

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ РАСПРЕДЕЛЕННЫХ ПАКЕТОВ ПРИКЛАДНЫХ ПРОГРАММ НА ОСНОВЕ МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ

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

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

УДК 004.4+519.6

В статье рассматривается актуальная проблема, связанная с разработкой распределенных пакетов прикладных программ (приложений) для решения крупномасштабных научных и прикладных задач. Предлагается новый подход, основанный на интеграции концептуального и модульного программирования в рамках гетерогенной распределенной вычислительной среды. Этот подход включает в себя следующие основные этапы: структурный анализ предметной области и построение ее концептуальной модели, разработка прикладного и настройка системного программного обеспечения пакета, базирующегося на использовании заложенных в модель знаний, выполнение приложений с мультиагентным управлением вычислительными процессами. Концептуальная модель описывает параметры предметной области, операции над полем параметров, программные модули (прикладное программное обеспечение), реализующие операции, вычислительные узлы и другие объекты как предметной области, так и вычислительной среды, включая отношения между объектами. Пользователи формулируют постановки решения задач в виде схем их решения. Понятие схемы решения задачи тесно коррелирует с термином *workflow*, часто используемым в зарубежной литературе. Системное программное обеспечение включает высокоуровневые средства для управления схемами решения задач как в однородных системах, так и в гетерогенной распределенной вычислительной среде. В рамках предлагаемого подхода представлены два инструментальных комплекса, обеспечивающих различные возможности для создания и применения распределенных пакетов прикладных программ с учетом особенностей их предметных областей.

Ключевые слова: распределенные вычисления, модульное программирование.

Введение

Развитие технологий параллельного программирования привело к существенным изменениям архитектуры пакетов прикладных программ (ППП) [1]. Современные ППП представляют собой распределенные приложения для решения крупномасштабных научных и прикладных задач в гетерогенной распределенной вычислительной среде (ГРВС). Такая среда состоит из вычислительных узлов, объединенных коммуникационной сетью. Вычислительный узел представляет собой программно-аппаратный ресурс, включающий модули оперативной памяти, один или несколько многоядерных процессоров, жесткий диск и системное программное обеспечение (ПО), представляющее собой совокупность программных средств, поддерживающих функционирование узла в рамках среды. ГРВС может включать кластеры, узлы которых имеют различные вычислительные характеристики. Кластеры могут быть географически распределены и принадлежать разным организациям. Как следствие, возникают новые требования к средствам описания и обработки схем решения задач пользователей распределенных ППП (РППП).

В общем случае архитектура среды выполнения РППП включает: его прикладное и программное наполнение; входные языки пакета; базы знаний и расчетных данных; интерфейсы пользователя; вычислительные ресурсы, в которых будут выполняться модули пакета; промежуточное ПО для поддержки взаимодействия

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).

между функциональным и системным наполнением пакета и вычислительными ресурсами; средства виртуализации компонентов пакета, обеспечивающие пользователям открытый и прозрачный доступ к этим компонентам.

Можно выделить два основных подхода к организации функционального наполнения пакета и решению прикладных задач в ГРВС. В рамках первого подхода производится синтез абстрактной программы по схеме решения задачи с последующей компиляцией на ее основе параллельной программы, которая выполняется в ГРВС. При втором подходе схема решения задачи выполняется в ГРВС в режиме интерпретации.

Появление развитого базового программного обеспечения [2], реализующего технологии организации расчетов в ГРВС, является основой для массовых параллельных и распределенных приложений. Однако анализ методов и средств организации такого рода вычислений выявил ряд проблем научно-технического характера: разработка распределенных и параллельных приложений выполняется зачастую “точечно”, в привязке к узкому классу задач из конкретной предметной области; создаваемые приложения плохо интегрируются, вследствие использования различного промежуточного ПО, форматов данных и протоколов их передачи, а также использования разных моделей программирования приложений, планирования вычислительных процессов и загрузки вычислительных ресурсов; круг потенциальных пользователей таких приложений достаточно ограничен, что обусловлено сложностью их освоения и применения прикладными специалистами для решения своих задач.

В частности, обеспечение эффективного масштабирования потоков заданий, порождаемых РППП, является в настоящее время одним из фундаментальных и практически важных направлений исследований по организации проблемно-ориентированных ГРВС [3]. Необходимость создания масштабируемых РППП возникает при выполнении многовариантных расчетов, решении широкого спектра переборных задач и многих других. Предполагается, что масштабируемое приложение включает набор прикладных программ для параллельного решения задачи с помощью различных вычислительных единиц (например, ядер) разнородных узлов ГРВС и порождает поток заданий для этих прикладных программ. При этом вычислительная нагрузка, связанная с решением задачи, распределяется между вычислительными единицами разнородных узлов ГРВС, а время выполнения потока заданий уменьшается обратно пропорционально числу используемых вычислительных единиц с учетом их производительности в составе конкретного узла ГРВС. Создание системы управления потоками заданий в ГРВС является нетривиальной и весьма актуальной проблемой [4].

В статье рассматриваются два инструментальных комплекса (ИК), предоставляющих различные возможности для создания и применения РППП в среде, организованной на базе ресурсов центра коллективного пользования (ЦКП), с учетом особенностей предметных областей разрабатываемых пакетов.

1 Средства разработки распределенных пакетов прикладных программ

РППП ориентируются, как правило, на класс задач, характеризующихся следующими свойствами: решение задачи требует использования больших объемов вычислительных ресурсов (процессорного времени, оперативной памяти, дискового пространства и других характеристик); возможна декомпозиция общей сложной задачи на более простые (с вычислительной точки зрения) подзадачи; процесс решения общей задачи подразумевает распределенное решение набора ее взаимосвязанных подзадач; не предполагается интенсивного взаимодействия между параллельными вычислительными процессами; задача допускает декомпозицию данных на блоки и независимую параллельную обработку этих блоков. К такому классу относятся, например, задачи с многовариантными расчетами, а также многие 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 предоставляют такие возможности гибкого управления вычислительным процессом решения задачи, как экспериментальное прогнозирование времени выполнения, обработка событий, обеспечение надежности и декомпозиция ее подзадач.

Известен широкий спектр средств управления workflow. В их числе известные системы Askalon, Condor DAGMan, Grid Ant, Grid Flow, Karajan, Kepler, Pegasus, Taverna, Triana, UNICORE и другие [9]. Однако анализ функциональных возможностей этих систем показывает, что проблемы формирования и обработки ими масштабируемых потоков заданий для ЦКП не решены в полной мере [10].

Ниже рассматриваются два ИК, базирующихся в отличие от вышеперечисленных средств на использовании концептуальной модели ГРВС, интегрирующей знания как о предметных областях решаемых задач, так

и среде выполнения РППП, а также на применении специализированной системы мета-мониторинга 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.

2 Концептуальная модель ГРВС

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 решаемой 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 и стандартизированном формате, который может быть использован различными программными комплексами, функционирующими в рамках данной среды.

Концептуальная модель проблемно-ориентированной ГРВС, используемая рассматриваемыми в статье ИК, позволяет осуществить взаимосвязанное представление алгоритмических знаний предметных областей решаемых задач, а также 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.

Модель обеспечивает фундаментальную основу для разработки алгоритмов управления вычислениями в ГРВС, включая алгоритмы классификации заданий, планирования вычислений и распределения ресурсов. Методы и средства ее построения представлены более детально в работе [11].

3 Инструментальный комплекс DISCOMP

ИК DISCOMP [13] предназначен для поддержки основных этапов разработки и применения РППП, характеризующихся перечисленными ниже особенностями.

Функциональное наполнение РППП 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 платформо-независимыми 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. Особенности функционирования мультиагентной системы для управления распределенными вычислениями приведены в работе [12].

Решение задачи выполняется, как правило, по одной слабо меняющейся схеме, требующей динамического управления процессом вычислений. Схема решения задачи строится в параллельно-ярусной форме на основе процедурной постановки задачи: “зная M выполнить Q ”, где M и Q — это соответственно модель ГРВС и множество модулей, необходимых для решения задачи.

Выполнение схемы решения задачи осуществляется в режиме интерпретации. Предварительная классификация заданий по выполнению схем решения задач позволяет оптимально назначать ресурсы модулям.

ИК DISCOMP поддерживает виртуализацию ресурсов ГРВС. Если программно-аппаратные ресурсы какого-либо узла среды не соответствуют требованиям модуля схемы решения задачи, то для выполнения такого модуля может быть запущена виртуальная машина.

Концептуальная модель ГРВС и схемы решения задач описываются в текстовом виде с использованием расширения языка XML. На рис. 1 представлены фрагменты текста описания модуля и его параметров (рис. 1 а), а также постановки задачи (1 б), включающей этот модуль.

<pre> <!-- секция спецификаций параметров --> <parameters> <param name='list' type='filelist' filepath='list_%1.txt' /> <param name='result' type='filelist' filepath='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>	<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>
---	--

Рис. 1: Фрагмент описания модели (1 а) и постановки задачи (1 б)

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 новой схемы решения задачи;
- завершение избыточных вычислительных процессов в случае нахождения решения задачи;
- идентификацию и устранение отказов ресурсов узла, а также другие операции, направленные на повышение надежности вычислительного процесса.

Интерпретатор схемы решения задачи получает данные о состоянии вычислительных процессов и ресурсов, необходимые при обработке событий, от специализированной системы метамониторинга [14].

В целом характерными особенностями ИК DISCOMP являются: представление пакетных знаний на языке XML; построение схемы решения задачи на основе процедурной постановки задачи; выполнение вычислительного процесса в режиме интерпретации; предоставление конечному пользователю гибких языковых средств управления процессом вычислений.

4 Инструментальный комплекс Orlando

ИК Orlando [15] предоставляет пользователям графический язык описания предметной области РППП и постановок задач, которые могут формулироваться как в процедурной форме, так и в непроцедурной форме “зная M вычислить Y по X ”, где X и Y — это соответственно множества исходных и целевых параметров задачи. В последнем случае производится автоматический синтез схемы решения задачи (абстрактной программы). Схема решения задачи, сформированная 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. Схема решения задачи выполняется в режиме интерпретации. На рис. 2 приведен пример графического описания схемы решения задачи в ИК Orlando.

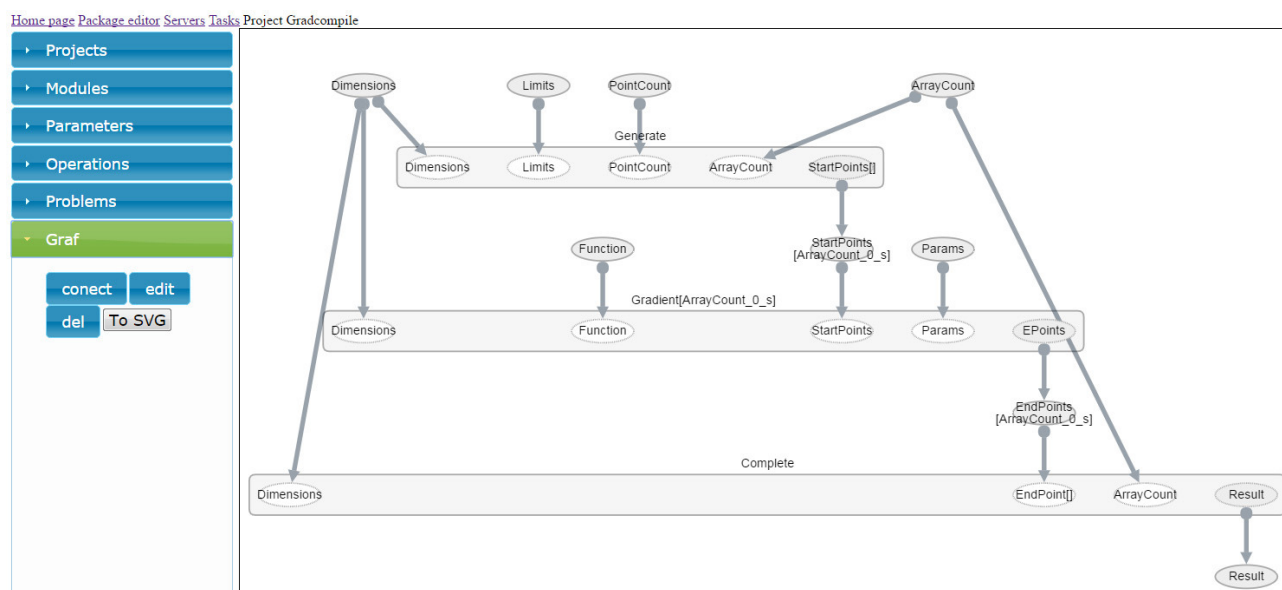


Рис. 2: Описание схемы решения задачи в ИК Orlando

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

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, имеющихся в других пакетах. Вследствие этого, сокращаются сроки разработки РППП и проведения вычислительных экспериментов.

5 Применение инструментальных комплексов

Разработанные ИК используются в ЦКП “Иркутский суперкомпьютерный центр СО РАН” [16] в качестве сред разработки РППП для решения научных и прикладных задач из разных предметных областей. В их числе можно выделить задачи построения множеств достижимости управляемых летательных объектов [17], выполнимости булевых формул [18], поиска глобального минимума многоэкстремальных функций [19], исследования биоресурсов озера Байкал [20], обеспечения безопасности энергетических систем [21] и оптимизации складской логистики [22].

Результаты экспериментов, приведенные в вышеперечисленных публикациях, показывают ускорение и эффективность вычислений близкие соответственно линейному ускорению и эффективности, равной единице. В разных экспериментах использовалось от 64 до 3500 ядер.

Для ряда задач были проведены модельные эксперименты с использованием ресурсов, включавших до 35000 ядер. Эксперименты показали, что результаты по ускорению и эффективности, аналогичные вышеупомянутым результатам, могут быть достигнуты при росте числа ядер моделируемых ресурсов за счет увеличения размерности задачи или уменьшения допустимой погрешности вычислений, приводящих к увеличению вычислительной нагрузки.

Заключение

В статье представлены высокоуровневые инструментальные средства создания и применения РППП. Методология выполненных работ базируется на комплексном применении методов и средств искусственного интеллекта: баз знаний, концептуального крупноблочного программирования и синтеза программ, мультиагентных технологий, а также методов оценки надежности, планирования и распределения вычислительных ресурсов.

ИК DISCOMP ориентирован преимущественно на подготовленных специалистов, способных самостоятельно описывать предметную область и постановки задач на входном языке программирования. ИК ORLANDO предоставляет пользователям и разработчикам пакетов визуальный графический интерфейс, облегчающий выполнение основных этапов создания и применения РППП.

РППП, разработанные с использованием предложенных инструментальных средств, продемонстрировали эффективность функционирования при решении научных и прикладных задач. Основными преимуществами их применения являются: сокращение времени разработки проблемно-ориентированных сред параллельных и распределенных вычислений; сокращение затрат на создание схем решения задач при проведении широкомасштабных вычислительных экспериментов.

Список литературы

- [1] Горбунов-Посадов М.М., Корягин Д.А., Мартынюк В.В. Системное обеспечение пакетов прикладных программ. М.: Наука, 1990.
- [2] Шамакина А.В. Обзор технологий распределенных вычислений // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2014. Т. 3, № 3. С. 51–85.
- [3] Соколинский Л.Б., Шамакина А.В. Методы управления ресурсами в проблемно-ориентированных вычислительных средах // Программирование. 2016. № 1. С. 26–38.
- [4] 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.
- [5] Foster I., Kesselman C. Globus: A metacomputing infrastructure toolkit // International Journal of High Performance Computing Applications. 1997. V. 11, № 2. P. 115–128.
- [6] Couvares P., Kosar T., Roy A., Weber J., Wenger K. Workflow management in condor // Workflows for e-Science. London: Springer, 2007.
- [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] Воеводин В.В. Решение больших задач в распределенных вычислительных средах // Автоматика и телемеханика. 2007. № 5. С. 32–45.
- [9] Talia D. Workflow Systems for Science: Concepts and Tools // ISRN Software Engineering. 2013. V. 2013. P. 1–15.
- [10] Tao J., Kolodziej J., Ranjan R., Jayaraman P., Buyya R. A Note on New Trends in Data-Aware Scheduling and Resource Provisioning in Modern HPC Systems // Future Generation Computer Systems. 2015. V. 51, № C. P. 45–46.

- [11] 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.
- [12] Бычков И.В., Опарин Г.А., Феоктистов А.Г., Богданова В.Г., Пашинин А.А. Сервис-ориентированный мультиагентный подход к управлению распределенными вычислениями // *Автоматика и телемеханика*. № 11. С. 118–131.
- [13] Сидоров И.А., Опарин Г.А., Феоктистов А.Г. Разработка и применение распределенных пакетов прикладных программ // *Программные продукты и системы*. 2010. № 2. С. 108–111.
- [14] Bychkov I., Oparin G., Novopashin A., Sidorov I. Agent-based approach to monitoring and control of distributed computing environment // *Lecture Notes in Computer Science*. 2015. V. 9251. P. 253–257.
- [15] Феоктистов А.Г., Горский С.А. Технология построения пакетов прикладных программ для вычислительных кластеров // *Параллельные вычисления и задачи управления: Тр. III Междунар. конф. М.: Изд-во ИПУ РАН, 2006. С. 498–504.*
- [16] Иркутский суперкомпьютерный центр СО РАН. [Электронный ресурс]. Режим доступа: <http://hpc.icc.ru> (дата обращения: 26.05.2017).
- [17] 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.*
- [18] Заикин О.С., Семенов А.А., Сидоров И.А., Феоктистов А.Г. Параллельная технология решения SAT-задач с применением пакета прикладных программ D-SAT // *Вестник ТГУ*. 2007. № 23. Приложение. С. 83–95.
- [19] Феоктистов А.Г., Горский С.А. Реализация метода мултистарта в пакете Градиент // *Вестник Новосибирского государственного университета. Серия: Информационные технологии*. 2007. Т. 5, вып. 2. С. 78–82.
- [20] 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, № 1. P. 84–88.
- [21] Edelev A.V, Sidorov I.A. Combinatorial Modeling Approach to Find Rational Ways of Energy Development with Regard to Energy Security Requirements // *Lecture Notes in Computer Science*. 2017. V. 10187. P. 310–317.
- [22] Bychkov I., Oparin G., Tchernykh A., Feoktistov A., Bogdanova V., Dyadkin Yu., Andrukhova V., Basharina O. Toolkit for Simulation Modeling of Logistics Warehouse in Distributed Computing Environment // *Информационные технологии и нанотехнологии: Сб. тр. III Междунар. конф. и молодежной шк. Самара: Новая техника, 2017. С. 1106–1111.*

*Александр Геннадьевич Феоктистов — к.т.н., ст. науч. сотр.
Института динамики систем и теории управления им. В.М. Матросова СО РАН;
e-mail: agf65@yandex.ru;*

*Иван Александрович Сидоров — к.т.н., науч. сотр.
Института динамики систем и теории управления им. В.М. Матросова СО РАН;
e-mail: ivan.sidorov@gmail.com;*

*Сергей Алексеевич Горский — к.т.н., науч. сотр.
Института динамики систем и теории управления им. В.М. Матросова СО РАН;
e-mail: gorskysergey@mail.ru.*

Дата поступления — 31 мая 2017 г.