

AUTOMATION OF DEVELOPMENT AND APPLICATION OF DISTRIBUTED APPLIED SOFTWARE PACKAGES

A. G. Feoktistov, I. F. Sidorov, S. A. Gorsky

Matrosov Institute for System Dynamics and Control Theory of SB RAS
664033, Irkutsk, Russia

We address relevant problems of constructing subject-oriented systems of modular programming (applied software packages) for solving large-scale problems in a heterogeneous distributed computing environment that integrates computational resources of a public access computer center where computational clusters are the main components.

Nowadays, the software that implements computational technologies in high-performance computing systems provides a potential foundation for the mass creation of parallel and distributed applications. However, the analysis of their application in practice makes it possible to detect the following important problems: the complexity of training and applying these tools by end-users (specialists in subject areas), which significantly narrows their number; insufficient support for taking into account the subject area specifics of the tasks being solved; difficulty in reconciling the determined criteria for the effectiveness of their solving (time, cost, security and other restrictions) and the required indicators of resource use (load balancing, reliability, energy consumption, etc.); lack of universal approaches to resource integration; the weak interconnecting of the created applications for data and management. In particular, the above-mentioned problems are actualized when solving tasks in a heterogeneous distributed computing environment, the main components of which are computational clusters, including hybrid clusters with heterogeneous nodes.

An actual approach to solving these problems is the development of distributed applied software packages. The development of distributed computing technologies actualizes the creation of new methods and tools that provide an efficiency degree increase of computational process management in such packages. To this end, we represent two toolkits for automating the creation and application of distributed applied software packages with multi-agent management of their computational jobs that is based on the integrated use of knowledge about both subject domains of tasks and the environment in which they are solved.

The package creation includes the following main stages: structural analysis of subject domains of tasks, conceptual model design for these domains, development of applied software libraries, configuring the system software of packages. The conceptual model describes parameters, operations over the parameter field, program modules (applied software) that realize the operations, computational nodes and other objects of subject domains and computational environment, and relationships between the objects. Thus, it includes several components of a comprehensive knowledge about the environment and subject domains. Using the conceptual model, users formulate their tasks in both procedural and non-procedural form, construct task solving plans and form computational jobs. The job describes the necessary requirements to the computational environment for solving the task. Represented toolkits provide both text and graphic mode of the conceptual model development.

A specialized meta-monitoring system collects information about the distributed computing environment and its resources. It processes and unifies this information. The knowledge thus obtained is transferred to the knowledge base. Package components can to use meta-monitoring system API for knowledge elicitation. The meta-monitoring system is also used for testing the environment nodes,

collecting data about their current state, detecting node faults, and diagnosing and to partially repairing these faults.

The system software of packages includes high-level tools for the job management in both homogeneous clusters and heterogeneous distributed computing environment. We propose the task formulation and algorithm of multi-agent management of jobs. The management efficiency is achieved through the resource allocation on the basis of a computational work tender. It is based on a model of the Vickrey auction with one-round bidding. Agents apply the algorithm for the static resource allocation based on economic mechanisms for regulation of resources demand and supply.

The developed multi-agent system has a hierarchical structure, which includes two or more functional levels, and operates on the basis of 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. Agents can represent users and owners of the resources, and interact themselves with aim to meet their interests. Agents are autonomous, and computing management is based on their interaction. They can be organized in virtual communities. In these communities, agents interact using a cooperation or competition. The formation of virtual communities allows agents to adapt the management process to the new challenges.

The developed toolkits are used in the public access computer center „Irkutsk supercomputer center of Siberian Branch of the Russian Academy of Science“ to develop distributed applied software packages for solving scientific and applied tasks from different subject domains. Examples of solving practical tasks illustrate study results. In particular, we design the distributed applied software package to support modeling of warehouse logistics. Using this package, we solved three optimization tasks for adjusting parameters of a warehouse management system. The practical experiments are focused on the refrigerated warehouse. The developed applications demonstrate high efficiency and scalability capabilities to optimize nine criteria to cope with different production demands.

Key words: distributed computing, modular programming, multi-agent management, toolkits.

References

1. Shamakina A. V. *Obzor tekhnologij raspredelennykh vychislenij* [Survey on Distributed Computing Technologies] // Bulletin of the South Ural State University, Series „Computational Mathematics and Software Engineering“. 2014. V. 3. N 3, P. 51–85 (in Russian).
2. Toporkov V. V., Bobchenkov A. V., Emelyanov D. M., Tselishchev A. S. *Metody i evristiki planirovaniya v raspredelennykh vychisleniyakh s neotchuzhdaemyimi resursami* [Scheduling methods and heuristics in distributed computing with non-dedicated resources]. Scheduling methods and heuristics in distributed computing with non-dedicated resources // Bulletin of the South Ural State University, Series „Computational Mathematics and Software Engineering“. 2014. V. 3. N 2. P. 43–62 (in Russian).
3. Sokolinsky L. B., Shamakina A. V. Methods of Resource Management in Problem-Oriented Computing Environment // Programming and Computer Software. 2016. V. 42. N 3. P. 17–26.
4. Il'in V. P., Skopin I. N. Computational programming technologies // Programming and Computer Software. 2011. V. 37. N 4. P. 210–222.
5. Foster I., Kesselman C. Globus: A metacomputing infrastructure toolkit // International Journal of High Performance Computing Applications. 1997. V. 11, N 2. P. 115–128.
6. Thain D., Tannenbaum T., Livny M. Distributed Computing in Practice: The Condor Experience // Concurrency and Computation: Practice and Experience. 2005. V. 17. N 2–4. P. 323–356.
7. Anderson D. Boinc: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, IEEE. 2004, P. 4–10.

8. Voevodin V. V. The solution of large problems in distributed computational media // Automation and Remote Control. 2007. V. 68. N 5. P. 773–786.
9. Belhajjame K., Embury S. M., Paton N. W., Stevens R., Goble A. C. Automatic Annotations of Semantic Web Services Based on Workflow Definitions // ACM Transactions on the Web, 2008. V. 2. N 2. P. 1–34.
10. Gil Y., Deelman E., Ellisman M., Fahringer T., Fox G., Gannon D., Gobl C., Livny M., Moreau L., Myers J. Examining the challenges of scientific workflows // Computer. 2007. V. 40. N 12. P. 24–32.
11. Yu J., Buyya R. A taxonomy of workflow management systems for grid computing // Journal of Grid Computing. 2005. V. 3, N 3–4. P. 171–200.
12. Talia D. Workflow Systems for Science: Concepts and Tools // ISRN Software Engineering. 2013. V. 2013. P. 1–15.
13. Oparin G. A. Osnovannaya na znaniyakh tekhnologiya resheniya vychislitel'nykh zadach [Knowledge-based technology for solving computational problems] // Informatsionnye tekhnologii kontrolya i upravleniya transportnymi sistemami. I. 6, Irkutsk: Irkutsk State Transport University Publ., 2000, P. 3–15 (in Russian).
14. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Gorsky S. Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management // Procedia Computer Science. 2017. V. 103. P. 162–167.
15. Garey M., Johnson D. Computers and Intractability. W. H. Freeman, San Francisco, 1979.
16. Bychkov I. V., Oparin G. A., Feoktistov A. G., Kanter A. N. *Mul'tiagentnyy algoritm raspredeleniya vychislitel'nykh resursov na osnove ekonomicheskogo mekhanizma regulirovaniya ikh sprosa i predlozheniya* [Multiagent algorithm for resources allocation based on the economic mechanism of regulating their supply and demand] // Herald of computer and information technologies. 2014. N 1. P. 39–45 (in Russian).
17. Vickrey W. Counterspeculation, Auctions, and Competitive Sealed Tenders // Journal of Finance. 1961. V. 16, N 1. P. 8–37.
18. Oparin G. A., Feoktistov A. G., Sidorov I. A. *Tekhnologiya organizatsii raspredelennykh vychislenij v instrumental'nom komplekse DISCOMP* [Technology of the distributed calculations organization in tool complex DISCOMP] // Modern technologies. System analysis. Modeling. 2009. N 2. P. 175–180 (in Russian).
19. Sidorov I. A. Methods and Tools to Increase Fault Tolerance of High-Performance Computing Systems // Proceedings of the 39th International Convention on information and communication technology, electronics and microelectronics. Riejk: MIPRO, 2016. P. 242–246.
20. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Dyadkin Yu., Andrukhova V., Basharina O. Simulation Modeling in Heterogeneous Distributed Computing Environments to Support Decisions Making in Warehouse Logistics // Procedia Engineering. 2017. V. 201. P. 524–533.
21. Gorsky S. A., Novopashin A. P., Oparin G. A., Feoktistov A. G. *Upravlenie asinkhronnym vychislitel'nyim protsessom v ORLANDO TOOLS* [Management of asynchronous computational process in IN ORLANDO TOOLS] // Herald of computer and information technologies. 2011. N 1. P. 44–51 (in Russian).
22. Feoktistov A. G., Gorsky S. A. Realizatsiya metoda mul'tistarta v pakete Gradient [Implementation of the multi-start method in the Gradient package] // Vestnik Novosibirsk State University, Series „Information Technologies“. 2007, V. 5, I. 2. P. 78–82 (in Russian).
23. Cho H., Olivera F., Guikema S. D. A derivation of the number of minima of the Griewank function // Applied Mathematics and Computation. 2008. V. 204. N 2. P. 694–701.
24. Irkutsk Supercomputer Centre of SB RAS. [Electron. res.]: <http://hpc.icc.ru>.
25. Sidorov I. A., Tyatyushkin A. I., Feoktistov A. G. *Raspredelennaya informatsionno-vychislitel'naya sreda modul'nogo programmirovaniya* [Distributed information-computing environment of modular programming] // Parallel'nye vychisleniya i zadachi upravleniya: Tr.

III Mezhdunar. konf. [Proceedings of the 3d International conference on Parallel Computations and Control Problems]. Moscow: ICSRAS Publ., 2006. P. 505–521 (in Russian).

26. Zaikin O. S., Sidorov I. A. *Technologiya krupnoblochnogo resparallelivaniya v kriptanalize nekotorykh generatorov dvoichnykh posledovatel'nostej* [Technology of large-block parallelization in the crypto-analysis of some generators of binary sequences] // *Parallel'nye vychislitel'nye tekhnologii: Tr. Mezhdunar. nauch. konf. [Proceedings of the International Scientific Conference on Parallel Computational Technologies]*. Chelyabinsk: SUSU Publ., 2007. V. 1. P. 158–169 (in Russian).

27. Galachyants Y. P., Zakharova Yu. R., Petrova D. P., Morozov A. A., Sidorov I. A., Marchenkov A. M., Logacheva M. D., Markelov M. L., Khabudaev K. V., Likhoshway Ye. V., Grachev M. A. Sequencing of the complete genome of an araphid pennate diatom *Synedraacus* subsp. *radians* from Lake Baikal // *Doklady Biochemistry And Biophysics*. 2015. V. 461. N 1. P. 84–88.

28. Bychkov I. V., Feoktistov A. G., Sidorov I. A., Kostromin R. O., Zorkalzev V. I., Edelev A. V. *Intellektnaya tekhnologiya upravleniya vychisleniyami v virtualizirovannoj klusternoj srede* [Intelligent technology for computations management in a virtualized clustered environment] // *Materialy XV seros. Mul'tikonferentsii po problemam upravleniya. [Proceedings of the 10th All Russian Multi-conference on Control Problems]*. Rostov-na-Donu; Tagenrog: Southern Federal University Publ. 2017. V. 3. P. 84–86 (in Russian).

АВТОМАТИЗАЦИЯ РАЗРАБОТКИ И ПРИМЕНЕНИЯ РАСПРЕДЕЛЕННЫХ ПАКЕТОВ ПРИКЛАДНЫХ ПРОГРАММ

А. Г. Феоктистов, И. А. Сидоров, С. А. Горский

Институт динамики систем и теории управления им. В. М. Матросова СО РАН,
664033, Иркутск, Россия

УДК 004.4+519.6

Рассматриваются проблемы построения предметно-ориентированных систем модульного программирования (пакетов прикладных программ) для решения крупномасштабных задач в гетерогенной распределенной вычислительной среде. Развитие технологий распределенных вычислений актуализирует разработку новых методов и средств, обеспечивающих повышение степени эффективности управления вычислительными процессами в таких пакетах. В этой связи в статье представлены два инструментальных комплекса для автоматизации создания и применения распределенных пакетов прикладных программ с мультиагентным управлением их заданиями, базирующимся на комплексном использовании знаний о предметных областях задач и среде, в которой они решаются. Сформулирована постановка задачи и предложен алгоритм мультиагентного управления заданиями. Эффективность управления достигается путем распределения ресурсов среды на основе тендера вычислительных работ. Примеры решения практических задач иллюстрируют результаты исследований.

Ключевые слова: распределенные вычисления, модульное программирование, мультиагентное управление, инструментальные средства.

Введение. В настоящее время базовые программные средства [1], реализующие технологии организации расчетов в высокопроизводительных вычислительных системах, обеспечивают потенциальную основу для массового создания параллельных и распределенных приложений. Однако анализ их применения на практике позволяет выявить следующие важные проблемы: сложность освоения и применения этих средств конечными пользователями (специалистами в предметных областях), что существенно сужает их круг; недостаточную поддержку учета специфики предметных областей решаемых задач; трудность согласования заданных критериев эффективности их решения (времени, стоимости, безопасности и других ограничений) и требуемых показателей использования ресурсов (балансировки загрузки, надежности, энергопотребления и тому подобных условий); отсутствие универсальных подходов к интеграции ресурсов; слабую комплексированность создаваемых приложений по данным и управлению.

В особенности вышеперечисленные проблемы актуализируются при решении задач в гетерогенной распределенной вычислительной среде (ГРВС), в качестве основных компонентов которой выступают вычислительные кластеры, в том числе гибридные кластеры с разнородными узлами, а высокопроизводительные серверы, системы хранения данных,

Работа выполнена при частичной финансовой поддержке РФФИ (номера проектов 15-29-07955 и 16-07-00931), а также Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ Российской Федерации (НШ-8081.2016.9).

персональные компьютеры и другие вычислительные элементы дополняют инфраструктуру среды. Кластеры организуются на базе как выделенных, так и невыделенных узлов. Узлы разных кластеров отличаются своими вычислительными характеристиками. На кластерах установлены системы управления прохождением заданий (СУПЗ), выполняющие обработку заданий, которые поступают в общие очереди этих систем и затем оттуда направляются на выполнение в назначенные ресурсы среды. Для решения задачи пользователь должен сформировать задание той или иной СУПЗ.

Такая ГРВС, организованная на базе ресурсов центра коллективного пользования (ЦКП), существенно отличается по своим характеристикам от однородных систем, представленных, например, суперкомпьютерами с мультипроцессорной конфигурацией, или специализированных программно-аппаратных комплексов, таких как реконфигурируемые вычислительные системы. Все ее ресурсы являются общими для пользователей ЦКП и распределяются на основе административных политик, определенных как на уровне среды в целом, так и на уровне отдельных ее узлов. Порядок прохождения заданий пользователей в среде жестко регламентирован. Административные политики в узлах среды могут различаться между собой.

В рамках ЦКП свои задачи решают как локальные пользователи отдельных кластеров среды, которым достаточно ресурсов кластера, так и глобальные пользователи, нуждающиеся в использовании ее интегрированных ресурсов при выполнении масштабируемых приложений, включающих наборы прикладных программ (модулей) для параллельного решения задач с помощью вычислительных единиц (например, ядер) узлов ГРВС и порождающих потоки заданий для этих модулей. При этом вычислительная нагрузка, связанная с решением задачи, распределяется между вычислительными единицами, а время выполнения потока заданий уменьшается обратно пропорционально числу используемых вычислительных единиц с учетом их производительности в составе конкретного узла. В процессе решения своих задач пользователи обеих категорий разделяют общие вычислительные ресурсы. Как правило, эффективное распределение ресурсов для выполнения заданий масштабируемых приложений является нетривиальной проблемой для известных систем управления распределенными вычислениями [2]. Обеспечение предметной ориентированности ГРВС позволяет повысить эффективность управления вычислениями [3].

Актуальным подходом к решению обозначенных выше проблем является разработка распределенных пакетов прикладных программ (РППП). В статье рассматриваются два инструментальных комплекса (ИК), предоставляющих широкий набор средств создания и применения РППП в ГРВС, организованной на базе ресурсов ЦКП, с учетом особенностей предметных областей разрабатываемых пакетов.

1. Распределенные пакеты прикладных программ. Развитие технологий организации распределенных вычислений привело к существенным изменениям архитектуры пакетов прикладных программ [4]. Сегодня в общем случае архитектура РППП включает: его прикладное и системное наполнение; входные языки пакета; базы знаний и расчетных данных; сервис-ориентированные интерфейсы пользователя; промежуточное программное обеспечение (ПО) для поддержки взаимодействия между компонентами пакета и вычислительными ресурсами, а также их виртуализации. Системное наполнение пакета обеспечивает постановку задач, планирование вычислений, распределение ресурсов, мониторинг состояния среды, обработку данных, выполнение планов решения задач и другие функции, необходимые в процессе управления вычислениями. План решения задачи выполняется либо в режиме его интерпретации, либо по нему осуществляется синтез параллельной

программы на целевом языке программирования. В обоих случаях в соответствии с планом формируется задание для СУПЗ, используемых в узлах среды.

Класс задач, подходящих для их решения в рамках РППП, имеет следующие особенности: допускается декомпозиция общей сложной задачи на более простые (с вычислительной точки зрения) взаимосвязанные подзадачи, а также разбиение исходных данных на блоки и независимая параллельная обработка этих блоков; решение подзадач реализуется с помощью набора прикладных программных модулей; порядок выполнения модулей определяется последовательно-параллельным планом решения задачи; не предполагается интенсивного взаимодействия между модулями в процессе их выполнения; требуется динамическое управление процессом вычислений при изменении состояния ресурсов и результатов вычислений; эффективность выполнения плана решения задачи определяется заданными пользователем критериями. К такому классу относятся, например, задачи с многовариантными расчетами, а также многие другие переборные задачи.

Известные средства организации распределенных вычислений, например, пакет Globus Toolkit [5], кластерная система HTCondor [6], программный комплекс BOINC [7] или инструментарий X-COM [8], как правило, не в полной мере реализуют вышеперечисленные особенности. В частности, в них отсутствуют средства описания предметных областей, а возможности построения планов решения задачи в лучшем случае ограничены взаимосвязанными заданиями, представимыми в форме направленного ациклического графа.

План решения задачи тесно коррелирует с понятием workflow, используемым при описании потока работ [9]. Для организации и управления потоками работ применяются специализированные системы Workflow Management Systems (WMS) [10]. К их числу относятся Askalon, Condor DAGMan, Grid Ant, Grid Flow, Karajan, Kepler, Pegasus, Taverna, Triana, UNICORE и другие системы [11].

Однако анализ функциональных возможностей вышеперечисленных WMS показывает, что проблемы формирования и обработки этими системами масштабируемых потоков заданий для ЦКП полностью не решены [12]. Ряд WMS (Grid Flow, Triana и UNICORE) используют собственное программное обеспечение для управления ресурсами или средства пакета Globus Toolkit (Askalon, Grid Ant и Pegasus), что неизбежно влечет дополнительные накладные расходы на управление вычислениями при организации их взаимодействия с СУПЗ, применяемыми в ЦКП. Для работы систем Condor DAGMan и Pegasus требуется установка системы управления ресурсами HTCondor, редко используемая в суперкомпьютерных ЦКП. Системы Karajan, Kepler и Taverna имеют централизованную архитектуру, не поддерживающую динамическую декомпозицию потоков заданий между ресурсами, что не позволяет эффективно использовать ее преимущества в случае изменения состояния используемых ресурсов. Возможности ряда WMS (Condor DAGMan, Grid Ant, Karajan и UNICORE) для описания предметной области решаемых задач существенно ограничены, а средства представления предметных знаний других систем (например, Taverna) ориентированы в основном на достаточно узкие классы задач. Ограниченное число WMS поддерживает автоматический синтез workflow. Отсутствие стандартизованного описания предметной области и потоков заданий не позволяет комплексировать пакеты, разработанные в разных WMS, по данным и управлению.

2. Представление знаний. Выбор оптимальной конфигурации вычислительной системы для решения крупномасштабной задачи определяется алгоритмами построения эффективного плана ее решения и назначения подходящих ресурсов. Успешная работа этих алгоритмов во многом зависит от степени полноты информации о специфике реша-

емой задачи, а также вычислительных характеристиках и правилах использования ресурсов системы. Таким образом, возникает необходимость разработки модели объектно-ориентированной ГРВС, которая бы позволила представить всесторонние знания об этой среде в некотором унифицированном формате, который может быть использован программными комплексами, функционирующими в рамках данной среды.

В отличие от вычислительных моделей (см., например, [13]) концептуальная модель проблемно-ориентированной ГРВС позволяет осуществить взаимосвязанное представление не только алгоритмических знаний предметных областей решаемых задач, но и знаний о программно-аппаратной инфраструктуре среды и административных политиках, определенных для ее ресурсов. *Вычислительные знания* содержат информацию о прикладных модулях для решения задач, системных модулях и вспомогательных модулях для препроцессорной и постпроцессорной обработки данных. Системные модули обеспечивают планирование вычислений, формирование заданий, распределение ресурсов, мониторинг вычислительных процессов и динамическую декомпозицию задач. *Схемные знания* включают множество объектов для описания модульной структуры модели и алгоритмов для исследования проблемной области. Основными объектами предметной области являются параметры и операции. *Продукционные знания* определяют правила применения операций и позволяют пользователям выбирать лучшие алгоритмы в текущей вычислительной ситуации. Порядок выполнения продукций определяется их приоритетами. *Знания о программно-аппаратной инфраструктуре* представлены характеристиками узлов, каналов связи, сетевых устройств, топологии сети и другими подобными сведениями. Эти характеристики включают в себя также информацию о сбоях программного обеспечения и аппаратных средств. *Знания об административных политиках* включают данные о пользователях и их заданиях, системах управления ресурсами и дисциплинах диспетчеризации вычислительных работ.

Технология построения концептуальной модели ГРВС предполагает создание ее ядра и его дальнейшее использование в различных РППП. Ниже приведен пример классов объектов ядра модели и их атрибутов, создаваемых с помощью рассматриваемых в статье инструментальных комплексов:

- параметры (имя, тип, область допустимых значений);
- операции (имя, тип, список входных параметров, список выходных параметров, программный модуль);
- формальные параметры (тип, назначение);
- программные модули (имя, список формальных параметров);
- продукции (имя, имя параметра, имя операции);
- постановки задач (имя, список исходных параметров, список целевых параметров, список операций, список критериев качества решения задачи; список ресурсов; список критериев эффективности использования ресурсов);
- планы решения задач (имя, список операций);
- характеристики заданий (имя, тип, область допустимых значений, ранг, вес);
- задания (имя, список характеристик задания);
- классы заданий (имя, список характеристик заданий);
- характеристики вычислительного устройства (имя, тип, область допустимых значений);
- вычислительные устройства (имя, тип, список характеристик вычислительного устройства, список классов заданий);

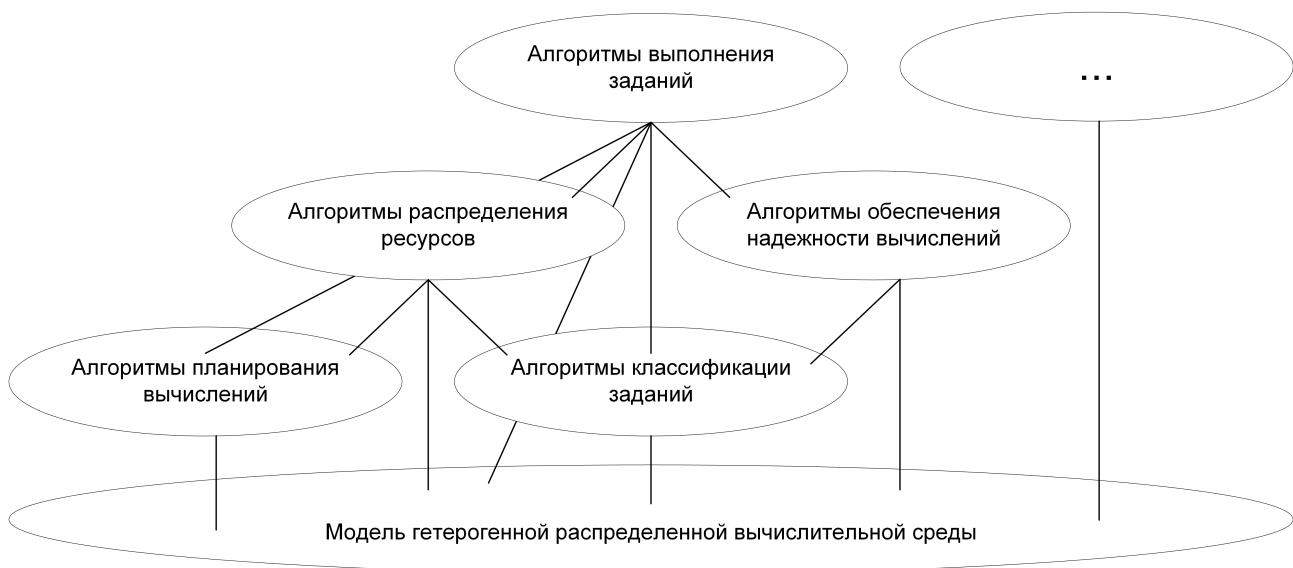


Рис. 1. Алгоритмы управления вычислениями

- характеристики канала связи (имя, тип, область допустимых значений);
- каналы связи (имя, тип, список характеристик канала связи);
- характеристики сетевого устройства (имя, тип, область допустимых значений);
- сетевые устройства (имя, тип, список характеристик сетевого устройства);
- пользователи (имя, логин, пароль);
- права пользователей (имя, тип, область допустимых значений);
- группы пользователей (имя, тип, список пользователей);
- административные политики (имя, список групп пользователей, список прав пользователей).

Концептуальная модель ГРВС обеспечивает фундаментальную основу для разработки алгоритмов управления вычислениями в этой среде, включая алгоритмы классификации заданий, планирования вычислений и распределения ресурсов (рис. 1). Эти алгоритмы реализуются системными модулями и могут также использоваться при построении планов решения задач в РППП. Методы и средства ее построения представлены в работе [14].

3. Мультиагентное управление заданиями. В общем случае на концептуальной модели ГРВС формулируется следующая постановка задачи управления распределенными вычислениями: „зная M , вычислить X по Y , выполнив F с заданными критериями C , используя ресурсы R “. Здесь M — это концептуальная модель, X и Y — множества исходных и целевых параметров, F — множество операций, C — заданные пользователем критерии эффективности решения задачи, R — множество ресурсов среды. Данная постановка задачи является полной и представляет собой план решения задачи. При возникновении неопределенности в одном из условий полной постановки задачи, заданных множествами X , Y , F или R , она сводится к сокращенной процедурной („зная M , вычислить X ? по Y ?, выполнив F с заданными критериями C , используя ресурсы R ?“) или непроцедурной („зная M , вычислить X по Y , выполнив F ? с заданными критериями C , используя ресурсы R ?“) постановке задачи. Символ „?“ означает наличие неопределенности в том или ином множестве. Неопределенность в условиях полной постановки задачи обуславливает многообразие сокращенных постановок задач.

Задача управления вычислениями на основе приведенных выше сокращенных постановок задач является NP-трудной [15]. В этой связи активно развиваются эвристические методы ее решения, обеспечивающие достаточно эффективные (рациональные) решения. В их число входят методы, использующие мультиагентные системы. Агент в такой системе представляет собой интеллектуальную (использующую элементы искусственного интеллекта) программную сущность, наделенную правами и обязанностями по обслуживанию вычислительного процесса и управления им. Агенты представляют интересы пользователей и владельцев ресурсов. Они взаимодействуют между собой в процессе выполнения заданий пользователей, координируя свои действия на основе кооперации или конкуренции в зависимости от поставленных им целей и своих ментальных свойств.

Для управления заданиями в ГРВС разработана мультиагентная система, включающая агентов постановки задачи, планирования вычислений, мониторинга и распределения ресурсов, классификации, конкретизации и выполнения заданий. Ниже основные этапы алгоритма мультиагентного управления заданиями [16].

— *Планирование вычислений.* На основе процедурной или не процедурной постановки задачи, сформулированной пользователем пакета на модели M , агентом планирования вычислений выполняется построение плана p решения задачи.

— *Формирование виртуального сообщества агентов.* Объединение агентов в виртуальное сообщество осуществляется на основе знаний о соответствии классов заданий и ресурсов ГРВС. Эти знания заносятся в модель M администраторами ГРВС. В виртуальное сообщество включаются агенты, представляющие ресурсы, в которых могут быть выполнены модули плана p .

— *Выбор координатора.* Избрание участниками виртуального сообщества агента координатора производится в процессе их локальных взаимодействий на основе модифицированного древесного алгоритма с учетом коммуникационной топологии сети агентов.

— *Проведение тендера вычислительных работ.* Формирование агентами предложений по выполнению модулей плана p и определение победителей торгов в рамках тендера реализуется на основе модели однораундового аукциона Викри второй цены [17]. Применение тендера вычислительных работ позволяет использовать, помимо их стоимости, являющейся единственным условием в рамках аукциона, дополнительные критерии качества выполнения заданий, такие как время решения задачи, надежность вычислений, безопасность информации и другие ограничения. В случае многовариантных расчетов агенты виртуального сообщества параллельно выполняют один и тот же модуль, а торги ведутся за право обработки вариантов входных данных с помощью этого модуля.

— *Подведение итогов торгов.* Определение агентов (победителей торгов), участвующих в выполнении модулей плана p , производится на основе лексикографического правила выбора с учетом заданных критериев эффективности решения задачи, упорядоченных по степени их значимости.

— *Процесс вычислений.* Выполнение модулей плана p в назначенных агентами ресурсах осуществляется в асинхронном режиме по готовности данных.

Разработанная мультиагентная система относится к классу метапланировщиков и может быть использована в качестве надстройки к СУПЗ. Агенты системы могут быть использованы в инструментальных средствах разработки РППП.

4. Инструментальные средства разработки распределенных пакетов прикладных программ. В данном разделе рассматриваются два ИК, базирующихся на использовании концептуальной модели ГРВС, которая интегрирует знания как о пред-

метных областях решаемых задач, так и о среде выполнения РППП, а также на применении специализированной системы метамониторинга ГРВС. При разработке и применении РППП в этих ИК предполагается, что узлы ГРВС используют различные программно-аппаратные платформы и операционные системы (ОС). В этой связи, сегменты ГРВС различаются степенью производительности и надежности. Языковые средства РППП предоставляют возможности для описания концептуальной модели предметной области, ее объектов и основных свойств, формулировки постановок задач, формирования заданий, подготовки исходных данных и других необходимых операций.

4.1. *Инструментальный комплекс DISCOMP.* Функциональное наполнение РППП В ИК DISCOMP [18] составляют модули, представленные в виде исполняемых в пакетном режиме программ, реализованных на различных языках программирования (например, С, Fortran, Pascal). Модули размещаются в разных узлах ГРВС. В каждом узле может быть установлено несколько модулей. Допустимо включение в состав функционального наполнения унаследованного ПО, а также нетиражируемых программных комплексов, жестко привязанных к узлам среды. Обмен данными между модулями осуществляется через файлы. Системная часть РППП включает мультиагентные компоненты ИК DISCOMP для управления распределенными вычислительными процессами и обменом данными. Эти компоненты являются платформонезависимыми и могут функционировать под управлением различных ОС (например, MS Windows, Linux и Mac OS X). Набор компонентов, включаемых в системную часть пакета, определяется его разработчиком.

Управление вычислениями реализуется системой агентов. Решение задачи выполняется, как правило, по одной слабо меняющейся схеме, требующей динамического управления процессом вычислений. План решения задачи строится в параллельно-ярусной форме на основе процедурной постановки задачи. Выполнение плана решения задачи осуществляется в режиме интерпретации. Предварительная классификация заданий специальным агентом позволяет рационально назначать ресурсы модулям. ИК DISCOMP поддерживает виртуализацию ресурсов ГРВС. Если программно-аппаратные ресурсы какого-либо узла среды не соответствуют требованиям модуля плана решения задачи, то для выполнения такого модуля может быть запущена виртуальная машина.

Концептуальная модель ГРВС и планы решения задач описываются в текстовом виде с использованием расширения языка XML. На рис. 2 представлены фрагменты текста описания модуля и его параметров (рис. 2, а), а также постановки задачи (рис. 2, б), включающей этот модуль. Языковые средства обеспечивают задание условий обработки событий, возникающих в процессе вычислений. В их числе такие события как запуск и завершение выполнения модуля или группы модулей, вычисление значений управляющих параметров, приближение характеристик ресурсов узла среды к их критическим значениям, отказ ресурсов узла и другие действия, совершенные в процессе вычислений.

Обработка событий включает: диагностику запуска и завершения выполнения модулей; рестарт модулей с аварийным завершением их работы; выбор нового плана решения задачи; завершение избыточных вычислительных процессов в случае нахождения решения задачи; идентификацию и устранение отказов ресурсов, а также другие операции, направленные на повышение надежности вычислительного процесса. Интерпретатор плана решения задачи получает данные о состоянии вычислительных процессов и ресурсов, необходимые при обработке событий, от системы метамониторинга [19].

Был разработан РППП для имитационного моделирования логистических складских комплексов путем организации многовариантных расчетов [20]. Он был использован на

<pre> <!-- секция спецификаций параметров --> <parameters> <param name='list' type='filelist' filepattern='list_%1.txt' /> <param name='result' type='filelist' filepattern='result_%1.txt' /> </parameters> <!-- секция спецификаций модулей --> <modules> <module name='solver'> <commands os='Linux'> <start>./solver</start> <stop>./stop.sh</stop> </commands> <parameters> <input> <param name='list' /> </input> <output> <param name='result' /> </output> </parameters> </module> </modules> </pre> <p style="text-align: center;">a)</p>	<pre> <scheme> <stage> <listmodule name='solver' onFinish='checkResult (\$el_num)' /> </stage> </scheme> <control><![CDATA[function checkResult (el_num) { // получить значение выходного элемента // параметра-списка result с порядковым // номером el_num var res = DiscompAPI.getLPV('result', el_num); // если значение элемента содержит подстроку // "RESULT FOUND", то остановить вычисления на // текущем ярусе и перейти к следующему if (res.match(/RESULT FOUND/gi)) { DiscompAPI.stopStageModules (); } }]]></control> </pre> <p style="text-align: center;">б)</p>
---	---

Рис. 2. Фрагменты описания модели: а — модуль и его параметры; б — постановка задачи

практике для решения трех оптимизационных задач для рефрижераторного склада ООО „Иркутский Хладокомбинат“. Этот склад является вторым в России по объему хранения от Урала до Дальнего Востока. Были решены следующие задачи:

- улучшение процессов погрузки и разгрузки, включая обработку возвратных отходов;
- реструктуризация процесса обслуживания клиентов;
- моделирование процессов переоборудования и сдачи в аренду дополнительных объектов коммерческой недвижимости.

Предметная область РППП включает в себя три модели на языке GPSS для решаемых задач. Каждая модель имеет множество входных переменных (параметры развертки) и наблюдаемые переменные (критерии функционирования хранилища). В процессе решения задачи пакет порождает поток независимых заданий. Каждое задание запускает модель с одним из вариантов значений входных переменных. Запуск моделей осуществляется с помощью виртуальных машин. Поток заданий распределяется по ресурсам агентами ИК DISCOMP. Независимость заданий обеспечивает высокую масштабируемость вычислений. Общая схема решения задачи включает следующие этапы: прогон модели со всеми вариантами сочетаний допустимых значений входных переменных; определение оптимального варианта значений наблюдаемых переменных модели с помощью многокритериального лексикографического правила выбора на множестве всех вариантов значений наблюдаемых переменных, упорядоченных по степени их значимости; многокритериальный выбор варианта значений входных переменных, который соответствует оптимальному варианту значений наблюдаемых переменных. Нахождение варианта значений входных переменных реализовано аналогично выбору варианта значений наблюдаемых переменных. Найденный вариант входных переменных является решением задачи. Лицо, принимающее решение, (менеджер) склада используют эти переменные для настройки параметров логистических процессов в системе управления складом. Результаты решения задач были

Таблица 1

Улучшение критериев функционирования склада

Критерий	Улучшение
Эффективность работы грузчиков	9 %
Среднее время выполнения погрузочно-разгрузочных операций	6 %
Средняя прибыль от погрузочно-разгрузочных операций	11 %
Среднее время обработки возвратных отходов	10 %
Прибыль от реализации возвратных отходов	75 %
Число кладовщиков	20 %
Число вызовов лифтов	12 %
Наполнение склада	8 %
Прибыль от сдачи в аренду объектов коммерческой недвижимости	23 %

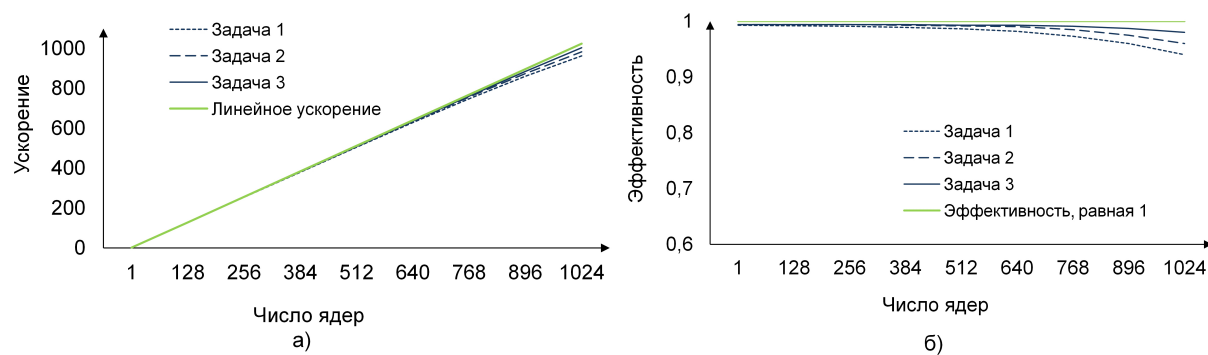


Рис. 3. Масштабируемость вычислений: а — ускорение; б — эффективность

применены для настройки параметров системы управления, используемой на складе. Это позволило улучшить девять критериев работы склада на величину от 6 % до 75 % (табл. 1).

Рис. 3, а, и 3, б, демонстрируют ускорение и эффективность вычислений в процессе решения задач относительно разного числа. Эти результаты подтверждают высокую масштабируемость вычислений под управлением инструментального комплекса DISCOMP.

4.2. *Инструментальный комплекс Orlando*. ИК Orlando [21] предоставляет пользователям как текстовый, так и графический языки описания предметной области РППП и постановок задач. Постановки задач могут формулироваться в процедурной и непроцедурной формах. В последнем случае производится автоматический синтез плана решения задачи (абстрактной программы). План решения задачи, сформированный на основе ее процедурной постановки, может включать управляющие конструкции ветвления, цикла и рекурсии.

Графическое описание предметной области РППП и постановок задач автоматически переводится во внутреннее представление в терминах языка XML. План решения задачи выполняется в режиме интерпретации. На рис. 4 приведен пример графического представления плана решения задачи в пакете Градиент для исследования многоэкстремальных функций методом мульгистарта [22].

Средства описания концептуальной модели в ИК Orlando позволяют также создавать системные параметры и операции, в том числе представляющие данные и модули API системы метамониторинга [19]. Это обеспечивает возможность тестового запуска модулей

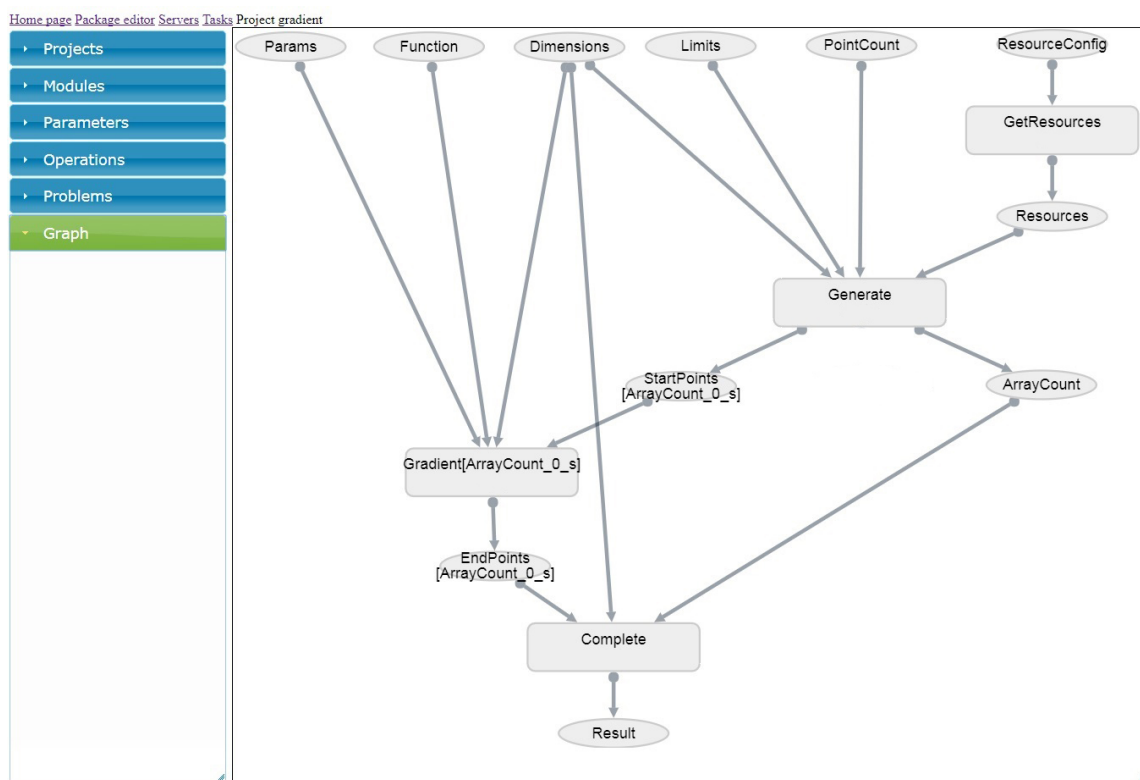


Рис. 4. Описание плана решения задачи в ИК Orlando

в узлах среды в процессе решения задачи с целью экспериментального прогнозирования времени их выполнения и динамического распределения нагрузки между гетерогенными узлами пропорционально их вычислительным характеристикам. Прогнозирование времени вычислений базируется на определении зависимости между показателями работы программы и характеристиками вычислительного узла, в котором она выполняется. Используются следующие показатели работы программы: время выполнения программы, общее число инструкций программы, число тактов процессора, информация по работе с кэш-памятью процессора первого и второго уровней. В качестве характеристик узла задействуются производительность процессора при выполнении целочисленных операций и операций с плавающей точкой, а также размер, пропускная способность и латентность для оперативной и кэш-памяти.

Приложение „Градиент“ для поиска глобального минимума многоэкстремальной функции методом мултистарта включает четыре операции. Операция *GetResources* получает на вход список операций с указанием минимального и максимального числа их экземпляров, порождаемых при решении задачи. На основе этой информации системный модуль, реализующий данную операцию, резервирует ресурсы необходимые для дальнейшего решения задачи и распределяет их для выполнения соответствующих операций. На выходе модуль возвращает число и производительность выделенных ресурсов. Он осуществляет измерение их производительности с помощью агентов системы метамониторинга. Операция *GetResources* реализует дополнительный этап схемы решения задачи по сравнению с классическим методом мултистарта и обеспечивает декомпозицию задачи операцией *Generate* на основе эвристической информации о производительности ресурсов.

Таблица 2

Показатели процесса решения задачи при использовании различных тактик

Показатель	Тактика 1	Тактика 2	Тактика 3
Время, с	1580	1380	1096
Ускорение	4,54	5,20	6,54
Средняя загрузка процессоров	0,60	0,69	0,87

Операция *Generate* генерирует множество начальных точек и сопоставляет каждому ресурсу массив начальных точек, пропорциональный его производительности. Это обеспечивает эффективность использования выделенных ресурсов. Операция *Gradient* реализует спуск из начальной точки в точку локального минимума методом градиента. При выполнении схемы решения задачи интерпретатор запускает множество экземпляров этой операции на выделенных ресурсах. Каждый экземпляр операции реализуется параллельной программой. Элементы параллельных списков *StartPoints* и *EndPoints* обрабатываются независимо друг от друга в отдельных процессах — экземплярах этой операции. Операция *Complete* объединяет результаты вычислений и находит минимальное значение функции, принимаемое в качестве ее глобального минимума.

В табл. 2 приведены результаты времени решения задачи оптимизации двухмерной функции Griewank [23] с 5 млн. начальных точек на 10 узлах двух кластеров с различной производительностью (по 5 узлов от каждого кластера), а также ускорения (относительно решения задачи на одном узле с максимальными характеристиками) и средней загрузки процессоров при использовании трех тактик распределения вычислительной нагрузки:

- выделение одного массива с равным числом начальных точек каждому ресурсу;
- формирование избыточного (по отношению к числу ресурсов) числа массивов с равным числом начальных точек и распределение этих массивов по ресурсам по мере их освобождения;
- назначение каждому ресурсу по одному массиву с числом начальных точек, пропорциональным производительности ресурса для данной задачи, определенной на дополнительном этапе схемы решения задачи (рис. 3).

Очевидно преимущество третьей тактики. Используемые в примере узлы имели разное число ядер (16 и 32) и отличались размером оперативной памяти (32 и 64 Гбайт). Эксперимент показал, что размер оперативной памяти не является практически значимым. Число выполняемых в секунду операций с плавающей точкой, напротив, существенно влияет на время решения задачи. В результате проведенного исследования были определены коэффициенты производительности узлов двух кластеров, используемые в процессе прогнозирования, равные для данной задачи соответственно 0,49 и 1.

Важной особенностью ИК Orlando является комплексирование разрабатываемых пакетов. Комплексирование обеспечивает возможность использования в процессе создания нового пакета фрагментов описания предметных областей, программных модулей, исходных данных и результатов вычислений, имеющих в других пакетах. Вследствие этого, сокращаются сроки разработки РППП и проведения вычислительных экспериментов.

Заключение. В статье рассмотрены два высокоуровневых инструментальных комплекса для создания РППП, предоставляющих в совокупности широкий спектр возможностей описания специфики решаемых задач и процесса управления вычислениями. Методология разработки РППП базируется на интегрированном применении методов и средств инженерии знаний, концептуального моделирования, крупноблочного программирования

и синтеза программ, мультиагентных технологий, а также методов оценки надежности и распределения вычислительных ресурсов, включая экономические механизмы регулирования их спроса и предложения.

Разработанные ИК используются в ЦКП „Иркутский суперкомпьютерный центр СО РАН“ [24] для решения научных и прикладных задач из разных предметных областей. Помимо рассмотренных выше примеров их практического использования, они также применялись при решении задач построения множеств достижимости управляемых летательных объектов [25], выполнимости булевых формул [26], исследования биоресурсов озера Байкал [27], обеспечения безопасности энергетических систем [28].

Примеры решения практических задач показали, что РППП, реализующие многовариантные расчеты, обеспечивают хорошую масштабируемость вычислений в ГРВС. В целом применение инструментальных комплексов позволяет сократить время разработки РППП и проведения с их помощью широкомасштабных вычислительных экспериментов.

Список литературы

1. Шамакина А. В. Обзор технологий распределенных вычислений // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2014. Т. 3. № 3. С. 51–85.
2. Топорков В. В., Бобченков А. В., Емельянов Д. М., Целищев А. С. Методы и эвристики планирования в распределенных вычислениях с неотчуждаемыми ресурсами // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2014. Т. 3. № 2. С. 43–62.
3. Sokolinsky L. B., Shamakina A. V. Methods of Resource Management in Problem-Oriented Computing Environment // Programming and Computer Software. 2016. V. 42. N 3. P. 17–26.
4. Il'in V. P., Skopin I. N. Computational programming technologies // Programming and Computer Software. 2011. V. 37. N 4. P. 210–222.
5. Foster I., Kesselman C. Globus: A metacomputing infrastructure toolkit // International Journal of High Performance Computing Applications. 1997. V. 11. N 2. P. 115–128.
6. Thain D., Tannenbaum T., Livny M. Distributed Computing in Practice: The Condor Experience // Concurrency and Computation: Practice and Experience. 2005. V. 17. N 2–4. P. 323–356.
7. Anderson D. Boinc: A system for public-resource computing and storage // Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing. IEEE, 2004. P. 4–10.
8. Voevodin V. V. The solution of large problems in distributed computational media // Automation and Remote Control. 2007. V. 68. N 5. P. 773–786.
9. Belhajjame K., Embury S. M., Paton N. W., Stevens R., Goble A. C. Automatic Annotations of Semantic Web Services Based on Workflow Definitions // ACM Transactions on the Web. 2008. V. 2. N 2. P. 1–34.
10. Gil Y., Deelman E., Ellisman M., Fahringer T., Fox G., Gannon D., Gobl C., Livny M., Moreau L., Myers J. Examining the challenges of scientific workflows // Computer. 2007. V. 40. N 12. P. 24–32.
11. Yu J., Buyya R. A taxonomy of workflow management systems for grid computing // Journal of Grid Computing. 2005. V. 3. N 3–4. P. 171–200.
12. Talia D. Workflow Systems for Science: Concepts and Tools // ISRN Software Engineering. 2013. V. 2013. P. 1–15.
13. Опарин Г. А. Основанная на знаниях технология решения вычислительных задач // Информационные технологии контроля и управления транспортными системами. Вып. 6. Иркутск: Изд-во ИрИИТ, 2000. С. 3–15.

14. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Gorsky S. Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management // *Procedia Computer Science*. 2017. V. 103. P. 162–167.
15. Garey M., Johnson D. *Computers and Intractability*. W. H. Freeman, San Francisco, 1979.
16. Бычков И. В., Опарин Г. А., Феоктистов А. Г., Кантер А. Н. Мультиагентный алгоритм распределения вычислительных ресурсов на основе экономического механизма регулирования их спроса и предложения // *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. N 1. P. 8–37.
18. Опарин Г. А., Феоктистов А. Г., Сидоров И. А. Технология организации распределенных вычислений в инструментальном комплексе DISCOMP // *Современные технологии. Системный анализ. Моделирование*. 2009. № 2. С. 175–180.
19. Sidorov I. A. Methods and Tools to Increase Fault Tolerance of High-Performance Computing Systems // *Proceedings of the 39th International Convention on information and communication technology, electronics and microelectronics*. Riejska: MIPRO, 2016. P. 242–246.
20. Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Dyadkin Yu., Andrukhova V., Basharina O. Simulation Modeling in Heterogeneous Distributed Computing Environments to Support Decisions Making in Warehouse Logistics // *Procedia Engineering*. 2017. V. 201. P. 524–533.
21. Горский С. А., Новопащин А. П., Опарин Г. А., Феоктистов А. Г. Управление асинхронным вычислительным процессом в ORLANDO TOOLS // *Вестник компьютерных и информационных технологий*. 2011. № 1. С. 44–51.
22. Феоктистов А. Г., Горский С. А. Реализация метода мультистарта в пакете Градиент // *Вестник Новосибирского государственного университета. Серия: Информационные технологии*. 2007. Т. 5. Вып. 2. С. 78–82.
23. Cho H., Olivera F., Guikema S. D. A derivation of the number of minima of the Griewank function // *Applied Mathematics and Computation*. 2008. V. 204. N 2. P. 694–701.
24. ЦКП Иркутский суперкомпьютерный центр СО РАН. [Электронный ресурс]. <http://hpc.icc.ru>.
25. Сидоров И. А., Тятюшкин А. И., Феоктистов А. Г. Распределенная информационно-вычислительная среда модульного программирования // *Параллельные вычисления и задачи управления: Тр. III Междунар. конф.* Москва: Изд-во ИПУ РАН, 2006. С. 505–521.
26. Заикин О. С., Сидоров И. А. Технология крупноблочного распараллеливания в криптоанализе некоторых генераторов двоичных последовательностей // *Параллельные вычислительные технологии: Тр. Междунар. науч. конф.* Челябинск: Изд-во ЮУрГУ, 2007. Т. 1. С. 158–169.
27. Galachyants Y. P., Zakharova Yu. R., Petrova D. P., Morozov A. A., Sidorov I. A., Marchenkov A. M., Logacheva M. D., Markelov M. L., Khabudaev K. V., Likhoshway Ye. V., Grachev M. A. Sequencing of the complete genome of an araphid pennate diatom *Synedra acus* subsp. *radians* from Lake Baikal // *Doklady Biochemistry And Biophysics*. 2015. V. 461. N 1. P. 84–88.
28. Бычков И. В., Феоктистов А. Г., Сидоров И. А., Костромин Р. О., Зоркальцев В. И., Еделев А. В. Интеллектуальная технология управления вычислениями в виртуализированной кластерной среде // *Материалы X Всерос. Мультиконференции по проблемам управления: в 3-х т.* Ростов-на-Дону; Таганрог: Изд-во Южного федерального университета, 2017. Т. 3. С. 84–86.



Феоктистов Александр Геннадьевич, канд. техн. наук, доцент, Институт динамики систем и теории управления им. В. М. Матросова СО РАН

(ИДСТУ СО РАН), старш. науч. сотр.; e-mail: agf65@yandex.ru, +79247116704, +73952453154.

Феоктистов Александр Геннадьевич является старшим научным сотрудником лаборатории „Параллельных и распределенных вы-

числительных систем“ и доцентом отдела аспирантуры Института динамики систем и теории управления им. В. М. Матросова СО РАН. Он получил ученую степень кандидата технических наук в 2000 г. и ученое звание доцента в 2003 г. В настоящее время Феоктистов А. Г. является руководителем ряда российских научно-исследовательских проектов. Областями его научных интересов являются вычислительные модели, параллельные и распределенные вычисления, мультиагентные технологии и имитационное моделирование.

Alexander Feoktistov is a senior research officer in the Laboratory of Parallel and Distributed Computing Systems and an associate professor in Graduate School Department of Matrosov Institute for System Dynamics and Control Theory of SB RAS. He received the Ph.D. degree in Engineering Sciences in 2000 and academic title of Associate Professor in 2003. He leads a number of national research projects. His main interests include computational models, parallel and distributed computing, multi-agent technologies, and simulation modeling.



Сидоров Иван Александрович, канд. техн. наук, Институт динамики систем и теории управления им. В. М. Матросова СО РАН (ИДСТУ СО РАН), науч. сотр.; e-mail: yvan.sidorov@gmail.com, +79027668364.

Сидоров Иван Александрович является научным сотрудником лаборатории „Параллельных и распределенных вычислительных систем“ Института динамики систем и теории управления им. В. М. Матросова СО РАН и системным администратором Центра коллективного пользования „Иркутский суперкомпьютерный центр СО РАН“. Он получил ученую степень кандидата технических наук в 2010 г. Областями его научных интересов являются си-

стемное программирование и администрирование, параллельные и распределенные вычисления, мультиагентные технологии и мониторинг.

Ivan Sidorov is a research officer in the Laboratory of Parallel and Distributed Computing Systems of Matrosov Institute for System Dynamics and Control Theory of SB RAS, and a system administrator in the public access computer center „Irkutsk supercomputer center of Siberian Branch of the Russian Academy of Science“. He received the Ph.D. degree in Engineering Sciences in 2010. His main interests include system programming and administration, parallel and distributed computing, multi-agent technologies, and monitoring.



Горский Сергей Алексеевич, канд. техн. наук, Институт динамики систем и теории управления им. В. М. Матросова СО РАН (ИДСТУ СО РАН), науч. сотр.; e-mail: gorsky@icc.ru, +73952453017.

Горский Сергей Алексеевич является научным сотрудником лаборатории „Параллельных и распределенных вычислительных систем“ и доцентом отдела аспирантуры Института динамики систем и теории управления им. В. М. Матросова СО РАН. Он получил ученую степень кандидата технических наук в 2009 г. Областями его научных интересов являются системное программирование, параллельные и распределенные вычисления, булево моделирование.

Sergey Gorsky is a research officer in the Laboratory of Parallel and Distributed Computing Systems and an associate professor in Graduate School Department of Matrosov Institute for System Dynamics and Control Theory of SB RAS. He received the Ph.D. degree in Engineering Sciences in 2009. His main interests include system programming, parallel and distributed computing, and Boolean modeling.

Дата поступления — 23.10.2017