



А. Г. Феоктистов, канд. техн. наук, **Р. О. Костромин**, **Ю. А. Дядькин** (Институт динамики систем и теории управления им. В. М. Матросова Сибирского отделения РАН, Иркутск, Россия);
e-mail: agf65@yandex.ru

УПРАВЛЕНИЕ ЗАДАНИЯМИ НА ОСНОВЕ ЗНАНИЙ В ГЕТЕРОГЕННОЙ РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ*

Представлена новая система агентов для управления вычислениями на уровне среды. В отличие от известных систем она основана на использовании модели, включающей знания как о самой среде, так и предметных областях решаемых задач. Данные знания извлекаются посредством комплексного применения методов и средств концептуального моделирования, классификации заданий и параметрической настройки алгоритмов работы агентов распределения ресурсов. Эксперименты подтверждают практическую значимость предложенного подхода к управлению вычислениями.

Ключевые слова: распределенные вычисления; мультиагентное управление; самоорганизация; извлечение знаний; имитационное моделирование.

A. G. Feoktistov, R. O. Kostromin, Yu. A. Dyadkin (Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia)

KNOWLEDGE BASED MANAGEMENT OF JOBS IN HETEROGENEOUS DISTRIBUTED COMPUTING ENVIRONMENT

Ensuring effective management of applications for solving large problems in a heterogeneous distributed computing environment is the non-trivial problem. User applications compete with each other for shared resources of the environment. Their jobs contain different criteria of a problem solving quality. Preferences of resource owners are to optimize their use and often are not consistent with user criteria. The relevant approach to their harmonization is to use multi-agent technologies in combination with elements of economic theory. In this paper, we present a new system of agents for application management at the environment level. In contrast to the known systems, our system is based on applying a special conceptual model of the environment that includes knowledge about both the environment itself and the subject domains of the problems being solved. This knowledge is elicited through integrated use of methods and tools of conceptual modeling, job classification and parameter adjustment of agent functioning algorithms for a resource allocation. The multi-agent system has a hierarchical structure, which can include two or more functional levels, and operate based on self-organization. At each level, agents play a variety of roles, and perform different functions. The roles may be permanent or temporal. Their changes occur at discrete times when agents need to solve new problems. Each level is related with the conceptual model knowledge layers. Experiments confirm the practical significance of the proposed approach to the computations management.

Keywords: Distributed computing; Multi-agent management; Self-organization; Knowledge elicitation; Simulation modeling.

Статья поступила в редакцию 08.09.2017 г.

* Работа выполнена при поддержке РФФИ, проект № 16-07-00931 и Президиума РАН, программа № 30, проект «Методы, алгоритмы и инструментальные средства децентрализованного группового решения задач в вычислительных и управляющих системах».

Введение

В настоящее время высокопроизводительные вычисления являются неотъемлемым звеном процесса решения сложных научных и прикладных задач широкого спектра, которые зачастую существенно различаются своими требованиями к вычислительным системам. Это приводит к разнообразию необходимых программно-аппаратных платформ и коммуникационных сред, а также актуализирует исследования, связанные с интеграцией данных компонентов в единую *гетерогенную распределенную вычислительную среду* (ГРВС). Управление заданиями в такой среде – нетривиальная проблема.

Выделяются два подхода к управлению в ГРВС: на уровне среды и уровне приложений [1]. В рамках *первого* подхода управление, как правило, ориентировано на обеспечение эффективности работы среды в целом и может ухудшать показатели (например, время или стоимость) выполнения заданий отдельных приложений. *Второй* подход в большей степени используется для оптимизации работы приложений. Большое число приложений обостряет конкуренцию за ресурсы, что зачастую приводит к снижению степени их полезного использования и, как следствие, ухудшению качества обслуживания самих приложений.

На уровне среды применяются различные метапланировщики, например, GridWay [2], обеспечивающие распределение заданий между ее узлами, где функционируют *системы управления прохождением заданий* (СУПЗ). К их числу относятся, например, такие известные зарубежные системы, как PBS (*Portable Batch System*) [3] и HTCondor (*High Throughput Computing with Condor*) [4]. Известные российские разработки – *система управления прохождением параллельных заданий* (СУППЗ) [5] и система управления заданиями Cleo [6]. Организация взаимодействия между метапланировщиками и СУПЗ, как правило, базируется на использовании компонентов *связующего программного обеспечения* (СПО), такого как пакет Globus Toolkit [7]. К средствам, реализующим управление вычислениями на уровне приложений, относят AppLeS (*Application-Level Scheduling*) [8], Nimrod/G [9] и другие системы.

Связующее программное обеспечение – достаточно сложное для применения пользователями ГРВС, т.е. специалистами в своих предметных областях, не обладающими необходимыми навыками системного программирования и администрирования. В этой связи активно разрабатываются системы управления потоками заданий, упоминаемые в зарубежной литературе как *WMS (Workflow Management System)* [10]. В таких системах вопросы организации взаимодействия с компонентами СПО, метапланировщиками и СУПЗ в процессе управления вычислениями по большей части экранированы от пользователей. В то же время WMS предоставляют пользователям в той или иной степени возможности учета специфики предметной области решаемых задач. К классу WMS относятся: Askalon, Kepler, Pegasus, Taverna, Triana и другие программные комплексы [11].

Анализ возможностей вышеперечисленных систем, а также тенденций развития средств и методов управления вычислениями в ГРВС [12, 13] позволяет сделать вывод о необходимости интеллектуализации процесса управления, как можно более полного учета специфики предметных областей решаемых задач и согласования предпочтений владельцев ресурсов среды с критериями качества выполнения заданий. Актуальным подходом к решению данных проблем является применение мультиагентных технологий, основанных на самоорганизации агентов [14] в сочетании с элементами экономической теории [15].

В статье представлена *мультиагентная система* (МАС) управления приложениями в ГРВС, базирующаяся на использовании знаний как о самой среде, так и предметных областях решаемых задач. Знания, необходимые для эффективной работы МАС, извлекаются посредством комплексного применения методов и средств концептуального моделирования, классификации заданий и параметрической настройки алгоритмов работы агентов.

Мультиагентная система

Мультиагентная система относится к классу метапланировщиков, имеет иерархическую структуру и функционирует на основе принципов самоорганизации. На каждом уровне агенты выпол-

няют разные функции. Управление вычислениями реализовано на локальных взаимодействиях агентов. Все агенты имеют свои базы знаний. На каждом уровне иерархии агенты могут объединяться в *виртуальные сообщества (ВС)*, кооперироваться и конкурировать в их рамках. Использование ВС позволяет адаптировать процесс управления к решению новых задач. Данная система применяет распределение ресурсов с помощью экономических механизмов регулирования их спроса и предложения. Агенты организуют тендер вычислительных работ [16], в основе которого лежит модель аукциона Викри [17]. В процессе решения задач возможно динамическое перераспределение работ.

Задания пользователей ГРВС направляются в общую очередь СУПЗ узлов среды, выбранных агентами МАС. Для получения оценки текущего состояния среды используется специализированная система метамониторинга [18], осуществляющая сбор данных с помощью локальных систем мониторинга и контрольно-измерительных средств, размещенных на кластерах.

Известны различные подходы к извлечению знаний [19]. В МАС реализуются следующие методы:

- концептуальное моделирование, позволяющее разработчикам приложений представлять результаты своего экспертного анализа предметных областей;
- классификация заданий, обеспечивающая учет специфики вычислительных работ и их сопоставление имеющимся ресурсам на основе практического опыта администраторов среды;
- имитационное моделирование, результаты которого используются для параметрической настройки алгоритмов работы агентов распределения ресурсов.

Концептуальное моделирование обеспечивает формализацию знаний об основных объектах предметной области и связях между ними, атрибутах объектов и их характеристиках, правилах и процессах функционирования объектов, событиях, фактах и явлениях, свойственных данной области, а также методах решения возникающих в ней задач. *Классификация заданий* представляет собой активный способ извлечения знаний и рассматривается как аналог анкетирования

в сочетании с сопоставлением. *Имитационное моделирование*, необходимое для параметрической настройки алгоритмов работы агентов, относится к пассивным методам извлечения знаний, таким как наблюдение. В свою очередь параметрическая настройка – элемент обучения агентов.

Концептуальная модель ГРВС [20], используемая МАС, включает в себя:

- вычислительные знания о программных модулях приложений для решения задач и работы с системными объектами среды;
- схемные знания о модульной структуре алгоритмов;
- производственные знания, обеспечивающие поддержку принятия решений для выбора оптимальных алгоритмов в зависимости от состояния среды;
- знания о программно-аппаратной инфраструктуре среды и административных политиках в ее узлах.

Для построения концептуальной модели разработан высокоуровневый инструментальный SIRIUS II [21].

Система классификации предназначена для улучшения процесса распределения ресурсов. Она позволяет получать знания от администраторов об их соответствии заданиям, включающим такие характеристики, как время решения задачи, объемы оперативной и дисковой памяти, число узлов, процессоров и ядер, используемые библиотеки программ и *операционную систему (ОС)*, а также другую необходимую информацию. Распознавание свойств заданий осуществляется с помощью набора характеристических функций. Классифицированное задание конкретизируется путем добавления в него директив для СУПЗ, определяющих ресурсы, допустимые для его выполнения. Обработанное задание направляется агентам ВС, представляющим данные ресурсы. Виртуальное сообщество проводит их окончательный выбор и распределение.

Алгоритмы работы агентов имеют следующие управляющие параметры: ограничения загрузки компонентов узла; бонусы за удовлетворение данным ограничениям; штрафы за их превышение; приоритеты классов; степень желания выполнять задания определенных классов.

Администраторы ГРВС задают начальные значения таких параметров при настройке МАС, которые в дальнейшем автоматически корректируются в процессе ее работы.

На каждом кластере ГРВС, интегрирующей ресурсы *Института динамики систем и теории управления им. В. М. Матросова СО РАН (ИДСТУ СО РАН), Иркутского суперкомпьютерного центра СО РАН (ИСКЦ) и Иркутского государственного университета (ИГУ)*, установлена своя СУПЗ. Схема параметрической настройки алгоритмов работы агентов показана на рис. 1. Кластеры представлены в среде агентами ВС; в наличии имеется несколько машин-шлюзов с агентами пользователей для постановки задач, формирования заданий, их классификации и конкретизации. Шлюзы служат точками ввода потоков работ в среду. База знаний системы классификации размещена в системном хранилище данных. Обработанные задания передаются агентам ВС, подходящим для выполнения таких заданий.

Параметрическая настройка осуществляется агентом, размещенным в управляющем узле

ГРВС (см. рис. 1) и использующим имитационную модель среды. Ее входные и наблюдаемые переменные, упорядоченные по степени их важности, представлены векторами x и y . Входные переменные модели включают: параметры обрабатываемого потока заданий (число заданий, интенсивность их поступления и время обработки, требуемые ресурсы); вычислительные характеристики узлов и коммуникационной сети ГРВС; конфигурационные параметры СУПЗ (число очередей, дисциплины и периодичность их обслуживания, квоты на использование ресурсов и другие параметры), а также управляющие параметры алгоритмов работы агентов. К наблюдаемым переменным относятся, например, показатели состояния очередей СУПЗ и загрузки ресурсов в ВС агентов.

На основе многовариантных расчетов, организуемых путем варьирования значений управляющих параметров алгоритмов работы агентов, определяются варианты (векторы) y_1, y_2, \dots, y_m значений наблюдаемых переменных (показателей эффективности работы среды).

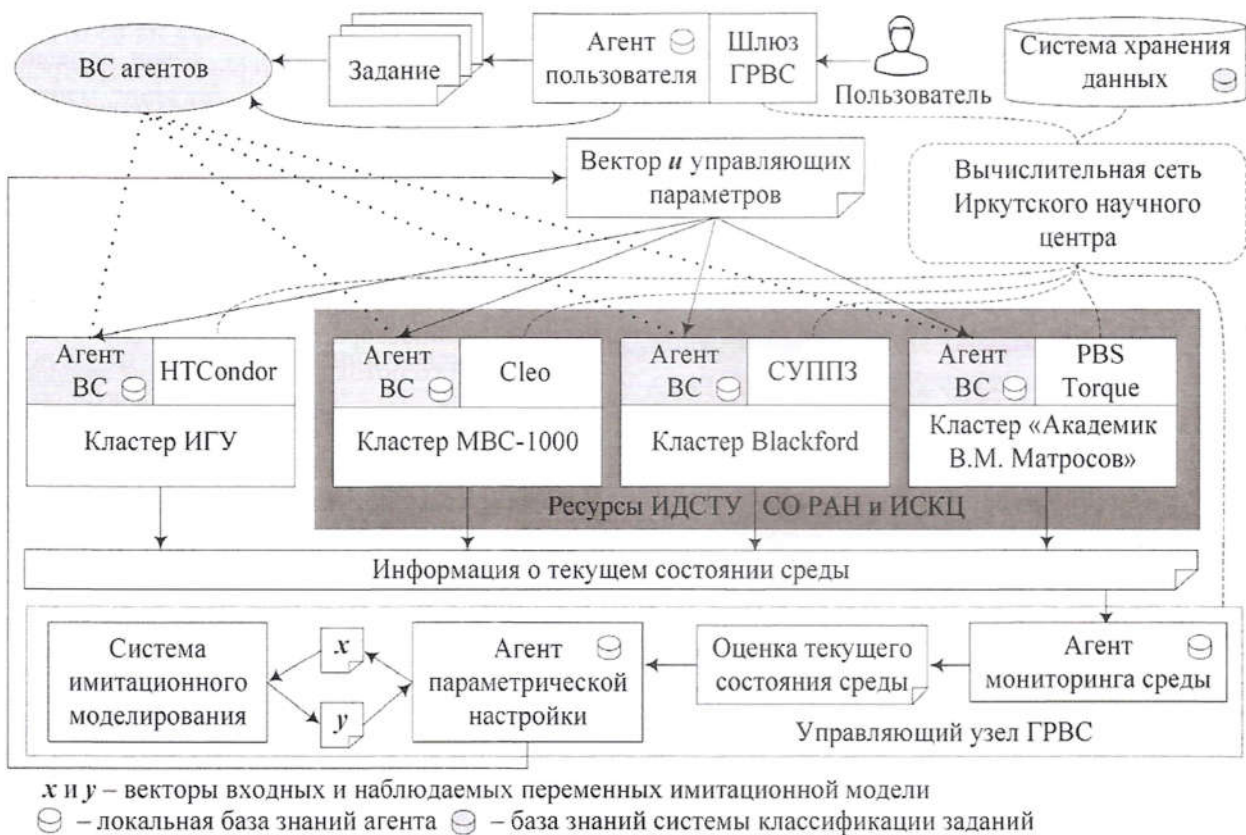


Рис. 1. Схема параметрической настройки алгоритмов работы агентов

Каждому вектору y_i соответствует вектор x_i значений входных переменных, $x_{i,k} \in D_k$, где D_k – область допустимых значений элемента $x_{i,k}$ вектора x_i , $k = \overline{1, p}$.

Далее решается прямая задача нахождения множества $S_{\text{опт}}^y$ оптимальных вариантов значений наблюдаемых переменных с помощью лексикографического правила многокритериального выбора:

$$S_{\text{опт}}^y = \{y_q : \forall y_r \exists s : (\hat{y}_{q,1} = \hat{y}_{r,1}) \wedge (\hat{y}_{q,2} = \hat{y}_{r,2}) \wedge \dots \wedge (\hat{y}_{q,s} = \hat{y}_{r,s}) \wedge (\hat{y}_{q,s+1} > \hat{y}_{r,s+1})\}; \quad (1)$$

$$y_{i,j}(x_i) \rightarrow \min(\max), \quad l_j^{y-\min} \leq y_{i,j} \leq l_j^{y-\max}, \quad (2)$$

где $\hat{y}_{i,j}$ – оценка значения $y_{i,j}$, формируемая автоматически с учетом условий (2);

$l_j^{y-\min}$, $l_j^{y-\max}$ – предельные значения j -го элемента вектора y , $i = \overline{1, m}$, $j = \overline{1, n}$, $q \in \overline{1, m}$, $r \in \overline{1, m}$, $q \neq r$ и $s \in \overline{1, n-1}$.

Если множество $S_{\text{опт}}^y$ содержит единственный вектор y_q , то варьируемые переменные соответствующего ему вектора x_q отождествляются с элементами вектора u (управляющими параметрами). В противном случае, решается обратная задача нахождения множества $S_{\text{опт}}^x$ оптимальных вариантов значений входных переменных:

$$S_{\text{опт}}^x = \{x_q : \forall x_r \exists s : (\hat{x}_{q,1} = \hat{x}_{r,1}) \wedge (\hat{x}_{q,2} = \hat{x}_{r,2}) \wedge \dots \wedge (\hat{x}_{q,s} = \hat{x}_{r,s}) \wedge (\hat{x}_{q,s+1} > \hat{x}_{r,s+1})\}; \quad (3)$$

$$x_{i,j} \rightarrow \min(\max), \quad l_j^{x-\min} \leq x_{i,j} \leq l_j^{x-\max}, \quad (4)$$

где $\hat{x}_{i,j}$ – оценка значения $x_{i,j}$ с учетом условий (4);

$l_j^{x-\min}$, $l_j^{x-\max}$ – предельные значения j -го элемента вектора x , $i = \overline{1, v}$, $j = \overline{1, p}$, $q \in \overline{1, v}$, $r \in \overline{1, v}$, $q \neq r$ и $s \in \overline{1, p-1}$.

Если множество $S_{\text{опт}}^x$ содержит единственный вектор x_q , то значения его варьируемых переменных принимаются за значения элементов вектора u . В противном случае, оптимальный вектор выбирается из множества $S_{\text{опт}}^x$ случайным образом.

Упорядочение входных и наблюдаемых переменных, а также задание для них типа экстремума и предельных значений в формулах (2) и (4) осуществляется администратором ГРВС. Для оценки значений векторов x и y по степени их близости к заданному типу экстремума разработан специальный алгоритм [18].

Вычислительные эксперименты

Преимущества применения разработанных средств извлечения знаний агентами демонстрируются на примере оценки возможности перераспределения реального потока из 83 075 заданий между тремя кластерами рассмотренной ГРВС. В табл. 1 приведены характеристики вычислительных кластеров, при этом использованы следующие обозначения: R_{max} – пиковая производительность; n_y , n_n и $n_{\text{я}}$ – число узлов, процессоров и ядер соответственно; $l_{\text{ср}}$ – средняя загрузка процессора за период реальной обработки данного потока. Распределение заданий по времени t их выполнения и числу $n_{\text{я}_j}$ запрашиваемых ядер отражены на рис. 2. Эти данные показывают существенное преобладание работ со временем t менее 5 мин или требующих для их обслуживания от 2 до 10 ядер.

1. Характеристики вычислительных кластеров

Кластер	R_{max} , Тфлопс	Процессор	$n_y/n_n/n_{\text{я}}$	ОС	СУПЗ	$l_{\text{ср}}$, %
Blackford (ИСКЦ)	1,50	Intel Xeon E5345	20/40/160	Linux	Cleo	70
MBC-1000 (ИСКЦ)	0,17	Intel Xeon	16/32/32		СУППЗ	41
Кластер ИГУ	0,77	AMD Athlon II X4	16/16/64	Windows	HTCondor	28

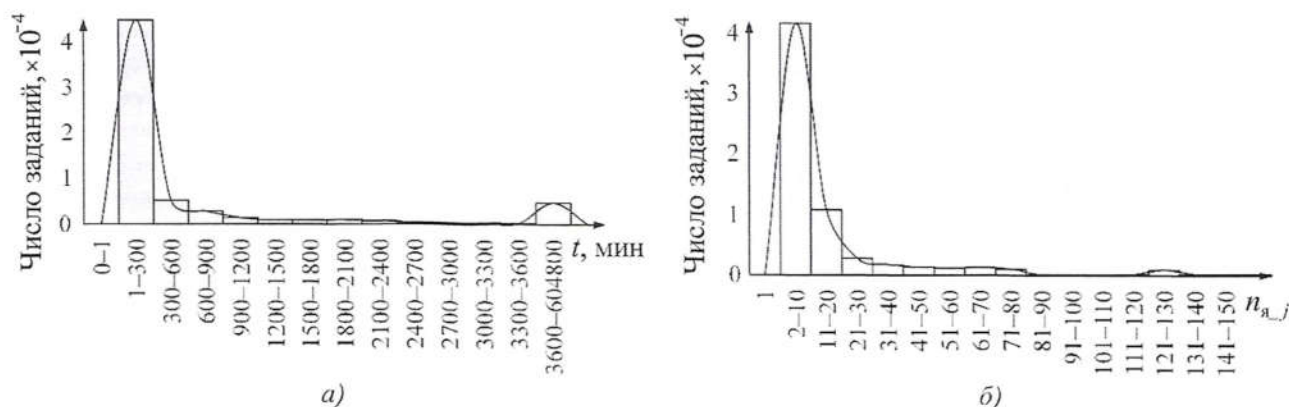


Рис. 2. Распределение заданий по времени их выполнения (а) и числу используемых ядер (б)

Кластеры Blackford и МВС-1000 включают узлы, используемые только в рамках данных вычислительных систем. Узлы кластера ИГУ задействуются как их владельцами, так и его пользователями. В системе Cleo все ядра узла выделяются заданию, независимо от числа запрошенных в нем ядер, что часто приводит к неэффективному использованию ресурсов. Дополнительные накладные расходы на виртуализацию узлов кластера Blackford возникают при запуске на нем Windows-приложений. В дневное время задания с малым и средним временем счета обладают более высокими приоритетами по сравнению с заданиями с большим временем их выполнения.

Эксперимент осуществлен на основе имитационного моделирования с учетом загрузки кластеров во время обработки рассматриваемого потока. Вначале проведен анализ вычислительной истории заданий и для них сформирован набор родительских классов $c_1 - c_{12}$, включающих характеристики t и $n_{я,j}$ (табл. 2). На базе родительских классов построены дополнительные классы с учетом требуемого размера оперативной памяти и дискового пространства, опера-

ционной системы, прикладного программного обеспечения и других сведений. Дополнительные классы сопоставлены тем кластерам, на которых задания данных классов могут быть выполнены наилучшим образом с точки зрения администраторов ГРВС. Для каждого агента распределения ресурсов определена степень его намерений обрабатывать задания того или иного класса.

Для проведения эксперимента с помощью комплекса SIRIUS II разработана модель ГРВС на языке GPSS (General Purpose Simulation System) [22]. В ней сформирован поток заданий на основе их вычислительной истории. Каждое задание потока классифицировано и передано СУПЗ кластера, выбранного агентами ВС в процессе торгов.

В процессе моделирования перераспределено более 10 000 заданий, сохранивших при этом допустимые значения характеристик своих классов. Средний коэффициент полезного использования процессоров кластера Blackford вырос на 19% в связи с отправкой в другие вычислительные системы части заданий с числом требуемых ядер, не кратным 8, а также заданий по выполнению Windows-приложений.

2. Характеристики родительских классов

Характеристика	Родительский класс											
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
$n_{я,j}$	1			2...32			2...64			> 64		
t , мин	< 5	5...60	> 60	< 5	5...60	> 60	< 5	5...60	> 60	< 5	5...60	> 60

В моделируемый период средняя загрузка процессоров кластеров (см. табл. 1) улучшена соответственно на 17, 24 и 29 %. Увеличение средней загрузки процессора для кластера Blackford обусловлено тем, что ряд больших заданий (с временем t , варьируемым в пределах нескольких суток, и $n_{я_j}=160$), снятых на практике в связи с превышением максимально допустимого времени пребывания в очереди, выполнен в процессе моделирования.

Заключение

Рассмотрена проблема управления заданиями в гетерогенной распределенной вычислительной среде. Представлена мультиагентная система для управления приложениями в такой среде, базирующаяся на использовании знаний как о самой среде, так и специфике решаемых задач. Эксперименты, проведенные на основе данных о применении ресурсов Иркутского суперкомпьютерного центра СО РАН [23], показали практическую значимость предложенного подхода к мультиагентному управлению заданиями на основе знаний.

Библиографический список

1. **Топорков В. В., Емельянов Д. М., Потехин П. А.** Формирование и планирование пакетов заданий в распределенных вычислительных средах // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2015. Т. 4, № 2. С. 44 – 57.
2. **GridWay** [Электронный ресурс]. URL: <https://www.gridway.org> (дата обращения: 05.09.2017).
3. **Global Leader in HPC Workload Management – PBS Works** [Электронный ресурс]. URL: <http://www.pbsworks.com/> (дата обращения: 05.09.2017).
4. **HTCondor** [Электронный ресурс]. URL: <http://research.cs.wisc.edu/htcondor> (дата обращения: 05.09.2017).
5. **Система управления прохождением параллельных заданий СУППЗ** [Электронный ресурс]. URL: <http://suppz.jssc.ru> (дата обращения: 05.09.2017).
6. **Система управления заданиями Cleo** [Электронный ресурс]. URL: <http://parcon.parallel.ru/cleo.html> (дата обращения: 05.09.2017).
7. **Globus Toolkit** [Электронный ресурс]. URL: <http://toolkit.globus.org/toolkit/> (дата обращения: 05.09.2017).
8. **Grid Computing Laboratory Home Page: AppLeS Group** [Электронный ресурс]. URL: <http://cseweb.ucsd.edu/groups/gcl> (дата обращения: 05.09.2017).
9. **Nimrod: A Tool for Performing Parameterized Simulations Using Distributed Workstations** / D. Abramson et al. // Proc. of the 4th IEEE Symp. on High Performance Distributed Computing. IEEE, 1995. P. 112 – 121.
10. **Examining the Challenges of Scientific Workflows** / Y. Gil, E. Deelman, M. Ellisman et al. // IEEE Computer. 2008. V. 40, No. 12. P. 24 – 32.
11. **Talia D.** Workflow Systems for Science: Concepts and Tools // ISRN Software Engineering. 2013. V. 2013. Article ID 404525. 15 p.
12. **Negotiation Mechanism for Self-Organized Scheduling System with Collective Intelligence** / A. Madureira et al. // Neurocomputing. 2014. V. 132. P. 97 – 110.
13. **A Note on New Trends in Data-Aware Scheduling and Resource Provisioning in Modern HPC Systems** / J. Tao et al. // Future Generation Computer Systems. 2015. V. 51, No. C. P. 45 – 46.
14. **Каляев А. И., Каляев И. А., Коровин Я. С.** Метод мультиагентного диспетчирования ресурсов в гетерогенной облачной среде при выполнении потока задач // Вестник компьютерных и информационных технологий. 2015. № 11. С. 31 – 40.
15. **Топорков В. В., Yemelyanov D. M.** Economic Model of Scheduling and Fair Resource Sharing in Distributed Computations // Programming and Computer Software. 2014. V. 40, No. 1. P. 35 – 42.
16. **Мультиагентный алгоритм распределения вычислительных ресурсов на основе экономического механизма регулирования их спроса и предложения** / И. В. Бычков и др. // Вестник компьютерных и информационных технологий. 2014. № 1. С. 39 – 45.
17. **Vickrey W.** Counterspeculation, Auctions, and Competitive Sealed Tenders // Journal of Finance. 1961. V. 16, No. 1. P. 8 – 37.
18. **Multiagent Control of Computational Systems on the Basis of Meta-Monitoring and Imitational Simulation** / I. V. Bychkov et al. // Optoelectronics, Instrumentation and Data Processing. 2016. V. 52, No. 2. P. 107 – 112.
19. **Cooke N. J.** Varieties of Knowledge Elicitation Techniques // Intern. Journal of Human-Computer Studies. 1994. V. 41, No. 6. P. 801 – 849.
20. **Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management** / I. Bychkov et al. // Procedia Computer Science. 2017. V. 103. P. 162 – 167.

21. Дядькин Ю. А., Фереферов Е. С. Инструментальный комплекс имитационного моделирования разнородной распределенной вычислительной среды // Вычислительные технологии. 2016. Т. 21, № 3. С. 18 – 32.

22. Боев В. Д. Моделирование систем. Инструментальные средства GPSS World. СПб.: БХВ-Петербург, 2004. 368 с.

23. ЦКП Иркутский суперкомпьютерный центр СО РАН [Электронный ресурс]. URL: <http://hpc.icc.ru/> (дата обращения: 05.09.2017).

References

1. Toporkov V. V., Emel'ianov D. M., Potekhin P. A. (2015). Forming and scheduling the tasks in distributed computing environments. *Vestnik Iuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naiia matematika i informatika*, 4(2), pp. 44-57. [in Russian language]
2. *GridWay*. Available at: <https://www.gridway.org> (Accessed: 05.09.2017).
3. Global leader in HPC workload management – PBS works. Available at: <http://www.pbsworks.com/> (Accessed: 05.09.2017).
4. *HTCondor*. Available at: <http://research.cs.wisc.edu/htcondor> (Accessed: 05.09.2017).
5. *Control system for the passage of parallel jobs SUPPZ*. Available at: <http://suppz.jssc.ru> (Accessed: 05.09.2017).
6. *Tasks control system Cleo*. Available at: <http://parcon.parallel.ru/cleo.html> (Accessed: 05.09.2017).
7. *Globus toolkit*. Available at: <http://toolkit.globus.org/toolkit/> (Accessed: 05.09.2017).
8. *Grid computing laboratory home page: AppLeS group*. Available at: <http://cseweb.ucsd.edu/groups/gcl> (Accessed: 05.09.2017).
9. Abramson D. et al. (1995). *Nimrod: a tool for performing parameterized simulations using distributed workstations*. Proc. of the 4th IEEE Symp. on High Performance Distributed Computing. IEEE, (pp. 112-121).
10. Gil Y., Deelman E., Ellisman M. et al. (2008). Examining the challenges of scientific workflows. *IEEE Computer*, 40(12), pp. 24-32.
11. Talia D. (2013). Workflow systems for science: concepts and tools. *ISRN Software Engineering*, 2013. Article ID 404525.
12. Madureira A. et al. (2014). Negotiation mechanism for self-organized scheduling system with collective intelligence. *Neurocomputing*, 132, pp. 97-110.
13. Tao J. et al. (2015). A note on new trends in data-aware scheduling and resource provisioning in modern HPC systems. *Future Generation Computer Systems*, 51(C), pp. 45-46.
14. Kaliaev A. I., Kaliaev I. A., Korovin Ia. S. (2015). Method of multiagent dispatching resources in heterogeneous cloud environments while performing flow of incoming tasks. *Vestnik komp'iuternykh i informatsionnykh tekhnologii*, (11), pp. 31-40. doi: 10.14489/vkit.2015.11.pp.031-040 [in Russian language]
15. Toporkov V. V., Yemelyanov D. M. (2014). Economic model of scheduling and fair resource sharing in distributed computations. *Programming and Computer Software*, 40(1), pp. 35-42.
16. Bychkov I. V., Oparin G. A., Feoktistov A. G., Kanter A. N. (2014). Multiagent algorithm for resources allocation based on the economic mechanism of regulating their supply and demand. *Vestnik komp'iuternykh i informatsionnykh tekhnologii*, (1), pp. 39-45. [in Russian language]
17. Vickrey W. (1961). Counterspeculation, Auctions, and competitive sealed tenders. *Journal of Finance*, 16(1), pp. 8-37.
18. Bychkov I. V. et al. (2016). Multiagent control of computational systems on the basis of meta-monitoring and imitational simulation. *Optoelectronics, Instrumentation and Data Processing*, 52(2), pp. 107-112.
19. Cooke N. J. (1994). Varieties of knowledge elicitation techniques. *Intern. Journal of Human-Computer Studies*, 41(6), pp. 801-849.
20. Bychkov I. V. et al. (2017). Conceptual model of problem-oriented heterogeneous distributed computing environment with multi-agent management. *Procedia Computer Science*, 103, pp. 162-167.
21. Diad'kin Iu. A., Fereferov E. S. (2016). Instrumental complex of simulation heterogeneous distributed computing environment. *Vychislitel'nye tekhnologii*, 21(3), pp. 18 – 32. [in Russian language]
22. Boev V. D. (2004). *Modeling the systems. GPSS World tools*. St. Petersburg: BKHV-Peterburg. [in Russian language]
23. *CCU Irkutsk Supercomputer Center of the SB RAS*. Available at: <http://hpc.icc.ru/> (Accessed: 05.09.2017).