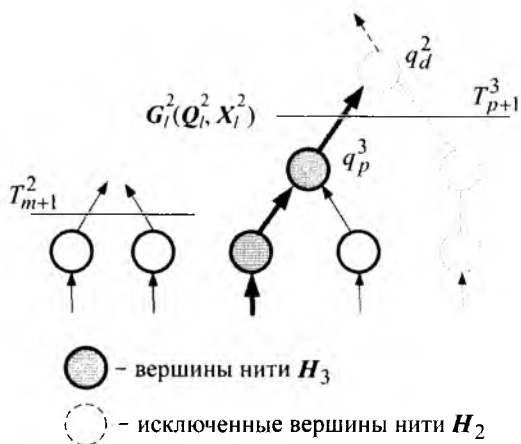


Рис. 5. Граф  $G_i^2(Q_i^2, X_i^2)$  задачи  $Z_i$  после модификации агентом  $R_c$

Далее в процесс распределения подзадач задачи  $Z_i$  включается следующий свободный агент, обнаруживший ее дескриптор на ДО, и так далее до тех пор, пока не окажется, что граф  $G_i^f(Q_i^f, X_i^f)$  стал пустым, т.е. все подзадачи задачи  $Z_i$  разобраны агентами, вошедшими в сообщество  $R_i$  по его выполнению.

После того, как некоторый агент  $R_p$  выбрал для исполнения нить  $H_f = \langle q_1^f, q_2^f, \dots, q_k^f \rangle$ , он приступает к реализации подзадач, приписанных ее вершинам. При этом перед началом выполнения очередной подзадачи  $A_d$ , приписанной вершине  $q_d^f \in H_f$  ( $d = 1, 2, \dots, k$ ), агент  $R_p$  должен проверить, во-первых, наличие всех исходных данных, необходимых для ее выполнения; во-вторых, соблюдение временного графика выполнения всей нити  $H_f$  в целом.

Если для выполнения подзадачи  $A_d$  требуются исходные данные, получаемые в результате выполнения смежной нити другим агентом  $R_c$ , то к моменту начала выполнения агентом  $R_p$  подзадачи  $A_d$  может оказаться, что они еще не поступили. В этом случае агент  $R_p$  должен перейти в режим ожидания поступления необходимых исходных данных, которое продолжается до тех пор, пока  $T_d^f \geq T_{тек}$ .

Если  $T_d^f < T_{тек}$ , то ресурс  $R_p$  уже не успевает выполнить оставшиеся подзадачи нити  $H_f$  к требуемому моменту времени. Поскольку в данном варианте принято, что производительность всех ресурсов одинакова, то никакой другой ресурс также не

успеет выполнить оставшуюся часть нити  $H_f$  к требуемому моменту времени. Поэтому агент  $R_p$  должен известить ДО о том, что произошло отставание от графика вычислительного процесса, и задача  $Z_i$  не может быть решена к установленному пользователем моменту времени. При этом задача снимается с ДО, пользователю сообщается о невозможности ее решения к заданному моменту времени, а всем агентам, вошедшим в сообщество  $R_i$  по ее решению, направляется сообщение о прекращении процесса решения задачи  $Z_i$ .

Если ресурс  $R_p$  успешно выполнил всю принятую на себя нить  $H_f$  задачи к заданному моменту времени, то агент  $R_p$  информирует об этом ДО. При этом идентификатор агента  $R_p$  ресурса исключается из списка участников сообщества  $R_i$  в дескрипторе задачи, и ему начисляются премиальные баллы за выполненную работу, число которых определяется следующим образом:

$$O_p = O V^p / V,$$

где  $O$  – премиальные баллы (бонусы), гарантируемые пользователем за решение его задачи  $Z_i$  к требуемому моменту времени;

$V^p$  – суммарная трудоемкость подзадач, выполненных агентом  $R_p$ ;

$V$  – суммарная трудоемкость всей задачи  $Z_i$ .

Процесс решения задачи  $Z_i$  в ОВС продолжается до тех пор, пока не станут пустыми список агентов-участников сообщества  $R_i$ , а также модифицированный граф данной задачи  $Z_i$ , что означает успешное выполнение всех нитей задачи. При этом условия задача снимается с ДО, а пользователю отправляется сообщение об успешном завершении процесса решения его задачи, после чего пользователь переводит оплату (премиальные баллы) всем агентам, принявшим участие в решении данной задачи в зависимости от объема выполненной ими работы (суммарной трудоемкости решенных подзадач).

Рассмотренному процессу (варианту 1) отвечает следующий укрупненный алгоритм функционирования агента, представляющего ресурс  $R_p$  в ОВС.

### Алгоритм 1

1. Агент свободного ресурса  $R_p$  опрашивает ДО.
2. При обнаружении на ДО задачи  $Z_i$  агент  $R_p$  считывает ее дескриптор и анализирует граф задачи

$G_i^j(Q_i^j, X_i^j)$ . Если  $G_i^j(Q_i^j, X_i^j) = \emptyset$ , то переход к пункту 1.

3. Агент  $R_p$  оценивает экономическую целесообразность своего участия в решении задачи  $Z_l$ , сравнивая предлагаемую пользователем относительную стоимость вычислений  $D = O/V$  с установленным для него порогом  $D_p$ . Если  $D < D_p$ , то переход к пункту 1.

4. Агент  $R_p$  согласно выражению (1) выделяет в графе  $G_i^j(Q_i^j, X_i^j)$  наиболее длинную нить  $H_j = \langle q_1^j, q_2^j, \dots, q_k^j \rangle$ , конечной вершине  $q_k^j$  которой приписано требуемое время ее исполнения  $T_{k+1}^j$  (если  $H_j = H_1$ , то  $T_{k+1}^j = T_{\max}^l$ ), и определяет согласно выражению (2) допустимый момент времени  $T_1^j$ , когда необходимо начать ее выполнение.

5. Если  $T_1^j < T_{\text{тек}}$ , то переход к пункту 14.

6. Агент  $R_p$  принимает на себя выполнение нити  $H_j$ , для чего модифицирует дескриптор задачи  $Z_l$  на ДО: записывает в список участников сообщества  $R_l$  свой идентификатор; модифицирует граф  $G_i^j(Q_i^j, X_i^j)$  задачи  $Z_l$  путем исключения из него вершин нити  $H_j$ , т.е.  $G_i^{j+1}(Q_i^{j+1}, X_i^{j+1}) = G_i^j(Q_i^j, X_i^j) / H_j$ ; приписывает вершинам  $q_f^{j+1}$  модифицированного графа  $G_i^{j+1}(Q_i^{j+1}, X_i^{j+1})$ , инцидентным вершинам  $q_b^j$  нити  $H_j$  идентификатор ресурса  $R_p$ , которому необходимо передать результаты исполнения вершины  $q_f^{j+1}$ , а также требуемое время их исполнения  $T_{f+1}^{j+1}$ , определяемое согласно выражению (4).

7. Агент  $R_p$  переходит к исполнению последовательности подзадач, приписанных вершинам нити  $H_j = \langle q_1^j, q_2^j, \dots, q_k^j \rangle$ ;  $d = 1$ .

8. Если  $T_d^j < T_{\text{тек}}$ , где  $T_d^j = T_{k+1}^j - \left( \sum_{i=d}^k \frac{V_i^j}{S_i^p} + t_n \right) -$

требуемое время начала выполнения подзадачи  $A_d$ , приписанной вершине  $q_d^j \in H_j$ , то переход к пункту 14.

9. Агент  $R_p$  проверяет наличие исходных данных, необходимых для выполнения подзадачи  $A_d$ . Если исходные данные еще не поступили, то переход к пункту 8.

10. Агент  $R_p$  решает подзадачу  $A_d$ , приписанную вершине  $q_d^j \in H_j$ , на «своем» вычислительном ресурсе.

11. Если агенту  $R_p$  поступило сообщение о прекращении решения задачи  $Z_l$ , то переход к пункту 1.

12.  $d = d + 1$ , если  $d \leq k$ , то переход к пункту 8.

13. Агент  $R_p$  сообщает на ДО об успешном завершении решения нити  $H_j$  задачи  $Z_l$ . При этом идентификатор агента  $R_p$  исключается из списка членов сообщества  $R_l$  по решению данной задачи, а также ему начисляются премиальные баллы  $O_p = OV^p / V$ . Переход к пункту 1.

14. Задача  $Z_l$  не может быть решена к требуемому пользователем моменту времени  $T_{\max}^l$ . Дескриптор задачи  $Z_l$  удаляется с ДО, пользователю направляется сообщение о невозможности решения его задачи к требуемому моменту времени, а всем агентам, идентификаторы которых записаны в списке участников сообщества  $R_l$  по выполнению задачи  $Z_l$ , передается сообщение о прекращении ее исполнения. Переход к пункту 1.

*Конец алгоритма 1.*

### Библиографический список

1. **Kepes Ben.** Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS [Электронный ресурс] // Rackspace. Support Network. URL: [http://www.rackspace.com/knowledge\\_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas](http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas) (дата обращения 01.09.2015).
2. **Kalyaev A. I.** Multiagent Approach for Building Distributed Adaptive Computing System [Электронный ресурс] // Procedia Computer Science: Intern. Conf. on Computational Science. 2013. V. 18. P. 2193 – 2202. doi:10.1016/j.procs.2013.05.390.
3. **Коннов А. Л.** Моделирование облачных технологий в вычислительных системах [Электронный ресурс]. URL: [http://conference.osu.ru/assets/files/conf\\_reports/conf9/436.doc](http://conference.osu.ru/assets/files/conf_reports/conf9/436.doc) (дата обращения: 01.09.2015).

4. **О проекте** [Электронный ресурс] // Облако СО РАН. URL: <http://cloud.sbras.ru/ru/about> (дата обращения: 01.09.2015).

5. **Каляев А. И.** Децентрализованная организация диспетчера GRID на базе сообществ агентов // Изв. ЮФУ. Технические науки. 2011. № 8 (121). С. 230 – 238.

6. **Каляев А. И.** Метод и алгоритмы адаптивной организации распределенных вычислений в децентрализованной GRID // Вестник компьютерных и информационных технологий. 2012. № 4. С. 28 – 33.

7. **Kalyaev A. I., Korovin Ya. S.** Adaptive Multiagent Organization of the Distributed Computations // AASRI Procedia. 2014. V. 6. P. 49 – 58. doi:10.1016/j.aasri.2014.05.008.

8. **Introduction to Algorithms** / Т. Н. Cormen et al. 2nd ed. Cambridge, Massachusetts London, England, 2001. 984 p. URL: <http://www.mif.vu.lt/~valdas/ALGORITMAI/LITERATURA/Cormen/Cormen.pdf> (дата обращения: 01.09.2015).

**References**

1. **Kepes Ben.** *Understanding the cloud computing Stack: SaaS, PaaS, IaaS.* Rackspace. Support Network. Available at: [http://www.rackspace.com/knowledge\\_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas](http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas) (Accessed 01.09.2015).

2. **Kalyaev A. I.** (2013). Multiagent approach for building distributed adaptive computing system. *Procedia*

*Computer Science: Intern. Conf. on Computational Science.* Vol. 18, pp. 2193 – 2202. doi:10.1016/j.procs.2013.05.390.

3. **Konnov A. L.** *Modeling of cloud technologies in computing systems.* Available at: [http://conference.osu.ru/assets/files/conf\\_reports/conf9/436.doc](http://conference.osu.ru/assets/files/conf_reports/conf9/436.doc) (Accessed: 01.09.2015).

4. *About the project.* Siberian Branch of the Russian Academy of Sciences cloud. Available at: <http://cloud.sbras.ru/ru/about> (Accessed: 01.09.2015).

5. **Kalyaev A. I.** (2011). Decentralized organization of GRID manager based on the agents communities. *Izvestiia IuFU. Tekhnicheskie nauki.* 121(8), pp. 230-238.

6. **Kalyaev A. I.** (2012). Method and algorithms of the adaptive organization for distributed calculations in decentralized GRID. *Vestnik komp'uternykh i informatsionnykh tekhnologii*, (4), pp. 28-33.

7. **Kalyaev A. I., Korovin Y. S.** (2014). Adaptive multiagent organization of the distributed computations. *AASRI Procedia.* Vol. 6, pp. 49-58. doi:10.1016/j.aasri.2014.05.008.

8. **Cormen T. H. et al.** (2001). *Introduction to algorithms.* (2nd ed.). Cambridge, Massachusetts London, England. Available at: <http://www.mif.vu.lt/~valdas/ALGORITMAI/LITERATURA/Cormen/Cormen.pdf> (Accessed: 01.09.2015).

*Продолжение следует*



**Шелихов Г.С.**

**МАГНИТОПОРОШКОВАЯ  
ДЕФЕКТОСКОПИЯ**

Под ред. В.В. Ключева. – М.: Издательский дом «Спектр», 2010. – 336 с.: ил. ISBN 978-5-904270-29-2

**Цена (с НДС10%) – 500 руб.**

Описаны технологические операции магнитопорошкового контроля. Приведены методики контроля конкретных деталей, расшифровки индикаторных рисунков дефектов, образуемых осаждением порошка над различными дефектами. Рассмотрены схемы и фотографии индикаторных рисунков мнимых дефектов и даны способы их расшифровки. Указаны области применения и технические характеристики дефектоскопов и приборов. Приведены примеры расчетов режимов намагничивания деталей, соответствующие требованиям стандарта по магнитопорошковой дефектоскопии.

Для специалистов промышленности, работников служб контроля, эксплуатации и ремонта, а также для специалистов, обслуживающих медицинскую и таможенную технику, может быть полезна преподавателям и студентам вузов.

ООО «Издательский дом «Спектр», 119048, Москва, ул. Усачева, д. 35, стр. 1

Заявки принимаются по почте, по телефону или по e-mail: [info@idspektr.ru](mailto:info@idspektr.ru), [izzakaz@rambler.ru](mailto:izzakaz@rambler.ru)

Контактные телефоны: **(495) 514-26-34, 514-76-50.** Подробная информация на сайте [www.idspektr.ru](http://www.idspektr.ru)