

## СПЕЦИАЛЬНЫЙ ВЫПУСК

### ПЕРСПЕКТИВНЫЕ ТЕХНОЛОГИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

*Под редакцией доктора технических наук, профессора А. В. Бухановского*

## СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ .....	5
<b>СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ</b>	
Бухановский А. В., Васильев В. Н., Виноградов В. Н., Смирнов Д. Ю., Сухоруков С. А., Яппаров Т. Г. CLAVIRE: перспективная технология облачных вычислений второго поколения .....	7
Бухановский А. В., Житников А. Н., Петросян С. Г., Слоот П. М. А. Высокопроизводительные технологии экстренных вычислений для предотвращения угрозы наводнений .....	14
<b>ТЕХНОЛОГИЯ IPSE СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ</b>	
Марьин С. В., Ковальчук С. В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде .....	21
Ковальчук С. В., Маслов В. Г. Интеллектуальная поддержка процесса конструирования композитных приложений в распределенных проблемно-ориентированных средах .....	29
Князьков К. В., Ларченко А. В. Предметно-ориентированные технологии разработки приложений в распределенных средах .....	36
<b>ТЕХНОЛОГИИ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД</b>	
Спельников Д. М., Гуськов А. А., Маслов В. Г., Бухановский А. В. Учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“ на основе Грид-среды .....	44
Чуров Т. Н., Насонов Д. А., Штейнберг Б. Я., Бухановский А. В. Особенности вероятностного анализа производительности и надежности проблемно-ориентированных сред облачных вычислений .....	51
<b>ПРИЛОЖЕНИЯ ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННЫХ СРЕД НА ОСНОВЕ IPSE</b>	
Безгодов А. А., Иванов С. В., Косухин С. С., Бухановский А. В. Виртуальный полигон для исследовательского проектирования морских объектов и сооружений .....	58
Иванов С. В., Болгова Е. В., Каширин В. В., Якушев А. В., Чугунов А. В., Бухановский А. В. Web-ориентированный производственно-исследовательский центр „Социодинамика“ .....	65

Болгова Е. В., Иванов С. В., Гринина Е. А., Слоот П. М. А., Бухановский А. В. Параллельные алгоритмы моделирования динамических процессов на комплексных сетях .....	72
--	----

## **ПРОДВИЖЕНИЕ ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ**

Бухановский А. В., Марьин С. В., Князьков К. В., Сиднев А. А., Жабин С. Н., Баглий А. П., Штейнберг Р. Б., Шамакина А. В., Воеводин В. В., Головченко Е. Н., Фалалеев Р. Т., Духанов А. В., Тарасов А. А., Шамардин Л. В., Моисеенко А. И. Результаты реализации проекта „Мобильность молодых ученых“ в 2010 году: развитие функциональных элементов технологии iPSE и расширение состава прикладных сервисов.....	80
--	----

## **КРАТКИЕ СООБЩЕНИЯ**

Штейнберг Б. Я., Кравченко Е. Н., Морылев Р. И., Нис З. Я., Петренко В. В., Скиба И. С., Шаповалов В. Н., Штейнберг О. Б., Штейнберг Р. Б. Особенности реализации распараллеливающих преобразований программ в системе ДВОР.....	87
Фадеев Д. А. Особенности численного решения эволюционных задач распространения коротких лазерных импульсов на системах архитектуры NUMA.....	89
Боголепов Д. К., Сопин Д. П., Ульянов Д. Я., Турлапов В. Е. Построение SAN BVN деревьев для трассировки лучей с использованием графических процессоров.....	92
Насонов Д. А., Иванов С. В., Бухановский А. В. Распределенная программная платформа для протипирования интеллектуальных систем управления сложными динамическими объектами .....	94
Князьков К. В. Особенности работы с потоками задач длительного исполнения в рамках концепции iPSE.....	97
Сысоев А. В. О построении семейства множественных разверток на основе кривых Пеано для параллельного решения задач глобально-оптимального поиска .....	100
Мостаманди С. В., Насонов Д. А., Калюжная А. А., Бухановский А. В. Ансамблевые прогнозы экстремальных гидрометеорологических явлений в распределенной среде CLAVIRE .....	102
SUMMARY (перевод Ю. И. Копилевича).....	105

## SPECIAL ISSUE

# ADVANCED DISTRIBUTED COMPUTING TECHNOLOGIES

*By Edition A. V. Boukhanovsky, Doctor of Technical Science, Professor*

## CONTENTS

PREFACE .....	5
<b>ACTUAL TENDENCIES FOR DISTRIBUTED COMPUTING</b>	
<b>Boukhanovsky A. V., Vasiliev V. N., Vinogradov V. N., Smirnov D. Yu., Sukhorukov S. A., Yapparov T. G.</b> CLAVIRE: Perspective Technology for Second Generation Cloud Computing .....	7
<b>Boukhanovsky A. V., Zhitnikov A. N., Petrosyan S. G., Sloot P. M. A.</b> High-Performance Technologies of Urgent Computing for Flood Hazard Prevention .....	14
<b>IPSE TECHNOLOGY FOR DISTRIBUTED PROBLEM-SOLVING ENVIRONMENTS FOR SIMULATION AND DATA PROCESSING</b>	
<b>Maryin S. V., Kovalchuk S. V.</b> Service-Oriented Platform for Execution of Composite Applications in Distributed Environment .....	21
<b>Kovalchuk S. V., Maslov V. G.</b> Intellectual Support of Composite Application Design in Distributed Problem-Oriented Environment .....	29
<b>Knyazkov K. V., Larchenko A. V.</b> Domain-Specific Technologies for Application Design in Distributed Environments .....	36
<b>LIFE CYCLE SUPPORT FOR DISTRIBUTED PROBLEM-SOLVING ENVIRONMENTS</b>	
<b>Spelnikov D. M., Guskov A. A., Maslov V. G., Boukhanovsky A. V.</b> Scientific-Training Complex “Computer Modeling in Nanotechnologies” on the Base of a Grid System .....	44
<b>Churov T. N., Nasonov D. A., Shteinberg B. Ya., Boukhanovsky A. V.</b> Features of Probabilistic Analysis of Efficiency and Reliability of Problem-Oriented Environments for Cloud Computing .....	51
<b>IPSE-BASED APPLICATIONS OF DISTRIBUTED ENVIRONMENTS</b>	
<b>Bezgodov A. A., Ivanov S. V., Kosukhin S. S., Boukhanovsky A. V.</b> Virtual Range for Research Design of Maritime Objects and Constructions .....	58
<b>Ivanov S. V., Bolgova E. V., Kashirin V. V., Yakushev A. V., Chugunov A. V., Bukhanovsky A. V.</b> Web-Oriented Research-Production Senter “Sociodynamics” .....	65
<b>Bolgova E. V., Ivanov S. V., Grinina E. A., Sloot P. M. A., Boukhanovsky A. V.</b> Parallel Algorithms of Dynamic Process Modeling in Complex Network .....	72

## DISSEMINATION OF DISTRIBUTED COMPUTING

- Boukhanovsky A. V., Maryin S. V., Knyazkov K. V., Sidnev A. A., Zhabin S. N., Bagly A. P., Shteinberg R. B., Shamakina A. V., Voevodin V. V., Golovchenko E. N., Phalaleev R. T., Dukhanov A. V., Tarasov A. A., Shamardin L. V., Moiseenko A. I.** Results of Project “Young Researchers Mobility” Realization In 2010: Development of Functional Elements of iPSE Technology and Applied Service Widening ..... 80

## BRIEF NOTES

- Shteinberg B. Ya., Kravchenko E. N., Morylev R. I., Nis Z. Ya., Petrenko V. V., Skiba I. S., Shapovalov V. N., Shteinberg R. B., Shteinberg O. B.** Features of Realization of Parallelizing Transformations in Interactive High-Level Optimizing Parallelizer of Codes..... 87
- Fadeev D. A.** Peculiarities of Digital Solution for Evolutional Problems of Short Laser Pulse Propagation with NUMA Architecture Systems ..... 89
- Bogolepov D. K., Sopin D. P., Ulyanov D. Ya., Turlapov V. E.** Construction of SAH BVH Trees for Ray Tracing with the Use of Graphic Processors ..... 92
- Nasonov D. A., Ivanov S. V., Boukhanovsky A. V.** Distributed Program Platform for Prototyping of Intellectual Systems for Control over Complex Dynamic Objects ..... 94
- Knyazkov K. V.** Peculiarities of Long-Term Execution Programs Flows in the Frames of iPSE Concept ..... 97
- Sysoev A. V.** On Family of Multiple Scans Based on Peano Curves Construction for Global Optimization Tasks Parallel Solving..... 100
- Mostamandi S. V., Nasonov D. A., Kalyuzhnaya A. A., Boukhanovsky A. V.** Ensemble Forecasts of the Extreme Hydrometeorological Phenomena in Distributed Environment CLAVIRE..... 102
- SUMMARY ..... 105

*Editor-in-Chief E. B. Yakovlev*

## ПРЕДИСЛОВИЕ

Перспективы развития технологий распределенных вычислений в настоящее время связаны с расширением покрываемых ими предметных областей в условиях глобализации вычислительных инфраструктур. Потенциальный потребитель таких систем — специалист-предметник, ориентированный на решение специфических задач в условиях широкого выбора доступного программного инструментария и вычислительных ресурсов. Как следствие, ставшая традиционной с конца 1990-х гг. задача консолидации и виртуализации распределенных вычислительных ресурсов отчасти теряет актуальность. На первый план выступает проблема развития технологий поддержки жизненного цикла исследований и разработок в существующих распределенных средах с ориентацией, в первую очередь, на мультидисциплинарные задачи в рамках парадигмы „электронной науки“ (eScience) и „электронного производства“ (eEngineering). Технологически это отчасти связано с развитием концепции облачных вычислений (cloud computing), систематизирующей принципы поддержки, использования и коммерциализации распределенных систем с точки зрения организации различных видов сервисов и производств на их основе.

В настоящем выпуске журнала представлены результаты исследований и разработок в рамках перспективной технологии iPSE (Intelligent Problem Solving Environment) создания распределенных проблемно-ориентированных сред. Они связаны с развитием отечественной многопрофильной инструментально-технологической платформы облачных вычислений CLAVIRE (CLOUD Applications VIRTUAL Environment), предназначенной для создания проблемно-ориентированных высокопроизводительных программных комплексов нового поколения в различных предметных областях.

Работы выполнены в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации.

*Доктор технических наук, профессор  
А. В. БУХАНОВСКИЙ*

## PREFACE

At the present moment the prospects of development of technologies of the distributed computing are connected with their subject domains' expansion under the condition of globalization of computing infrastructures. The potential consumer of such systems is Subject Matter Expert focused on the decision of specific tasks and who is provided with a wide range of accessible software tools and computing resources. As consequence, the problem of consolidation and virtualization of the distributed computing resources which became traditional since the end the 1990s partly loses its applicability.

The problem of development of support technologies of Research and Development cycle in already existing distributed environments focused, first of all, on multidisciplinary tasks in the frame of eScience paradigm and eEngineering moves on the foreground. Technologically it is partly connected with development of the cloud computing concept which systematizes the principles of support, usage and commercialization of the distributed systems from the point of view of the organization of various kinds of services and manufacturing on their basis.

This issue presents the results of research and development in frame of perspective technology iPSE (Intelligent Problem Solving Environment) for distributed domain-specific environments building. They are related to the evolution of domestic multidisciplinary cloud computing platform CLAVIRE (CLoud Applications VIRtual Environment) are presented. It is designated for creation of problem-oriented new generation high-performance software solutions in various subject domains. The work is done in the frame of projects on implementation of Decrees № 218 and 220 of the Government of the Russian Federation.

*Doctor of Technical Science, Professor  
A. V. BOUKHANOVSKY*

---

---

# СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

---

---

УДК 681.3.069, 681.324

А. В. БУХАНОВСКИЙ, В. Н. ВАСИЛЬЕВ, В. Н. ВИНОГРАДОВ, Д. Ю. СМИРНОВ,  
С. А. СУХОРУКОВ, Т. Г. ЯПАРОВ

## CLAVIRE: ПЕРСПЕКТИВНАЯ ТЕХНОЛОГИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ ВТОРОГО ПОКОЛЕНИЯ

Рассмотрены особенности технологий облачных вычислений второго поколения, реализующих модель SaaS/AaaS. Описана многофункциональная технологическая платформа CLAVIRE, предназначенная для создания и исполнения композитных приложений на основе распределенных облачных сервисов.

*Ключевые слова:* облачные вычисления, сервисы, ресурсы, виртуализация, композитное приложение.

**Введение.** Парадигма облачных вычислений (cloud computing) ориентирована на модели, методы и технологии, позволяющие предоставлять пользователю удобный доступ к массиву конфигурируемых компьютерных и информационных ресурсов, которые могут быть быстро зарезервированы и высвобождены их провайдером [1]. Она обобщает и систематизирует известные бизнес-модели IaaS (инфраструктура как сервис), PaaS (платформа как сервис) и SaaS (программное обеспечение как сервис), рассматривая их как вложенные механизмы, реализуемые в облаке [2]. При этом в качестве предоставляемых ресурсов могут выступать вычислительные системы, хранилища данных, системные приложения, средства разработки, прикладные программы и композитные приложения на их основе. Для развитых сред облачных вычислений характерна интеграция различных бизнес-моделей, что связано, в первую очередь, с ориентацией на конкретные потребности пользователей в различных предметных областях. Для их классификации и анализа используется модель становления облачных вычислений ССММ (Cloud Computing Maturity Model) [3]. Она декларирует пять этапов развития облачных вычислений.

I. Консолидация и модернизация доступных ресурсов.

II. Виртуализация доступных ресурсов в рамках облачной парадигмы.

III. Автоматизация процессов использования виртуальных ресурсов с обеспечением адаптивности, безопасности и повторяемости.

VI. Обеспечение поддержки автоматизированных сервисов: проведение аудита, проверка отказоустойчивости, обеспечение качества.

V. Полнофункциональная реализация облачной инфраструктуры на основе объединения сервисов, находящихся в различных облачных средах.

В настоящее время развитие облачных инфраструктур (и реализующих их технологий) в полной мере обеспечивает прохождение *первых трех* этапов. Четвертый этап, связанный с проведением аудита, измерением производительности и полным контролем над выполнением

сервисов, к сожалению, находится в зачаточном состоянии, как в силу специфики измерения характеристик сервисов в распределенной среде, так и из-за отсутствия развитых сред облачных вычислений, требующих полномасштабного решения данных задач.

В настоящей статье представлена перспективная разработка, ориентированная на покрытие всех пяти этапов ССММ — многофункциональная инструментально-технологическая платформа CLAVIRE (CLOUD Applications VIRtual Environment).

**Облачные технологии второго поколения.** Технологии облачных вычислений имеет смысл разделить на два класса (по аналогии с технологиями Грид [4]): облачные вычисления первого (С-1) и второго поколения (С-2). Технологии первого поколения соответствуют этапам I—III ССММ и покрывают все три бизнес-модели (IaaS, PaaS, SaaS). Рынок таких решений уже достаточно насыщен, хотя продолжает эволюционировать весьма быстро [5]. Технологии С-2, напротив, во многом остаются предметом исследовательских проектов. Это связано с тем, что требования этапов IV—V ССММ реализуемы только при совокупном использовании возможностей современных информационных технологий и априорных знаний предметных областей, на которые ориентируется разработка. Иными словами, облачные вычисления второго поколения основываются на проблемно-ориентированных технологиях решения комплексных задач в распределенной вычислительной среде. При этом синергетическое сочетание моделей IaaS, PaaS, SaaS приводит к возникновению новой бизнес-модели — AaaS (приложение как сервис). Основной услугой в рамках данной модели является разработка и использование *композиционных приложений* — совокупности взаимодействующих сервисов в облачной среде, ориентированных на решение общей задачи.

В настоящее время развитию технологий С-2 препятствует ряд технологических аспектов:

— разнообразие и неоднородность предметно-ориентированных сервисов в неструктурированном облаке, что делает необходимым развитие интеллектуальных технологий аннотирования, поиска, применения сервисов (формализация знаний);

— декларативная запись композиционных приложений, которая не допускает явной алгоритмической интерпретации. Для решения этой задачи требуются технологии эффективного управления их исполнением в распределенной среде;

— использование коммуникационных сетей общего назначения со случайными вариациями загрузки сказывается на специфике учета, распределения и балансировки нагрузки в облачной среде. Как следствие, целесообразно осуществлять планирование исполнения, квотирование и тарификацию на вероятностной основе;

— высокая инерционность среды облачных вычислений, что требует, в частности, развития специальных технологий сопряжения с системами, работающими в режиме реального времени (например, для интерактивной визуализации в ситуационных центрах).

Совокупное преодоление указанных факторов становится возможным на основе симбиотического подхода в рамках концепции iPSE (Intelligent Problem Solving Environment) [6, 7], которая ориентирована на развитие интеллектуальных технологий поддержки жизненного цикла проблемно-ориентированных сред распределенных вычислений. Данная концепция воплощена в многофункциональной инструментально-технологической платформе (МИТП) CLAVIRE, предназначенной для эффективного управления вычислительными, информационными и программными ресурсами распределенных неоднородных вычислительных инфраструктур в рамках модели облачных вычислений второго поколения. МИТП поддерживает процессы создания, исполнения, управления и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композиционным приложениям, функционирующим на основе облака распределенных прикладных сервисов.



**Принципы функционирования МИТП CLAVIRE.** МИТП может быть использована в основе инновационных производств, предоставляющих пользователям через Интернет высокопроизводительные предметно-ориентированные сервисы в рамках модели облачных вычислений для различных нужд науки, промышленности, бизнеса и социальной сферы, с возможностью использования ресурсов уже существующих распределенных вычислительных инфраструктур (выделенных суперкомпьютеров, Грид, облачных сред первого поколения). Особенности применения и конфигурирования МИТП в конкретных предметных областях определяются в каждом случае общими потребностями данной области в широкодоступных высокопроизводительных вычислениях и наличием прикладного ПО, допускающего предоставление в форме облачных сервисов.

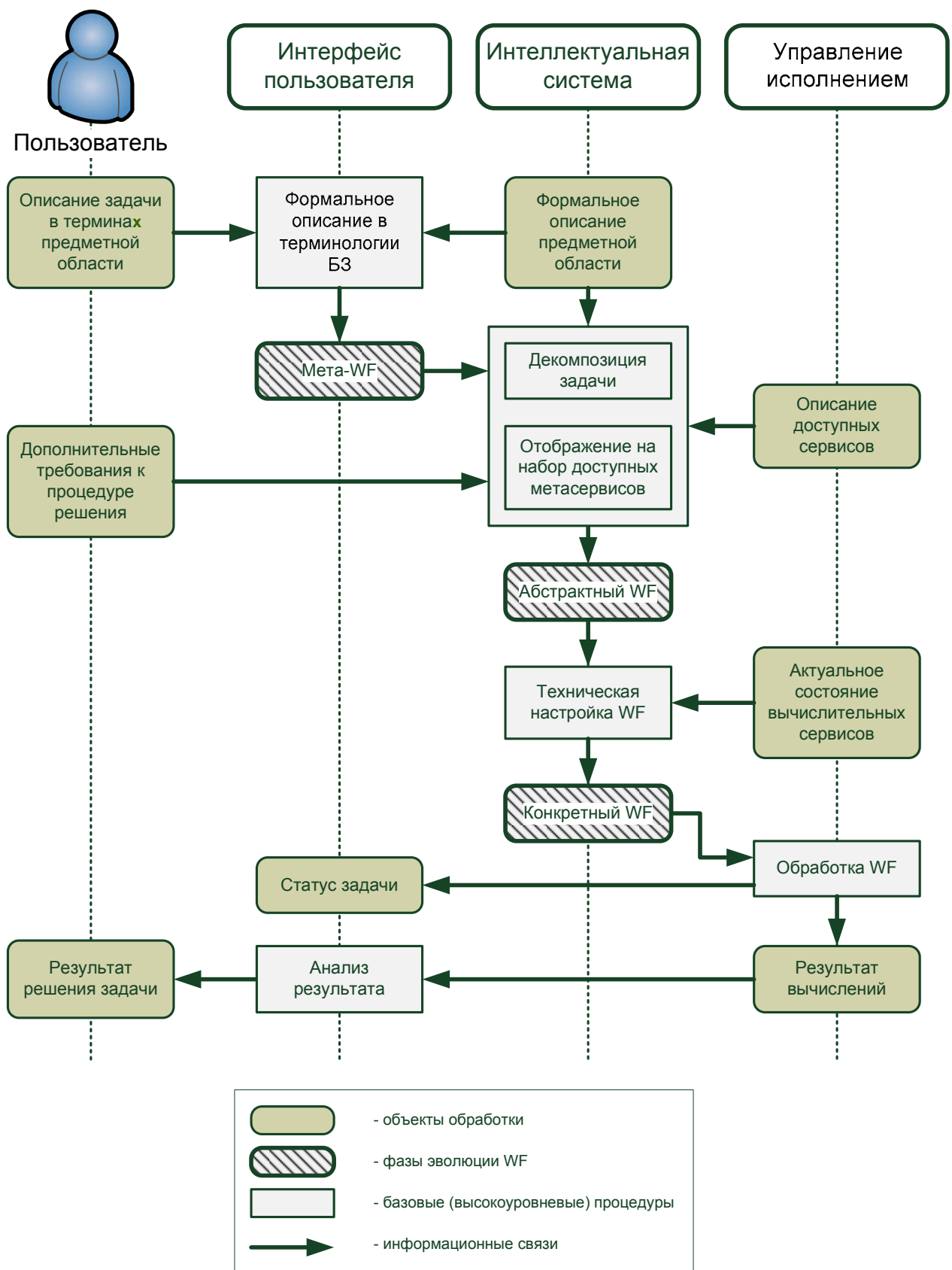
Организация процесса создания и исполнения композитного приложения под управлением МИТП в рамках концепции iPSE сводится к последовательной формализации наборов описаний в терминах потоков заданий (workflow, WF). Верхним уровнем описания приложения является мета-WF (далее MWF). В нем отдельные блоки содержат лишь описания расчетных задач, нужных пользователю, в виде неявных указаний. Таким образом, составленный MWF представляет собой формальное описание пользовательской задачи в терминах предметной области, без указаний на условия ее реализации. Помимо описания действий и данных, необходимых для вычислений, пользователь может задавать критерии, по которым будут подбираться конкретные сервисы, ресурсы, данные (например, время выполнения или высокая надежность), а также дополнительные ограничения и параметры отдельных действий (например, требуемая точность результата).

Проектирование композитного приложения при таком задании исходных данных будет представлять собой процесс поэтапного уточнения MWF через стадии абстрактного (AWF) и конкретного WF (CWF) вплоть до создания конкретных сценариев запуска сервисов в облачной среде и дальнейшего их выполнения. На первом этапе процесса проектирования композитного приложения создается MWF. Пользователь может, например, выбирать классы сервисов, которые доступны в облачной среде, по мере ввода дополнительной информации. Указанные пользователем классы сервисов будут использоваться на следующем этапе для подбора конкретных сервисов. Далее создается WF, в котором уже зафиксированы конкретные реализации вычислительных сервисов, однако еще ничего не известно об условиях их выполнения (AWF). Следующим этапом процесса проектирования является построение расписания и создание сценария выполнения в терминах CWF, который представляет собой WF с полностью определенными блоками. Для блоков действий указаны сервисы и узлы для исполнения, а для блоков данных — конкретное местоположение необходимых данных. На рисунке представлена принципиальная схема создания и исполнения композитного приложения под управлением МИТП, общая для всех способов организации облачной среды.

Более подробно модели, методы и технологии, используемые в МИТП CLAVIRE, рассмотрены в работах [8—11].

**Технологические платформы на основе МИТП CLAVIRE.** МИТП представляет собой инструментальную оболочку, которая позволяет строить различные технологические платформы поддержки облачных вычислений исходя из специфики конкретной задачи.

Технологическая платформа МИТП-К предназначена для создания распределенных корпоративных программных приложений для компьютерного моделирования сложных систем и поддержки принятия решений по управлению комплексными объектами. Специфической особенностью МИТП-К является то, что она использует исключительно корпоративные вычислительные ресурсы. Иными словами, МИТП-К представляет собой программную платформу корпоративного уровня, которая разворачивается на вычислительных ресурсах заказчика (предприятия, имеющего потребность в соответствующих расчетах) и далее используется исключительно его сотрудниками через соответствующую вычислительную сеть.



Такая схема относится к вырожденной модели SaaS/AaaS: само технологическое решение предоставляется предприятию по традиционным вендорским схемам; аналогичным образом предприятием закупается необходимое прикладное ПО для предоставления через МИТП. Предприятие выполняет обязанности провайдера сервисов (исключительно своим сотрудникам); основное преимущество такого подхода связано не с продажами сервисов (как в полной модели SaaS), а с реструктуризацией затрат на обеспечение корпоративной ИТ-инфраструктуры.

Технологическая платформа МИТП-М является инструментальной надстройкой над Грид-средами I поколения, которая обеспечивает высокоуровневый интерфейс к прикладным пакетам, сервисам и композитным приложениям на их основе. За функционирование МИТП-М (в отличие от МИТП-К) отвечает специальная организация — провайдер услуг МИТП-М. В ряде случаев в качестве такой организации может выступать сам оператор Грид-среды, однако это требует приобретения и инсталляции ПО МИТП-М на его собственной инфраструктуре. МИТП-М позволяет реализовать модель SaaS в рамках „публичного“ облака, однако имеет определенные ограничения на способы взаиморасчетов за услуги: в данном случае основным владельцем транспортной сети и прикладных сервисов является оператор Грид-сети, а не провайдер услуг МИТП-М, который и определяет соответствующие правила.

Технологическая платформа МИТП-Ц является специализированным средством для создания инфраструктуры проблемно-ориентированных центров компетенции в различных предметных областях. Задачей центров компетенции является, в частности, интеграция прикладного ПО, покрывающего решение определенного класса задач, а также необходимых для этого вычислительных ресурсов и источников данных. При этом обоснование выбора и принципов взаимосвязи элементов ПО является задачей специалистов-экспертов центра компетенции. Поэтому пользователь такого центра в рамках модели SaaS может рассчитывать не только на доступ к конкретным приложениям (в форме сервисов), но и на услуги в области информационной и интеллектуальной поддержки по выбору и настройке сервисов, а также создания на их основе композитных приложений. Центр компетенции должен обеспечить возможность исполнения выбранного сервиса или приложения на необходимых для этого ресурсах: на собственных выделенных системах или на „арендованных“ мощностях в Грид-средах или облачных средах I поколения. Таким образом, центр компетенции играет роль провайдера услуг МИТП-Ц в рамках модели SaaS. Управляющая инфраструктура МИТП-Ц, равно как и МИТП-К, развертывается на ресурсах внешней организации (в данном случае — центра компетенции), однако, в отличие от МИТП-К, вычислительная инфраструктура ориентирована на использование как собственных, так и внешних ресурсов. МИТП-Ц позволяет реализовать полномасштабную модель SaaS в рамках „публичного“ облака; в отличие от МИТП-М, в данном случае правила предоставления услуг определяются только провайдером МИТП-Ц (самим центром компетенции).

Технологическая платформа МИТП-Э является средством поддержки ситуационных центров, специализирующихся на поддержке принятия решений в различных критических ситуациях, в условиях неопределенности и неполноты исходных данных. В отличие от МИТП-К, МИТП-М, ситуационный центр помимо клиентских (персональных) компьютеров для доступа в Интернет обладает специализированными средствами поддержки принятия решений — широкоэкранными системами визуализации и виртуальной реальности. Потому задача МИТП-Э сводится, в первую очередь, к обеспечению эффективного процесса построения и исследования модельных сценариев развития экстремальных ситуаций с возможностью их интерактивной визуализации. Спецификой экстренных вычислений является то, что соответствующая вычислительная архитектура формируется динамически, с выбором ресурсов под решение конкретной задачи за определенное время (не превышающее допустимого времени принятия решений). Учитывая относительную редкость возникновения критических

ситуаций, оснащать ситуационные центры центрами обработки данных (ЦОД), по аналогии с центрами компетенции, нецелесообразно. Потому МИТП-Э разворачивается и эксплуатируется специальной организацией — провайдером услуг МИТП-Э. Ее потребителями являются ситуационные центры, специализированные в различных областях.

Технологическая платформа МИТП-Д является аналогом МИТП-Ц, позволяющим решать задачу поиска и обработки больших объемов данных в распределенных хранилищах. Поскольку эта задача может быть связана с различными предметными областями, провайдером МИТП-Д является не предметно-ориентированный центр компетенции (как для МИТП-Ц), а специализированная организация, предоставляющая разнообразные сервисы работы с данными в Интернете и содержащая соответствующий ЦОД. Предметом деятельности такой организации может быть (помимо собственно SaaS/AaaS) консалтинг по различным направлениям, основанный на анализе и мониторинге „живых“ публичных данных в Интернете. В качестве основного источника публичных данных, обеспечивающих процессы работы с МИТП-Д, рассматриваются социальные сети в Интернете. Несмотря на то что физически эти данные могут не требовать размещения в распределенном хранилище (достаточно использовать серверы оператора социальной сети), при выборе способа их сбора, обработки и анализа исходят из логической распределенности, обусловленной связями между фрагментами данных — узлами социальной сети. Поскольку объемы информации в современных социальных сетях огромны, такой подход оправдан, потому что позволяет отказаться от ресурсоемких операций обработки всей базы данных в пользу связеориентированных технологий (например, обхода базы по соответствующим связям между узлами). Кроме того, это позволяет обеспечить доступ к данным стандартными средствами социальной сети (через Интернет), без специальных договоренностей с оператором о полном доступе к его базе данных (что в большинстве случаев является невозможным).

**Особенности коммерциализации МИТП.** Внедрение и коммерциализация МИТП и технологических платформ на ее основе связаны с задачей организации учета (биллинга) и платежных отношений, поскольку исполнение WF является комплексной услугой, в которой один или несколько образцов ПО используются в рамках модели SaaS/AaaS. При любой системе учета специфика расчетов за пользование композитным приложением состоит в том, что пользователи могут рассматривать обращение к МИТП как разовый процесс, не предполагающий наличия предварительных договорных обязательств. Потому (в идеальной ситуации) все расчеты должны проводиться в электронной форме в режиме реального времени. Дифференцированная форма учета услуг SaaS/AaaS предполагает наличие различных тарифов — от безлимитной подписки с заданным уровнем качества (SLA) до оплаты конкретных вычислительных затрат. Возможны различные формы отношений (от договоров на постоянное обслуживание, заключаемых традиционным путем, до электронных договоров „по факту“ в реальном времени). Кроме того, должны поддерживаться различные механизмы оплаты (традиционные, с помощью средств eCommerce и mCommerce), поддерживающие как банковские, так и небанковские формы платежей. В целом, это позволит сформировать однозначные и легитимные схемы распределения выручки между всеми участниками бизнес-процесса функционирования МИТП (провайдер МИТП, операторы распределенных сред, провайдеры ресурсов и сервисов, разработчики приложений и пр.), учитывающие не только фактические затраты на исполнение конкретного композитного приложения, но и интересы владельцев интеллектуальной собственности (ПО и баз данных), используемой в WF.

**Заключение.** МИТП CLAVIRE предназначена для создания, исполнения и предоставления сервисов доступа к предметно-ориентированным высокопроизводительным композитным приложениям, функционирующим в облаке неоднородных вычислительных ресурсов

корпоративного уровня, уровня центров компетенции, центров обработки данных, инфраструктур экстренных вычислений и систем распределенного хранения и обработки данных. На ее основе реализуется ряд успешных проектов создания проблемно-ориентированных сред облачных вычислений [12, 13].

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации.

#### СПИСОК ЛИТЕРАТУРЫ

1. NIST.gov — Computer Security Division — Computer Security Resource Center [Электронный ресурс]: <<http://csrc.nist.gov/groups/SNS/cloud-computing/>>.
2. Defining the Cloud Computing Framework // Cloud Computing Journal [Электронный ресурс]: <<http://cloudcomputing.sys-con.com/node/811519>>.
3. GTSI Cloud Computing Maturity Model [Электронный ресурс]: <<http://www.gtsi.com/cms/documents/White-Papers/Cloud-Computing.pdf>>.
4. Foster I., Kesselman C. The Grid2: Blueprint for a New Computing Infrastructure. Morgan-Kaufman, 2004. 748 p.
5. Twenty-One Experts Define Cloud Computing // Cloud Computing Journal [Электронный ресурс]: <<http://cloudcomputing.sys-con.com/node/612375/>>.
6. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.
7. Бухановский А. В., Васильев В. Н. Современные программные комплексы компьютерного моделирования e-Science // Изв. вузов. Приборостроение. 2010. Т. 53, № 3. С. 60—64.
8. Ковальчук С. В., Маслов В. Г. Интеллектуальная поддержка процесса конструирования композитных приложений в распределенных проблемно-ориентированных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 29—36.
9. Марьин С. В., Ковальчук С. В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде // Там же. С. 21—29.
10. Князьков К. В., Ларченко А. В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Там же. С. 36—43.
11. Князьков К. В. Особенности работы с потоками задач длительного исполнения в рамках концепции IPSE // Там же. С. 97—100.
12. Иванов С. В., Болгова Е. В., Каширин В. В., Якушев А. В., Чугунов А. В., Бухановский А. В. Web-ориентированный производственно-исследовательский центр „Социодинамика“ // Там же. С. 65—72.
13. Спельников Д. М., Гуськов А. А., Маслов В. Г., Бухановский А. В. Учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“ на основе Грид-среды // Там же. С. 44—50.

#### Сведения об авторах

- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор; E-mail: avb\_mail@mail.ru
- Владимир Николаевич Васильев** — д-р техн. наук, профессор; Санкт-Петербургский государственный университет информационных технологий, механики и оптики; ректор
- Валерий Николаевич Виноградов** — ЗАО „Фирма АйТи. Информационные технологии“, Центр управления проектами, Москва; директор; E-mail: VVinogradov@it.ru
- Дмитрий Юрьевич Смирнов** — ЗАО „Фирма АйТи. Информационные технологии“, Дирекция по работе с государственным и образовательным секторами, Москва; руководитель; E-mail: DSmirnov@it.ru

*Сергей Анатольевич Сухоруков*

— ЗАО „Фирма АйТи. Информационные технологии“, Дирекция по работе с государственным и образовательным секторами, Москва; заместитель руководителя; E-mail: SSukhorukov@it.ru

*Тагир Галеевич Яппаров*

— ЗАО „Фирма АйТи. Информационные технологии“, Москва, председатель совета директоров; E-mail: TYapparov@it.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 681.3.069, 681.324

А. В. БУХАНОВСКИЙ, А. Н. ЖИТНИКОВ, С. Г. ПЕТРОСЯН, П. М. А. СЛООТ

## **ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ТЕХНОЛОГИИ ЭКСТРЕННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПРЕДОТВРАЩЕНИЯ УГРОЗЫ НАВОДНЕНИЙ**

Рассматриваются вопросы применения технологий экстренных вычислений при организации и разработке систем предотвращения угрозы наводнений на примере Санкт-Петербурга.

*Ключевые слова:* экстренные вычисления, поддержка принятия решений, суперкомпьютеры, Грид, прогноз уровня и течений, усвоение данных.

Эффективность предотвращения природных и техногенных катастроф зависит не только от возможности их заблаговременного предсказания, но и от выработки адекватных и своевременных мер, позволяющих минимизировать возможный ущерб. Поэтому в современных технологиях активно используется ресурсоемкое компьютерное моделирование различных сценариев действий в экстремальных ситуациях, что дает возможность исследовать различные варианты эволюции событий, приводящих к неблагоприятным последствиям. Процесс поддержки принятия решений сводится к определению наиболее вероятных сценариев развития ситуации и выявлению „узких мест“, критичных с точки зрения дальнейшего негативного влияния на ее общий исход [1]. Это дает возможность рассматривать формирование мер противодействия, позволяющих уменьшить или вообще устранить последствия экстремальной ситуации, как решение задачи оптимизации.

В целом при таком подходе высокая вычислительная сложность компьютерного моделирования отдельных сценариев совмещается с жестко заданным сроком получения результатов расчетов, напрямую связанным с допустимым временем принятия решений. При этом основной проблемой является существенная неоднородность вычислительных задач, поскольку, в зависимости от развития ситуации, для реализации того или иного сценария могут быть использованы различные вычислительные приложения, предъявляющие различные требования к вычислительным ресурсам и организации данных. Преодолеть эту проблему позволяет специфическое направление суперкомпьютерных технологий — технологии экстренных вычислений (ЭВ) [2].

В настоящей статье рассматриваются аспекты применения технологий ЭВ для организации системы предотвращения угрозы наводнений в Санкт-Петербурге.

**Концепция экстренных вычислений.** Под ЭВ понимается процесс организации высокопроизводительных вычислений в целях компьютерного моделирования экстремальных ситуаций для поддержки принятия решений в условиях ограниченного времени решений. Основная задача ЭВ состоит в том, чтобы динамически сформировать и контролировать пул

ресурсов (данных, прикладных пакетов, вычислительных мощностей) в распределенной среде, достаточный для решения задачи пользователя (в форме задаваемого им сценария) за определенное время. При этом пользователь не участвует в процессе управления данными ресурсами: оно осуществляется низкоуровневыми средствами ЭВ; после завершения задачи используемые ресурсы освобождаются автоматически.

По сравнению с традиционными технологиями высокопроизводительных вычислений для компьютерного моделирования сложных систем, технологии ЭВ имеют свои специфические особенности.

— В системах ЭВ наравне с вычислительными сервисами используются сервисы доступа к данным измерений в режиме реального времени (интерфейсы датчиков, контроллеров технических средств и пр.). В процессе вычислений производится согласование данных измерений и результатов расчетов (например, данные ассимилируются в моделях [3]).

— Вычислительные процессы в рамках модели ЭВ обычно описываются в форме разветвленных (в соответствии с деревом возможных рисков) сценариев развития событий, представляемых в форме потоков заданий (*workflow*, *WF*).

— Задача оптимизации нагрузки в распределенной среде в рамках концепции ЭВ может быть представлена следующим образом: сформировать такой пул ресурсов, который обеспечивал бы решение задачи за время не более заданного. Это требует анализа вариантов распараллеливания вычислений на всех уровнях иерархии (от ядер и процессоров до выбора отдельных целевых систем в распределенной среде).

Как следствие, в технологиях ЭВ отражен ряд противоречивых требований к организации вычислительной инфраструктуры для поддержки принятия решений в экстремальной ситуации [4]. С одной стороны, ограничения по допустимому времени принятия решения требуют использования суперкомпьютеров для реализации отдельных вычислительных задач, в то время как задача консолидации разнородных вычислительных приложений в составе сценариев приводит к необходимости применения технологий класса Грид. Аналогично — необходимость выделения динамических ресурсов по требованию ассоциируется с архитектурой облачных вычислений, а обеспечение надежности получения результата в срок требует резервирования ресурсов в монопольной распределенной системе с географически разнесенными узлами. Поскольку потребность в экстренных вычислениях может возникать достаточно редко (по мере появления экстремальных ситуаций), система ЭВ по сути является динамической: она не располагает собственной выделенной вычислительной инфраструктурой, однако обладает механизмами для того, чтобы обеспечить приоритет исполнения своих задач в распределенных средах общего назначения.

Спецификой ЭВ также является интерактивность процесса поддержки принятия решений, в котором могут участвовать несколько лиц, принимающих решения (ЛПР), имеющих доступ к одним и тем же инструментальным средствам. При этом ЛПР могут самостоятельно формулировать и исследовать разные сценарии развития ситуации в процессе обоснования принимаемого решения. Как следствие, процесс поддержки принятия решений может быть представлен в виде последовательности *сессий* [5]. Для эффективной организации процесса исполнения сессий необходимо наличие гибких механизмов управления вычислительными ресурсами. Они должны учитывать, с одной стороны, специфику самой вычислительной инфраструктуры, а с другой — различные приоритеты решаемых задач, связанные как с организацией процесса принятия решений, так и с предметной областью. Таким образом, решение задач ЭВ не может быть реализовано в полной мере в рамках программно-аппаратного обеспечения высокопроизводительных вычислений широкого профиля и требует развития специфических технологий.

**Предотвращение наводнений как задача экстренных вычислений.** В настоящее время технологии ЭВ находят свое применение в системах раннего предупреждения о

природных катастрофах [6], в том числе связанных с наводнениями, вызванными воздействием длинных волн (цунами и штормовых нагонов). Однако предсказание волн цунами осложнено неопределенностью локализации землетрясения, что в ряде случаев не позволяет обеспечить требуемую заблаговременность даже при использовании ЭВ с практически неограниченным доступом к вычислительным ресурсам. Напротив, возникновение длинных волн синоптической природы (морских нагонов) во многом определяется физико-географическими особенностями конкретных географических районов, что в общем случае позволяет прогнозировать их в оперативном режиме. Такие наводнения достаточно распространены в мире: им подвержены мелководные побережья Балтийского, Северного, Адриатического морей, Мексиканского и Бенгальского заливов, некоторых крупных озер. В России печальную известность снискали нагонные наводнения в Санкт-Петербурге — самые грозные стихийные бедствия города со дня его основания. По сути дела, история Санкт-Петербурга — это история борьбы с наводнениями [7]. Только во время царствования Петра I с 1703 по 1725 г. их было одиннадцать. В 1824 г. уровень воды достиг самой высокой точки в истории Санкт-Петербурга (410 см) — в городе были разрушены 462 дома и 3681 — поврежден. В 1924 г. водой, поднявшейся на 369 см, было затоплено более половины города, снесено 19 мостов, повреждено свыше 5000 домов, выброшено на берег 95 судов.

С учетом тенденций развития городской застройки для противодействия наводнениям уже в 1982 г. началось строительство комплекса защитных сооружений (КЗС). Он расположен в 25 км западнее Санкт-Петербурга; сооружения рассчитаны на подъем воды в Финском заливе до 5 метров. КЗС представляет собой систему каменно-земляных дамб, в теле которых находятся шесть водопропускных и два судопропускных сооружения, оборудованные затворами, которые могут закрываться при угрозе наводнения. Несмотря на то что КЗС сам по себе ограничивает распространение нагонной волны в Невской губе, в полной мере предотвращение наводнения может быть обеспечено только за счет своевременного закрытия затворов судо- и водопропускных сооружений. Поскольку темпы роста уровня воды при наводненческих ситуациях зависят от совокупности природных факторов, оптимальное время и порядок закрытия затворов невозможно указать заранее, исходя только из прогностической информации. Для решения задачи информационной поддержки процессов управления КЗС в условиях наводненческой ситуации в настоящее время разработана и внедрена Система предупреждения угрозы наводнений (СПУН), реализующая элементы концепции ЭВ. Ее основной задачей является построение оптимального плана маневрирования затворами КЗС, который должен обеспечивать наименьшее время нахождения затворов КЗС в закрытом состоянии, достигая при этом полного предотвращения наводнения (уровень воды не поднимается выше 160 см). Заблаговременно закрыть затворы по первым признакам надвигающейся наводненческой ситуации тоже не представляется возможным, поскольку в этом случае ограничивается сток Невы (в свою очередь, это также может привести к подтоплению города). Кроме того, длительное нахождение всех затворов КЗС в закрытом состоянии ограничивает деятельность морского порта Санкт-Петербурга.

**Система предотвращения угрозы наводнений.** В основе функционирования СПУН заложена технология оперативного прогноза наводнений в Санкт-Петербурге. В СПУН используется численная модель динамики водных масс Балтийского моря, воспроизводящая синоптическую изменчивость уровня и морских течений; географические особенности региона позволяют ограничиться двумерной моделью (приближение „мелкой воды“) в криволинейной системе координат. В качестве входных данных для модели динамики водных масс применяются данные метеорологических прогнозов атмосферного давления и ветра, а также водообмена Балтийского и Северного морей через Датские проливы. Модель функционирует в оперативном режиме совместно с моделью синоптической изменчивости морского волнения на основе уравнения баланса волновой энергии в спектральной форме. Информация о



морском волнении необходима не только для определения нагрузок на технические устройства КЗС, но и при использовании в составе модели динамики водных масс с целью уточнения механизма воздействия ветра на водную поверхность. В качестве параметрической характеристики расчетной акватории также используется прогноз ледовой обстановки.

Принципиальной составляющей технологии оперативного прогноза является процедура оперативного усвоения данных наблюдений за гидрометеорологическими процессами в Финском заливе, позволяющая на 30—70 % (в зависимости от заблаговременности) повысить качество прогностических оценок уровня. При использовании автоматизированной системы гидрометеорологических наблюдений в Восточной части Финского залива, входящей в состав КЗС, учитывая региональную специфику, возможно ограничиться локальной процедурой ассимиляции, корректирующей профиль распространяющейся нагонной волны на акватории Финского залива.

На рис. 1 приведен пример корректировки модельных расчетов уровня ( $h$ ) в п. Горный институт за счет усвоения данных, на примере наводнения 16 ноября 2008 г. в Санкт-Петербурге, т.е. ретроспективный прогноз (1 — результаты измерений, 2 — низкочастотная составляющая ошибки, 3 — прогноз с использованием ассимиляции только по пространству, 4 — прогноз с использованием ассимиляции по пространству и времени). Видно, что усвоение данных только в текущий момент времени не позволяет произвести корректировку поведения модели, в частности, модельное решение в течение 6 ч возвращается на прежнюю траекторию, а на графике хода уровня возникают осцилляции. С учетом временной предыстории в процедуре усвоения ход уровня становится более гладким, наблюдается хорошее совпадение расчетов и измерений, ошибка не превышает 10—15 см.

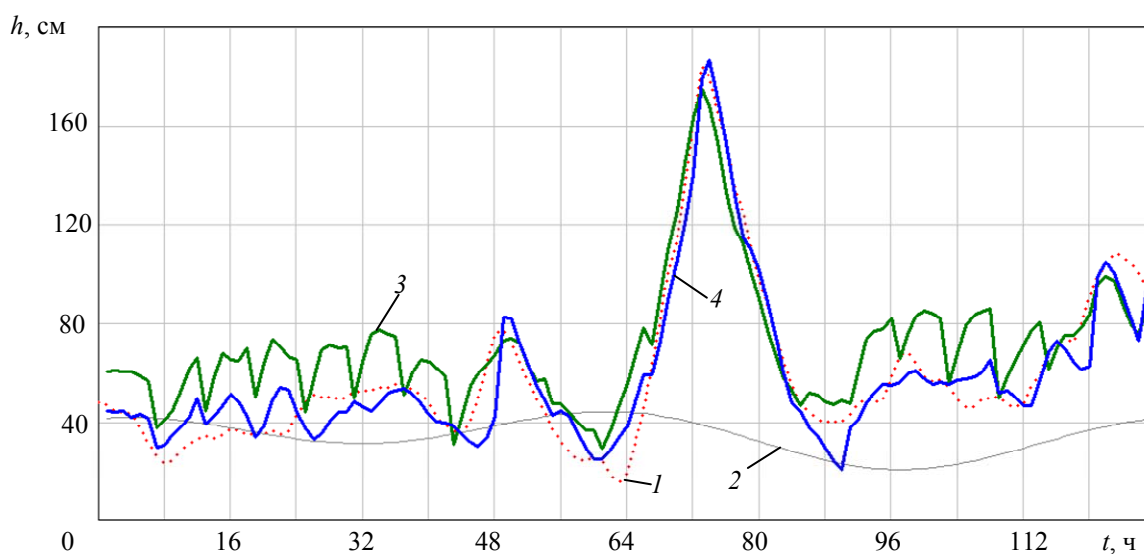


Рис. 1

При создании СПУН отдельное внимание уделяется обеспечению надежности ее функционирования, достигаемой как резервированием вычислительных ресурсов в распределенной системе, так и функциональным резервированием отдельных задач. В частности, технология допускает использование различных (альтернативных) источников метеорологических прогнозов (как отечественных, так и зарубежных), а также корректировку влияния отсутствующих источников входных данных посредством пространственно-временного усвоения текущих наблюдений. Для контроля качества прогностических данных используется альтернативная технология краткосрочных прогнозов уровня на основе вероятностной модели.

Оперативный прогноз уровня воды вырабатывается каждые 6 часов на 48 часов вперед; он лежит в основе функционирования СПУН в целом. В том случае, когда прогнозируется

наступление наводнения, система исходя из данных прогноза формирует план маневрирования затворами КЗС. При этом решается обратная задача: определяется время закрытия затворов КЗС при условии, чтобы за время нахождения затворов в закрытом состоянии не произошло подтопление города за счет ограничения стока Невы; оперативные данные измерений речного стока также доступны в составе СПУН. Для построения плана требуется учет локальных гидродинамических эффектов, связанных с тем, что процедура закрытия затворов КЗС в целом занимает несколько часов; при этом возможно образование дополнительных волновых движений в акватории Невской губы с амплитудой до нескольких десятков сантиметров. Расчет и анализ плана выполняются с учетом неопределенности прогностических оценок на основе интервального подхода.

На рис. 2 приведена концептуальная схема принятия решений при использовании СПУН (СППР — система поддержки принятия решений).

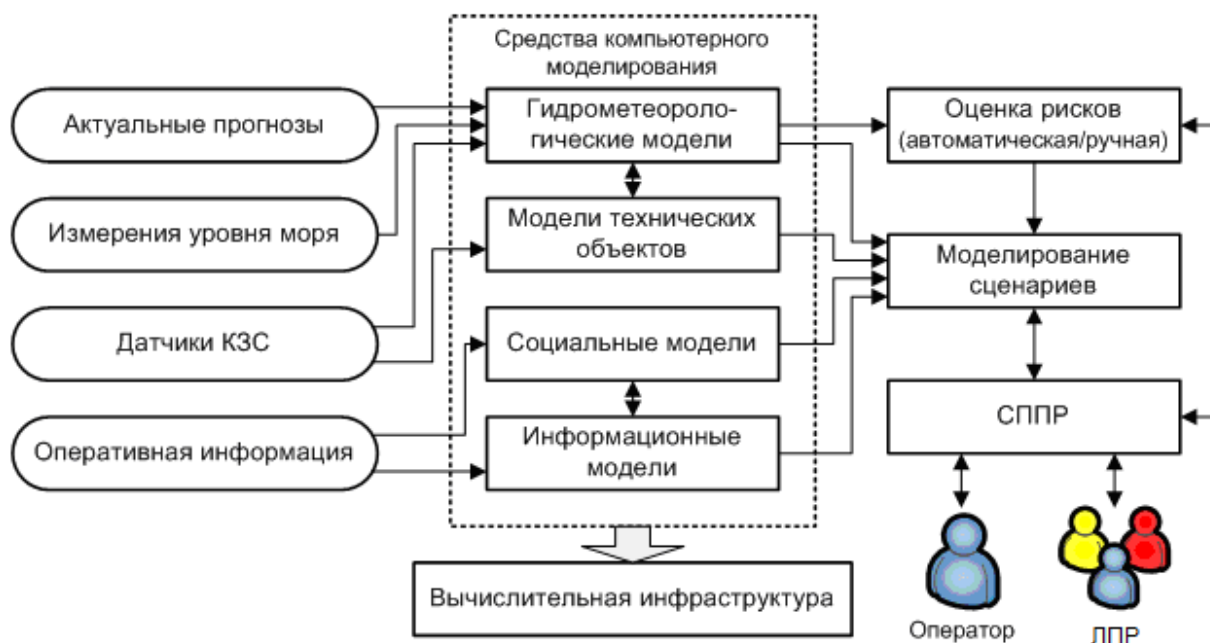


Рис. 2

СПУН в полной мере реализует функции интерактивной советующей системы: несмотря на то что она автоматически формирует непротиворечивый план, пригодный для исполнения КЗС, решение по маневрированию затворами принимает и воплощает диспетчер КЗС (как ЛПР) на основе интегральной информации о состоянии объекта и окружающей среды. Она включает в себя данные альтернативных источников о гидрометеорологической обстановке, сведения о текущем состоянии отдельных технических устройств КЗС и нахождении судов на акватории КЗС, информацию, поступающую от оперативных служб города. В ряде случаев план, предлагаемый системой, может быть отклонен ЛПР. В такой ситуации информационная поддержка ЛПР осуществляется на основе ручного формирования, расчета и сопоставления различных модельных сценариев развития ситуации, с оценкой соответствующих рисков для КЗС и города в целом, для этого применяются технологии ЭВ. В перспективе такой подход требует учитывать не только технические характеристики исполнимости плана, но и специфические аспекты взаимодействия системы „природная среда—КЗС—город“. К ним относятся, в частности, модели развития затопления городских территорий, сценарии, позволяющие координировать действия служб оперативного реагирования, а также модели развития экологической катастрофы.

**Оценка ресурсоемкости вычислений в СПУН.** В целом расчетные компоненты в составе СПУН неоднородны по требованиям к вычислительным ресурсам, периодичности применения и ограничениям на допустимое время получения результата. Например, объем вычислений, необходимых для оперативного прогноза уровня на 48 часов, составляет всего 9 ТФлоп, с использованием пространственно-временного усвоения — 20 ТФлоп. Поскольку время принятия решения в данном случае ограничивается десятками минут (учитывая, что заблаговременность прогноза — 48 часов), для решения данной задачи не требуются специализированные суперкомпьютерные технологии. Однако при построении оптимального плана (решение обратной задачи) объем вычислений составляет уже около 260 ТФлоп. В том случае, когда ЛПР вручную исследуют и сопоставляют различные сценарии оперирования затворами КЗС и время принятия решений составляет минуты, без использования суперкомпьютерных технологий обойтись не представляется возможным. Отдельные требования к ресурсам предъявляют перспективные компоненты, использование которых связано с анализом альтернативных источников данных: например, модель регионального метеорологического прогноза (320 ТФлоп) или процедура ансамблевого пространственно-временного усвоения данных (650 ТФлоп). Учитывая их спорадическое использование, обеспечивать такой компонент постоянно функционирующим вычислительным ресурсом нецелесообразно; по-видимому, оптимален в данной ситуации хостинг инфраструктуры, например, в рамках одной из моделей облачных вычислений. Таким образом, это отражает современные тенденции эволюции систем поддержки принятия решений: для относительно сложных, но локализованных объектов допустимо использовать неспецифические вычислительные платформы; однако учет множественных взаимосвязей объекта с внешней средой делает необходимым применение ЭВ как разновидности суперкомпьютерных технологий.

**Заключение.** Использование технологий ЭВ открывает новые возможности для развития и улучшения существующей системы предотвращения угрозы наводнений в Санкт-Петербурге. Это обеспечивается как экстенсивными факторами (увеличение точности прогнозов, повышение эффективности планов маневрирования затворами), так и расширением состава информационной и интеллектуальной поддержки ЛПР, включая оценку степени неопределенности оценки текущей ситуации и возможность моделирования различных сценариев развития неблагоприятных последствий. В целом такой подход может быть без потери общности перенесен на другие технические объекты аналогичного назначения.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации.

#### СПИСОК ЛИТЕРАТУРЫ

1. Интеллектуальные системы в морских исследованиях и технологиях / Под ред. Ю. И. Нечаева. СПб: ГМТУ, 2001. 352 с.
2. Beckman P., Nadella S., Trebon N., Beschastnikh I. SPRUCE: A System for Supporting Urgent High-Performance Computing // Grid-Based Problem Solving Environments. IFIP 2.5 Conf. Proc. Springer Press. P. 295—316.
3. Gill M., Malanotte-Rizzoli P. Data assimilation in meteorology and oceanography // Adv. Geophys. 1991. Vol. 33. P. 141 — 266.
4. Cencerrado A., Senar M.Á., Cortés A. Support for Urgent Computing Based on Resource Virtualization // Lecture Notes in Computer Sci. 2009. Vol. 5544/2009. P. 227—236.
5. Wolski R., Nurmi D., Brevik J., Obertelli G. Using Batch Controlled Resources to Support Urgent Computing // Urgent Computing Workshop. 2007 [Электронный ресурс]: <<http://spruce.teragrid.org/workshop/talks/Wolski.pdf>>.
6. Allen G., Bogden P., Kosar T., Kulshrestha A. et al. Cyberinfrastructure for Coastal Hazard Prediction // CTWatch Quarterly. Urgent Computing: Exploring Supercomputing's New Role. 2008. Vol. 4, N 1. P. 17—26.
7. Померанец К. С. Три века петербургских наводнений. СПб: Искусство-СПб, 2005. 142 с.

**Сведения об авторах**

- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
E-mail: avb\_mail@mail.ru
- Андрей Николаевич Житников** — ЗАО „Бизнес компьютер центр“, Департамент консалтинговых и программных проектов, Санкт-Петербург; заместитель директора;  
E-mail: azhitnikov@bcc.ru
- Сергей Георгиевич Петросян** — ЗАО „Бизнес компьютер центр“, Департамент консалтинговых и программных проектов, Санкт-Петербург; руководитель проекта;  
E-mail: spetrosyan@bcc.ru
- Петрус Мария Арнольдус Слоот** — доктор философии в области компьютерных наук, профессор; Университет города Амстердам; E-mail: p.m.a.sloot@uva.nl

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

---

---

# ТЕХНОЛОГИЯ iPSE СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ОБРАБОТКИ ДАННЫХ

---

---

УДК 004.75;004.453

С. В. МАРЬИН, С. В. КОВАЛЬЧУК

## СЕРВИСНО-ОРИЕНТИРОВАННАЯ ПЛАТФОРМА ИСПОЛНЕНИЯ КОМПОЗИТНЫХ ПРИЛОЖЕНИЙ В РАСПРЕДЕЛЕННОЙ СРЕДЕ

Разработана интеллектуальная платформа управления исполнением композитных приложений в распределенной вычислительной среде на основе сервисно-ориентированной архитектуры. Платформа адаптирована для использования в составе программного комплекса HPC-NASIS для многомасштабного моделирования в нанотехнологиях.

*Ключевые слова:* интеллектуальное управление, планирование, параметрическая модель, эвристика, сервисно-ориентированная архитектура.

**Введение.** Методология электронной науки (eScience) предусматривает обеспечение коллективного доступа исследователей к разнообразному программному инструментарию компьютерного моделирования и обработки данных посредством распределенной вычислительной среды. Это, в свою очередь, требует развития специализированного класса программного обеспечения — платформ распределенных вычислений (ПРВ) для разработки и исполнения *композитных* приложений, состоящих из нескольких взаимодействующих прикладных сервисов, описываемых в форме потока заданий (workflow, WF). При этом ПРВ в общем случае должна обеспечивать не только исполнение композитных приложений на заданном наборе вычислительных систем, но и *управление* процессом исполнения отдельных сервисов в составе WF с целью обеспечения эффективного использования ресурсов и минимизации общего времени решения задачи. Процесс управления сводится к построению расписания, обеспечивающего синхронизацию работы отдельных сервисов в условиях неоднородности вычислительных ресурсов и стохастической изменчивости параметров коммуникационных сетей и вычислительных систем.

В настоящее время исследование проблемы управления композитными приложениями в форме WF в распределенных вычислительных средах связано с развитием нескольких параллельных направлений. В качестве иллюстрации в табл. 1 приведены характеристики ведущих отечественных и зарубежных решений в данной области [1]. В таблице приведены сведения о модели WF (абстрактный, конкретный [2]), а также способе задания потока: графический, текстовый либо автоматический (по неполному пользовательскому описанию). Также приведены

характеристики планировщика, осуществляющего исполнение WF, и указана целевая функция планирования (время исполнения или квоты на использование ресурсов).

Результаты анализа табл. 1 в целом демонстрируют, что в настоящее время еще не сложилось единого подхода к вопросам управления процессом исполнения композитного приложения в распределенной среде.

Таблица 1

**Характеристики программных платформ управления  
композитными приложениями в распределенной вычислительной среде**

Название системы	Компоновка WF		Планирование			
	модель	составление	архитектура планировщика	уровень планирования	схема	цель
CAEBeans	WF фиксирован*		Централизованная	Задача	Динамическая	Время
СУС ИСА РАН	Абстрактная	Графическое	Планирование вырожденное: каждому вычислительному компоненту априори поставлен в соответствие ровно один вычислительный ресурс			
GridMD	WF задается непосредственно в тексте запускаемой программы (C++)		Используется планировщик той Грид-системы, на которой запускается приложение			
DAGMan	Абстрактная	Текстовое	Централизованная	Задача	Динамическая	Время
Pegasus	Абстрактная	Текстовое Автоматически	Централизованная	Задача WF	Статическая от пользователя Динамическая	Время
Triana	Абстрактная	Графическое	Распределенная	Задача	Динамическая	Время
ICENI	Абстрактная	Текстовое Графическое	Централизованная	WF	Динамическая с предсказанием	Время Квоты
Taverna	Абстрактная Конкретная	Текстовое Графическое	Централизованная	Задача	Динамическая	Время
GrADS	Абстрактная	Текстовое	Централизованная	Задача WF	Динамическая с предсказанием	Время
GridFlow	Абстрактная	Графическое Текстовое	Иерархическая	Задача	Статическая	Время
UNICORE	Конкретная	Графическое	Централизованная	**	Статическая от пользователя	**
Gridbus workflow	Абстрактная Конкретная	Текстовое	Иерархическая	Задача	Статическая от пользователя Динамическая	Квоты
Askalon	Абстрактная	Графическое Текстовое	Распределенная	WF	Динамическая Динамическая с предсказанием	Время Квоты
Karajan	Абстрактная	Графическое Текстовое	Централизованная	**		
Kepler	Абстрактная Конкретная	Графическое	Централизованная	**		

\* В системе CAEBeans WF задается разработчиком конечного комплекса и остается фиксированным для пользователя.

\*\* Архитектура системы подразумевает явную реализацию части стратегии планирования разработчиком конечного комплекса.

Так, часть решений требует явного задания расписания исполнения или использует результаты статического планирования. Вместе с тем ряд платформ (например, Askalon, GrADS, ICENI) позволяет не только динамически планировать процесс исполнения, но и прогнозировать время исполнения с целью дальнейшего мониторинга хода решения задачи. Однако адекватность и достоверность такого прогноза дискуссионны в силу того, что он основывается только на экстраполяции фактических данных измерений времени расчетов и не использует в полной мере априорных знаний предметной области о производительности отдельных предметно-ориентированных сервисов в составе композитного приложения.

В настоящей работе предложен новый подход к управлению процессом исполнения композитного приложения в распределенной среде в условиях неопределенности с использо-

ванием экспертных знаний в форме параметрических моделей производительности сервисов заданной предметной области.

**Модель процесса исполнения композитного приложения в распределенной вычислительной среде.** Подход к решению задачи управления композитными приложениями развивается в рамках концепции iPSE [3]. Концепция предусматривает такой способ описания сервисов в распределенной среде, когда уже на этапе создания сервисной оболочки разработчики прикладных сервисов предоставляют информацию не только об интерфейсах их взаимодействия, но и о характеристиках их производительности. Фактически эта информация также представляет собой экспертное знание, заданное в форме уравнения (параметрической модели) или табличной функции (профиля приложения). Эффективное взаимодействие сервисов в этом случае организуется самой оболочкой управления, которая выполняет операцию логического вывода (строит субоптимальное расписание) на основе знаний о производительности, заложенных в функциональных сервисах, и данных о функционировании распределенной системы в целом, получаемых посредством ее мониторинга в режиме реального времени. Это позволяет выбрать субоптимальную схему исполнения WF за счет управления распределением отдельных сервисов на ресурсах, способами их распараллеливания и маршрутами передачи данных.

Формальный механизм построения описания композитного приложения сводится к последовательности преобразований описания абстрактного WF в конкретный (или частично-конкретный) WF. В качестве модели абстрактного WF выступает ориентированный ациклический граф

$$W_a = \{w_a = (V_a, E_a)\},$$

где множество вершин  $V_a$  — решаемые подзадачи, а множество ребер  $E_a$  — зависимости между ними по данным. Промежуточным этапом построения приложения является частично-конкретный WF:

$$\begin{aligned} W_i &= \{(w_i = (V_i, E_i), state, resource)\}, \\ state : V_i &\rightarrow \{done, running, scheduled, not\_scheduled\}, \\ resource : V_i &\rightarrow C \cup \{\emptyset\}, \end{aligned} \quad (1)$$

где  $state$  — функция отображения множества решаемых подзадач на множество состояний планирования, включающего такие состояния, как „выполнено“, „запущено“, „спланировано“, „не спланировано“;  $resource$  — функция отображения множества решаемых задач на множество доступных ресурсов  $C$  (в случае, если задача находится в состоянии, отличном от „не спланировано“);  $i$  — шаг частично-конкретного WF.

Для составления расписания используется процедура планирования, которая может быть представлена в виде функции следующего вида:

$$sched : W_i \times T'_0 \times H \rightarrow W_i, \quad (2)$$

где  $T'_0$  — множество, содержащее характеристики времени исполнения основных сервисов в составе WF,  $H$  — характеристики распределенной среды. Ход исполнения WF в целом может быть представлен в виде последовательности частично-конкретных WF:

$$\begin{aligned} W_c(w_a \in W_a, sched, t'_0, h \in H) &= \{(w_i)\}, \\ w_0 &= \{w_a, state(v) = not\_scheduled, resource(v) = \emptyset\}, \\ w_i &= sched(w_{i-1}, t'_0, h), i > 0, \end{aligned} \quad (3)$$

при этом функция оценки времени окончания счета на вычислительном ресурсе  $t'_0$  (как основная характеристика процесса синхронизации) представляет собой отображение вида

$$t'_0 : C \rightarrow R^+. \quad (4)$$

Значения  $t'_0$  могут быть получены различными способами, в том числе путем профилировки. Однако в рамках концепции iPSE они интерпретируются как априорные знания предметной области, формой представления которых являются параметрические модели производительности, ассоциированные с доступными вычислительными сервисами предметной области. На рис. 1 приведены графики, иллюстрирующие основные аспекты построения параметрических моделей производительности на примере трех сервисов в области квантовой химии, реализуемых вычислительными пакетами GAMESS, ORCA и MOLPRO (1, 2 и 3 соответственно).

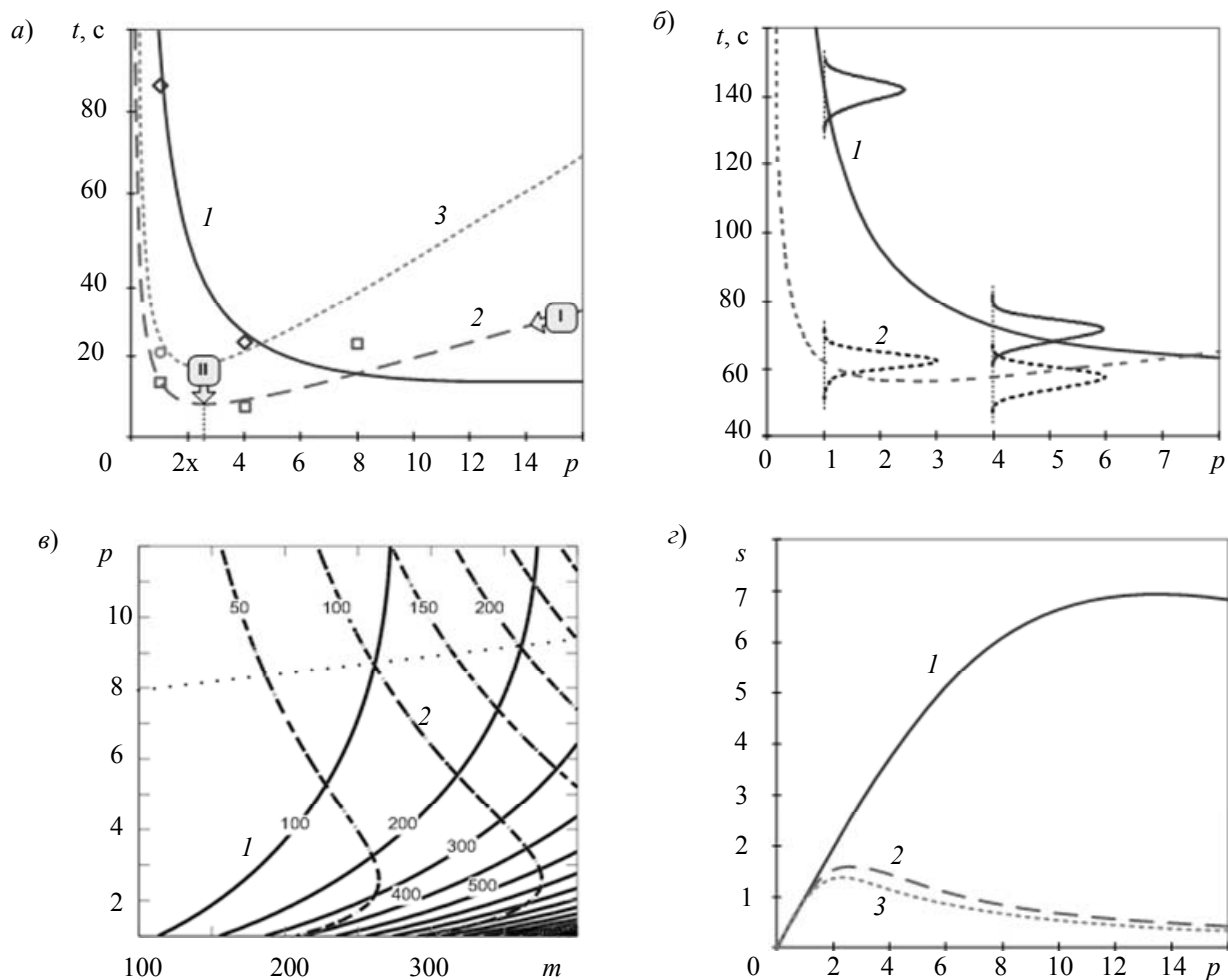


Рис. 1

На рис. 1, а приведен пример модели зависимости времени решения задачи  $t$  от количества вычислительных ядер  $p$ , на которых она распараллелена. Из соображений минимизации времени работы и с учетом ограничения числа вычислителей может быть произведен предварительный выбор вычислительного сервиса (I на рис. 1, а): использование пакета ORCA (I), функционирующего на двух вычислителях (II). На рис. 1, б представлено распределение времени работы пакетов (реализуемых сервисами), полученное на основании экспериментов в среде распределенных вычислений. Время исполнения учитывает накладные расходы на запуск сервиса в распределенной среде, что выражается в параллельном сдвиге графиков вдоль



оси ординат, по сравнению с рис. 1, а. Кроме того, пересечение распределений при  $P = 4$  свидетельствует о неоднозначности решения, построенного по детерминированным моделям производительности. На рис. 1, в приведен график, иллюстрирующий зависимость времени работы пакетов (составляющих основу сервисов) от двух величин: количества базисных функций  $m$  (параметр предметной области) и количества вычислительных ядер (технический параметр). Все пространство изменения этих переменных можно разделить на области, характеризующиеся минимизацией времени при использовании какого-либо из пакетов (что и является критерием выбора). На рис. 1, г приведены графики производного параметра  $s$  (параллельного ускорения), получаемого в процессе моделирования. Как можно заметить, выбор по этому параметру (максимизация ускорения) привел бы к иным результатам (выбору пакета  $I$  — GAMESS). Как следствие, одной из задач, решаемых при построении схемы выполнения, является корректное определение критериев оптимизации в соответствии с потребностями пользователя.

**Интеллектуальная технология планирования процесса исполнения композитного приложения.** Параметрические модели производительности позволяют эффективно описывать лишь характеристики отдельных прикладных сервисов в составе композитного приложения. Определение времени работы WF в целом требует использования специфических подходов, основанных на численном построении алгоритмов планирования на основе различных эвристик, входными данными для которых, в соответствии с (2), (4), являются значения времени работы отдельных сервисов. Для исследования эффективности решения задачи управления процессом исполнения композитного приложения в распределенной среде рассмотрены эвристические алгоритмы планирования MaxMin, MinMin и Sufferage [4]. В результате анализа, проведенного посредством имитационного моделирования, было установлено, что в реальных распределенных системах, вследствие наличия стохастических факторов в изменчивости характеристик вычислительных ресурсов и коммуникационных каналов, возможно только интервальное сопоставление различных сценариев исполнения; при этом в зависимости от конкретного состояния среды может выигрывать та или иная эвристика. Таким образом, нельзя однозначно декларировать целесообразность использования того или иного алгоритма планирования, и необходимо в каждом конкретном случае рассматривать конкурирующие эвристики, вводя при этом критерии их ранжирования. Это позволяет обосновать общую процедуру планирования исполнения композитного приложения в распределенных вычислительных средах в рамках концепции iPSE (рис. 2).

Процедура включает в себя следующие этапы:

— формализация композитного приложения: формирование структуры абстрактного WF исходя из пользовательского описания, состава данных  $\Xi$  и ограничений на режимы исполнения отдельных сервисов;

— определение актуальных параметров распределенной среды (состава и текущих характеристик доступных ресурсов) с использованием инструментов мониторинга вычислительных ресурсов;

— формирование набора активных фактов: оценка характеристик производительности отдельных прикладных сервисов по параметрическим моделям (как форме представления знаний, ассоциированных с элементами WF), а также определение накладных расходов, связанных с вызовом сервисов ( $T_C$ ), передачей ( $T_N$ ) и конвертированием ( $T_D$ ) данных;

— имитационное моделирование сценариев исполнения WF на основе набора конкурирующих эвристик: с использованием априорных знаний о стохастической изменчивости параметров распределенной среды методом Монте-Карло генерируются модельные ансамбли вариантов исполнения композитного приложения;

— интервальное оценивание: по каждой конкурирующей эвристике строится распределение времени исполнения, после чего численно проверяется гипотеза о сходстве-различии результатов для эвристик; в результате выбирается отделимая эвристика, с минимальным средним временем исполнения и ограничением на разброс в сторону увеличения времени исполнения. В том случае, если сценарии исполнения статистически неразделимы, к реализации предлагается схема с минимальным средним временем исполнения.

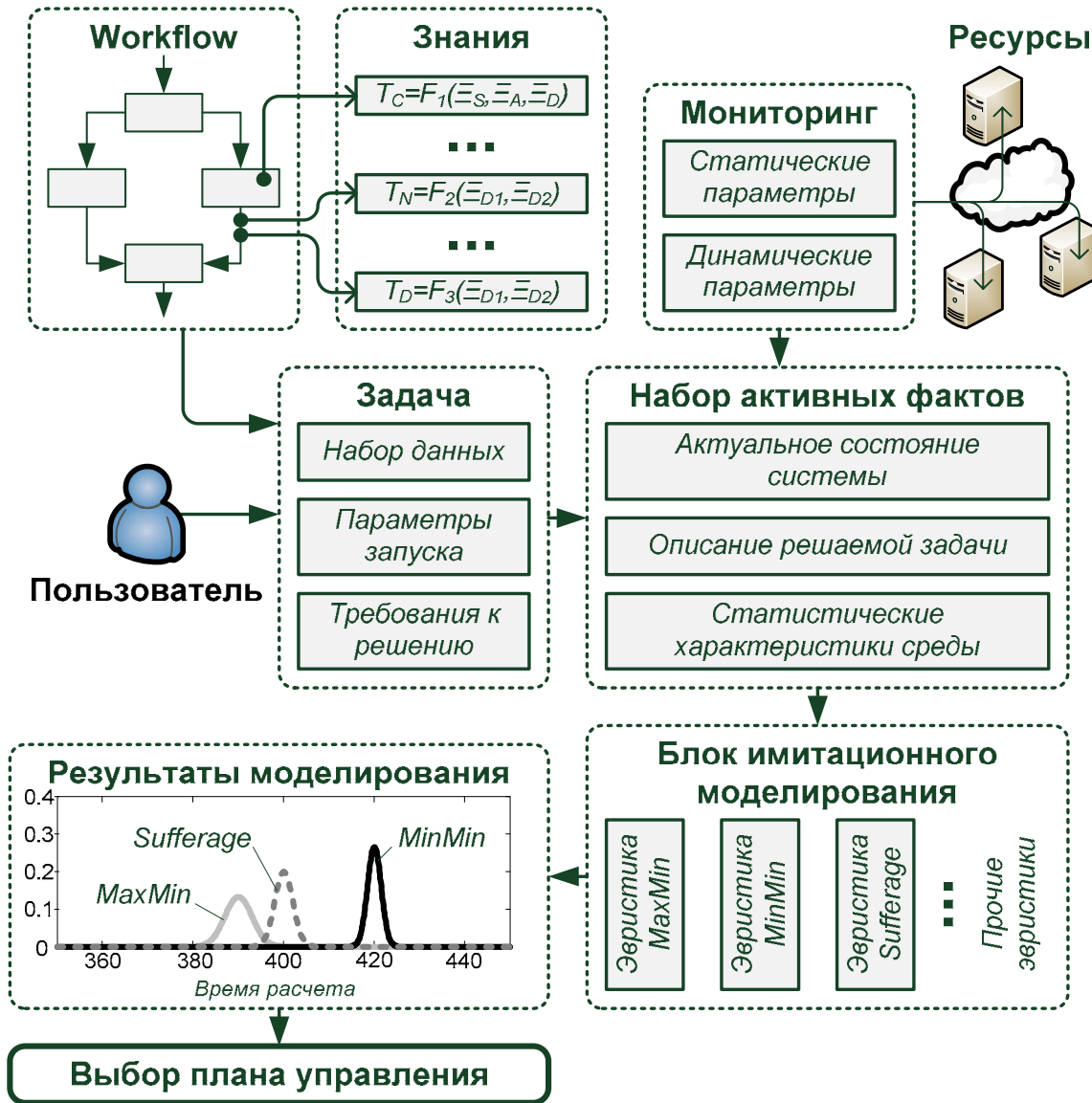


Рис. 2

Таким образом, предложенная интеллектуальная технология позволяет совокупно учесть стохастическую изменчивость характеристик распределенной среды и априорные знания о производительности прикладных сервисов в ходе планирования исполнения композитного приложения.

**Архитектура и реализация сервисно-ориентированной платформы.** Общая архитектура интеллектуальной платформы управления композитными приложениями приведена на рис. 3. Основная работа с компонентами (системными сервисами) в составе платформы осуществляется через интерфейс управляющего ядра, предназначенного для осуществления операций с пользовательскими WF и консолидации работы прочих системных сервисов. В интерфейс управляющего ядра входят базовые команды работы с WF: компоновка и доопределение описания композитного приложения в форме абстрактного

WF; запуск и остановка выполнения WF; получение информации о текущем состоянии WF, включая идентификаторы файлов входных и выходных данных в соответствующем хранилище.

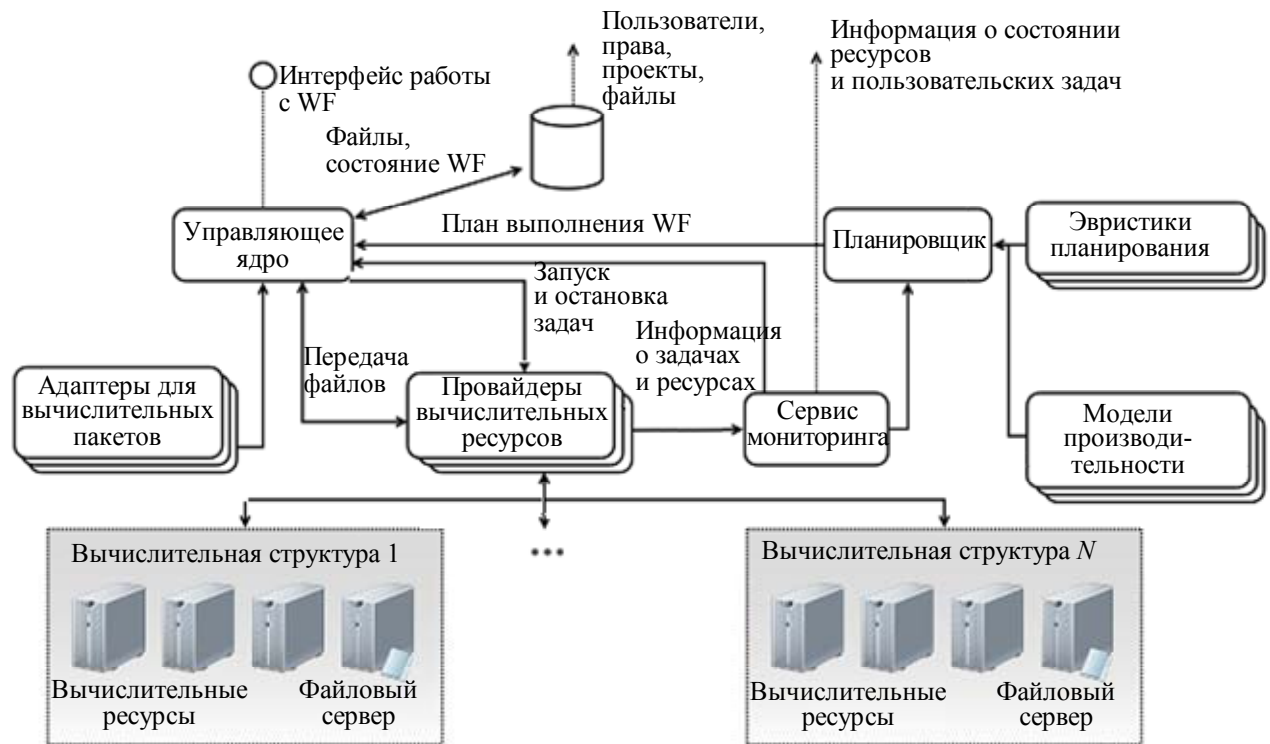


Рис. 3

Основным содержательным элементом платформы является сервис планирования (планировщик), предназначенный для составления расписания запусков, т.е. для отображения списка текущих задач, поступивших от управляющего ядра, на вычислительные ресурсы, информация о состоянии которых поступает от сервиса мониторинга. При планировании используются знания о вычислительных сервисах в форме параметрических моделей производительности и результаты имитационного моделирования в соответствии с процедурой, описанной в предыдущем разделе. Результатом планирования является расписание исполнения отдельных сервисов в составе WF.

Согласно полученному расписанию управляющим ядром осуществляются запуск заданий на соответствующих вычислительных ресурсах, контроль их исполнения, а также пред- и постобработка данных, выполняемая адаптерами вычислительных пакетов. К типичным задачам предобработки относится формирование входного файла для конкретного вычислительного пакета в соответствии с пользовательским описанием в терминах предметной области. К постобработке относится, например, конвертирование данных в необходимый формат для использования другими сервисами в составе WF или более удобного представления пользователю.

Для учета специфики запуска и сбора информации о ходе выполнения в конкретных вычислительных средах используется расширяемый набор провайдеров вычислительных ресурсов. Каждый провайдер — это подпрограмма, поставляемая вместе с платформой или написанная системным программистом, реализующая базовую функциональность взаимодействия с определенным набором вычислительных ресурсов. Такими ресурсами могут быть отдельные кластеры, группа кластеров или ресурсы Грид. После запуска задачи на конкретном вычислительном ресурсе информация о ее состоянии периодически запрашивается сервисом мониторинга. Кроме информации о состоянии задач в сервис мониторинга также поступают данные о

конфигурации и текущей загруженности вычислительных ресурсов. Эти данные передаются другим сервисам: планировщику для использования в процессе выбора ресурсов и построения расписания, управляющему ядру для выполнения необходимых действий при смене статуса задачи (например, при ее окончании или сбое), а также компонентам, взаимодействующим с пользователем для отображения хода выполнения задачи и загруженности ресурсов.

В табл. 2 приведены результаты экспериментального исследования производительности разработанной платформы в составе высокопроизводительного программного комплекса HPC-NASIS [5] для квантово-механических расчетов и моделирования наноразмерных атомно-молекулярных структур. Расчеты выполнялись в режиме метакомпьютинга (выделенные кластеры под управлением HPC-NASIS) и в среде Грид Национальной нанотехнологической сети [6]. Представлены статистические характеристики (среднее время  $M_x$ , СКО  $S_x$ , коэффициент вариации  $V_x = S_x / M_x$ ) по отдельным составляющим накладных расходов платформы управления и среды распределенных вычислений в целом.

Таблица 2

Статистические характеристики составляющих времени накладных расходов (секунды) при исполнении композитного приложения HPC-NASIS

Временные характеристики запуска приложения	Режим метакомпьютинга			Режим Грид		
	$M_x$	$S_x$	$V_x$	$M_x$	$S_x$	$V_x$
Время выбора вычислительного ресурса	14,92	2,10	0,14	—	—	—
Время работы адаптеров платформы управления	0,90	0,06	0,07	0,32	0,08	0,26
Время передачи расчетных данных в хранилище платформы управления	3,62	0,13	0,03	3,13	0,66	0,21
Собственные накладные расходы распределенной среды	10,09	1,90	0,19	186,36	123,97	0,67
Накладные расходы на управление исполнением сервисов в платформе управления	9,16	2,68	0,29	9,70	0,92	0,09
Накладные расходы на управление исполнением WF в платформе управления	6,97	0,31	0,04	0,58	0,20	0,35

Проведенные экспериментальные исследования демонстрируют, что накладные расходы платформы управления сопоставимы с накладными расходами инфраструктуры распределенных вычислений в режиме метакомпьютинга и на порядок меньше накладных расходов в среде Грид, что подтверждает возможность практического использования разработанной платформы управления композитными приложениями в распределенных вычислительных средах, без оказания при этом существенного влияния на общую производительность вычислительной инфраструктуры.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации, при частичной поддержке государственного контракта № 16.647.12.2025 „Создание функционирующего в режиме удаленного доступа интерактивного учебно-методического комплекса для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе, в распределенной вычислительной среде“ и ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Марьин С. В. Интеллектуальная платформа управления композитными приложениями в распределенных вычислительных средах. Дис... канд. техн. наук. СПб: СПбГУ ИТМО, 2010.
2. Deelman E. et al. Pegasus: Mapping scientific workflows onto the grid // Europ. Across Grids Conf. 2004. P. 11—20.

3. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.
4. Yu J. et al. Workflow Scheduling Algorithms for Grid Computing, Metaheuristics for Scheduling in Distributed Computing Environments / Ed. by F. Xhafa and A. Abraham. Berlin: Springer, 2008.
5. Свидетельство о регистрации ПС ЭВМ №2009615970. Ядро высокопроизводительного программного комплекса для квантово-механических расчетов и моделирования атомно-молекулярных наноразмерных структур и комплексов / А. В. Бухановский и др. 26.10.2009.
6. Грид Национальной нанотехнологической сети (ГридННС) [Электронный ресурс]: <<http://www.ngrid.ru/trac/>>.

#### Сведения об авторах

- Сергей Владимирович Марьин** — канд. техн. наук; НИИ Научно-технологических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: sm.niinkt@gmail.com
- Сергей Валерьевич Ковальчук** — канд. техн. наук; НИИ Научно-технологических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник;  
E-mail: kovalchuk@mail.ifmo.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 681.3.069, 681.324

С. В. КОВАЛЬЧУК, В. Г. МАСЛОВ

## ИНТЕЛЛЕКТУАЛЬНАЯ ПОДДЕРЖКА ПРОЦЕССА КОНСТРУИРОВАНИЯ КОМПОЗИТНЫХ ПРИЛОЖЕНИЙ В РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕДАХ

Рассмотрены способы организации интеллектуальной поддержки пользователя в процессе конструирования композитных приложений на основе онтологического подхода. На их основе разработана система интеллектуальной поддержки пользователей при решении задач компьютерного моделирования в области квантовой химии.

**Ключевые слова:** интеллектуальная система, экспертные знания, онтология, логический вывод, композитное приложение, распределенная среда.

**Введение.** Современный этап развития парадигмы eScience характеризуется тенденцией к объединению географически удаленных друг от друга групп исследователей для решения комплексных научных задач, что требует консолидации распределенных вычислительных ресурсов и хранилищ данных. Консолидация позволяет создавать и использовать композитные приложения, которые включают в себя распределенные компоненты, разработанные и поддерживаемые различными группами специалистов и доступные как сервисы в сети Интернет. Поэтому процесс разработки композитных приложений осложняется не только технологическими особенностями стыковки отдельных компонентов (совместимостью форматов и структуры входных и выходных данных), но и методологическими аспектами интерпретации результатов их работы (совместимостью научных подходов, методов, моделей и алгоритмов). По-видимому, экстенсивным путем преодолеть проблему стыковки отдельных компонентов возможно лишь: а) в рамках конкретной задачи, для решения которой используется ограниченное число компонентов, ранее

известных пользователю; б) в ситуациях, когда все компоненты принадлежат одному и тому же разработчику, т.е. их технологическая и методологическая совместимость изначально обеспечена. В остальных случаях эффективное решение данной проблемы возможно только с использованием интеллектуальных технологий, основанных на отчуждаемых знаниях о природе и логической структуре программных компонентов композитного приложения [1].

Современные подходы к интеллектуальной поддержке процесса разработки композитных приложений на основе готовых компонентов (доступных в форме сервисов через Интернет) во многом наследуют особенности методологии Semantic Web [2]: отдельные компоненты рассматриваются в рамках единого пространства знаний в форме онтологии, что позволяет единообразно описывать правила их использования и сопряжения [3, 4]. Однако при таком подходе не уделяется должного внимания интеллектуальной поддержке пользователя в процессе построения самого композитного приложения. Оно может разрабатываться вручную (на основе аннотированных сервисов), с помощью шаблонов, а в ряде случаев — автоматизированно в ходе диалога с пользователем. Поскольку практическая цель построения композитного приложения определяется пользователем исходя из специфики предметной области, дальнейшее развитие технологий интеллектуальной поддержки возможно лишь в рамках симбиотического подхода, учитывающего особенности изучаемых процессов, специфику применяемых методов, алгоритмов и данных, а также отдельные аспекты их программной реализации и исполнения в распределенной вычислительной среде [5].

В настоящей работе подход к созданию системы интеллектуальной поддержки для разработки композитных приложений рассматривается на примере проблемной области, связанной с квантово-химическими расчетами и *ab initio* моделированием атомно-молекулярных наноразмерных структур и комплексов. Данный выбор обусловлен как практической значимостью этого направления в свете развития nanoиндустрии в целом, так и разнообразием входящих в него подходов, методов и технологий, обеспечивающих различные конкурирующие пути решения одной и той же задачи. В качестве среды для разработки и управления композитными приложениями используется многофункциональная инструментально-технологическая платформа CLAVIRE. Использование в ней проблемно-ориентированного языка описания композитных приложений EasyFlow [6] позволяет ограничить задачу интеллектуальной поддержки автоматизированным формированием соответствующего шаблона, на основе которого автоматически генерируется сценарий запуска в распределенной среде.

**Онтологический подход к описанию задач предметной области.** Первопринципное моделирование атомно-молекулярных наноразмерных структур и комплексов, в первую очередь, ориентировано на методы расчета оптических свойств, термодинамических характеристик, электрофизических и механических свойств материалов на их основе. Это приводит к необходимости использования различных методов описания свойств возбужденных состояний и электронных переходов, в том числе метода нестационарной теории функционала плотности (TDDFT), методов связанных кластеров применительно к возбужденным состояниям (EOM(LR)-CCSD), методов решения задач о нормальных колебаниях атомно-молекулярных систем методами квантовой химии, а также метода неравновесной функции Грина (NEGF). Учитывая механические свойства нанообъектов, необходимо также рассмотреть задачи оптимизации геометрии с ограничениями и обсудить особенности применения таких задач для моделирования поведения нанообъектов при механическом воздействии. Для систем очень большого размера целесообразно использовать полуэмпирические методы, а также методы QM/MM (квантовой механики/молекулярной механики), фрагментов и DFTB (функционала плотности сильной связи).

Обилие специфических задач и реализующих их методов приводит к необходимости выделить базовые понятия (*концепты*), позволяющие структурировать соответствующие знания. Исходя из общих принципов компьютерного моделирования можно выделить пять базовых концептов.

1. **Объект моделирования** — ключевой концепт компьютерного моделирования, нацеленного на определение некоторых характеристик данного объекта за счет использования одной или нескольких моделей, ассоциированных с ним. Объект моделирования описывается набором характеристик, которые условно можно разделить на две группы: *определяющие* — характеристики, фиксирующие объект моделирования и отличающие его от других объектов данного класса; *переменные* — характеристики, описывающие состояние объекта моделирования и чаще всего являющиеся входными или выходными данными для процесса моделирования.

2. **Модель** — служит для получения знаний об объекте моделирования. Представляет собой некоторую математическую или логическую структуру, отражающую заданный набор характеристик исследуемого объекта. Для работы с моделью могут быть применены различные методы моделирования, обеспечивающие, например, возможность получения по одним характеристикам модели других.

3. **Метод** — императивное описание процедуры работы с моделью для получения характеристик, проверки гипотезы и т.п. Именно методы чаще всего используются в качестве базового объекта для реализации в составе вычислительных пакетов и представляют наибольший интерес для пользователей этих пакетов. Очевидно, что может существовать несколько реализаций одного и того же метода.

4. **Реализация метода** — алгоритм, включенный в состав доступного пользователю вычислительного пакета и ассоциированный с некоторым методом.

5. **Вычислительный сервис** — вычислительный пакет, установленный на выделенном ресурсе в распределенной среде, используемый в режиме удаленного доступа.

На рис. 1 частично приведена структура онтологии, построенной на основе перечисленных выше концептов. Можно выделить два основных слоя: описание концептов (классов) и индивидов, реализующих концепты. Индивиды могут быть связаны отношениями, определенными на уровне концептов. Кроме того, допустимы отношения между отдельными концептами (например, отношение генерализации). В простейшем случае возможно ограничиться двумерными отношениями, что зачастую приводит к необоснованному усложнению структуры множества отношений за счет декомпозиции семантически связанных многомерных отношений на совокупность двумерных отношений.

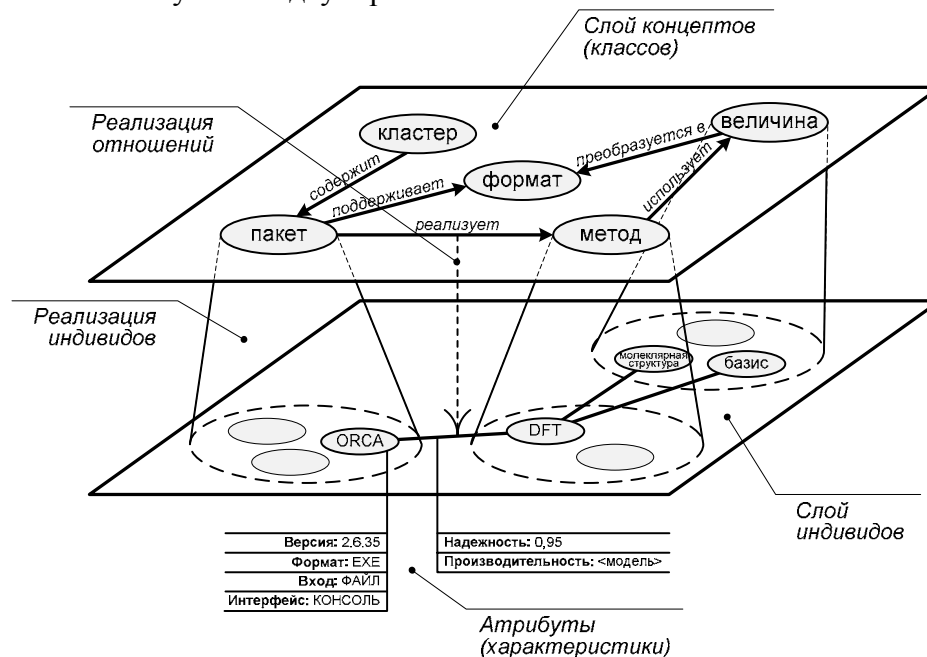


Рис. 1

Одним из элементов онтологии являются атрибуты (характеристики) индивидов, детализирующие их описание. Кроме того, одним из возможных расширений классического подхода

является ассоциация характеристик не только с индивидами (как реализациями классов), но и со связями между ними (как реализациями классов допустимых связей). Такое расширение применимо при введении операций второго порядка (например, анализе „неидеальности“ различного рода для связей, интерпретируемых как экспертные знания). В частности, из рис. 1 видно, что пакет ORCA [7] реализует метод функционала плотности (DFT), использующий в качестве параметров такие объекты, как собственно молекулярная структура и базис, по которому раскладывается решение. Характеристики связи между методом и пакетом позволяют оценить качество реализации (в данном случае в соответствии с критерием надежности), а также получить доступ к модели производительности, соответствующей данной реализации, для последующего использования в процессе моделирования.

Формально слой классов онтологии определяется как граф  $O = \langle C, R \rangle$ , где  $C$  — множество классов,  $R$  — множество абстрактных отношений, связывающих классы. Аналогично слой индивидов онтологии определяется как граф  $\tilde{O} = \langle \tilde{C}, \tilde{R} \rangle$ , где  $\tilde{C}$  — множество индивидов, а  $\tilde{R}$  — множество отношений между индивидами. Для каждого элемента слоя индивидов определены:

а) отношение генерализации  $gn^{(C)} : \tilde{C} \rightarrow C$ ,  $gn^{(R)} : \tilde{R} \rightarrow R$ , определяющее связь индивидов и связей между ними с соответствующими классами и связями классов;

б) „сторожевые“ условия (guard condition), определяющие применимость элементов в данных условиях

$$gc^{(C)}(F) : \tilde{C} \rightarrow \{0,1\}, \quad gc^{(R)}(F) : \tilde{R} \rightarrow \{0,1\},$$

где  $F$  — множество активных фактов, определенных для текущей задачи;

в) функция критериальной оценки

$$k^{(C)}(F) : \{ \tilde{c} \in \tilde{C} \mid gc^{(C)}(\tilde{c}) = 1 \} \rightarrow \Psi^{(C)}, \quad k^{(R)}(F) : \{ \tilde{r} \in \tilde{R} \mid gc^{(R)}(\tilde{r}) = 1 \} \rightarrow \Psi^{(R)},$$

где  $\Psi^{(C)}$  и  $\Psi^{(R)}$  — соответственно пространство критериев оценки индивидов и отношений между ними.

В общем случае сторожевые условия могут рассматриваться как один из критериев оценки. Здесь они выделены в отдельный класс в целях упрощения процедуры критериальной оценки, оперирующей сокращенными множествами  $\tilde{C}$  и  $\tilde{R}$ , для обеспечения логического вывода — построения шаблона композитного приложения на основе анализа описанной выше онтологии.

**Организация логического вывода.** Используя рассмотренную выше онтологию, можно получить структурированную информацию о параметрах, наиболее значимых для решения поставленной задачи, а также оценить доступные пользователю варианты решения поставленной задачи. При этом вариант решения определяется как кортеж  $S = (s_1, s_2 \dots s_N)$  фиксированной структуры,  $i$ -м элементом которого являются множества вида

$$s_i = \{ \tilde{c} \in \tilde{C} \mid gn^{(C)}(\tilde{c}) = c_i \},$$

где последовательность классов  $c_i \in C$  и требований к множествам  $s_i$  определяет общую структуру решения. Для оценки построенного решения по системе критериев применяется анализ графа

$$\tilde{O}' = \langle \tilde{C}', \tilde{R}' \rangle : \tilde{C}' = \bigcup_i s_i \cup \tilde{C}_S,$$

где  $\tilde{C}_S$  — присоединенная система классов

$$\tilde{C}_S = \left\{ \tilde{c}_S \mid \tilde{c}_S \notin \bigcup_i s_i, \exists \tilde{c}_1 \in \bigcup_i s_i : rch(\tilde{c}_S, \tilde{c}_1) \right\},$$



$rch(\tilde{c}_1, \tilde{c}_2)$  — отношение достижимости на графе  $\tilde{O}'$ . Оценка  $\tilde{O}'$  производится в пространстве критериев  $\Psi$ , определяемом пересечением множеств критериев, описывающих пространства  $\Psi^{(C)}$  и  $\Psi^{(R)}$ . С другой стороны, задача ранжирования способов запуска требует определения отображения  $\psi: \Psi \rightarrow R$ , позволяющего вычислить интегральный критерий качества оцениваемого решения.

Используя логический вывод на базе онтологической структуры, возможно построить описания композитных приложений с привлечением экспертных знаний различных уровней абстракции. Представим шаги логического вывода.

1. Производится *формализация требований* пользователя к решаемой задаче: требования формируют базовый набор активных фактов, записанный в терминологии используемой базы знаний. Факты, предоставляемые пользователем, можно разделить на две допускающие пересечение группы: *определяющие* — набор фактов, на основании которого формируется базовая цепочка решения (достаточный набор для однозначного определения решаемой задачи); *дополнительные* — набор требований и ограничений, более точно определяющих специфику решаемой задачи. Базовый набор активных фактов расширяется на основании правил логического вывода, доступных системе. Полученный таким образом набор активных фактов (или несколько альтернативных наборов) используется в последующих процедурах.

2. *Фильтрация*. На основании построенного набора активных фактов производится оценка элементов онтологии в пространстве критериев качества, по которой проводится первоначальная фильтрация онтологии, позволяющая отсеивать элементы, неприменимые для решения поставленной задачи. Отфильтрованная таким образом онтология представляет собой ориентированный граф, в котором каждой вершине или ребру сопоставлено значение в пространстве критериев качества. В простейшем случае этот шаг позволяет выбрать допустимые в данной ситуации элементы онтологической структуры.

3. *Определение доступных методов*. В составе построенного графа могут быть выделены связанные подграфы, каждый из которых состоит из элементов онтологии, определяющих один из доступных методов предметной области. При этом доступность метода определяется: а) реализацией этого метода хотя бы одним из сервисов, доступных в распределенной среде; б) соответствием метода сформированному набору активных фактов (определяется на основании процедуры фильтрации (см. шаг 2)).

4. *Композиция*. На основании полученного набора доступных методов должен быть построен абстрактный сценарий решения поставленной задачи. Структурная связь методов определяется: а) комбинацией типовых схем взаимодействия (последовательный запуск, цикл, параллельный перебор, ветвление и пр.); б) соответствием зависимостей по данным (входные данные для каждого из реализуемых методов должны быть либо предоставлены пользователем, либо получены в ходе выполнения предыдущих методов). Простейшим случаем можно назвать последовательный запуск сервисов, позволяющий в конечном итоге получить требуемую характеристику.

5. *Оценка*. Каждый из построенных вариантов композитного приложения должен быть подвергнут автоматическому анализу, результатом которого является интегральная оценка, позволяющая: а) провести ранжирование альтернативных вариантов решения для автоматического выбора лучшего из них (по совокупности критериев) или предоставления выбора пользователю; б) отсеять заведомо варианты решения, не удовлетворяющие заданным критериям качества. При этом входными данными для построения интегральной оценки служат оценки элементов онтологии, входящих в состав подграфов, описывающих отдельные методы, используемые в решении (см. шаг 3).

В общем случае на каждом этапе могут порождаться дополнительные варианты решения. Данная процедура проводится итеративно на каждом из концептуальных уровней базы

знаний. Таким образом, строится дерево решений, структура которого определяется базовой иерархией. На заключительном этапе производится оценка каждого из листьев полученного дерева, на основании которой можно либо провести автоматический выбор оптимального (с точки зрения качества) решения, либо предоставить пользователю осмысленные и обоснованные результаты оценки с предложением осуществить самостоятельный выбор (например, установив какой-либо из критериев качества как приоритетный).

На рис. 2 приведен пример формирования дерева принятия решений, позволяющего выбрать вариант решения квантово-химической задачи одной точки методом Хартри—Фока с помощью пакета GAMESS [8] в среде Грид Национальной нанотехнологической сети (конкурирующего с пакетами ORCA и MOLPRO).

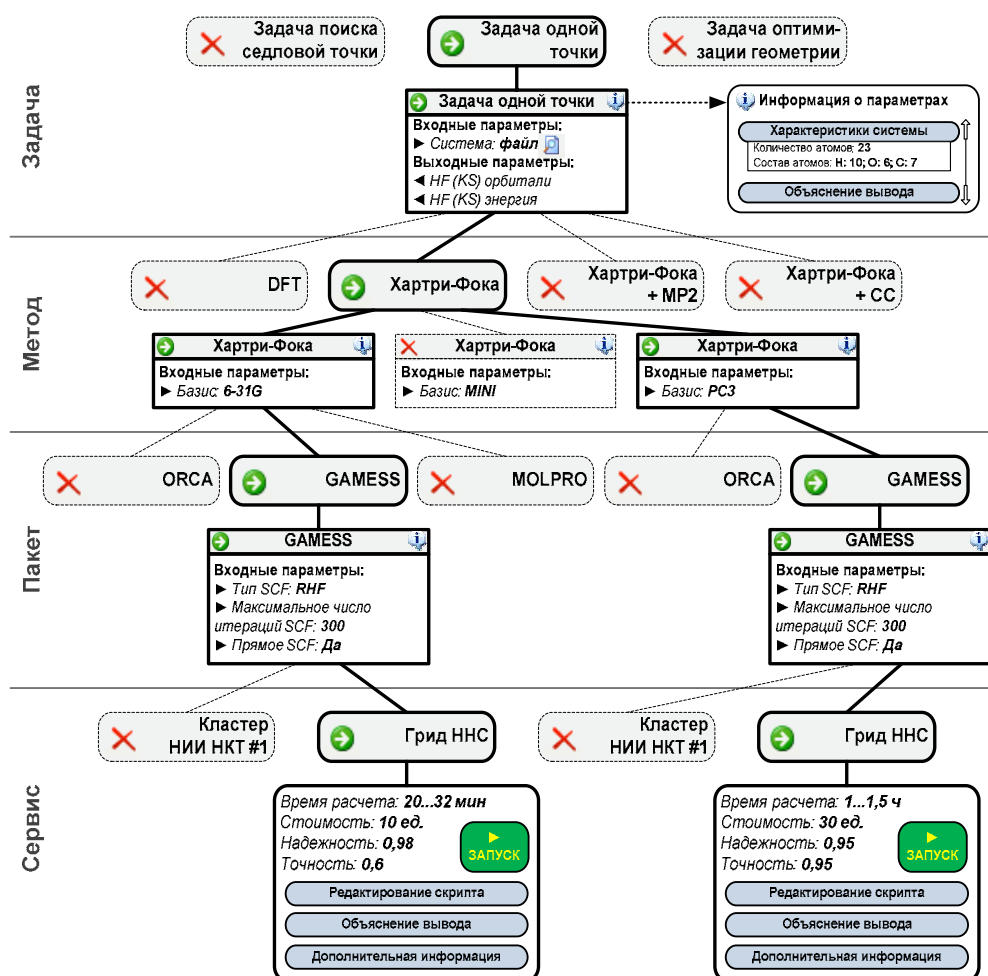


Рис. 2

На нижнем уровне рис. 2 приведены критерии качества формируемых решений.

1. *Точность предлагаемого решения*, определяемая с помощью экспертных знаний о: а) уровне точности используемых методов, б) качестве реализации этих методов в составе вычислительных пакетов.

2. *Время работы*, прогнозируемое на основании моделей производительности [9].

3. *Надежность решения*, основанная на истории запусков вычислительных сервисов, входящих в состав решения (в первую очередь, принимается во внимание доля отказов сервиса на конкретном ресурсе в распределенной среде).

Дополнительной характеристикой выбора может являться стоимость использования сервиса; на рис. 1 она определена в относительных единицах, которые могут иметь как временной, так и финансовый эквивалент.

**Программная реализация.** Описанный подход реализован в виде совокупности web-сервисов, которые могут использоваться для интеллектуальной и информационной поддержки специалистов в области квантовой химии и моделирования в нанотехнологиях, включая создание соответствующих композитных приложений, в частности, они обеспечивают:

- 1) логический вывод с целью поддержки принятия решения пользователя по выбору сервисов и формированию структуры композитных приложений;
- 2) информационно-справочную поддержку в области методов, задач и программного обеспечения в данной предметной области;
- 3) методическую поддержку, обучение специалистов рассматриваемой предметной области, осуществляемую в ходе анализа дерева принятия решений.

Программная реализация допускает два уровня хранения знаний — непосредственно в форме онтологической структуры и в форме реляционной базы данных. Онтологическая структура реализована с использованием возможностей языка OWL для описания базовых структур классов и индивидов и языка SWRL — для описания правил построения динамических связей между индивидами. База данных, хранящая фиксированный упрощенный вариант онтологической структуры и обеспечивающая более быстрый доступ к хранимым знаниям, функционирует на основе СУБД Microsoft SQL Server Compact Edition для хранения упрощенной онтологии, построенной на базе концептуальной иерархии индивидов (рис. 1). Использование базы данных позволяет за счет некоторого снижения гибкости обеспечить возможность ускорения взаимодействия с базой знаний. Автоматическое формирование базы знаний, синхронизированной по наполнению с текущей версией онтологической структуры, является одной из задач модуля обновления знаний.

Модуль логического вывода реализован в виде WCF-сервиса, предназначенного для построения дерева принятия решений на основе диалога с пользователем; для визуализации дерева принятия решений (например, в форме рис. 2) можно использовать различные инструментальные средства, в том числе в составе интерфейса прикладных комплексов и проблемно-ориентированных сред.

**Заключение.** Разработанный подход и программная реализация системы интеллектуальной поддержки позволяют строить и ранжировать различные варианты композитных приложений в ходе диалога с пользователем в различных предметных областях. Данная система применена в составе учебно-научного комплекса „Компьютерное моделирование в нанотехнологиях“ для поддержки разработки приложений комплекса HPC-NASIS II. Для этого использована база знаний в области квантово-химических расчетов и компьютерного моделирования в нанотехнологиях [10].

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации, при частичной поддержке государственного контракта № 16.647.12.2025 „Создание функционирующего в режиме удаленного доступа интерактивного учебно-методического комплекса для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе, в распределенной вычислительной среде“ и ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. The Fourth Paradigm. Data-Intensive Scientific Discovery / Ed. by T. Hey, S. Tansley, K. Tolle. Microsoft, 2009. 252 p.
2. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American Magazine. 2001. May. P. 29—37.
3. K-Wf Grid – Home [Электронный ресурс]: <<http://www.kwfgrid.net/>>.

4. Chin G., Ruby Leung L., Schuchardt K., Gracio D. New Paradigms in Problem Solving Environments for Scientific Computing // Proc. of the 7th Intern. Conf. on Intelligent User Interfaces. 2002. P. 39—46.
5. Бухановский А. В., Васильев В. Н. Современные программные комплексы компьютерного моделирования e-Science // Изв. вузов. Приборостроение. 2010. Т. 53, № 3. С. 60—64.
6. Марьин С. В., Ларченко А. В., Ковальчук С. В., Князьков К. В., Болгова Е. В., Бухановский А. В. Интеллектуальные технологии распределенных вычислений для моделирования сложных систем // Науч.-техн. вестн. СПбГУ ИТМО. 2010. Вып. 70. С. 123—124.
7. Welcome to the ORCA [Электронный ресурс]: <<http://www.thch.uni-bonn.de/tc/orca/>>.
8. Gordon Group/GAMESS Homepage [Электронный ресурс]: <<http://www.msg.ameslab.gov/games/>>.
9. Марьин С. В., Ковальчук С. В. Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 21—29.

#### Сведения об авторах

- Сергей Валерьевич Ковальчук** — канд. техн. наук; НИИ Научеомких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: kovalchuk@mail.ifmo.ru
- Владимир Григорьевич Маслов** — д-р физ.-мат. наук; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра ЦИОТ; E-mail: maslov04@bk.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 004.434;004.453

К. В. КНЯЗЬКОВ, А. В. ЛАРЧЕНКО

## ПРЕДМЕТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ ПРИЛОЖЕНИЙ В РАСПРЕДЕЛЕННЫХ СРЕДАХ

Предлагается подход к построению и исполнению композитных приложений в распределенных вычислительных средах на основе предметно-ориентированных описаний. Для его реализации разработаны предметно-ориентированные языки EasyPackage и EasyFlow. На их основе предложены технологии создания предметно-ориентированных интерфейсов, конструирования и исполнения композитных приложений, реализованные в платформе CLAVIRE.

**Ключевые слова:** поток заданий, прикладной пакет, композитное приложение, предметно-ориентированный язык, предметно-ориентированный интерфейс.

**Введение.** Непрерывно возрастающая сложность современных систем распределенных вычислений ограничивает возможности их широкого использования специалистами предметных областей. Как следствие, необходимо развивать специализированные подходы к разработке распределенных приложений с максимальным уровнем абстрагирования от специфики их технической реализации и исполнения на вычислительных ресурсах.

К настоящему времени хорошо зарекомендовал себя подход, основанный на описании сложных вычислительных процессов в форме потоков заданий (workflow, WF). Под WF понимается ориентированный граф, узлами которого, как правило, являются запуски отдельных задач, а ребрами — зависимости между ними, соответствующие операциям обмена данными и управляющими параметрами в распределенной вычислительной среде. В отличие от алго-

ритма, WF не определяет последовательность выполнения операций; он регламентирует только их состав и зависимости.

Существует ряд систем распределенных вычислений, построенных на технологиях работы с WF (их подробную классификацию можно найти в работе [1]). Некоторые системы используют графическую нотацию WF (например, YAWL [2] и Taverna [3]), другие — текстовую в виде скриптов или xml-разметки (например, Condor DAGMan [4], JSDL [5], ADL [6]). Однако возможности эффективного использования и расширения систем в целом ограничены, поскольку в настоящее время не существует унифицированных способов их описания, интерпретации и исполнения в терминах соответствующей предметной области [7].

В настоящей статье предлагается подход к построению и реализации композитных приложений в распределенных вычислительных средах на основе предметно-ориентированных описаний WF. Подход основывается на концепции iPSE [8], что подразумевает необходимость обеспечить максимальное абстрагирование пользователя от технических аспектов исполнения приложения. Пользователь должен описать в форме WF только содержательную составляющую своего приложения; при этом выбор условий выполнения (вычислительных ресурсов, необходимого количества процессов и пр.) осуществляется автоматически [9]. Поэтому принципиальной задачей является разработка способов человеко-компьютерного взаимодействия, удобных для пользователя и однозначно воспринимаемых системой. Ниже описывается ее реализация посредством двух взаимосвязанных языков: описаний прикладных пакетов (EasyPackage) и описаний структуры WF (EasyFlow).

**Язык описаний прикладных пакетов в распределенной среде.** Простейшая форма WF представляет собой описание исполнения одного вычислительного пакета в распределенной среде с загрузкой входных данных и получением выходных. Однако унифицированное описание этого действия осложнено тем, что разные вычислительные пакеты используют свою стратегию работы с данными (использование конфигурационного файла, командной строки аргументов, переменных окружения, проектов, хранящихся в структуре директорий и файлов). Ситуация осложняется требованием единообразных принципов работы с одним и тем же пакетом, установленным на ресурсах с различными операционными системами, средами управления и исполнения и пр.

В настоящей работе предлагается предметно-ориентированный язык (Domain Specific Language, DSL), позволяющий описывать пакеты в наглядной форме, понятной специалистам-предметникам, и поддающийся программной обработке. EasyPackage разработан на основе реализации языка Ruby (IronRuby) и является интерпретируемым со строгой динамической типизацией и явным приведением типов. Его базовые элементы идентичны элементам Ruby.

Описание пакета представляет собой один или несколько текстовых файлов. Оно использует следующие понятия: пакет, входной/выходной параметр, входной/выходной файл, режим запуска. *Пакет* — это исполняемое приложение, запускаемое в пакетном режиме (модель IPO — Input—Process—Output), которое принимает на входе определенный набор файлов, параметров командной строки, переменных окружения и других источников данных, а на выходе генерирует набор выходных файлов. *Параметр пакета* — это элемент данных, имеющий имя, тип и значение. Параметр может быть входным или выходным, а также может быть параметром исполнения. Тип параметра может быть одним из базовых: строка, логический тип, число с плавающей точкой, перечислимый тип, целое число, список. Режим запуска характеризуется набором используемых в нем параметров.

Структура описания пакета состоит из раздела объявления расширений, общего описания пакета, секционного описания входных и выходных данных пакета (секции *inputs* и *outputs*), описания параметров исполнения (рис. 1). Раздел объявления расширений предназначен для определения процедур, позволяющих расширить функциональные возможности

базовой библиотеки языка. Общее описание пакета включает в себя набор полей, несущих общую информацию о пакете: имя, версия, лицензия, поставщик и т.д. (строки 1—6). Раздел секционного описания содержит определение входных и выходных параметров и файлов. Параметры характеризуются следующим набором полей: тип, значение по умолчанию, процедура проверки значения параметра на корректность (например, параметр в строках 15—21). Параметры могут быть вычислимыми (строки 22—27), тогда для них указывается процедура вычисления из рабочего контекста — *evaluator* (строка 26).

```

1  name "TESTP"
2  display_as "Testp"
3  vendor "SPbSU ITMO"
4  url "http://escience.ifmo.ru"
5  license "GPLv3"
6  description "Simple package example"
7  inputs {
8    raw file {
9      name "inf"
10     filename "arg.txt"
11     place "/"
12     extractor IntegerFileExtractor.new("in")
13     assembler ObjectToSAssembler.new("in")
14   }
15   meta param {
16     name "in"
17     required
18     type int
19     validator lambda { |val, ctx| val > 0 and val < 10000 }
20     validation_error_msg "num have to be in [0; 10000]"
21   }
22   meta param {
23     name "abs_plus_3"
24     required
25     type int
26     evaluator { |ctx| ctx.in.abs + 3 }
27   }
28   cmdline lambda { |ctx| "{0} arg.txt out.txt" }
29 }
30 outputs {
31   auto file {
32     name "output_num"
33     required
34     filename "out.txt"
35     place "/"
36     extractor IntegerFileExtractor.new("out")
37   }
38   auto param {
39     name "out"
40     required
41     display_as "Output number"
42     type int
43   }
44 }
45 prepare_package

```

Рис. 1

Контекст работы представляет собой набор уже вычисленных значений параметров (*ctx*). Файловые параметры дополнительно имеют следующий набор полей (строки 8—14): имя файла, путь до файла, процедура извлечения данных из файла (*extractor*), процедура сборки файла (*assembler*). Процедура сборки файла позволяет создавать входной файл, основываясь на значениях входных параметров. На практике используются стандартные процедуры, например, сборка файла по шаблону (библиотека ERB). Процедура извлечения данных из файла, как правило, применяется для выходных файлов пакета с целью определения значений выходных параметров (строка 12). Базовый набор процедур извлечения значений из файлов и их сборки из пара-

метров можно дополнять за счет написания своих процедур в секции расширений. Последним в файле описания является раздел параметров исполнения, который позволяет при работе с пакетом не учитывать неоднородность ресурсов (различных ОС, архитектур). К параметрам данного раздела относятся: скрипт запуска пакета (точнее, процедура его сборки), командная строка, переменные окружения.

Таким образом, описание на языке EasyPackage позволяет не только задать правила обращения к конкретному пакету в распределенной вычислительной среде, но и корректно интерпретировать его входные и выходные данные (посредством процедур *extractor* и *assembler*). Это обеспечивает совместимость (по данным) пакетов различных разработчиков в составе WF.

**Язык описаний WF в распределенной среде.** Описание композитных приложений, состоящих из нескольких прикладных пакетов, требует определения не только правил работы с пакетами, но и структуры взаимодействия между ними. Специализированный язык EasyFlow позволяет упростить процедуру задания композитных приложений. Он предоставляет конечному пользователю гибкие возможности по заданию различных форм WF, в рамках которых выполняются различные прикладные пакеты, происходят генерация выходных данных, их получение, конвертация и обработка.

Характерной чертой языка является полное абстрагирование от особенностей распределенной вычислительной среды, в которой работает пользователь. Фактически EasyFlow — это высокоуровневый язык описания абстрактных workflow (AWF). Такой подход позволяет описывать саму решаемую задачу, а не способ ее исполнения на конкретной вычислительной архитектуре. На рис. 2 приведен пример описания простого AWF, представляющего собой скрипт. Тело скрипта состоит из описания вызовов прикладных пакетов — *шагов*, которые задаются с помощью директивы *step* и представляют собой узлы графа WF. Для описания каждого шага необходимо задать его имя (в примере это *Step1*, *Step2*, *Step3*), название запускаемого пакета (*EmptyPackage*, *Package1* и *Package2*) и перечень предметных параметров этого пакета, который описан в базе пакетов, рассмотренной выше.

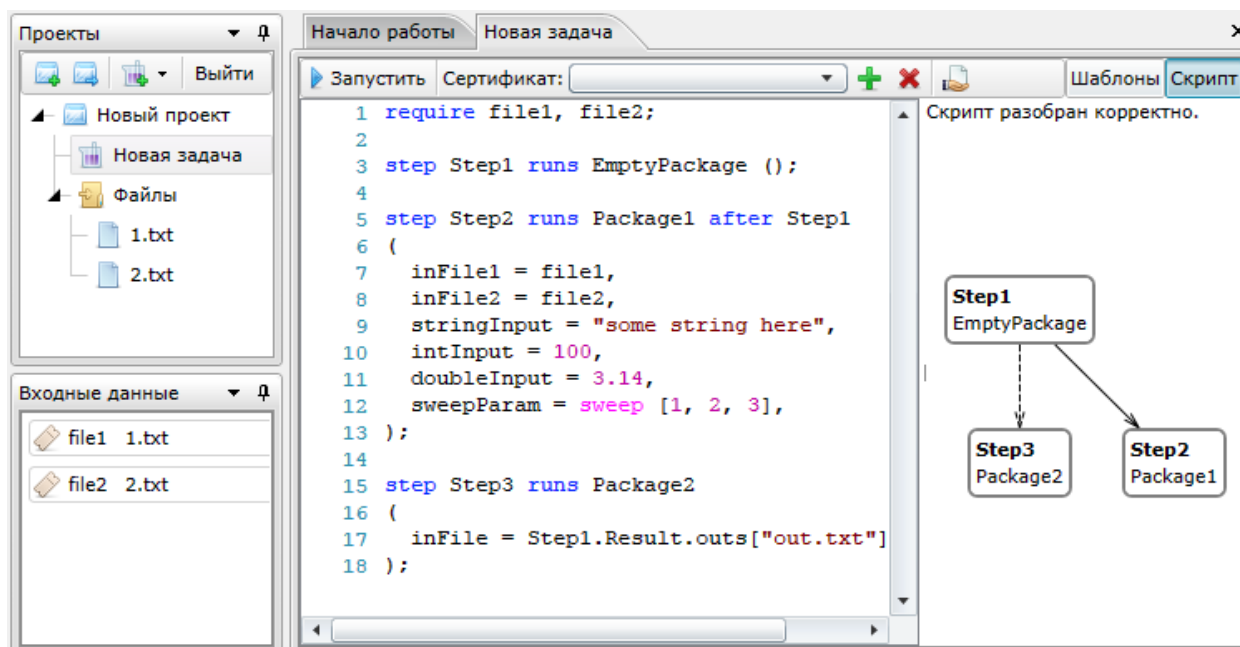


Рис. 2

Язык EasyFlow позволяет задавать параметры для следующих типов данных: целое число, строка, число с плавающей точкой, список, структура, указание на использование файла (см. описание шага *Step3*).

Большинство прикладных пакетов, помимо параметров, принимает и генерирует входные и выходные файлы, поэтому в EasyFlow предусмотрена поддержка работы с файлами. Их задание в скрипте представляет собой лишь абстрактное указание с помощью директивы *require*, что освобождает пользователя от необходимости указания абсолютных путей к файлам. В этой директиве через запятую перечислены файловые переменные, которые могут быть указаны в качестве значений параметров при описании шага (см. параметры *inFile1* и *inFile2* в описании шага *Step2*). В рамках одного скрипта директива требования файлов может появляться неограниченное число раз.

Так как WF представляет собой ориентированный граф, в EasyFlow введены механизмы определения порядка выполнения шагов, позволяющие организовать его структуру: зависимости по управлению и зависимости по данным.

*Зависимости по управлению* представляют собой явные указания на то, что один шаг должен начать свое исполнение после завершения другого. Это делается с помощью директивы *after* (см. рис. 2, шаг *Step2*).

*Зависимости по данным* представляют собой неявные указания на зависимости между шагами, которые анализируются при интерпретации скрипта EasyFlow. Они выражаются в том, что некоторые шаги могут использовать данные других шагов, что неявно влияет на последовательность их запуска. Такие зависимости могут присутствовать в описываемом WF одновременно с зависимостями по управлению, что позволяет очень гибко настраивать порядок выполнения шагов. Пример зависимостей по данным содержится в описании шага *Step3* (строка 17), где указано, что в качестве входного файла используется файл „out.txt“, полученный в результате выполнения шага *Step1*.

Еще одной полезной возможностью EasyFlow является автоматическое варьирование параметров (*parameter sweep*). Такая задача часто возникает, когда необходимо запустить один и тот же вычислительный пакет, закрепив одни и варьируя другие параметры. Для этого в язык введена директива *sweep*, которая принимает список параметров для варьирования и из одного шага создает  $N$  шагов, где  $N$  соответствует числу элементов в декартовом произведении списков варьирования для различных параметров.

Пример варьирования параметра приведен на рис. 2 (строка 12). В этом примере будут запущены три шага *Step2* с параметром *sweepParam*, равным соответственно единице, двум и трем при прочих зафиксированных параметрах.

Таким образом, WF, описанные на языке EasyFlow, полностью независимы от конкретной архитектуры вычислений и хранения данных, что позволяет пользователям распределенной среды беспрепятственно обмениваться ими и запускать их на различных вычислительных ресурсах.

**Предметно-ориентированные интерфейсы к распределенным приложениям.** При использовании представленных выше языков для описания пакетов и WF возможно ввести еще один уровень абстракции в представлении вычислительных задач для пользователей — предметно-ориентированные интерфейсы (ПОИ). Такой подход основан на разделении этапов конструирования WF и его использования, когда экспертом предметной области заранее задаются некие шаблоны применения пакетов, которые затем автоматически трансформируются в простые динамически генерируемые графические интерфейсы, доступные пользователям посредством сети Интернет. С их помощью пользователь может задать необходимые параметры расчета, запустить задачу на исполнение и получить результаты. Примером использования такого подхода может являться выполнение множества однотипных инженерных расчетов, когда от запуска к запуску требуется варьировать только ограниченный набор параметров.

При автоматическом построении ПОИ могут быть целиком использованы формальные описания пакетов на языке EasyPackage: набор параметров, их возможные значения и ограни-

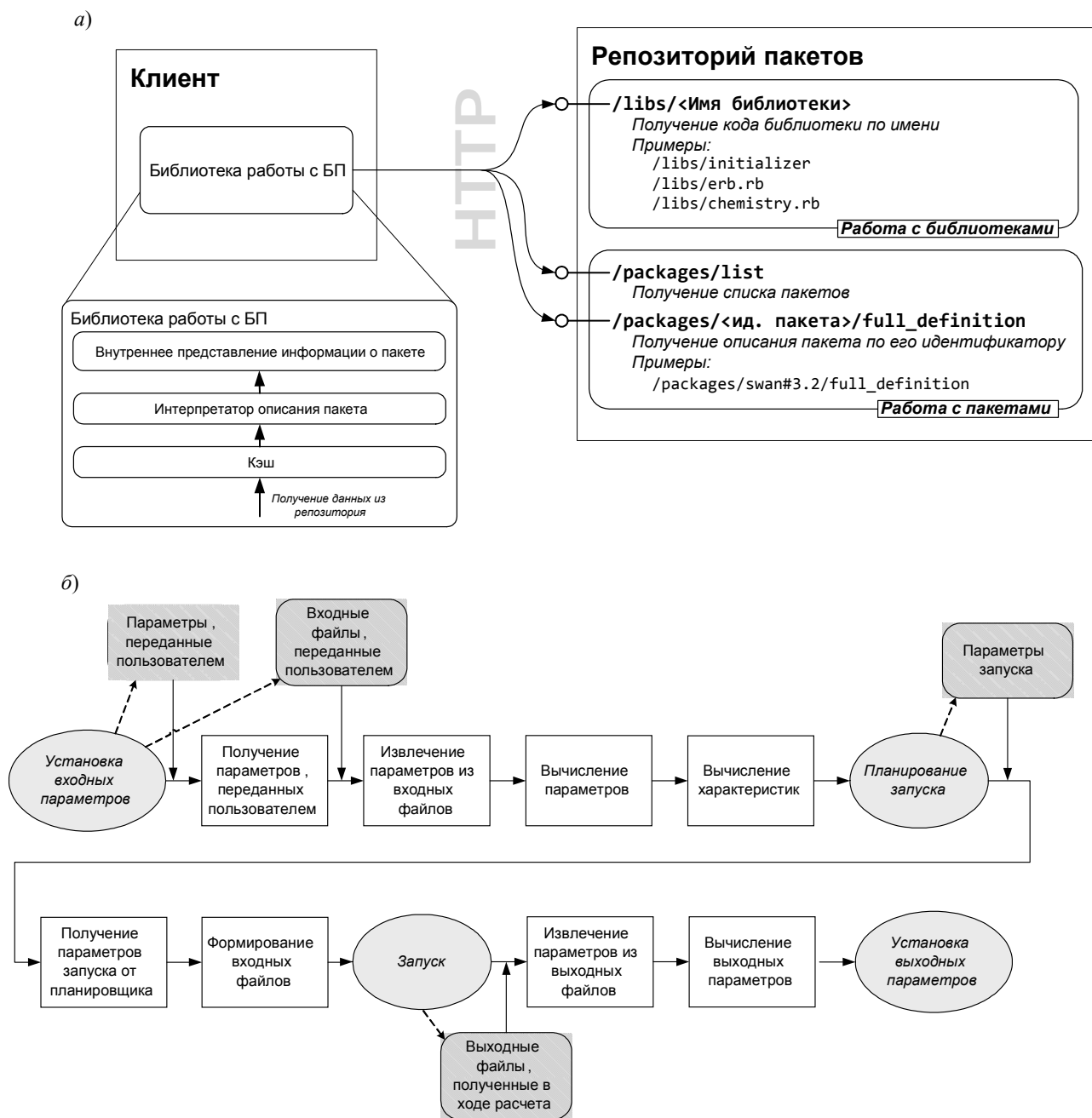


чения на них, взаимосвязь между параметрами и пр., что позволяет реализовать следующие возможности: отображение альтернатив при задании параметров, подсказки при выборе значений параметров, проверка значений и полноты состава параметров, отображение результатов вычислений. Таким образом, проблемно-ориентированный интерфейс приобретает все необходимые функциональные возможности, свойственные графическим интерфейсам самих прикладных пакетов.

Принципиальной особенностью технологии ПОИ, основанной на использовании языков EasyFlow и EasyPackage, является автоматизация процесса построения и поддержки интерфейсов, применимая для пакетов из различных предметных областей. Как следствие, такой подход наиболее предпочтителен для коллаборативных и глобальных распределенных систем (например, Грид-сред), в которых появление новых ресурсов и пакетов определяется интересами самих пользователей. При этом обеспечивается унификация не только по описаниям пакетов, но и по визуальным формам интерфейсов, что принципиально для быстрого освоения пользователями новых пакетов.

**Исполнение композитных приложений в распределенной среде.** Рассмотренные выше подходы к описанию пакетов, композитных приложений и интерфейсов работы с ними реализуются многофункциональной инструментально-технологической платформой CLAVIRE [10]. Композитное приложение в рамках платформы CLAVIRE представляется в виде скрипта описания WF на языке EasyFlow, который может быть параметризован набором входных параметров и файлов, а также параметров исполнения, т.е. один и тот же WF может быть исполнен для разного набора входных данных, а также в различных условиях исполнения. За разбор скрипта WF и за исполнение WF в целом отвечает компонент интерпретации. После получения скрипта данный компонент „разбирает“ его и преобразует во внутреннее представление (предварительно проверив его корректность). Обработка представления WF производится непрерывно, согласно событийной модели функционирования, т.е. интерпретация WF происходит в рамках цикла обработки поступающих событий. При запуске отдельной задачи происходит интерпретация параметров узла WF и формируется описание запуска задачи, после чего сформированное описание передается в очередь компонента исполнения WF. Для определения параметров исполнения данной задачи привлекается компонент планирования, который для определения очередности исполнения задач использует данные моделей производительности пакета и данные о текущей загруженности вычислительных ресурсов. Далее компонент исполнения подготавливает данные для пакета и производит запуск пакета на конкретном ресурсе, после чего обрабатывает выходные данные. Для подготовки пакета к запуску и обработки его результатов используется база пакетов, которая позволяет сопоставить абстрактные и фактические правила работы с каждым пакетом в составе WF. На рис. 3, а приведена структура компонента — базы пакетов. Он состоит из интерфейсной библиотеки и репозитория пакетов. Описание пакета в такой схеме хранится в виде файла со скриптом EasyPackage в репозитории. При этом компоненты системы для работы с данным описанием получают скрипты и сопутствующие файлы и интерпретируют их на своей стороне, используя лишь необходимую им информацию. Данный подход выгодно отличается от применения централизованного хранилища информации о пакетах, построенного на сервисно-ориентированной модели, так как позволяет легко масштабировать систему на большее количество пользователей за счет перераспределения нагрузки. На рис. 3, б представлен сценарий использования базы пакетов при запуске задачи. Необходимо отметить, что характеристиками на данной схеме считаются вычисляемые параметры, необходимые только для планирования запуска в распределенной среде. После получения и обработки результатов данные о завершении работы WF передаются обратно в интерпретатор,

там они становятся доступными пользователю (через соответствующий интерфейс — графическую оболочку CLAVIRE или ПОИ).



**Заключение.** В настоящей работе предложены подходы к описанию предметно-ориентированных пакетов и композитных приложений в форме WF. Они обеспечивают эргономичность, гибкость и расширяемость распределенных сред для компьютерного моделирования и обработки данных в рамках парадигмы eScience. Данные подходы реализованы и апробированы в составе многофункциональной инструментально-технологической платформы CLAVIRE.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

СПИСОК ЛИТЕРАТУРЫ

1. Yu J., Вууа R. A Taxonomy of Workflow Management Systems for Grid Computing // J. of Grid Computing. 2005. Vol. 3, N. 3—4. P. 171—200.
2. Van der Aalst W.M.P., ter Hofstede A.H.M. YAWL: Yet Another Workflow Language (Revised version) [Электронный ресурс]: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.3819&rep=rep1&type=pdf>>.
3. Taverna — open source and domain independent Workflow Management System [Электронный ресурс]: <<http://www.taverna.org.uk/>>.
4. DAGgMan Applications [Электронный ресурс]: <[http://www.cs.wisc.edu/condor/manual/v7.0/2\\_10DAGMan\\_Applications.html](http://www.cs.wisc.edu/condor/manual/v7.0/2_10DAGMan_Applications.html)>.
5. Job Submission Description Language (JSDL) Specification, Version 1.0 [Электронный ресурс]: <<http://www.gridforum.org/documents/GFD.56.pdf>>.
6. Guidolin M., Brady T., and Lastovetsky A. How Algorithm Definition Language (ADL) Improves the Performance of SmartGridSolve Applications // The 7th High-Performance Grid Computing Workshop. Atlanta, USA, 2010.
7. Goderis A., Sattler U., Lord P., Goble C. Seven Bottlenecks to Workflow Reuse and Repurposing // The Semantic Web – ISWC 2005- 2005. P. 323—337.
8. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные программные комплексы компьютерного моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 3. С. 5—24.
9. Бухановский А. В., Васильев В. Н. Современные программные комплексы компьютерного моделирования e-Science // Там же. 2010. Т. 53, № 3. С. 60—64.
10. Бухановский А. В., Васильев В. Н., Виноградов В. Н., Смирнов Д. Ю., Сухоруков С. А., Яппаров Т. Г. CLAVIRE: перспективная технология облачных вычислений второго поколения // Там же. 2011. Т. 54, № 10. С. 7—14.

*Сведения об авторах*

- Константин Валерьевич Князьков** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: [constantinvk@gmail.com](mailto:constantinvk@gmail.com)
- Алексей Викторович Ларченко** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник;  
E-mail: [aleksey.larchenko@gmail.com](mailto:aleksey.larchenko@gmail.com)

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

---

---

# ТЕХНОЛОГИИ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА РАСПРЕДЕЛЕННЫХ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД

---

---

УДК 681.3.069, 681.324

Д. М. Спельников, А. А. Гуськов, В. Г. Маслов, А. В. Бухановский

## УЧЕБНО-НАУЧНЫЙ КОМПЛЕКС „КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В НАНОТЕХНОЛОГИЯХ“ НА ОСНОВЕ ГРИД-СРЕДЫ

Разработан учебно-научный комплекс для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе в режиме удаленного доступа к ресурсам распределенной вычислительной среды Грид Национальной нанотехнологической сети.

*Ключевые слова:* нанотехнологии, сервис, распределенная среда, удаленный доступ, портал, наноразмерные системы, учебно-методические материалы.

**Введение.** В настоящее время для изучения сложных систем активно используется так называемая „третья парадигма научного познания“ — изучение на основе компьютерного моделирования [1]. По сравнению с двумя традиционными парадигмами (теорией и экспериментом), она предоставляет широкие возможности для исследования явлений, полномасштабное наблюдение которых ограничено, затруднено или в принципе невозможно. К таким явлениям, в частности, относятся нанотехнологические процессы. Их специфика состоит в том, что в рамках одного процесса необходимо связать характеристики микроуровня (свойства отдельных атомно-молекулярных структур) и макроуровня (свойства конкретных устройств, материалов и явлений на их основе). Как следствие, полномасштабное экспериментальное изучение таких процессов на всех уровнях сопряжено с существенными трудностями. Потому компьютерное моделирование является необходимым инструментом не только изучения и расчета свойств нанотехнологических процессов, но и предметно-ориентированного обучения, связанного с различными направлениями развития nanoиндустрии. Данный факт объясняется тем, что в процессе компьютерного моделирования обучаемый имеет возможность ознакомиться с комплексными свойствами нанотехнологических процессов, прослеживая ключевую специфику нанотехнологий — формирование макрообъектов с заданными свойствами на основе манипулирования характеристик микрообъектов (атомно-молекулярных систем).

Эклектичность способов формирования модельно-методической базы в области нанотехнологий, вызванная их многомасштабностью, породила специфический класс программного инструментария компьютерного моделирования — проблемно-ориентированные оболочки, обеспечивающие доступ к различным прикладным пакетам (и их комбинациям) для решения задач по различным направлениям развития nanoиндустрии. К таким инструментам, в частности, относится комплекс Accelrys [2], портал информационной и инструментальной

поддержки NANOhub [3], а также отечественная разработка Nanomodel.ru [4]. Несмотря на достаточно мощную технологическую основу для научных расчетов, перечисленные комплексы не в полной мере подходят для модельного изучения различных нанотехнологических процессов, поскольку не обеспечивают возможности наблюдения моделируемого процесса на всех этапах или уровнях иерархии.

В настоящей работе рассматривается специализированный учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“. Его основу составляет аппаратно-программный комплекс (АПК) HPC-NASIS II, который реализует концепцию iPSE [5] и предоставляет доступ к вычислительным сервисам и композитным приложениям в рамках бизнес-модели SaaS (Software as a Service, программное обеспечение как услуга). Вычисления выполняются на ресурсах Грид-среды Национальной нанотехнологической сети [6].

**Методы и технологии компьютерного моделирования.** При компьютерном моделировании в нанотехнологиях используется многомасштабный подход. На нижнем уровне иерархии находится уровень моделирования свойств самих атомно-молекулярных систем на основе квантово-механических *ab initio* расчетов (из первых принципов). На всех последующих уровнях применяются результаты таких расчетов в качестве параметров системы вложенных моделей, определяющих свойства мезомасштабных и макромасштабных явлений.

Основным функциональным назначением учебно-научного комплекса является организация доступа и возможности использования распределенных вычислительных сервисов при компьютерном моделировании в нанотехнологиях. Каждый сервис представляет собой прикладной пакет, развернутый на ресурсах среды ГридHPC и доступный в режиме удаленного доступа. Комплекс включает в себя три группы прикладных сервисов:

- 1) для квантово-механического моделирования и *ab initio* расчетов атомно-молекулярных систем,
- 2) для расчетов характеристик различных процессов нанопластики и наноплазмоники,
- 3) для моделирования с помощью методов молекулярной динамики.

Первая группа сервисов позволяет выполнять высокоточные *ab initio* расчеты из свойств малых атомно-молекулярных систем, включая свойства основного и возбужденных состояний, и моделирование мезосистем (до 10 000 атомов) посредством прикладных пакетов GAMESS, ORCA и PRIRODA (неэмпирические методы), SEMP (полуэмпирический метод ZINDO/S-CI) и DPIMC (квантовый метод Монте-Карло).

Вторая группа сервисов ориентирована на решение специфических задач расчета линейных и нелинейных свойств ансамблей наночастиц, погруженных в матрицу или нанесенных на поверхность, выполняемого с помощью пакетов UpConversion, QDLaser, Plasmon и JAggregate [7]. Эти сервисы позволяют моделировать оптические и фотоэлектрические свойства ансамблей наночастиц на подложках и в объеме диэлектриков, степень усиления или подавления комбинационного рассеяния, флуоресценции, переноса возбуждения, нелинейно-оптические свойства ориентированных молекулярных J-агрегатов, энергетические, спектральные и угловые характеристики излучения твердотельных и полупроводниковых лазеров с наноструктурами.

Третья группа сервисов ориентирована на моделирование процессов наногидродинамики и формирования наночастиц, нанопластин и нанотрубок. Она включает в себя пакеты NanoFlow, NAEN и NAMD [8].

Учебно-научный комплекс предоставляет пользователю возможность не только выбора вариантов применения различных прикладных пакетов, но их комбинирования в составе так называемых композитных приложений — набора сервисов, скоординированно выполняемых в составе распределенной среды для решения общей прикладной задачи.

**Вычислительная платформа и программный инструментарий.** Ядром учебно-научного комплекса является аппаратно-программный комплекс HPC-NASIS II, в котором

развиты идеи, заложенные в высокопроизводительном программном комплексе для квантово-механических расчетов и моделирования наноразмерных атомно-молекулярных систем и комплексов HPC-NASIS [9]. АПК выгодно отличается от HPC-NASIS следующим:

- ориентация на технологии удаленного взаимодействия с пользователем через веб-браузер;
- представление пользовательских приложений в форме потоков заданий (workflow, WF) на предметно-ориентированном языке EasyFlow;
- применение новой онтологической базы знаний в области моделирования наноразмерных атомно-молекулярных систем;
- возможность эффективно совокупно использовать ресурсы выделенных суперкомпьютеров и среды Грид;
- поддержка когнитивной визуализации результатов расчетов в режиме удаленного доступа (включая режимы поддержки стереоизображений);
- обеспечение безопасности многопользовательского применения, поддержка операций квотирования, биллинга и тарификации;
- обеспечение коллективной работы над проектами пользователей.

Комплекс HPC-NASIS II разработан на основе многофункциональной инструментально-технологической платформы CLAVIRE (Cloud Applications Virtual Environment), его принципиальная архитектура приведена на рис. 1.

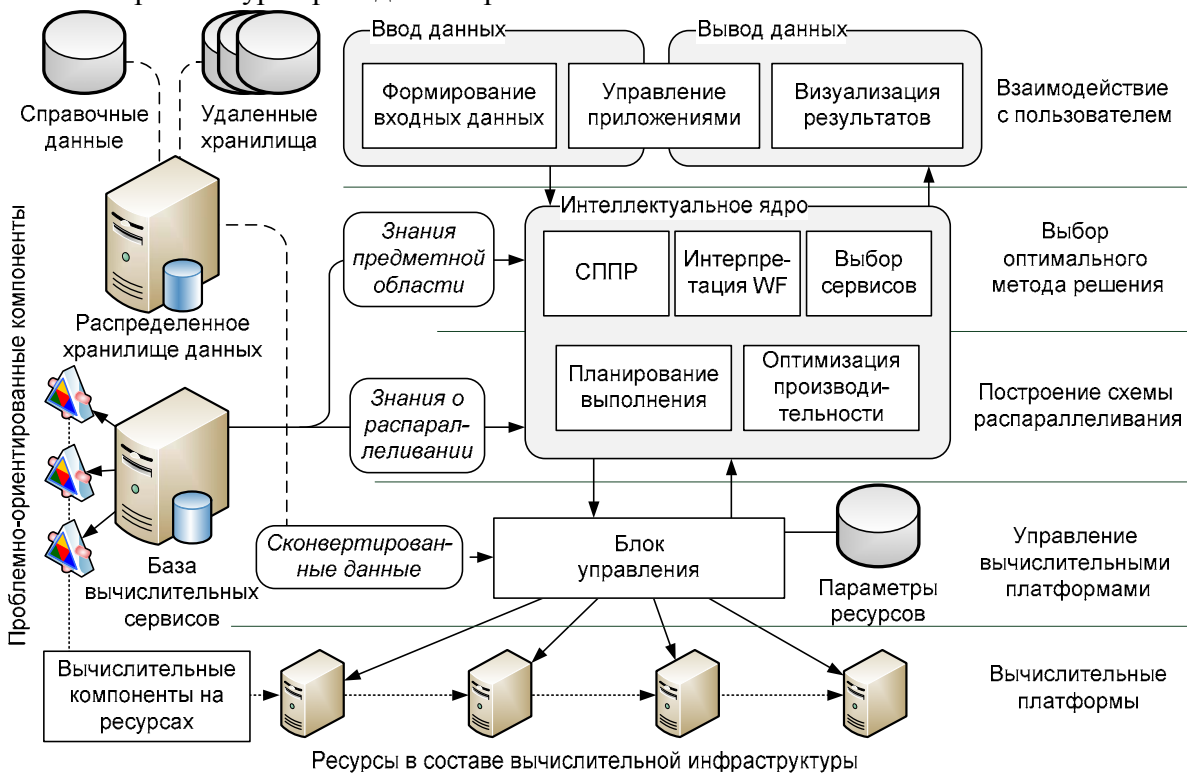


Рис. 1

На уровне взаимодействия с пользователем выполняются процессы разработки и отладки композитных приложений, подготовки входных данных, а также интерпретации и визуализации результатов расчетов. Уровень выбора оптимального метода решения включает в себя процессы поддержки принятия решений пользователя в ходе конструирования композитного приложения, выбора определенных прикладных сервисов, а также интерпретации пользовательского WF в исполнимую форму. На уровне построения схемы распараллеливания реализуются задачи планирования исполнения блоков композитного приложения в распределенной вычислительной среде. Уровень управления вычислительными платформами обеспечивает запуск отдельных задач (в составе композитного приложения) на исполнение, а также мониторинг их

работы, обмен входными и выходными данными. Уровень вычислительных платформ включает низкоуровневую инфраструктуру среды распределенных вычислений (например, ГридННС).

Комплекс HPC-NASIS II предоставляет пользователю четыре режима организации вычислительного процесса с использованием прикладных сервисов в распределенной среде:

1) режим интеллектуальной поддержки, в рамках которого пользователь формулирует постановку задачи на языке предметной области на основе опроса экспертной системы. В результате пользователю предлагается наиболее отвечающий его целям вариант использования прикладного пакета, на основе которого автоматически генерируется сценарий запуска;

2) режим ручного создания композитных приложений: подразумевает описание композитного приложения в окне редактирования на языке EasyFlow, с непосредственным определением режимов вызова прикладных пакетов, необходимых пользователю;

3) режим использования шаблонов, соответствующих типовому применению вызовов отдельных прикладных пакетов. В этом случае пользователь отвечает за подготовку входных данных; выбор вычислительного ресурса и запуск пакета выполняются комплексом автоматически;

4) режим использования сценариев позволяет применять заранее подготовленные описания композитных приложений с ассоциированными входными данными в форме проектов HPC-NASIS II. Сценарии могут содержаться как в хранилище данных пользователя HPC-NASIS II, так и во внешних хранилищах в составе Национальной нанотехнологической сети, с возможностью загрузки в комплекс. При этом пользователь (по необходимости) может изменять только отдельные значения параметров запуска.

Все варианты использования комплекса позволяют обеспечить доступ к вычислительным сервисам и композитным приложениям на их основе в рамках бизнес-модели SaaS, поэтому пользователь может не устанавливать специализированное программное обеспечение на свой компьютер, а ограничиться использованием стандартного web-браузера.

**Учебно-методические материалы.** В состав учебно-научного комплекса входят гипертекстовые материалы, необходимые для освоения приемов работы с комплексом HPC-NASIS II, а также для обучения по таким направлениям развития nanoиндустрии, как нанобиотехнологии, nanoинженерия и nanoэлектроника. В состав учебно-методических материалов входят методики использования комплекса, учебные модули, лабораторные работы, методические указания для исследовательских работ и электронные тьюторы.

Методики использования комплекса HPC-NASIS II (технологические инструкции) описывают возможности применения комплекса для решения различных предметных задач в области нанотехнологий, порядок работы с сервисами и композитными приложениями в режиме удаленного доступа, а также особенности использования комплекса в режиме симулятора нанотехнологических процессов.

Учебные модули охватывают отдельные разделы знаний по направлениям „nanобиотехнологии“, „nanoинженерия“, „nanoэлектроника“. В них представлены теоретические основы расчетных методов и моделей, реализуемых комплексом HPC-NASIS II, а также предметные направления, освоение которых требует интенсивного применения средств компьютерного моделирования в нанотехнологиях:

- фотоника наноструктур и наноструктурных материалов;
- квантово-химические методы;
- моделирование оптических и электрофизических свойств наноструктур методами квантовой химии;
- оптические солитоны и солитонные линии связи;
- компьютерное моделирование лазеров и лазерно-оптических систем;
- течения жидкости в наноканалах.

Лабораторные работы предназначены для исследования различных явлений и свойств атомно-молекулярных систем, процессов, материалов и устройств на их основе с целью прак-

тического закрепления знаний и навыков, полученных в ходе освоения материалов учебных модулей. Все лабораторные работы выполняются на данном портале в режиме удаленного доступа. Методические указания для исследовательских работ способствуют стимулированию научной активности студентов и аспирантов в области нанотехнологий. Они ориентированы на поддержку самостоятельного решения практических задач с использованием комплекса НРС-NASIS II. Электронные тьюторы представляют собой анимированные демонстрации работы с комплексом НРС-NASIS II. Они воспроизводят как технические аспекты работы в режиме удаленного доступа, так и методические аспекты применения комплекса для решения конкретных прикладных задач.

Программная документация на комплекс НРС-NASIS II и прикладные пакеты в его составе доступна в составе учебно-методических материалов. Ее целесообразно использовать при самостоятельном выполнении исследовательских работ с применением новых пакетов и разработкой собственных композитных приложений.

**Средства взаимодействия с пользователем в режиме удаленного доступа.** Основным средством работы с учебно-научным комплексом является web-портал, который реализует интерфейс, обеспечивающий удаленный доступ пользователей к учебно-методическим материалам и сервисам НРС-NASIS II. Web-портал построен с применением системы управления контентом (CMS), которая разработана с использованием архитектуры MVC (Model—View—Controller, Модель—Представление—Контроллер). Структура портала представлена несколькими группами, каждая из которых содержит набор специализированных сервисов (хранения, мониторинга, вычислительные, персональные). Блоки „Модель“ и „Контроллер“ компонента реализованы с помощью языка программирования PHP 5.2, блок „Представление“ — с помощью технологий Javascript, html DOM, Ajax (библиотека jquery), Silverlight, Flash. Связь с web-сервисами осуществляется на основе стандарта SOAP. Для хранения данных может использоваться сервер базы данных MySQL или PostgreSQL. Для связи приложений с сервером базы данных используется язык запросов SQL. Хостинг сервисов осуществляется с помощью web-сервера Apache на UNIX-платформе.

Процесс работы с учебно-научным комплексом организован в рамках модели профессионального интернет-сообщества пользователей НРС-NASIS II. Сообщество организовано на основе виртуального рабочего пространства в рамках специализированной социальной сети. Основным источником мотивации для членов сообщества является возможность доступа к вычислительным сервисам, композитным приложениям и учебно-методическим материалам. Членом сообщества автоматически становится каждый зарегистрированный пользователь НРС-NASIS II. Как следствие, он приобретает следующие возможности:

- предоставлять информацию о себе другим членам профессионального интернет-сообщества;
- обмениваться профессиональными материалами с другими членами профессионального интернет-сообщества в рамках виртуальных групп социальной сети;
- предоставлять свои проекты НРС-NASIS II для общего доступа (для коллаборативной работы или независимого использования).

Учебно-научный комплекс обеспечивает возможность коллаборативной работы географически разделенных пользователей над одним проектом. Для обеспечения этой функции задействованы возможности симулятора нанотехнологических процессов (СНП) — репозитория композитных приложений.

На рис. 2 приведена схема организации коллаборативной работы пользователей над одним проектом с использованием СНП. В ней участвуют два пользователя. Пользователь „А“ создает, настраивает и исследует собственное композитное приложение, убеждаясь в его работоспособности. В том случае, если пользователь желает сделать проект доступным, выполняется отчуждение проекта с записью в репозиторий СНП. При этом пользователю „А“



предлагается создать краткую аннотацию приложения и сформировать графический образ, ассоциируемый с ним (например, в виде WF). В том случае, если проект сохранен в репозитории СНП, его описание становится доступным другим зарегистрированным пользователям HPC-NASIS II через web-портал удаленного доступа, поэтому пользователь „Б“ по описанию может выбрать приложение из каталога. При этом автоматически запускается сеанс работы с комплексом HPC-NASIS II и загружается соответствующий (готовый) проект от имени пользователя „Б“.

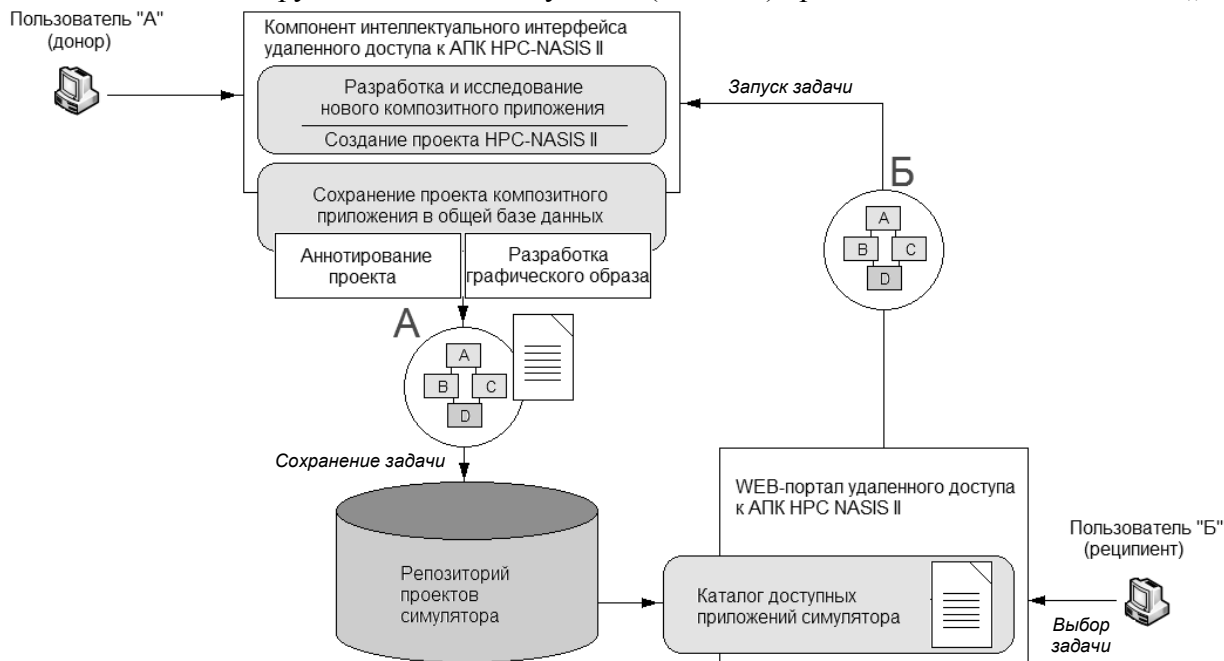


Рис. 2

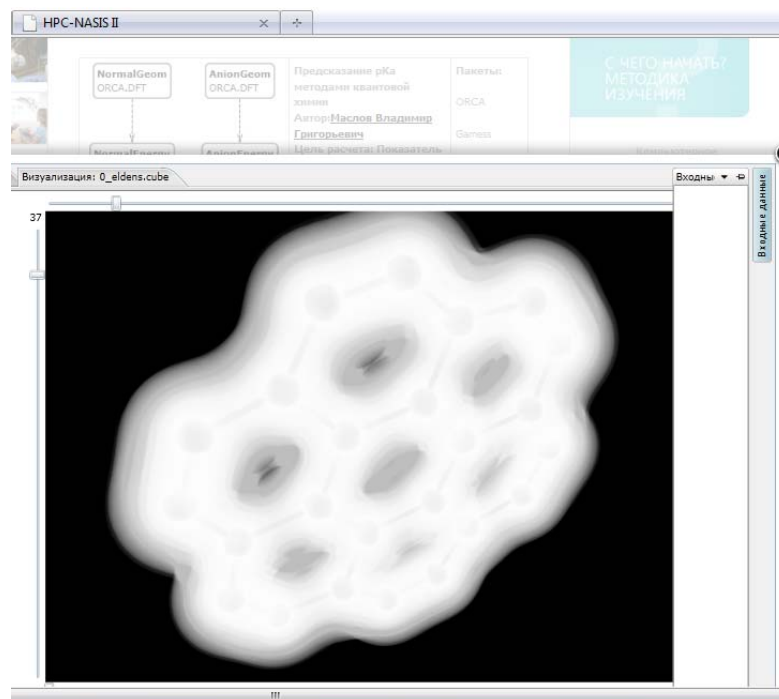


Рис. 3

Учебно-научный комплекс предоставляет пользователю мощный инструмент для предметно-ориентированной визуализации, позволяя формировать изображение на web-сервере с последующей передачей пользователю растрового графического файла. Визуализация выполняется как с помощью специализированных приложений, ориентированных на атомно-

молекулярные системы, так и посредством пакетов обработки данных общего назначения (например, SciLab [10]). На рис. 3 приведены примеры визуализации результатов моделирования.

**Заключение.** Учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“ доступен членам Национальной нанотехнологической сети (см. <<http://hpc-nasis.ifmo.ru>>).

Работа выполнена в рамках государственного контракта № 16.647.12.2025 „Создание функционирующего в режиме удаленного доступа интерактивного учебно-методического комплекса для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе, в распределенной вычислительной среде“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Naumov L. Modelling with Cellular Automata: Problem Solving Environments and Multidimensional Applications. Amsterdam, 2011. 136 p.
2. Accelrys [Электронный ресурс]: <<http://accelrys.com>>.
3. Nanomodel [Электронный ресурс]: <<http://nanomodel.ru>>.
4. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.
5. ГридННС [Электронный ресурс]: <<http://www.ngrid.ru/trac>>.
6. Розанов Н. Н. Диссипативные оптические солитоны. М.: Физматлит, 2011. 536 с.
7. NAMD [Электронный ресурс]: <<http://www.ks.uiuc.edu/Research/namd>>.
8. Васильев В. Н. и др. Высокопроизводительный программный комплекс моделирования атомно-молекулярных наноразмерных систем // Науч.-техн. вестн. СПбГУ ИТМО. 2008. Вып. 54. С. 3—12.
9. SciLab [Электронный ресурс]: <<http://www.scilab.org>>.

#### Сведения об авторах

- Дмитрий Михайлович Спельников** — НИИ Научно-технологических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: pilule@ya.ru
- Александр Александрович Гуськов** — НИИ Научно-технологических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: aleksander.guskov@gmail.com
- Владимир Григорьевич Маслов** — д-р физ.-мат. наук; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра ЦИОТ; E-mail: maslov04@bk.ru
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технологических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

Т. Н. ЧУРОВ, Д. А. НАСОНОВ, Б. Я. ШТЕЙНБЕРГ, А. В. БУХАНОВСКИЙ

## ОСОБЕННОСТИ ВЕРОЯТНОСТНОГО АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ И НАДЕЖНОСТИ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ СРЕД ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Предложена вероятностная модель функционирования среды облачных вычислений; рассмотрены способы оценки производительности и надежности предоставления сервисов с ее использованием.

**Ключевые слова:** облачные вычисления, сервисы, распределенная среда, производительность, время наработки на отказ.

**Введение.** Концепция облачных вычислений предполагает, что пользователю предоставляются вычислительные ресурсы и сервисы с гарантированным уровнем качества, априори обеспеченным провайдером. Мера качества определяется бизнес-моделью предоставления сервиса. В частности, в рамках модели SaaS (Software as a Service) она связана как с надежностью функционирования, так и с производительностью (временем предоставления сервиса). Особенность определения этих характеристик состоит в том, что среда облачных вычислений представляет собой распределенный программно-аппаратный комплекс, включающий в себя управляющую инфраструктуру среды облачных вычислений, вычислительные ресурсы, а также пакеты прикладных программ, на основе которых формируются сервисы. По сравнению с нормативами автоматизированных систем для традиционных видов [1], оценка надежности сред облачных вычислений требует учета таких факторов, как распределенность, иерархичность и многовариантность использования. Как следствие, применение аппарата оценки показателей надежности программного обеспечения [2—4] требует определенной модификации в соответствии с принципами работы среды облачных вычислений как системы массового обслуживания.

В настоящей работе рассматриваются вероятностная модель функционирования среды проблемно-ориентированных облачных вычислений, связанные с ней показатели производительности прикладных сервисов и среды в целом, а также интегральный подход к оценке временных характеристик надежности.

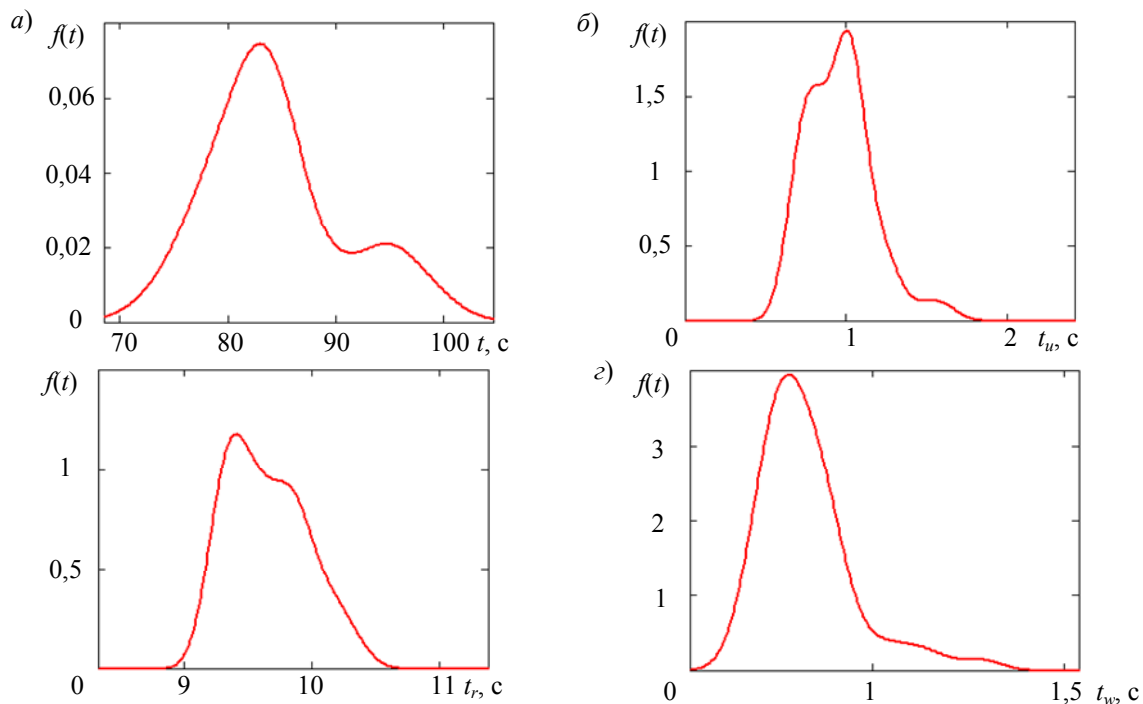
**Вероятностная модель среды облачных вычислений.** Основой для функционирования среды является облако вычислительных ресурсов  $\Omega = \{\Omega_i\}_{i=1}^n$ ; где  $\Omega_i$  — физические ресурсы — отдельные вычислительные системы (серверы приложений, хранилища данных и пр.). Каждый вычислительный ресурс описывается кортежем, определяющим основные его технические характеристики:

$$\Omega_i = \langle B, (q, p, \Pi)_B, (m, \mu), (d, D), \mathfrak{F}_t \rangle_i, \quad (1)$$

здесь  $B$  — элемент из множества классов вычислительных архитектур (SMP, MPP, GPGPU, FPGA, ...), определяющих свойства, отражающие целесообразность запуска тех или иных пакетов. Производительность каждого ресурса характеризуется триплетом  $(q, p, \Pi)_B$ , где  $q$  — количество ядер,  $p$  — количество процессоров,  $\Pi$  — количество узлов; марка  $B$  определяет их иерархию в конкретной архитектуре. Оперативная память описывается набором  $(m, \mu)$  объемных характеристик памяти на узле  $m$  и в целом в системе  $\mu$  (могут быть пересчитаны через  $p, q, \Pi$ ). Дисковая память характеризуется аналогичными по смыслу величинами  $(d, D)$ .

Многомерное вероятностное распределение  $\mathfrak{Z}_l(T)$ , где  $T = \{t_1, \dots, t_z\}$ , описывает составляющие случайной изменчивости временных характеристик накладных расходов работы с данной вычислительной системой как с физическим ресурсом  $\Omega_i$  в распределенной среде. В простейшем случае они могут включать в себя время  $t_w(\Xi)$  на передачу данных  $\Xi$  на физический ресурс  $\Omega_i$  и накладные расходы  $t_r(\Omega)$  системы управления вычислительным ресурсом  $\Omega_i$ . Такая высокоуровневая детализация позволяет использовать модель (1) как для „настоящих“ физических ресурсов (серверов и суперкомпьютеров), так и для виртуальных систем (например, Грид-инфраструктур), доступных в „облаке“. В случае виртуальных систем значение  $t_r$  может быть представлено в форме суммы нескольких составляющих накладных расходов, связанных с работой отдельных управляющих сервисов Грид.

В качестве иллюстрации на рисунке приведены ядерные оценки маргинальных распределений времени  $f(t)$  накладных расходов, возникающих при выполнении заданий в среде облачных вычислений HPC-NASIS II [5], использующей ресурсы Грид Национальной нанотехнологической сети [6]. На рисунке, *a* показана плотность распределения времени выполнения задания в Грид; видно, что при среднем времени выполнения около 85 с разброс может составлять от 70 до 110 с, что характеризуется латентностью самой Грид-среды. Из этого следует, что составляющая  $t_r(\Omega)$  может достигать 40 с. На рисунке, *б* представлена плотность распределения нахождения задания в очереди, *в* — суммарные издержки среды облачных вычислений на основе платформы CLAVIRE (CLOUD Applications VIRTUAL Environment). Видно, что их разброс невелик; в среднем они в четыре раза меньше, чем издержки, связанные с Грид. На рисунке, *г* приведена плотность распределения времени передачи данных; для рассмотренного примера оно незначительно. В целом из рисунка видно, что латентность Грид является существенной по сравнению с остальными накладными расходами среды облачных вычислений, что требует ее отдельного учета в распределении  $\mathfrak{Z}_l(T)$ .



При предоставлении пользователю сервиса в процессе его запуска в облаке происходит выделение (резервирование) виртуального ресурса  $\Omega_i^*$  на физическом ресурсе  $\Omega_i$ , при этом оставшийся ресурс  $\tilde{\Omega}_i = \Omega_i \setminus \Omega_i^*$  может быть задействован для другой задачи. Эта операция

позволяет рассматривать  $\tilde{\Omega}_i$  как физический ресурс, однако — с „ухудшенными“ (за счет выделения под заданную задачу) характеристиками по сравнению с  $\Omega_i$ . Следует отметить, что изменение касается не только числовых характеристик вычислителей (количество процессоров, узлов, ядер) и памяти, но и свойств распределения  $\mathfrak{Z}_i(T)$ : при увеличении количества независимых приложений объем накладных расходов может возрастать (хотя бы в силу загрузки управляющего узла).

**Вероятностная модель производительности сервисов в рамках SaaS.** Модель (1) описывает ресурсную составляющую облачной среды в рамках модели IaaS (Infrastructure as a Service). Для описания процесса запуска на ней прикладных пакетов следует ввести модель сервиса

$$\Sigma_j = \langle \Xi_j, S_j(\Xi_j, P, \Omega_i), \kappa_{ij}, T_{0j} \rangle, \quad (2)$$

который ассоциирован с каждым из физических ресурсов  $\Omega_i$  в (1). Здесь  $\Xi_j$  — характеристики (объем, структура) входных и выходных данных сервиса,  $S_j(\Xi_j, P, \Omega_i)$  — формализация „внутреннего“ параллельного ускорения (например, в виде параметрической модели [7]) на физической системе  $\Omega_i$  в зависимости от количества занимаемых вычислительных ресурсов  $P = (q, p, \Pi)$ ,  $\kappa_{ij}$  — относительная пиковая производительность системы  $\Omega_i$  для заданного сервиса  $\Sigma_j$ , а  $T_{0j}$  — абсолютное время исполнения сервиса на „эталонном“ (т.е. выбранном для сопоставления) ресурсе облачной среды в последовательном режиме. Следует отметить, что сервисы могут исполняться на тех системах, где они установлены, или развертываться на новых целевых системах (если они совместимы). На каждом ресурсе могут быть доступны все сервисы (или только несколько).

Время исполнения конкретного сервиса в рамках модели облачной среды (1)—(2) может быть представлено выражением:

$$T_{ij} = t_{ij} + \kappa_{ij} T_{0j} / S(\Xi_j, P, \Omega_i). \quad (3)$$

Здесь индексы  $i, j$  обозначают целевой ресурс и тип сервиса. Величина  $t$  в (3) является характеристикой накладных расходов на запуск сервиса, задаваемой распределением  $\mathfrak{Z}_i(T)$ , и может быть представлена следующим образом:

$$t = t_u(\Omega, \Xi) + t_r(\Omega) + t_w(\Xi), \quad (4)$$

где  $t_u$  — накладные расходы на подготовку задания к исполнению в распределенной среде  $\Omega$  с данными  $\Xi$ ,  $t_r(\Omega)$  — внутренние накладные расходы распределенной среды,  $t_w(\Xi)$  — совокупное время подготовки и передачи данных в распределенной среде.

При запуске каждого сервиса создается *сессия*  $\Sigma_{ij}^{(k)} = \Sigma_j(\Xi_j^{(k)}, S_j(\Xi_j^{(k)}, P^*, \Omega_i^*), \kappa_{ij})$ . Ее окончание включает в себя прикладной пакет  $j$  с заданными характеристиками исполнения, функционирующий на выделенном виртуальном ресурсе  $\Omega_i^*$ . Здесь  $P^*$  в общем случае соответствует оптимальному режиму исполнения сервиса (для максимума функции  $S_j(\Xi_{ij}, P^*, \Omega_i)$  при ограничении на предельное число вычислителей на ресурсе  $\tilde{\Omega}_i$ ).

Таким образом, временной срез „общего“ процесса функционирования облачной среды в рамках модели SaaS представляется в виде

$$\Sigma = \left\langle \left\{ \Sigma_{ij}^{(k)}(\Omega_i^*) \right\}_{\substack{i=1, N \\ j \in W}}^{k=1, M} \middle| \bigcup_l \Omega_l \setminus \bigcup_i \Omega_i^* \right\rangle. \quad (5)$$

Здесь  $k$  — индекс сквозной нумерации сессий пользователей,  $i$  — индекс сквозной нумерации виртуальных ресурсов, выделяемых под предоставление им сервисов, а  $j \in W$  — идентификаторы конкретных содержательных сервисов. При этом допускается одновременное наличие нескольких сессий одних и тех же пользователей с одинаковыми сервисами, различающимися, например, только параметрами запуска  $\Xi_{ij}^{(k)}$ .

Множество  $\bar{\Omega} = \bigcup_l \Omega_l \setminus \bigcup_i \Omega_i^*$  определяет доступное облако ресурсов, которое может быть представлено в виде

$$\bar{\Omega} = \bigcup_l \tilde{\Omega}_l = \bigcup_l \left( \Omega_l \setminus \left\{ \Omega_{al}^*, \Omega_{bl}^*, \dots \right\} \right),$$

т.е. с учетом того, что виртуальный ресурс выделяется на конкретной физической целевой системе. Таким образом, облако ресурсов может быть описано набором свободных физических систем  $\bar{\Omega}$  с меньшим количеством вычислителей и иными характеристиками доступной оперативной и дисковой памяти, по сравнению с исходной средой  $\Omega$ .

Модель (3)—(4) определяет размерные характеристики производительности сервиса. Однако для анализа эффективности применения среды облачных вычислений в рамках модели SaaS, по сравнению с традиционными моделями использования программного обеспечения, целесообразно ввести ряд безразмерных характеристик:

— технологическая эффективность  $\mathcal{G}_1$  использования среды облачных вычислений в сопоставлении с выделенной вычислительной системы с предустановленным пакетом (учитываются накладные расходы на управление выделенной системой);

— пользовательская эффективность  $\mathcal{G}_2$  использования среды облачных вычислений в сопоставлении с „чистым“ временем исполнения пакета на идеальной вычислительной системе (накладные расходы отсутствуют);

— абсолютная эффективность  $\mathcal{G}_3$  технических решений среды облачных вычислений в сопоставлении с техническими решениями низкоуровневой вычислительной инфраструктуры (кластеры, Грид);

— относительная эффективность  $\mathcal{G}_4$  технических решений среды облачных вычислений в сопоставлении с техническими решениями низкоуровневой вычислительной инфраструктуры для решения конкретной прикладной задачи, требующей передачи данных определенного объема.

Поскольку выражение (3) определяет случайную функцию в соответствии с  $\mathfrak{Z}_l(T)$ , определяемые через него безразмерные характеристики  $\mathcal{G}_1 - \mathcal{G}_4$  являются случайными величинами. В табл. 1 приведены вероятностные характеристики эффективности среды облачных вычислений НРС-NASIS II для примера, соответствующего рисунку.

Таблица 1

Характеристика	Моменты, с		Квантиль 10 %-ной обеспеченности, с		
	$m_x$	$\sigma_x$	$q_{10\%} - I_{95\%}$	$q_{10\%}$	$q_{10\%} + I_{95\%}$
$\mathcal{G}_1$	88,0	0,9	88,9	89,4	89,8
$\mathcal{G}_2$	72,3	4,5	76,2	78,1	80,0
$\mathcal{G}_3$	159,4	64,5	235,8	275,6	315,4
$\mathcal{G}_4$	58,3	9,7	67,6	72,0	76,3

Из таблицы следует, что оценка технологической эффективности  $\mathcal{G}_1$  достаточно устойчива (около 88—89 %). Разброс пользовательской эффективности  $\mathcal{G}_2$  существенно выше, а

среднее значение — ниже (72 %), что можно объяснить учетом накладных расходов среды ГридННС. Абсолютная эффективность технических решений  $\vartheta_3$  значительно превышает 100 % вследствие того, что накладные расходы платформы CLAVIRE, используемой в НРС-NASIS II, существенно меньше накладных расходов ГридННС (см. рисунок, в). Относительная эффективность технических решений  $\vartheta_4$  меньше, чем пользовательская эффективность  $\vartheta_2$ , в силу сходных затрат на передачу данных.

**Вероятностная модель надежности предоставления сервисов.** В соответствии с моделью (1)—(2), среда облачных вычислений может иметь два режима функционирования:

— ожидания, когда в среде не выполняется ни одной сессии; все ресурсы свободны,  $\bar{\Omega} = \Omega$ ;

— предоставления сервисов в соответствии с (5), когда  $\bar{\Omega} \subset \Omega$ : в этой ситуации может быть одновременно запущено  $k$  сессий с различным временем выполнения.

Каждая сессия задействует ресурс  $\Omega_i$ , который может включать несколько вычислителей (узлов, процессоров, ядер). Как следствие, вероятность отказа пропорционально увеличивается. Это позволяет использовать для оценки надежности *условное* время работы  $T_{0j}$  параллельной программы на ресурсе  $\Omega_i$ , которое равно времени работы этой программы на одном процессоре с данными характеристиками  $\Xi_{ij}$  (3). Совокупное время работы в режиме предоставления сервисов является случайной величиной, зависящей от количества пользователей, работающих в данный момент в среде, выбора сервисов, исходных данных и режимов работы. В предположении о достаточности вычислительных ресурсов среднее время работы в режиме предоставления сервисов может быть представлено в виде:

$$\tilde{T} = \sum_{l=1}^R C_l \bar{T}_{0l} f_l(P_l^*) (1 - \pi_l). \quad (6)$$

Здесь  $C_l$  — весовой коэффициент, отражающий потребительский интерес к использованию прикладного пакета  $l$ ,  $P_l^*$  — оптимальное количество вычислителей (ядер, процессоров, узлов),  $\bar{T}_{0l}$  — среднее время использования пакета (по разным наборам исходных данных),  $\pi_l$  — вероятность отказа конкретного сервиса. Функция  $f_l(P_l^*) \geq 1$  отражает дополнительные накладные расходы на поддержку параллельного выполнения пакета. Она может быть представлена в различных параметризациях, например, в форме закона Амдала. Однако применительно к задаче определения среднего времени наработки на отказ можно полагать  $f \equiv 1$  с ошибкой в безопасную сторону. Величина  $T_{0l}$  определяется путем тестового исполнения типовых задач. Величины  $C_l$  определяются экспертно специалистами предметной области, а  $\pi_l$  — путем тестирования соответствующих прикладных пакетов в процессе их разработки и опытной эксплуатации. Например, пусть на момент тестирования пакета имеется  $K$  ошибок. На первом этапе тестирования было произведено  $\tau_1$  запусков программного обеспечения и обнаружено  $k_1$  ошибок. После проведения мероприятий, направленных на устранение ошибок, в пакете остается  $K - \alpha k_1$  ошибок. Коэффициент  $\alpha$  связан с тем, что при внесении исправлений (устранении обнаруженных при тестировании ошибок) могут появиться новые ошибки. Соответственно после проведения  $m$ -го этапа тестирования, потребовавшего  $\tau_m$  запусков программы, в ней остается  $K - \alpha \sum_{j=1}^m k_j$  ошибок.

В предположении, что число ошибок  $c\lambda$  в программе распределено по закону Пуассона, где  $\lambda$  — частота появления ошибки, а  $c$  — коэффициент пропорциональности, вероятность безотказной работы  $p(0) = 1 - \pi$  и среднее время наработки на отказ  $ET$  имеют вид

$$p(0) = \exp\left(-\left(\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i\right)t\right), ET = \frac{1}{1 - \exp\left(-\left(\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i\right)\right)}, ET \approx \frac{1}{\hat{\gamma} - \hat{\beta} \sum_{i=1}^m k_i}, \quad (7)$$

где  $\gamma = cK, \beta = c\alpha$  — приведенные параметры, оцениваемые методом максимального правдоподобия. В общем случае оценки  $\hat{K}$  и  $\hat{K} - \hat{\alpha}k_i$  могут принимать нецелые значения, так как число ошибок до проведения тестирования — случайная величина, и в модели в качестве параметра присутствует математическое ожидание числа ошибок. Для оценки частоты возникновения ошибок по данным поэтапного тестирования используется метод наименьших квадратов.

Учитывая, что среда облачных вычислений используется разными пользователями, запускающими различные сервисы, можно определить среднее время безотказной работы среды в целом как среднюю длину серии успешных независимых запусков до первого отказа. Поскольку каждый запуск подчиняется распределению Бернулли, то длина всей серии будет распределена в соответствии с геометрическим законом  $f(n) = (1 - \pi_\Sigma)^n \pi_\Sigma$ . Среднее значение этого распределения  $\langle n \rangle = 1/\pi_\Sigma$ , таким образом:

$$T_\Sigma = \hat{T}/\pi_\Sigma. \quad (8)$$

Модель (8)—(9) является обобщенной по отношению к исходному разнообразию функциональных возможностей среды облачных вычислений и вариантов использования прикладных сервисов, в силу того что рассматривает последовательность запуска одинаковых „усредненных“ сервисов со средним временем выполнения  $T_\Sigma$ .

В качестве примера в табл. 2 приведены значения параметров выражения (6) для предметно-ориентированных сервисов среды HPC-NASIS II.

Таблица 2

Прикладной пакет	Параметры			
	$C_i, \%$	$T_i, \text{мин}$	$P_i^*$	$\pi_i, \%$
SEMP	10	600	1	1
PRIRODA	15	30	8	0,5
NDDO	10	5	8	3
DPIMC	0,1	300	8	1
MD- kMC	15	600	8	1,5
NTDMFT	5	300	8	1
JAGGREGATE	0,1	3	8	1
QDLaser	0,1	3	8	1
PLASMON	0,1	1	8	1
NANOIMAGE	10	10	24	1
NANOFLOW	4,4	60	24	5
NAEN	0,1	3	24	2
MAGNET	0,1	10	24	3
GAMESS	15	60	8	0
ORCA	15	60	8	0

В табл. 2 для компонентов GAMESS и ORCA принято  $\pi_i = 0$ , в силу того что эти компоненты являются внешними подключаемыми пакетами и не подвергались тестированию в ходе разработки. Используя данные из табл. 1 в (6)—(8), получим точечную оценку среднего



времени безотказной работы всего комплекса —  $T_{\Sigma} = 965$  ч. При этом выборочная изменчивость такой оценки достаточно велика. В частности, для верхней границы 90 % доверительного интервала  $\pi_{\Sigma}$ , в силу того что в (8) выполняется деление на относительно малую величину, получается следующая оценка снизу времени безотказной работы:  $T_{\Sigma}^{-90\%} = 643$  ч.

**Заключение.** Вероятностные модели производительности (1)—(5) и надежности (6)—(8) ориентированы на выполнение поверочных расчетов для оценки качества функционирования сервисов в ходе разработки и опытной эксплуатации проблемно-ориентированных сред облачных вычислений.

Работа выполнена при поддержке проектов Минобрнауки по реализации Постановлений № 218 и 220 Правительства Российской Федерации (2010—2012 гг.).

#### СПИСОК ЛИТЕРАТУРЫ

1. Надежность автоматизированных систем управления. ГОСТ 24.701 86.
2. Shootan M. L. Operational Testing and Software Reliability Estimation During Program Development, Record // IEEE Symp. on Computer Software Reliability. 1973. N 73. P. 51—76.
3. Благодатских В., Волик В., Посакалов К. Стандартизация разработки программных средств. М.: Финансы и статистика, 2005. 288 с.
4. Майерс Г. Искусство тестирования программ. М.: Финансы и статистика, 1982. 174 с.
5. Спельников Д. М., Гуськов А. А., Маслов В. Г., Бухановский А. В. Учебно-научный комплекс „Компьютерное моделирование в нанотехнологиях“ на основе Грид-среды // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 44—50.
6. [Электронный ресурс]: <www.ngrid.ru>.
7. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.

#### Сведения об авторах

- Тимофей Николаевич Чуров** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: tchurovtim@gmail.com
- Денис Александрович Насонов** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: denis.nasonov@gmail.com
- Борис Яковлевич Штейнберг** — д-р техн. наук; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; зав. кафедрой;  
E-mail: borsteinb@mail.ru
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

---

---

# ПРИЛОЖЕНИЯ ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННЫХ СРЕД НА ОСНОВЕ iPSE

---

---

УДК 004.925;004.942;004.946

А. А. БЕЗГОДОВ, С. В. ИВАНОВ, С. С. КОСУХИН, А. В. БУХАНОВСКИЙ

## ВИРТУАЛЬНЫЙ ПОЛИГОН ДЛЯ ИССЛЕДОВАТЕЛЬСКОГО ПРОЕКТИРОВАНИЯ МОРСКИХ ОБЪЕКТОВ И СООРУЖЕНИЙ

Рассмотрена задача организации виртуального полигона для задач исследовательского проектирования в судостроении. Разработан метод формирования динамических сцен поведения морских объектов на трехмерном волнении при экстремальных условиях эксплуатации, адаптированный для использования в широкоэкранных системах виртуальной реальности.

*Ключевые слова:* виртуальный полигон, исследовательское проектирование, экстремальная ситуация, трехмерное волнение.

**Введение.** Исследовательское проектирование морских объектов (МО) связано с обеспечением конструктора необходимой информацией при разработке и принятии конкретных решений в процессе оценки характеристик проектируемого объекта для различных условий эксплуатации, в том числе — экстремальных ситуаций при неопределенности и неполноте исходной информации [1]. Несмотря на то что развитие экстремальной ситуации является сложным многовариантным процессом, ретроспективный анализ известных инцидентов позволяет выявить обстоятельства, способствующие их возникновению, например, параметрические резонансы разной природы [2], потеря управляемости на гребне волны, захват судна волной (брочинг) [3]. В свою очередь, каждая из вышеперечисленных ситуаций может усложняться за счет внутренних факторов (смещение навалочного груза, затопление отсеков, интенсивное обледенение и пр.). Как следствие, разнообразие и неоднозначность влияния экстремальных условий эксплуатации ограничивают возможности физического эксперимента, поэтому требуется развивать соответствующие средства компьютерного моделирования динамики МО под воздействием внешних возмущений.

Подходы к компьютерному моделированию динамики морских объектов можно условно разделить на два конкурирующих класса: использование нелинейных моделей на основе интегрирования уравнений движения [4] и имитационных моделей динамики судна в прямой гидромеханической постановке решения задач гидромеханики [5]. Модели на основе уравнений движения в основном адаптированы под отдельные классы экстремальных задач за счет введения соответствующих нелинейных членов; они не обеспечивают необходимой гибкости управления при проведении компьютерного эксперимента. Напротив, полные гидродинамические модели, основанные на численном решении уравнений Навье—Стокса, лишены этого недостатка; однако их реализация требует существенно больших ресурсов.

Проблема выбора и применения математической модели для реализации инструментальных средств поддержки исследовательского проектирования связана с используемой парадигмой человеко-компьютерного взаимодействия. Так, технология виртуальной реальности (ВР) обеспечивает визуальное восприятие конструктором особенностей поведения МО, создавая эффект погружения в моделируемое явление с возможностью всестороннего наблюдения и анализа воспроизводимых закономерностей [6]. Технологическая реализация такого подхода стимулирует развитие нового класса комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента — виртуальных полигонов (ВП) [7, 8]. В настоящей статье рассматриваются особенности ВП для исследовательского проектирования МО.

**Компьютерное моделирование динамики морского объекта под воздействием внешних возмущений.** Реалистичная визуализация динамики МО в режиме реального времени требует применения адаптированных моделей и методов, ориентированных на воспроизведение качественных эффектов с разумным уровнем точности для широкого диапазона условий эксперимента, с временными затратами, обеспечивающими использование в интерактивном режиме. Поскольку структура традиционных моделей динамики МО в форме идеализированных уравнений движения для воспроизведения разных экстремальных ситуаций существенно различается (что ограничивает области их применения), при создании ВП был использован прямой метод, основанный на вычислении главных сил и моментов непосредственным интегрированием по мгновенной погруженной поверхности корпуса объекта  $S$ :

$$\mathbf{F} = - \left[ \iint_S p \mathbf{n} dS \right] - \left[ \iint_S H \mathbf{v} dS \right] + \mathbf{D} + \mathbf{F}_{\text{ext}}, \quad (1)$$

$$\mathbf{M} = - \left[ \iint_S (p \mathbf{n})(\mathbf{r} - \mathbf{p}) dS \right] - \left[ \iint_S (H \mathbf{v})(\mathbf{r} - \mathbf{p}) dS \right] + \mathbf{M}_{\text{ext}}. \quad (2)$$

Здесь  $\mathbf{D}$  — весовое водоизмещение объекта,  $p$  — гидростатическое и гидродинамическое давление воды в заданной точке,  $\mathbf{n}$  — нормаль к поверхности,  $\mathbf{r}$  — радиус-вектор точки поверхности в глобальных координатах,  $\mathbf{p}$  — положение объекта в пространстве, определяемое шестью степенями свободы,  $H \mathbf{v}$  — демпфирующая сила, действующая на единицу поверхности корпуса объекта,  $\mathbf{v}$  — скорость частиц вдоль поверхности судна в точке интегрирования,  $\mathbf{F}_{\text{ext}}$  и  $\mathbf{M}_{\text{ext}}$  — дополнительные сила и момент (внешнее управление).

Соответствующие интегралы рассчитываются численно по квадратурным формулам типа Маркова с локально распределенными случайными узлами, перестраиваемыми на каждом шаге  $t$ . Такой подход позволяет компенсировать невязку, вызванную дискретным характером сетки, при аппроксимации непрерывно изменяющегося профиля мгновенной ватерлинии.

Сила  $\mathbf{F}$  и момент  $\mathbf{M}$  рассматриваются относительно редуцированных масс и моментов инерции объекта, включающих в себя собственные и присоединенные составляющие. Структура выражений для демпфирующей силы и гидродинамической составляющей давления соответствует ньютоновским принципам гидродинамики; коэффициенты пропорциональности определяются экспериментально по записям затухающих колебаний.

В каждый момент времени  $t$  граница погруженной поверхности  $S(t)$  определяется исходя из мгновенного профиля взволнованной поверхности моря, задаваемой моделью Лонге-Хиггинса [9], идентифицируемой по частотно-направленному спектру волнения  $E(\mathbf{k})$ , где  $\mathbf{k}$  — волновой вектор. Достоинством модели (несмотря на ее относительно слабую сходимость) является возможность наглядной гидродинамической интерпретации с точки зрения определения волновых давлений в терминах вертикальных смещений частиц жидкости на глубине  $d$ :

$$p(t, \mathbf{p}, d) = \sum_{i,j} \gamma \exp(-|\mathbf{k}_{ij}|d) a_{ij} \cos(\mathbf{p} \cdot \mathbf{k}_{ij} - \omega(\mathbf{k}_{ij})t + \delta_{ij}) + d\gamma. \quad (3)$$

Здесь  $a_{ij}$  — коэффициенты модели Лонге-Хиггинса, которые определяются по спектру волнения;  $\delta_{ij}$  — равномерно распределенные случайные фазы,  $\omega = \omega(\mathbf{k})$  — дисперсионное соотношение для волн малой амплитуды;  $\gamma$  — объемный вес воды;  $\mathbf{p}$  — радиус-вектор произвольной точки в плоскости тихой воды;  $\mathbf{k}$  — волновой вектор.

Значения главных сил (1) и моментов (2) в каждый момент времени  $t$  позволяют определить пространственные линейную и угловую скорость перемещения судна. При этом модель (1)—(3) является достаточно удобной для интерактивной визуализации: она позволяет в режиме реального времени с приемлемой точностью воспроизводить динамику МО с учетом всех шести степеней свободы.

Дополнительные сила и момент  $\mathbf{F}_{\text{ext}}$  и  $\mathbf{M}_{\text{ext}}$  в выражениях (1), (2) используются для моделирования внешних воздействий (ветровой шквал, удар разрушающейся волны) и управляющих сил (перекладка руля, работа движителя). Как следствие, это позволяет придать задаче управления процессом визуализации в рамках (1), (2) ясную физическую интерпретацию: нестационарные изменения режимов движения объекта (например, поворот на заданный курс) осуществляются за счет приложения дополнительных сил, направление и интенсивность которых задаются в сценарии модельного эксперимента либо наблюдателем посредством манипулятора системы ВР.

**Технология формирования динамических сцен в виртуальном полигоне.** ВП представляет собой совокупность программных механизмов для проведения виртуальных экспериментов, которые обеспечивают:

- настройку параметров моделирования,
- создание сценариев модельных экспериментов,
- интерактивную визуализацию результатов компьютерного эксперимента в моно- и стереорежиме,
- экспорт расчетных данных для последующей обработки в пакетах математического моделирования и проектирования.

Специфика интерактивной визуализации связана с обеспечением эффекта погружения конструктора в ВП, что в данном случае требует отображать (помимо МО) визуально бесконечную водную поверхность, простирающуюся от точки наблюдения до горизонта. С этой целью была применена технология неравномерных сеток, привязанных к камере (рис. 1, *a* — общая структура сетки, *b* — сетка области интенсивного волнения). Она позволяет компоновать сеточную область из трех частей.

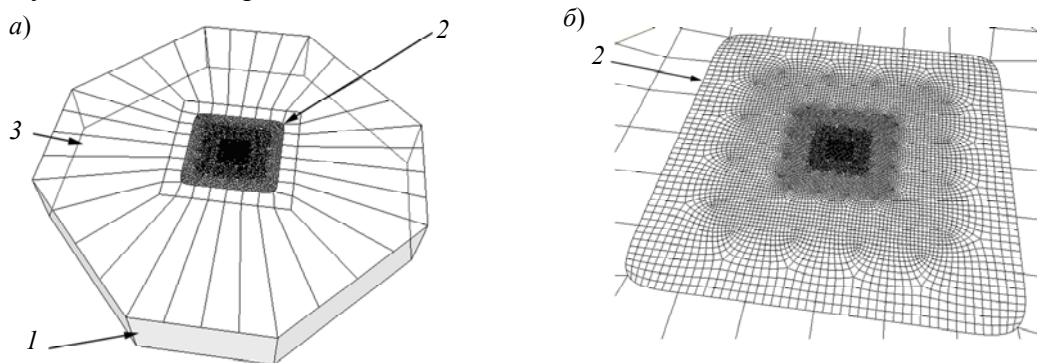


Рис. 1

Область 1 (дно) используется для маркировки буфера трафарета при отображении раздела сред, область 2 (т.н. область интенсивного волнения) применяется для отображения собственно волн в соответствии с моделью (3), а 3 (область горизонта) — находится

достаточно далеко от наблюдателя, в ней видимой высотой волн в процессе визуализации можно пренебречь.

В процессе визуализации центр такой сетки всегда находится под камерой или над ней. Повороты камеры на ориентацию сетки в пространстве не влияют. Вертикальная координата вершин в области 2 модифицируется вершинным шейдером в соответствии с моделью волнения. При этом высота волн плавно уменьшается по мере увеличения расстояния от наблюдателя. Область интенсивного волнения формируется путем рекурсивного подразбиения центральной части сетки размером  $4 \times 4$  ячейки с последующим сглаживанием методом Катмулла—Кларка. Такой подход обеспечивает непрерывный переход от областей с большей детализацией поверхности к областям с меньшей детализацией. Ординаты взволнованной поверхности вычисляются по модели (3) с использованием быстрого преобразования Фурье (БПФ) по технологии CUDA на GPU и передаются в вершинный шейдер как двумерная текстура. Дополнительно при закрасивании водной поверхности учитываются частичное отражение Френеля (только небо, рис. 2, б) и частичное преломление с затуханием по глубине (рис. 2, в). В рамках используемой технологии визуализации камера может находиться не только над, но и под водой, а также на границе сред. Для корректного отображения границы сред при визуализации морской поверхности используется буфер трафарета для маркировки погруженной в воду плоскости экрана, с последующим применением эффекта затуманивания (см. рис. 2, а).

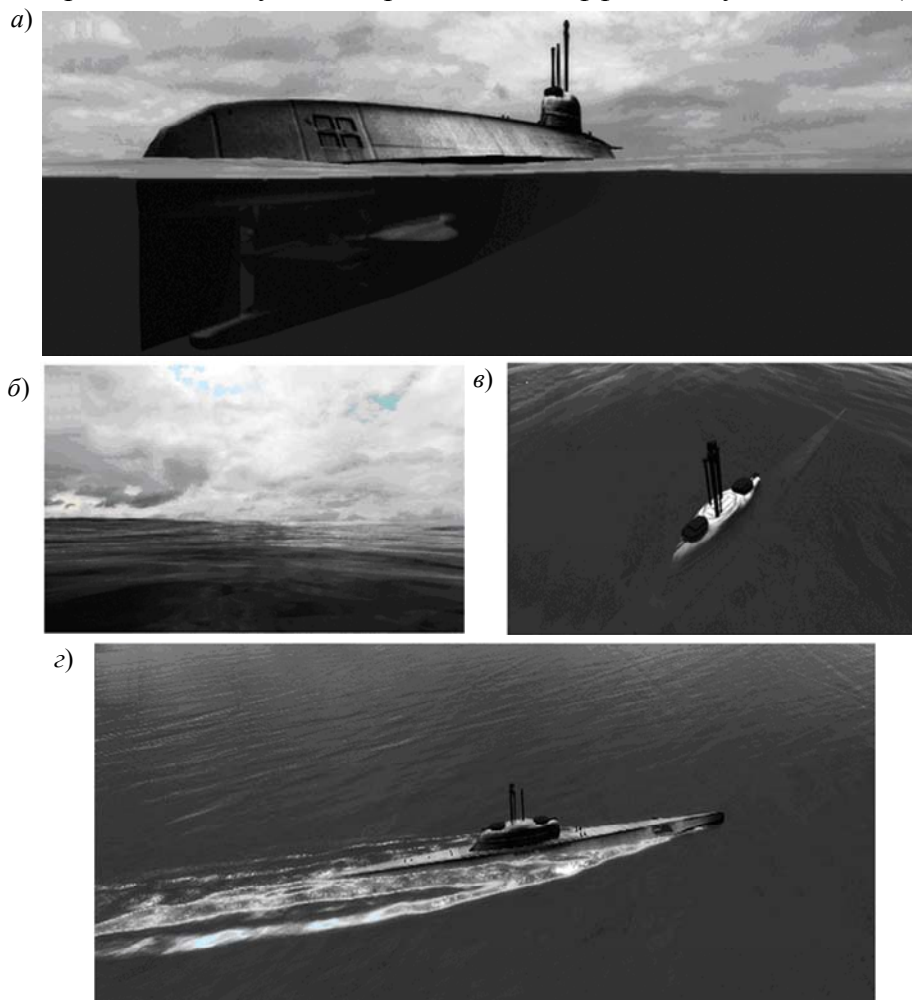


Рис. 2

Дополнительным аспектом отображения визуальных свойств водной поверхности является воспроизведение расходящихся корабельных волн при движении плавучего МО. Поскольку нелинейная задача расчета корабельных волн в полной постановке является

существенно более ресурсоемкой, по сравнению с моделью (1)—(3), для создания визуального эффекта используется упрощенная модель, основанная на решении линейного уравнения колебаний на регулярной сетке. Начальное возмущение формируется в точках пересечения корпуса судна и морской поверхности, а величина возмущения определяется в зависимости от локальной линейной скорости точки судна относительно поверхности воды. Уравнение решается на GPU, результат решения уравнения как набор значений в текстуре передается в вершинный шейдер, и высота колебаний добавляется к высоте волн. В тех областях, где скорость частиц колеблющейся поверхности выше определенного значения, поверхность моря окрашивается в белый цвет, что создает визуальный эффект пенообразования (рис. 2, з).

**Моделирование динамики судна в режимах параметрического резонанса и брочинга с помощью виртуального полигона.** Программный комплекс ВП развернут и апробирован на базе системы ВР Центра ситуационного моделирования и визуализации СПбГУ ИТМО. Система имеет недеполяризующий стереоэкран обратной проекции размером 3,35×2,0 м, изображение на котором формируется посредством шести HDTV DLP-проекторов Rohar Projectiondesign. Интерактивное взаимодействие с наблюдателем в системе ВР обеспечивается с помощью манипулятора с шестью степенями свободы производства 3dconnexion серии Space Pilot. Создание и подготовку стереоизображения обеспечивает рабочая станция HP Z800 с видеокартой nVidia Quadro FX 5800.

ВП позволяет моделировать эксперименты, связанные, в первую очередь, с возникновением сильной качки МО, вызванной резонансными явлениями разного рода. Основной резонанс качки обусловлен близостью характерных периодов волнения и собственных колебаний судна и хорошо предсказуем даже в линейном приближении. Напротив, параметрические резонансы имеют более тонкую природу, связанную с взаимовлиянием различных видов качки; условия их возникновения не всегда очевидны. К аналогичному классу явлений относится брочинг — неуправляемый разворот судна лагом к волне, сопровождаемый сильным динамическим креном. Поскольку брочинг является нестационарным процессом, для анализа его воспроизводимости в условиях ВП (рис. 3, а) необходимо создавать выборку экспериментов с одинаковыми начальными условиями (но различными случайными реализациями поля волнения), что позволило классифицировать и описать основные условия его возникновения исходя из диаграммы курса  $\varphi$ , угла крена  $\theta$  (рис. 3, в), угловой скорости  $\dot{\varphi}$  (рис. 3, д) и высоты волны в точке, где находится судно (рис. 3, з). На рис. 3, б представлена траектория судна, цифры у кривой — время (в секундах).



Рис. 3, а

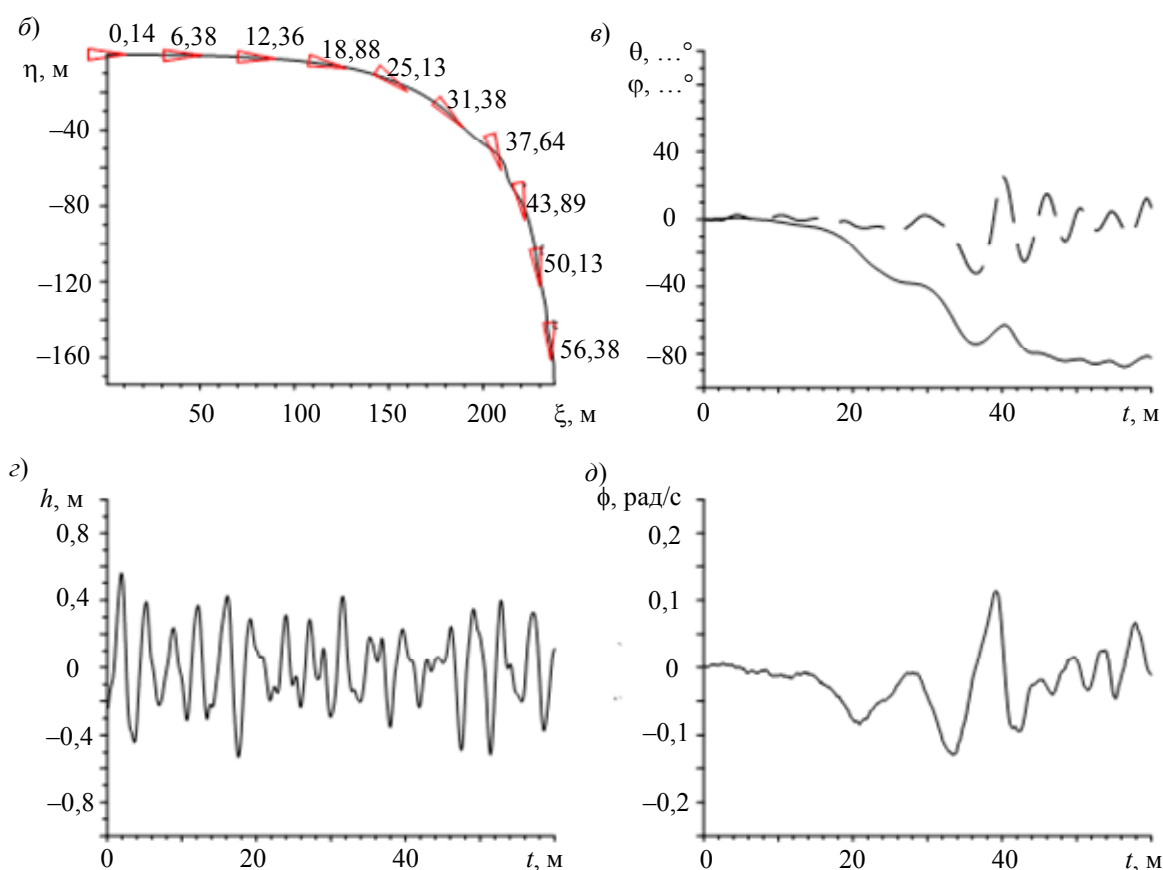


Рис. 3

Таким образом, разработанная технология ВП позволяет эффективно формировать динамические сцены на основе численного моделирования динамики внешней среды и МО, с учетом графических эффектов визуализации взволнованной поверхности моря и ее взаимодействия с корпусом объекта. Она адаптирована для применения в широкоэкранных системах ВР и дает возможность проведения компьютерных экспериментов по оценке динамических характеристик проектируемого объекта в экстремальных условиях эксплуатации.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Интеллектуальные системы в морских исследованиях и технологиях / Под ред. Ю. И. Нечаева. СПб: ГМТУ, 2001. 352 с.
2. France W. N., Levadou M., Treacle T. W., Paulling J. R., Michel R. K., Moore C. An Investigation of Head-Sea Parametric Rolling and Its Influence on Container Lashing Systems // Marine Technology. 2003. Vol. 40, N 1. P. 1—19.
3. Нечаев Ю. И., Завьялова О. П. Экстремальная ситуация „бродячий“: анализ и критериальные оценки условий „захвата“ и опрокидывания судна // Морской вестник. 2004. № 1(9). С. 87—92.
4. Sutulo S., Guedes Soares C. An implementation of the method of auxiliary state variables for solving seakeeping problems // Intern. Shipbuild. Progr. 2005. Vol. 52, N 4. P. 357—384.
5. Wu G. X., Hu Z. Z. Simulation of nonlinear interactions between waves and floating bodies through a finite-element-based numerical tank // Proc. Royal Soc. A: Math. Phys. and Eng. Sci. 2004. Vol. 460 (2050). P. 2797—2817.

6. Клименко С. В., Никитин И. Н., Никитина Л. Д. Аванго — система разработки виртуальных окружений. М.: Изд-во МФТИ — М.-Протвино, 2006. 252 с.
7. Charles W. Brice, Levent U. Gokdere, Roger A. Dougal. The Virtual Test Bed: An Environment for Virtual Prototyping // Proc. Intern. Conf. on Electric Ship (ElecShip'98). 1998. P. 27—31.
8. Bardina J., Rajkumar T. Intelligent Launch and Range Operations Virtual Test Bed (ILRO-VTB) // Proc. Enabling Techn. for Simulation Sci. VII. Vol. 5091. P.141—148.
9. Лонге-Хиггинс М. С. Статистический анализ случайной движущейся поверхности // Ветровые волны. М.: ИЛ, 1962. С. 112—230.

**Сведения об авторах**

- Алексей Алексеевич Безгодов** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: demiurghg@gmail.com
- Сергей Владимирович Иванов** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: Sergey.v.ivanov@rambler.ru
- Сергей Сергеевич Косухин** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: skosukhin@gmail.com
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор; E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.



С. В. ИВАНОВ, Е. В. БОЛГОВА, В. В. КАШИРИН, А. В. ЯКУШЕВ,  
А. В. ЧУГУНОВ, А. В. БУХАНОВСКИЙ

## WEB-ОРИЕНТИРОВАННЫЙ ПРОИЗВОДСТВЕННО-ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР „СОЦИОДИНАМИКА“

Рассмотрены концепция и принципиальная архитектура проблемно-ориентированной среды облачных вычислений, обеспечивающей функционирование производственно-исследовательского центра, для исследования социодинамики и ее приложений, в рамках парадигмы web 2.0.

*Ключевые слова:* социодинамика, социометрия, облачные вычисления, прикладные сервисы, социальная сеть, краулер, распространение слухов.

**Введение.** Развитие информационных технологий стимулирует появление новых методов и направлений исследований в различных предметных областях. В частности, бурный рост числа пользователей социальных сетей в Интернете обеспечивает информационную базу, позволяющую на качественно новом уровне обеспечить исследования в области социодинамики — раздела социологии, посвященного количественным методам моделирования взаимоотношений между индивидами или группами. Традиционно развитие социодинамики ограничивалось социометрическим фактором — возможностью наблюдения (измерения) соответствующих процессов в обществе, поскольку измерения и анализ парных или групповых взаимодействий индивидов гораздо более сложны, чем их индивидуальных характеристик в рамках выборочного подхода. Однако в глобальных социальных сетях подобные взаимоотношения виртуализируются, формируя, таким образом, слепок общественной структуры в пространстве Интернета [1, 2]. Несмотря на то что поведение и характеристики пользователей социальных сетей в ряде аспектов могут существенно отличаться от реальных, во многом эти факторы имеют систематический характер, что позволяет их учитывать (путем смещения определенных параметров) при обработке и анализе соответствующих данных [3]. Таким образом, социальные сети в настоящее время, по-видимому, составляют основу социометрических исследований нового поколения.

В настоящей статье рассматриваются концепция и принципиальная архитектура специализированного web-ориентированного производственно-исследовательского центра, который предоставляет возможности сбора, анализа и использования в моделях социометрических данных социальных сетей в Интернете, на основе технологий облачных вычислений, в соответствии с бизнес-моделью SaaS (Software as a Service, программное обеспечение как услуга).

**Концепция web-ориентированного центра.** Специфика выполнения количественных исследований на основе социальных сетей связана с рядом особенностей, ограничивающих доступность таких данных для широкого круга исследователей; отметим некоторые из них.

— Доступ к данным глобальных социальных сетей регламентируется политикой оператора сети и соответствующим законодательством в области персональных данных. Для масштабного сбора и анализа соответствующих данных необходимо наличие предварительных соглашений с оператором.

— Социальные сети имеют технологически различные интерфейсы доступа к данным, существуют разные принципы обхода сетей. Как следствие, проведение измерений характеристик

различных сетей требует разработки специализированных средств сбора данных, для отдельных исследователей это весьма трудоемкий процесс.

— Сбор данных в социальных сетях является достаточно ресурсоемкой операцией и требует соответствующих выделенных вычислительных ресурсов. Регулярное выполнение таких операций различными пользователями увеличивает нагрузку на инфраструктуру оператора сети, что нежелательно.

— Алгоритмы обработки и анализа таких социальных сетей во многих случаях имеют нелинейную сложность, поскольку описывают взаимоотношения „каждого с каждым“. Поэтому исследование сетей достаточно большого объема требует применения соответствующих вычислительных ресурсов и программного обеспечения, допускающего эффективное распараллеливание.

— Визуализация результатов исследований на основе социальных сетей связана с применением достаточно сложных когнитивных алгоритмов, позволяющих наглядно представить различные процессы на непланарных структурах данных большого объема, это приводит к необходимости использования специализированного программного обеспечения.

В совокупности перечисленные проблемы препятствуют развитию методов и технологий современной социометрии на основе социальных сетей в Интернете, это усугубляется тем, что потенциальные пользователи сетей (специалисты в науках об обществе) в большинстве случаев не обладают специализированными навыками для их самостоятельного решения. Выходом из сложившейся ситуации является эксплуатация концепции облачных вычислений — создание проблемно-ориентированной среды, обеспечивающей доступ к соответствующим сервисам (ресурсам, данным и процедурам их обработки и моделирования) через web-интерфейс. В качестве основы для развития среды рассматривается многофункциональная инструментально-технологическая платформа CLAVIRE (CLOUD Applications VIRTUAL Environment) обеспечения доступа к сервисам и композитным приложениям в среде облачных вычислений.

Проблемно-ориентированная среда включает следующие элементы:

— управляющая оболочка (ядро платформы), которая состоит из взаимодействующих системных web-сервисов, функционирующих с низкоуровневой вычислительной структурой, осуществляющих поддержку образа облачной среды, мониторинг и управление ресурсами, конструирование и исполнение сценариев, а также поддержку пользовательских интерфейсов. Управляющая оболочка реализуется на основе концепции iPSE (Intelligent Problem Solving Environment) [4]. Фактически web-центр представляет собой открытую интеллектуальную проблемно-ориентированную среду, объединяющую распределенные сервисы вычислений и доступа к данным и позволяющую эффективно управлять параллельными вычислительными процессами в распределенной иерархической среде на основе интеллектуальных технологий [5];

— набор прикладных сервисов вычислений и доступа к данным. В него входят как различные прикладные пакеты для обработки данных и социодинамического моделирования, доступные в рамках концепции облачных вычислений, так и специализированные инструменты для поиска и извлечения данных сбора данных из социальных сетей — краулеры [6]. Все сервисы строятся на основе предметно-ориентированных описаний пакетов на языке EasyPackage, регистрируемых в базе пакетов управляющей оболочки [7];

— дополнительные средства, обеспечивающие поддержку виртуального профессионального сообщества пользователей в рамках концепции web 2.0. Они включают в себя интерактивные средства общения, совместного выполнения проектов, поддержки единого рабочего пространства, а также средств, позволяющих вести дискуссии в режиме реального времени с использованием графических и текстовых средств общения. Кроме того, имеются сервисы интерактивной консультации экспертов, предусматривается возможность сохранения результатов выполненных задач для последующего использования другими членами сообщества

или для совместного обсуждения и исправления. В качестве отдельной задачи рассматриваются сбор, обработка и анализ текущей и ретроспективной информации о процессах в виртуальном профессиональном сообществе (включая ряд показателей индивидуальной и коллективной активности пользователей, характеристики востребованности сервисов и пр.).

Аппаратная составляющая поддержки web-ориентированного центра формируется в составе распределенной иерархической среды облачных вычислений, включающей в себя выделенные суперкомпьютеры, виртуальные машины в „облаке“ и целевые системы в составе Грид. Единообразный способ работы с центром, равно как и оптимизация распределения вычислительной нагрузки, осуществляется средствами управляющей оболочки без привлечения пользователя.

На рис. 1 представлена схема функционирования web-центра, демонстрирующая работу сервисов доступа к данным и приложениям в области социодинамики.

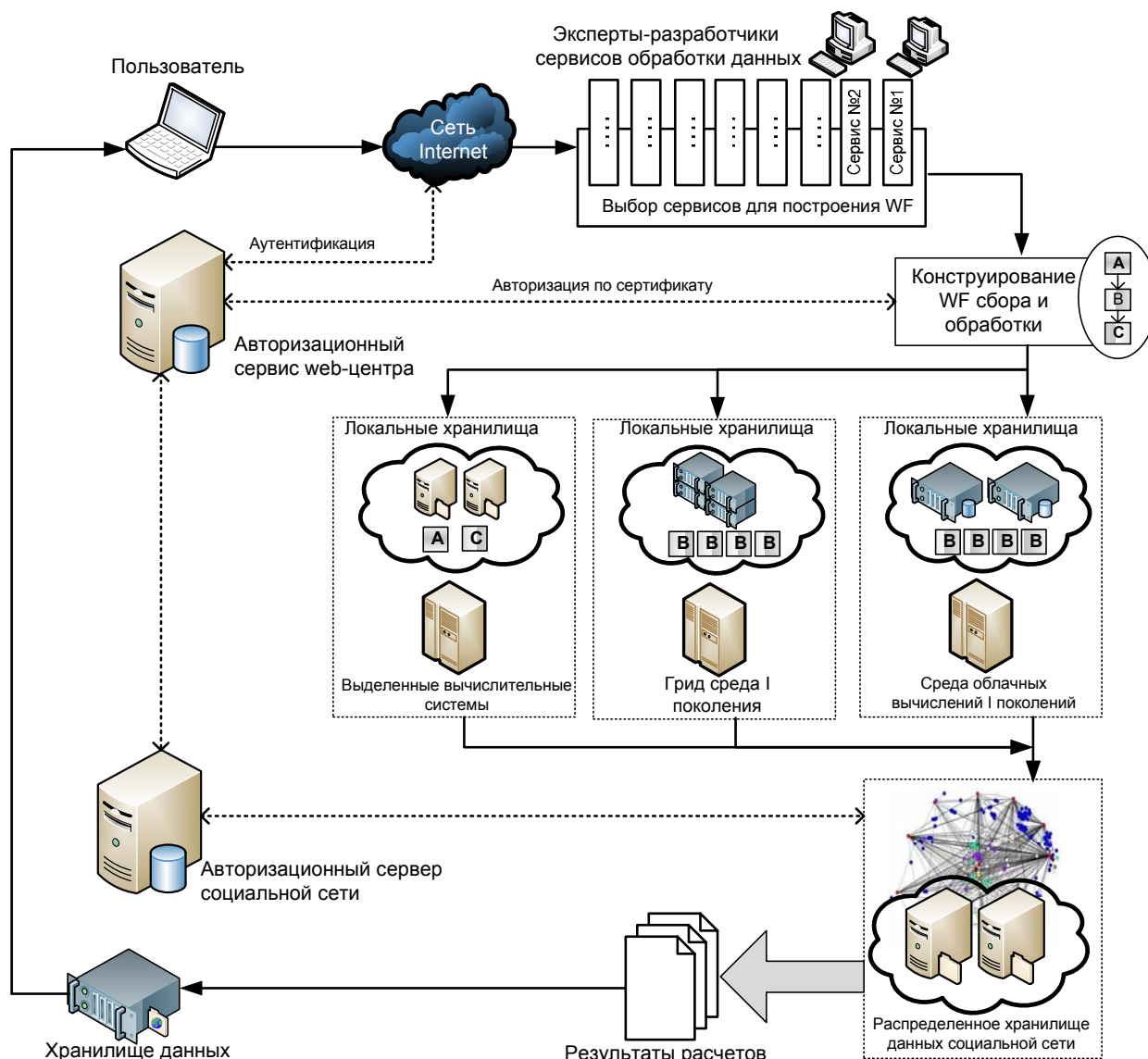


Рис. 1

Пользователь авторизуется в проблемно-ориентированной среде через портал провайдера. Через соответствующий web-интерфейс он может выбрать конкретные сервисы или шаблоны композитных приложений в форме потоков заданий (workflow, WF), а также получить (при необходимости) доступ к технической и эксплуатационной документации. Выбрав необходимые ему сервисы, пользователь средствами управляющей оболочки конструирует

соответствующее композитное приложение в форме WF, которое определяет правила сбора, обработки и анализа данных. При этом может использоваться как графическая, так и текстовая форма представления композитных приложений. Для подготовленного описания композитного приложения пользователь конфигурирует условия вычислений: определяет требуемые параметры WF, редактирует (при необходимости) его описание, готовит и загружает в хранилище среды входные данные для расчетов. В ряде случаев такие данные (например, учебные базы данных фрагментов социальных сетей) могут предоставляться провайдером web-центра.

Затем пользователь определяет режим исполнения задачи в среде (утверждает предлагаемые ему варианты) в соответствии с требованиями к временным характеристикам расчета и правилами доступа к различным источникам данных. При этом пользователю предлагаются различные тарифные варианты, связанные с использованием разных социальных сетей и привлечением ресурсов сред распределенных вычислений. Окончательно пользователь запускает задачу на исполнение в проблемно-ориентированной среде. Использование вычислительных ресурсов и сервисов сбора данных в социальных сетях производится с учетом единого сертификата, который обеспечивает права пользователя, делегированные провайдером. В процессе вычислений пользователь (при необходимости) осуществляет мониторинг процесса исполнения (в форме динамического отображения WF); при этом прогнозируется время завершения вычислений. Когда все расчеты завершены, результаты помещаются в хранилище данных web-центра; пользователю отправляется соответствующее уведомление (sms, e-mail и пр.). Пользователь может получить доступ к результатам расчетов через интерфейс проблемно-ориентированной среды.

**Прикладные сервисы в составе web-ориентированного центра** в соответствии с выполняемыми ими функциями можно условно разделить на четыре группы: сбора данных в социальных сетях, статистической обработки и анализа данных, моделирования сценариев, а также визуализации.

*Сервисы сбора данных в социальных сетях* реализуют различные модели краулинга (обход в глубину, в ширину) с оценкой общности по различным факторам, включая семантический профиль узлов сети. В рамках распределенной среды эффективно распараллеливание сетевого канала в рамках модели облачных вычислений, когда запросы к базе отправляются одновременно с разных целевых систем. На каждой целевой системе функционирует рабочий агент краулера. Он получает задание на просмотр определенного множества узлов сети, после выполнения задания передает данные в централизованное хранилище. Действия отдельных агентов не синхронизируются. Управляющий узел (мастер) определяет порядок, в котором будут обходиться пользователи сети, тем самым он реализует политику обхода краулера. Архитектура краулера с централизованным управлением позволяет динамически добавлять и удалять агентов, обеспечивая масштабируемость системы в целом. Помимо классических политик обхода (обход в ширину и глубину) таким образом может быть дополнительно реализована политика обхода по степени влияния, согласно которой сначала посещаются те узлы, на которые идет самое большое число ссылок. Эта эвристика позволяет обходить сеть по топологическим сообществам — множествам тесно связанных друг с другом вершин.

Для эффективного сбора информации в социальных сетях важно обеспечить высокую производительность краулера, что достигается за счет баланса операций по просмотру и записи данных в социальной сети, а также операций по их передаче в Интернет. Например, в социальной сети Live Journal (Живой Журнал, ЖЖ) за один день краулер обрабатывает данные около 700 тысяч пользователей сети со средней скоростью 490 пользователей в минуту. При этом выполняется около 270 итераций (которые соответствуют заданиям для отдельных агентов). Анализ структуры временных затрат показал, что наиболее ресурсоемки операции с базой данных (около 70 %), в частности — сохранение связей между пользователями (18,6 %)

и списков интересов пользователей (39,4 %). Временные затраты на работу с сетью не превышают 27 %, что указывает на необходимость оптимизации доступа к базе данных.

Сервисы статистической обработки и анализа данных используют общий подход к описанию многомерных комплексных сетей — стохастических графов с многомерными характеристиками вершин. Под комплексной сетью [8] понимается граф с достаточно большим числом узлов, характеризующихся, в том числе, многомерным кортежем признаков, и динамически изменяющимися связями; распределение признаков узлов и характеристик связей может быть описано вероятностной моделью (многомерным распределением). Для их оценки используются методы многомерного статистического анализа, что обусловлено неопределенностью перехода к уравнениям относительно вероятностных характеристик сети в многомерном случае. Это требует совокупного применения формальных способов снижения размерности (обобщение метода главных компонент для графов), методов дискриминантного анализа для выявления характерных структур в сети, а также методов ординации (шкалирования) для учета многомерности признаков, описывающих узлы сети. При этом интерпретация результатов осложняется тем, что социальные сети могут включать в себя как формальные, так и неформальные сообщества. Формальным сообществом можно назвать группу индивидов, объединенных по какому-либо (формальному) признаку. Например, такие сообщества составляют взаимосвязанные пользователи, которые указали одним из своих интересов „книги“, „музыку“, „эзотерику“ и др. Напротив, неформальное сообщество включает индивидов с общими интересами, однозначно не отраженными в профиле. Поскольку пользователи социальных сетей зачастую указывают неполную или искаженную информацию в своих профилях, для исследования неформальных сообществ требуется совокупно использовать статистическую информацию о топологической структуре сети и наборе характеристик каждого индивида.

На рис. 2 представлен пример анализа предпочтений сообщества индивидов из социальной сети ЖЖ, в профиле интересов которых присутствуют упоминания о наркотиках; приведены интервальные оценки вероятности наличия интереса ( $P$ ) для сети в целом и точечные — для выборки наркоманов.

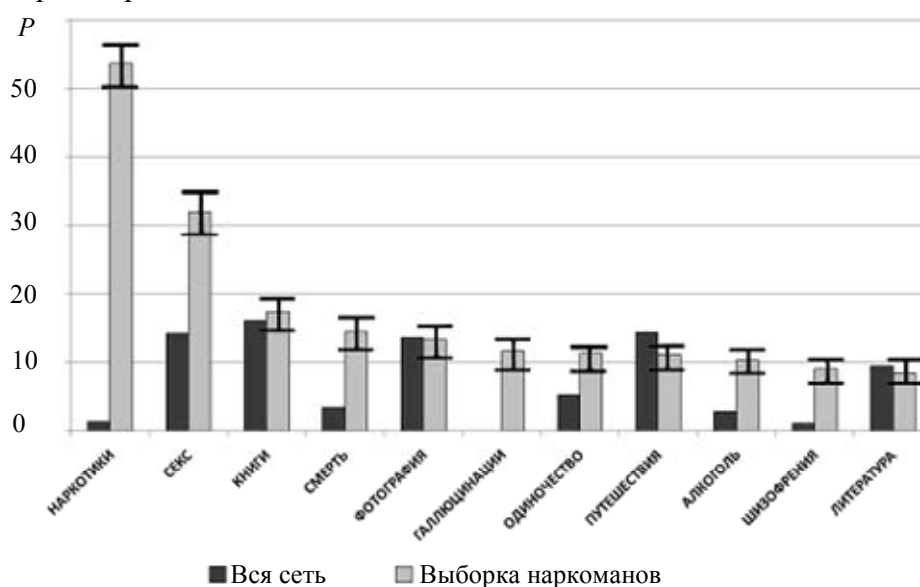


Рис. 2

Из рисунка следует, что проявление таких интересов у пользователей, как „музыка“, „книги“, „путешествия“, практически не связано с тем, что пользователь относит себя к формальному сообществу наркоманов. Наоборот, такие интересы, как „наркотики“, „смерть“, „одиночество“, характерны для пользователей из сообщества наркоманов и могут быть использованы в дальнейшем для выявления потенциальных потребителей наркотиков (группы риска).

Сервисы моделирования сценариев реализуют различные вероятностные модели социодинамики в терминах микро- и (или) макропараметров социальных сетей. Модель динамики комплексной сети задается в форме композиции стохастических эволюционных операторов над графом заданной структуры; каждый из операторов отражает определенный класс процессов в сети (присоединение новых пользователей, повышение рейтинга пользователя, расширение сферы научных интересов и пр.). Комплексная сеть характеризуется набором макропараметров (коэффициентов операторов), которые могут быть идентифицированы только по результатам измерений (посредством краулинга и обработки полученных данных). Посредством осреднения по ансамблю общее уравнение над графом сводится к системе обыкновенных дифференциальных уравнений, описывающих изменчивость отдельных вероятностных характеристик сети. Это позволяет исследовать чувствительность модели к изменению параметров на основе анализа фазовых портретов, в ходе чего могут быть выделены макропараметры, существенные для процесса мониторинга и управления.

В качестве иллюстрации на рис. 3 представлены результаты расчета распространения слухов в социальной сети на основе модели Далея—Кендалла [9]. В модели используются три множества вершин: неинформированные ( $I$ ), распространяющие ( $S$ ) и неактивные ( $R$ ). На каждом шаге, при взаимодействии на вершины типа  $I$  с вершиной типа  $S$ ,  $I$  с заданной вероятностью переходит во множество  $S$ , а при взаимодействии с вершиной из  $S$  или  $R$  — во множество  $R$ . Значения вероятностей можно задавать для всей сети, для каждого класса вершин и также для каждой пары классов, вершины которых вступают во взаимодействие. За шаг алгоритма можно взять сутки (что связано с цикличностью просмотра страниц пользователями социальных сетей), а взаимодействие между двумя вершинами определяется наличием связи между ними.

Сеть включает в себя два класса вершин. Распределение степеней первого класса характеризуется степенным законом с показателем степени 4, а второго — законом Пуассона с показателем 7. Вершины первого класса (80 %) определяют традиционную структуру социальной сети, тогда как второго (20 %) соответствуют сплоченному сообществу с большим количеством связей между участниками.

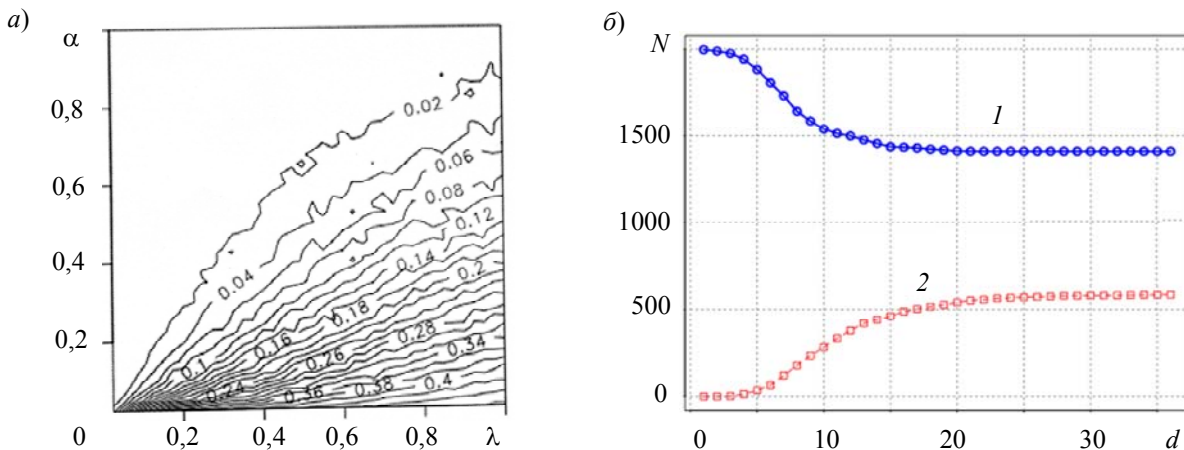


Рис. 3

На рис. 3, а приведен график, демонстрирующий степень покрытия вершин сети слухом в зависимости от параметров процесса распространения: вероятности передачи слуха ( $\alpha$ ) и его забывания ( $\lambda$ ). Видно, что наилучшее покрытие обеспечивается при высокой вероятности передачи слуха и низкой вероятности его забывания. На рис. 3, б приведена динамика распространения слуха, выраженная через число вершин ( $N$ ) во множествах  $I$  (1) и  $R$  (2) на определенном шаге работы алгоритма: видно, что с течением времени

(d) процесс выходит на определенную асимптоту — количество „осведомленных“ вершин не возрастает.

Сервисы визуализации ориентируются на применение моделей анализа и представления комплексных сетей, рассмотренных выше. В рамках web-центра используются пакеты *Rajec*, *Enronic* и *JUNG*, адаптированные к задачам серверной визуализации. В ходе выполнения расчетов визуализация выполняется на сервере, поддерживающем хранилище данных проблемно-ориентированной среды. Пользователь при этом имеет возможность просмотра статической картинки и (или) видеопотока средствами web-браузера (в зависимости от специфики решаемой задачи).

**Заключение.** Несмотря на то что, используя перечисленные сервисы, возможно выполнять разного рода расчеты в области социометрии и социодинамики, основным назначением web-центра является решение комплексных задач, требующих совокупного применения сервисов сбора, анализа, моделирования и визуализации. При этом сценарий решения задачи не является жестко заданным, а описывается пользователем в форме композитного приложения на языке *EasyFlow* [7]. Описания типовых задач (в форме соответствующих WF) могут быть представлены в репозитории для общего использования. К ним, в частности, относятся:

- построение социограммы неформального сообщества, анализ скорости и каналов распространения информации;
- анализ и прогноз индексов общественных настроений;
- выявление групп влияний в социальной сети и определение „лидеров мнений“;
- мониторинг манипуляций мнениями;
- выявление призывов к общественным/экстремистским акциям;
- обнаружение источников умышленной дезинформации.

Перечисленный перечень задач не является полным: поскольку проблемно-ориентированная среда в рамках концепции *iPSE* обладает открытой архитектурой, пользователи могут добавлять в базу пакетов собственные сервисы и самостоятельно пополнять репозиторий собственными композитными приложениями. Таким образом, это обеспечивает дальнейшее развитие и востребованность web-центра в области социодинамики и ее приложений.

Работа выполнена в рамках реализации Постановлений № 218 и 220 Правительства Российской Федерации при поддержке ФЦП „Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2012 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Mika P.* Social Networks and the Semantic Web (Semantic Web and Beyond). Springer, 2007. 234 p.
2. *Hu D., Kaza S., Chen H.* Identifying Significant Facilitators of DarkNetwork Evolution // *J. of the American Soc. for Inf. Sci. and Techn.* 2009. Vol. 60, N 4. P. 655—665.
3. *Hanneman R. A. and Riddle M.* Introduction to social network methods. Department of Sociology at the University of California, Riverside [Электронный ресурс]: <textbook available at <http://faculty.ucr.edu/~hanneman/nettext/>>.
4. *Бухановский А. В., Ковальчук С. В., Марьин С. В.* Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // *Изв. вузов. Приборостроение.* 2009. Т. 52, № 10. С. 5—24.
5. *Марьин С. В., Ковальчук С. В.* Сервисно-ориентированная платформа исполнения композитных приложений в распределенной среде // Там же. 2011. Т. 54, № 10. С. 21—29.
6. *Chau D. H., Pandit S., Wang S., Faloutsos C.* Parallel crawling for online social networks // *Proc. of 16th Intern. Conf. on World Wide Web – WWW.* 2007.

7. Князьков К. В., Ларченко А. В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 36—43.
8. Newman M. E. J. The Structure and Function of Complex Networks // Soc. for Industrial and Appl. Mathematics. 2003. Vol. 45, N 2. P. 167—256.
9. Daley D., Kendall D. Epidemics and rumours // Nature. 1964. Vol. 240, N 4963. P. 1118.

**Сведения об авторах**

- Сергей Владимирович Иванов** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: Sergey.v.ivanov@rambler.ru
- Екатерина Владимировна Болгова** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: katerina.bolgova@gmail.com
- Виктор Валерьевич Каширин** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: kashirin.victor@gmail.com
- Андрей Владимирович Якушев** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: yaja30@gmail.com
- Андрей Владимирович Чугунов** — канд. политич. наук; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, Центр технологий электронного правительства; директор; E-mail: chugunov@egov-center.ru
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор; E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 004.021, 004.492

Е. В. БОЛГОВА, С. В. ИВАНОВ, Е. А. ГРИНИНА,  
П. М. А. СЛОТ, А. В. БУХАНОВСКИЙ

## ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ ПРОЦЕССОВ НА КОМПЛЕКСНЫХ СЕТЯХ

Предложен способ формального описания динамики комплексных сетей. Рассмотрены схемы распараллеливания алгоритмов при моделировании динамических процессов в комплексных сетях.

**Ключевые слова:** комплексная сеть, динамический процесс, случайный граф, схема распараллеливания.

Комплексная сеть представляет собой граф с динамически эволюционирующей во времени структурой, при этом число состояний графа предполагается достаточно большим, чтобы интерпретировать закономерности процессов на нем средствами статистической механики [1]. Примерами комплексных сетей служат социальные сети (знакомств,



соавторства ученых [2]), информационные (цитирования в научных статьях [3], интернет-ссылок [4]), технологические (Интернет как сеть компьютеров, транспортные и электрические сети) и биологические (сети нейронов мозга, взаимодействующих протеинов, генетические сети). Развитие методов прямого моделирования процессов на комплексных сетях во многом обусловлено возможностями их реализации на параллельных вычислительных архитектурах, как для ускорения вычислительных операций, так и для работы со структурами данных большого объема.

В настоящей работе, продолжающей исследования авторов [5], предлагается формальный математический аппарат, позволяющий описывать динамические системы на основе моделей комплексных сетей в различных предметных областях, и рассматриваются аспекты параллельной реализации таких сетей.

**Микромасштабная вероятностная модель динамики комплексной сети.** Комплексная сеть представляется взвешенным неориентированным графом  $G$ , который определяется как совокупность  $(V, E)$  конечного множества вершин  $V$ ,  $\dim(V) = N$ , и множества ребер  $E$ , состоящего из неупорядоченных пар  $(u, v)$ , где  $u, v \in V$  и  $u \neq v$ . Каждая вершина характеризуется своей степенью, т.е. числом инцидентных ей ребер. Микромасштабная модель описывает динамику комплексной сети на основе знаний о состоянии каждой из вершин (и соответствующих им ребер) сети.

Комплексные сети от других графовых структур отличает возможность их вероятностного описания. Она не ограничивается частотным определением вероятности, пригодным для сетей с очень большим количеством узлов, но формально позволяет ввести вероятностное пространство  $(\Omega, B_\Omega, P_\Omega)$ , включающее в себя следующие элементы.

1.  $\Omega$  — пространство элементарных событий. Пусть  $V_i$  — множество всех вершин веса  $i$  (возможно, бесконечное), а  $E_{i,k}$  — множество всевозможных графов-звеньев, инцидентных паре вершин, одна из которых имеет вес  $i$ , а другая —  $k$ :  $E_{i,k} = \{e = \{u, v\} : u \in V_i, v \in V_k\}$ , тогда  $\Omega = \{v \in V_i, i = 1, \dots, N_1; e \in E_{i,k}, i, k = 1, \dots, N_1\}$ .

2.  $B_\Omega$  — сигма-алгебра подмножеств  $\Omega$ . Любой граф, содержащий вершины весов  $i = 1, \dots, N_1$  может быть составлен из элементов множества  $\Omega$  и соответственно рассмотрен как подмножество множества  $\Omega$ .

3.  $P_\Omega$  — сигма-аддитивная мера на множестве  $\Omega$  (вероятностная мера), отражающая вероятностные закономерности формирования топологии комплексной сети.  $P_\Omega(G) = p_G$  — вероятность того, что из всех возможных графов (элементов  $B_\Omega$ ) комплексная сеть представляется графом  $G$ .

Использование формального представления вероятностного пространства позволяет описывать динамические процессы на сетях даже в тех случаях, когда число узлов графа ограничено. В целом эволюция комплексной сети может быть описана посредством оператора  $\Gamma$  сети, действующего на множестве  $B_\Omega$ :

$$\begin{aligned} (V, E, f)_{t+1} &= \Gamma(V, E, f)_t, \\ (V, E, f)_{t=0} &= (V_0, E_0, f_0). \end{aligned} \tag{1}$$

Оператор  $\Gamma$  может быть представлен как совокупность нескольких различных операторов

$$\Gamma = \bigotimes_{m=1}^M \Gamma_m, \tag{2}$$

каждый из которых соответствует определенному динамическому компоненту эволюции сети, в общем случае это добавление новых вершин ( $\Gamma_1$ ), удаление из сети вершин ( $\Gamma_2$ ), добавление новых связей ( $\Gamma_3$ ), разрушение существующих связей, удаление ребер ( $\Gamma_4$ ). В общем случае эти операторы некоммутативны. Например, при эволюции комплексной сети по типу случайного графа

$$\Gamma = \Gamma_3 \otimes \Gamma_1 \otimes \Gamma_2 \otimes \Gamma_4. \quad (3)$$

Будем предполагать, что из сети на каждом шаге (в каждый момент времени) могут быть удалены только те вершины, которые не имеют инцидентных им ребер. При этом всякая вершина, не имеющая инцидентных ребер, может быть удалена из сети с вероятностью  $q_V$ , а любое ребро может быть удалено из сети с вероятностью  $q_E$ . Далее, пусть  $p_V$  — вероятность появления одной новой вершины в сети, а  $p_E$  — вероятность появления ребра, инцидентного двум произвольным вершинам. Компоненты эволюционного оператора (3) можно представить, используя описание комплексной сети как вероятностного пространства.

Пусть  $B_e$  — множество ребер, удаляемых из графа  $G$  в момент времени  $t$  ( $G_t$ ),  $B_e \subset B_\Omega$ . Тогда для оператора  $\Gamma_4$ , учитывая  $P(G_t \cap B_e) = q_E^{\#B_e}$ , получаем

$$\Gamma_4 = \langle G_t / (G_t \cap B_e), B_e \subset E_t | P(G_t \cap B_e) \rangle = \langle G_t / (G_t \cap B_e), B_e \subset E_t | q_E^{\#B_e} \rangle. \quad (4)$$

Здесь индекс # означает мощность множества.

Обозначим через  $B_v$  множество вершин, удаляемых из графа  $G_t$ ,  $B_v \subset B_\Omega$ . Тогда для оператора  $\Gamma_2$ , учитывая  $P(G_t \cap B_v) = q_V^{\#B_v}$ , получаем

$$\Gamma_2 = \langle G_t / (G_t \cap B_v), B_v \subset V_t | P(G_t \cap B_v) \rangle = \langle G_t / (G_t \cap B_v), B_v \subset V_t | q_V^{\#B_v} \rangle, \quad (5)$$

пусть теперь  $B_v$  — множество новых вершин, добавляемых в граф  $G_t$ ,  $B_v \subset B_\Omega$ , тогда оператор  $\Gamma_1$ , с учетом  $P(B_v) = p_V^{\#B_v}$ , имеет вид

$$\Gamma_1 = \langle G_t \cup B_v, B_v \subset V_t | P(B_v) \rangle = \langle G_t \cup B_v, B_v \subset V_t | p_V^{\#B_v} \rangle, \quad (6)$$

$B_e$  — множество новых ребер, добавляемых в граф  $G_t$ ,  $B_e \subset B_\Omega$ , оператор  $\Gamma_3$ , с учетом  $P(B_e) = p_E^{\#B_e}$ , имеет вид

$$\Gamma_3 = \langle G_t \cup B_e, B_e \subset E_t | P(B_e) \rangle = \langle G_t \cup B_e, B_e \subset E_t | p_E^{\#B_e} \rangle. \quad (7)$$

Дополнительно в (3) могут участвовать дополнительные операторы  $\Theta$ , отвечающие за эволюцию отдельных узлов сети (переход из состояния в состояние); могут использоваться операторы  $\Gamma_2$  и  $\Gamma_4$ .

В качестве примера на рис. 1 приведены три формы развития процессов в контактной сети носителей ВИЧ  $l \in N_1$  (○ — „здоровый“, ⊗ — „инфицированный“, ● — „больной/умерший“, ⊙ — новый узел, которого не было) [6]. Каждый узел может быть „инфицирован“; по истечении инкубационного периода он выбывает из сети; при этом соответствующие связи удаляются из сети. На рис. 1, а представлен процесс в „замороженной“ сети со статическими связями и постоянным количеством узлов. На рис. 1, б количество узлов сети постоянно, но связи между ними могут изменяться в течение времени вне зависимости от

действия оператора  $\Theta$ . На рис. 1, в сеть имеет переменное количество узлов и изменяемые связи между ними, т.е. задействованы все составляющие (3).

Модель (3)—(7) описывает эволюцию сети как случайного графа. Однако в ряде случаев такое приближение не является удовлетворительным. Это требует перейти от описания событий, определяемых (4)—(7) в форме модели Бернулли, к более сложным вероятностным схемам, например, с использованием условных распределений добавления (удаления) связи у вершины, которая уже имеет  $k$  связей (preferential attachment [7]).

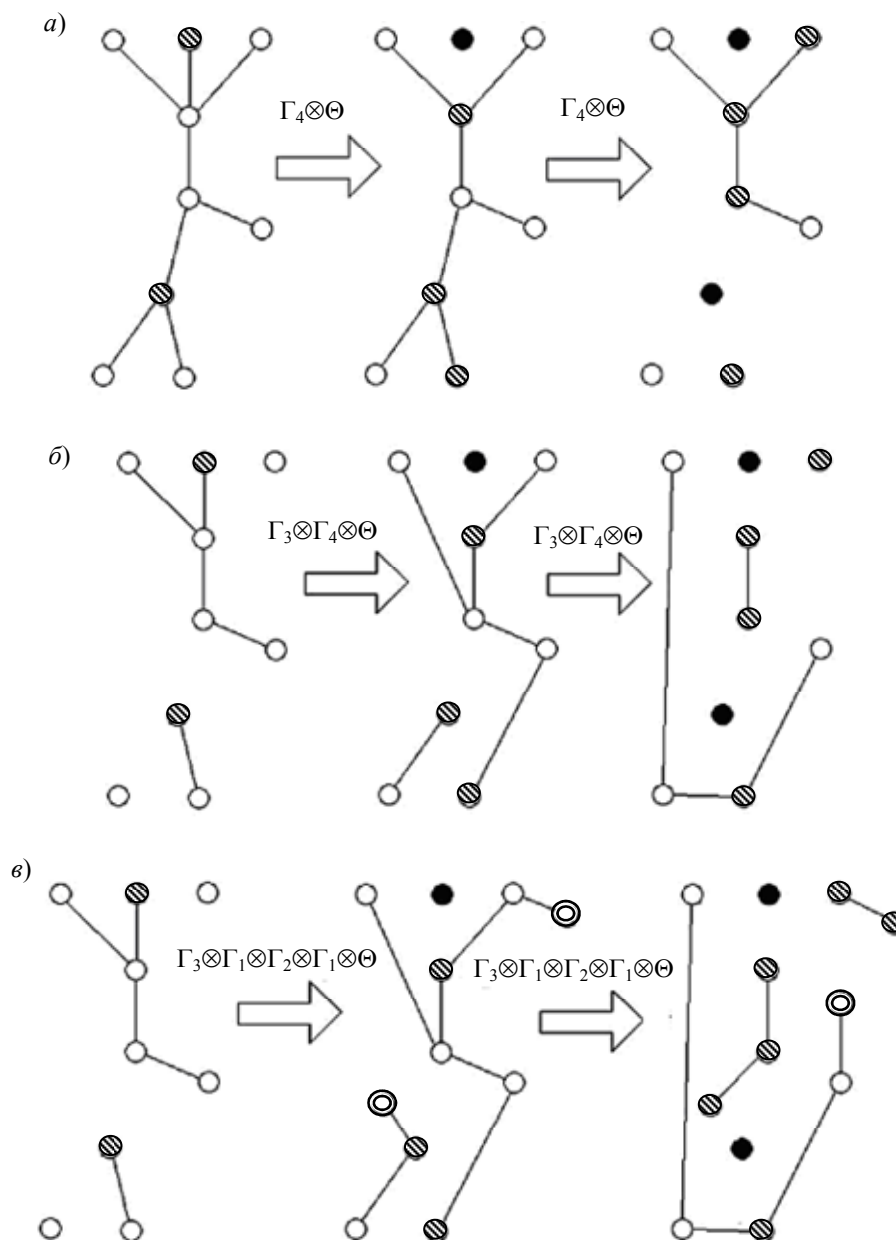


Рис. 1

**Макромасштабная вероятностная модель динамики комплексной сети.** Макроскопическая модель описывает динамику процесса в терминах вероятностных характеристик сети (общее количество вершин, общее количество ребер, количество вершин каждого конкретного веса), которые, в свою очередь, могут быть получены усреднением значений величин, характеризующих микроскопические свойства сети. Эти величины являются интегральными количественными характеристиками воздействия эволюционного оператора (2) на сеть.

Применяя к (1) оператор определения числа вершин заданного веса  $\rho_l = \rho_l(t)$  — количество вершин веса  $l$ ,  $l \in N_1$ , можно получить систему дифференциальных уравнений

$$\frac{d\rho_l(t)}{dt} = F(\rho_1(t), \rho_2(t), \dots, \rho_{N_1}(t), \lambda_1, \lambda_2, \dots, \lambda_p), \quad (8)$$

где  $\lambda_i$  — макропараметры, характеризующие процессы в конкретной сети, определяемые, например, через вероятности в соотношениях (4)—(7). Следует отметить, что сама по себе такая модель является детерминированной и не отражает стохастического характера процессов в комплексной сети.

Для приведенного выше примера эволюции сети (3)—(7) в форме случайного графа система (8) вырождается в единственное дифференциальное уравнение, описывающее зависимость  $\rho(t)$ , которая определяется балансом между количеством удаленных из сети вершин и количеством новых в каждый момент времени  $t$ . Всякая вершина сети, имеющая степень  $k$ , в результате действия оператора  $\Gamma_2 \otimes \Gamma_4$  удаляется из сети с вероятностью  $q_E^k q_V$ , тогда как под действием оператора  $\Gamma_3 \otimes \Gamma_1$  с вероятностью  $p_E^k p_V$  появляется новая, имеющая степень  $k$ :

$$\frac{d\rho(t)}{dt} = \rho(t) \sum_{k=0}^K (p_E^k p_V - q_E^k q_V) p_k(t), \quad (9)$$

где  $K$  — максимальная степень вершины в сети, а  $p_k = p_k(t)$  — закон распределения степеней.

В общем случае вид уравнений (8) зависит от характеристик распределения связей в структуре сети. Например, для эпидемиологической сети ВИЧ система (8) приобретает вид:

$$\left. \begin{aligned} \frac{ds_k(t)}{dt} &= -s_k(t) \sum_{l=1}^k f(l, \lambda) L(k, l, p), \\ \frac{d\rho_k(t)}{dt} &= -\mu \rho_k(t) + s_k(t) \sum_{l=1}^k f(l, \lambda) L(k, l, p), \\ \frac{dr(t)}{dt} &= \sum_k \mu \rho_k(t). \end{aligned} \right\} \quad (10)$$

Здесь  $s$  обозначает число индивидов,  $r$  — число изолированных от контактов (заболевших или умерших),  $\rho_k$  — количество инфицированных узлов, у которых есть ровно  $k$  связей. При этом  $L(k, l, p) = C_k^l p^l (1-p)^{k-l}$  задает биномиальное распределение контактов, где  $k$  — общее число узлов в группе, а  $l$  — доля узлов, подверженных заражению. Функция  $f(l, \lambda) = 1 - (1-\lambda)^l$  определяет вероятность изоляции узла (переход в группу  $r$ ). Здесь  $\lambda$  — вероятность изоляции, а  $\mu$  — вероятность заражения, интерпретируемые как макропараметры сети.

Модели в форме (8)—(10) являются более простыми с вычислительной точки зрения, однако менее гибкими, чем модель (1)—(7). Однако они позволяют анализировать основные характеристики процессов на сетях в терминах, присущих классическим динамическим системам. Например, на рис. 2 приведены результаты моделирования (кривая) развития эпидемии ВИЧ в США посредством модели (10), точки — результат измерений. На рис. 2, а, б представлено количество заболевших ( $r$ ) и фазовый портрет эпидемии в среде гомосексуального населения ( $dr$ ), а на рис. 2, в, г — в среде гетеросексуального населения. Из рисунка

видно, что фазовые портреты для различных популяций имеют принципиально разную форму, это является, в первую очередь, следствием различия структуры контактной сети.

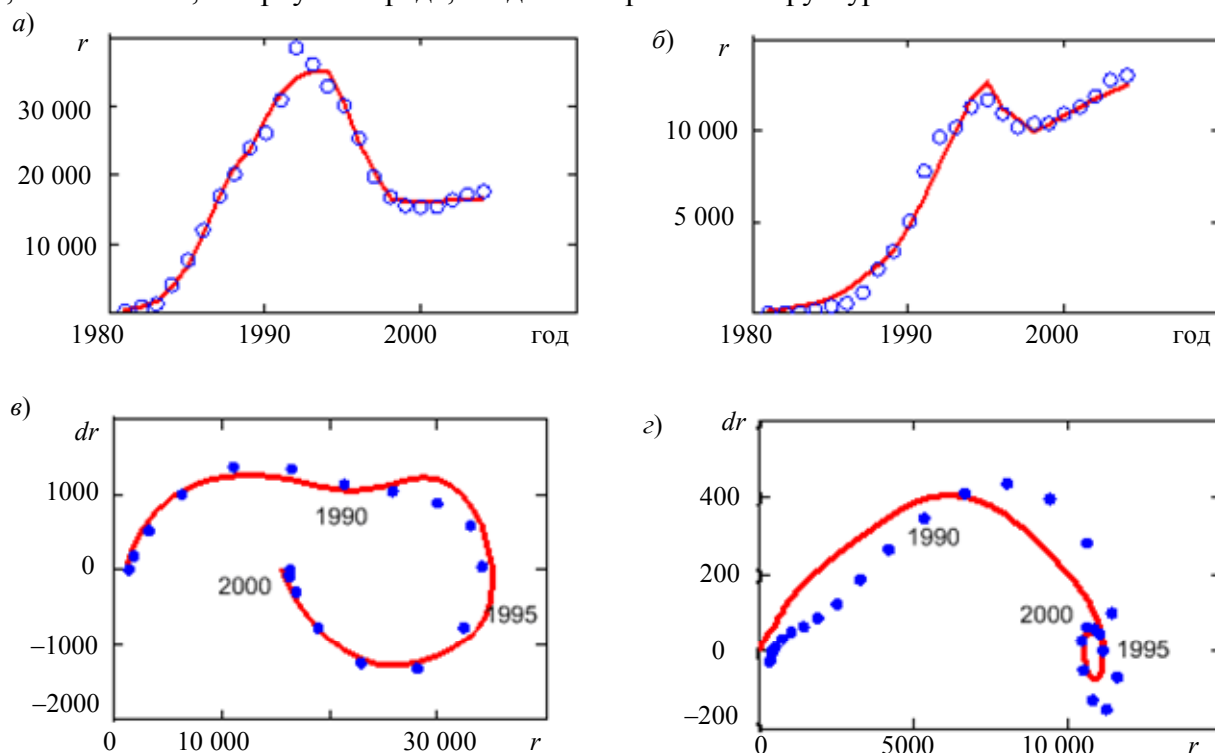


Рис. 2

**Параллельные алгоритмы моделирования процессов на комплексных сетях.** В последовательном алгоритме моделирования динамики комплексной сети по модели (1)—(7) условно можно выделить следующие этапы: формирование сети в начальный момент времени, эволюция сети, сбор и анализ необходимых данных (расчет макропараметров). Параллельные алгоритмы формирования сети заданной структуры подробно рассмотрены в работе [5]. Для моделирования эволюционной динамики сети можно рассмотреть следующие схемы распараллеливания для  $p$  вычислителей.

*Островная схема.* На каждом из вычислителей независимо формируется сеть размером  $N_i = N/p$ , после чего к ней применяется оператор (2). Результаты расчета макропараметров усредняются по всем  $p$  вычислителям. Такая схема отличается простотой реализации, однако имеет ограниченную эффективность, поскольку скорость сходимости статистических оценок (как следствие — параллельное ускорение) в данном случае будет пропорциональна  $\sqrt{p}$ . Однако ее применение может быть оправдано в том случае, когда макропараметры расчета варьируются от вычислителя к вычислителю (т.е. моделируется неэргодический процесс).

*Блочная схема.* Для применения оператора эволюции (2) необходимо сеть разделить на блоки (количество блоков  $2p$ ) с одинаковым числом узлов сети  $N_i = N/p$  в каждом, при этом любой из полученных блоков будет связан со всеми остальными. Применение оператора (2) осуществляется за одну итерацию для пары блоков на одном вычислителе, после чего происходит пересылка блоков между процессорами, этот процесс повторяется до тех пор, пока блоки не провзаимодействуют во всех возможных сочетаниях в соответствии с заранее построенной „турнирной“ таблицей. Реализация блочной схемы требует существенных накладных расходов на передачу данных между вычислителями на каждом шаге моделирования.

Для сопоставления производительности параллельных алгоритмов в рамках островной и блочной схем рассмотрим кривую изоэффективности, определяемую через равенство параллельных ускорений для обоих алгоритмов:

$$p(nm(1+\varphi) - 2N\bar{d}n\varphi) = N\bar{d} + 24N + nm. \quad (11)$$

Здесь  $N$  — количество вершин в сети,  $\bar{d}$  — средняя степень вершины,  $n$  — число шагов эволюционного алгоритма,  $m$  — количество макропараметров сети, подлежащих исследованию,  $\varphi = V_0\lambda$ , где  $\lambda$  — безразмерный коэффициент, определяющий отношение времени, затрачиваемого на пересылку единицы одного объема данных  $V_0$  между вычислителями, ко времени, затрачиваемому на одну арифметическую операцию. При этом оптимальное количество вычислителей, приводящих к наибольшему ускорению, вычисляется как

$$p_{\max}^{\text{б.с}} = \sqrt{\frac{\delta + n\tau}{2n\varphi}}, \quad p_{\max}^{\text{о.с}} = \sqrt{\frac{24N + N\bar{d}(1 + \delta + n\tau) + nm}{nm(1 + \varphi)}}, \quad (12)$$

$\delta$  — безразмерный параметр, характеризующий отношение времени, затрачиваемого на создание ребра, к времени на одну вычислительную операцию,  $\tau$  — безразмерный коэффициент, характеризующий отношение числа перестраиваемых вершин сети к общему количеству узлов  $N$  в сети. Из сопоставления (11) и (12) видно, что во всех случаях до достижения пика ускорения производительность островной схемы будет выше, чем у блочной. Однако этот приоритет является условным, поскольку при использовании островной модели не учитываются связи между отдельными подсетями, которые моделируются независимо друг от друга на различных вычислителях, число таких связей может быть  $N\bar{d}/p$ . Поэтому выбор схемы распараллеливания должен целиком определяться целями исследования и учитывать не только формальное ускорение, но и соответствие целей расчета предметной постановке задачи (например, моделирования связной сети не менее определенного размера).

**Заключение.** Динамически изменяющиеся комплексные сети являются эффективным инструментом описания сложных физических и социальных явлений. Для описания динамических моделей на базе комплексных сетей в работе предложен формальный математический аппарат, учитывающий основные аспекты сетевых моделей (стохастический характер сети, иерархию уровней моделирования и т.п.). Для проведения эффективных исследований сетевых моделей проанализированы основные подходы к распараллеливанию алгоритмов построения и эволюции комплексных сетей. В качестве примера для демонстрации удобства описания моделей на базе предложенного формализма и подтверждения эффективности параллельных алгоритмов была использована сетевая модель распространения инфекции ВИЧ.

Работа выполнена в рамках реализации Постановлений № 218 и 220 Правительства Российской Федерации при частичной поддержке ФЦП „Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2012 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Newman M. E. J. The Structure and Function of Complex Networks // SIAM Rev. 2003. Vol. 45, N 2. P. 167—256.
2. Redner S. How popular is your paper? An empirical study of the citation distribution // Europ. Phys. J. B. 1998. Vol. 4. P. 131—134.
3. Price D. J. de S. A general theory of bibliometric and other cumulative advantage processes // J. of the American Soc. for Information Sci. 1976. Vol. 27. P. 292—306.
4. Broder A., Kumar R., Maghoul F. et al. Graph structure in the web // Computer Networks. 2000. Vol. 33. P. 309—320.
5. Иванов С. В., Колыхматов И. И., Бухановский А. В. Параллельные алгоритмы моделирования комплексных сетей // Изв. вузов. Приборостроение. 2008. Т. 51, №10. С. 5—12.
6. Sloot P. M. A., Ivanov S. V. et al. Stochastic simulation of HIV population dynamics through complex network modeling // Intern. J. of Computer Mathematics. 2008. P. 1175—1187.

7. Barabási A.-L., Albert R. Emergence of scaling in random networks // Science. 1999. Vol. 286 (5439). P. 509—512.

**Сведения об авторах**

- Екатерина Владимировна Болгова** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: katerina.bolgova@gmail.com
- Сергей Владимирович Иванов** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: Sergey.v.ivanov@rambler.ru
- Екатерина Александровна Гринина** — канд. физ.-мат. наук; Санкт-Петербургский государственный университет информационных технологий, механики и оптики; E-mail: grinina@tprs.ru
- Петрус Мария Арнольдус Слоот** — доктор философии в области компьютерных наук, профессор; Университет города Амстердам; E-mail: p.m.a.sloot@uva.nl
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор; E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

---

---

# ПРОДВИЖЕНИЕ ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

---

---

УДК 004

А. В. БУХАНОВСКИЙ, С. В. МАРЬИН, К. В. КНЯЗЬКОВ, А. А. СИДНЕВ, С. Н. ЖАБИН,  
А. П. БАГЛИЙ, Р. Б. ШТЕЙНБЕРГ, А. В. ШАМАКИНА, В. В. ВОЕВОДИН,  
Е. Н. ГОЛОВЧЕНКО, Р. Т. ФАЛАЛЕЕВ, А. В. ДУХАНОВ, А. А. ТАРАСОВ,  
Л. В. ШАМАРДИН, А. И. МОИСЕЕНКО

## РЕЗУЛЬТАТЫ РЕАЛИЗАЦИИ ПРОЕКТА „МОБИЛЬНОСТЬ МОЛОДЫХ УЧЕНЫХ“ В 2010 ГОДУ: РАЗВИТИЕ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ ТЕХНОЛОГИИ iPSE И РАСШИРЕНИЕ СОСТАВА ПРИКЛАДНЫХ СЕРВИСОВ

Представлены результаты коллективных исследований в области применения и развития технологий проблемно-ориентированных распределенных инструментальных сред нового поколения для компьютерного моделирования сложных систем. Исследования выполнены в рамках мероприятия „Мобильность молодых ученых“ ФЦП „Кадры инновационной России на 2009—2013 гг.“ в 2010 г.

*Ключевые слова:* распределенные вычисления, компьютерное моделирование, сложная система.

**Введение.** Современный этап эволюции представлений о программных комплексах компьютерного моделирования и обработки данных научных экспериментов тесно связан с продвижением парадигмы „электронной науки“ (eScience). Понятие eScience ассоциируется с проблемой организации разнородными группами специалистов совместных научных исследований, требующих консолидации вычислительных и программных ресурсов для решения сложных междисциплинарных задач на основе технологий распределенных вычислений и систем. Инструментарий компьютерного моделирования, ориентированного на персональные компьютеры и вычислительные кластеры монопольного применения, в настоящее время претерпевает существенные изменения, связанные, в первую очередь, с глобализацией сферы его использования, ограниченной охватом сети Интернет. Это касается особенностей архитектуры программных комплексов, специфики их развертывания на вычислительных платформах, характеристик информационного обеспечения, поддержки жизненного цикла и продвижения на рынке [1].

Принципиальным аспектом развития и эффективного использования прикладных программных комплексов в рамках парадигмы eScience является введение единой технологической *проблемно-ориентированной* платформы, адаптируемой под специфику различных *предметных* областей. Характерным примером такой платформы является уникальная для Российской Федерации разработка — технология HPC-NASIS, заложенная в основу высокопроизводительного программного комплекса нового поколения для квантово-механических расчетов и моделирования наноразмерных структур и комплексов [2].



В настоящей статье рассматриваются результаты исследований, выполненных на основе технологии HPC-NASIS в рамках мероприятия „Мобильность молодых ученых“, проект „Развитие технологии проблемно-ориентированных распределенных инструментальных сред нового поколения для компьютерного моделирования сложных систем на основе высокопроизводительного программного комплекса HPC-NASIS“ в 2010 г. в рамках ФЦП „Кадры инновационной России на 2009—2013 гг.“.

**Технология iPSE как основа мультидисциплинарных распределенных исследований.** В отличие от большинства пакетов программ для моделирования сложных систем, комплекс HPC-NASIS основан на перспективном подходе к организации высокопроизводительного программного обеспечения сложных систем — технологии iPSE. Технология iPSE (Intelligent Problem Solving Environment) [3] строится на развитии традиционных проблемно-ориентированных оболочек и определяет принципы построения композитных приложений для eScience как интеллектуальной оболочки управления параллельными вычислительными процессами в распределенной иерархической среде, включающей в себя системы различной архитектуры. Такой подход позволяет расширить возможности традиционных проблемно-ориентированных оболочек за счет обеспечения эффективного параллельного исполнения композитных приложений, в силу того что использует для управления параллельными вычислениями симбиотические знания об особенностях предметной области и специфике процесса.

— В рамках iPSE формализуются не только методы и вычислительные алгоритмы, но и экспертные знания об изучении данного явления средствами компьютерного моделирования. Другими словами, iPSE реализует функции интеллектуальной системы поддержки принятия решений исследователя, что важно для практического внедрения такого комплекса.

— iPSE предоставляет единый интерфейс взаимодействия для предметно-ориентированных программных модулей и компонентов, которые могут разрабатываться различными коллективами и иметь разные условия распространения и использования.

— iPSE ориентирована на поддержку высокопроизводительных вычислений, причем не только для суперкомпьютерных систем с традиционной (кластерной) архитектурой, но и для неоднородных систем, например, гиперкластеров (суперкомпьютеров, объединенных высокоскоростным каналом). При этом управление эффективностью выполнения сценария является прерогативой iPSE, что позволяет избежать конфликтных ситуаций при разделении ресурсов между различными вычислительными модулями и разными пользователями.

Оболочка iPSE может быть отнесена к классу интеллектуальных систем, ориентированных на учет и формализацию неопределенности (информационная сложность), обладающих распределенной структурой (структурная сложность) и многоцелевым характером преобразования информации (функциональная сложность).

Концепция создания систем класса iPSE базируется на принципах, определяющих архитектуру системы и уровни ее управления. Наряду с традиционными математическими методами построения расписаний здесь широко применяются достижения искусственного интеллекта, определяющие принципы функционирования систем, основанных на знаниях. Важная роль принадлежит принципу открытости, позволяющему обеспечить наиболее сложные уровни иерархической структуры системы — самоорганизацию и самообучение, благодаря чему возможна интерпретация сложных процессов и явлений при анализе альтернатив и принятии решений, а также при моделировании действий системы в процессе решения задачи и „обучении“ на своем опыте.

Фактически программный комплекс на основе iPSE представляет собой интеллектуальную интегрирующую оболочку, которая позволяет встраивать, комбинировать и эффективно запускать предметно-ориентированные модули, решающие задачи, разработанные разными группами специалистов и функционирующие (как сервисы) на различных вычислительных системах в рамках

модели метакомпьютинга. Это обеспечивает открытость архитектуры комплекса и коммунальный характер его использования как ASP-приложения (Application Service Provider).

**Развитие функциональных компонентов технологии iPSE.** Программный комплекс HPC-NASIS представляет собой открытую информационную систему, динамично развиваемую усилиями разработчиков и виртуального сообщества пользователей [4]. Отдельные сервисы могут исполняться как на отдельных вычислительных узлах (серверах приложений или даже суперкомпьютерах), так и в распределенных вычислительных средах. Архитектура комплекса позволяет включать в его состав сервисы на основе систем следующих архитектур: SMP, MPP (кластеры), GPGPU (и кластеры на их основе), FPGA, а также СВЕА. Такие возможности в целом определяют пути дальнейшего развития комплекса и разработки его новых функциональных элементов в рамках концепции iPSE. В частности, элементы могут быть ориентированы на:

— обеспечение механизмов, позволяющих сторонним разработчикам удобно, эффективно и безопасно (как с технологической, так и с правовой точки зрения) интегрировать предметно-ориентированные модули в комплекс HPC-NASIS. Это включает выбор и адаптацию автоматизированного тестирования и отладки программных интерфейсов, средств для оптимизации структур хранилищ данных, а также средств виртуализации;

— развитие дополнительных системных инструментов в рамках технологии HPC-NASIS, повышающих надежность, реактивность, защищенность и эргономичность программных комплексов на ее основе.

В рамках исследований по проекту „Развитие технологии проблемно-ориентированных распределенных инструментальных сред нового поколения для компьютерного моделирования сложных систем на основе высокопроизводительного программного комплекса HPC-NASIS“ был спроектирован и реализован ряд экспериментальных образцов дополнительных программных компонентов HPC-NASIS, а также на основе результатов исследований показана их применимость и полезность.

— На основе анализа и развития программного компонента управления вычислительными платформами предложены и обоснованы принципы усовершенствования технологии мониторинга в составе HPC-NASIS, позволяющие расширить число контролируемых характеристик доступных платформ исполнения, реализовать более гибкий механизм предоставления актуальных данных о состоянии платформ, расширить возможности технической поддержки и частично автоматизировать ее, а также увеличить число доступных для интеграции стандартных средств мониторинга и управления платформами.

— Обоснована целесообразность тестирования проблемно-ориентированных модулей в HPC-NASIS по методу „белого ящика“ и „черного ящика“ (с варьированием входных данных сложной структуры), реализован набор соответствующих инструментов. Это позволит применять автоматическое тестирование программных сервисов при включении новых проблемно-ориентированных модулей в комплекс HPC-NASIS, тем самым расширяя его функциональность и круг решаемых с его помощью проблем.

— Предложены механизмы разработки и модификации высокопроизводительных программных сервисов, применяемые при организации работы с памятью ЭВМ для повышения производительности приложений. В частности, такие подходы могут быть применены при разработке новых сервисов в составе программного комплекса HPC-NASIS и модификации существующих.

— Предложены и обоснованы подходы и методы интеграции проблемно-ориентированных модулей в комплекс HPC-NASIS, которые основаны на парадигме Грид 2.0 и подходе REST для организации сервисной инфраструктуры. Это позволит расширять и модифицировать сервисную инфраструктуру HPC-NASIS с целью снижения накладных расходов на передачу данных (без использования протокола SOAP) и повышения прозрачности и простоты удаленных вызовов.

— Продемонстрирована возможность применения технологий облачных вычислений согласно нескольким сценариям (использование существующих облачных решений, предоставление SaaS или PaaS услуг). Сформулированы рекомендации для развития подсистем комплекса HPC-NASIS в соответствии с этими сценариями.

— Разработан механизм интеграции вычислительной инфраструктуры Национальной нанотехнологической сети (ННС) Грид в состав комплекса HPC-NASIS. Предложенное решение обеспечивает эффективный доступ к высокопроизводительным сервисам в рамках Грид-инфраструктуры, который отличается низкими накладными расходами, вносимыми управляющим ядром программного комплекса.

Проведенные исследования показали нетипичное распределение времени собственных накладных расходов среды ГридННС при испытании интегрирующего решения. На рис. 1 приведены ядерные оценки распределения: *a* — собственного времени исполнения пакета ORCA и *b* — общего времени исполнения композитного приложения в ГридННС (*1* — бензол, *2* — нафталин, *3* — пирен).

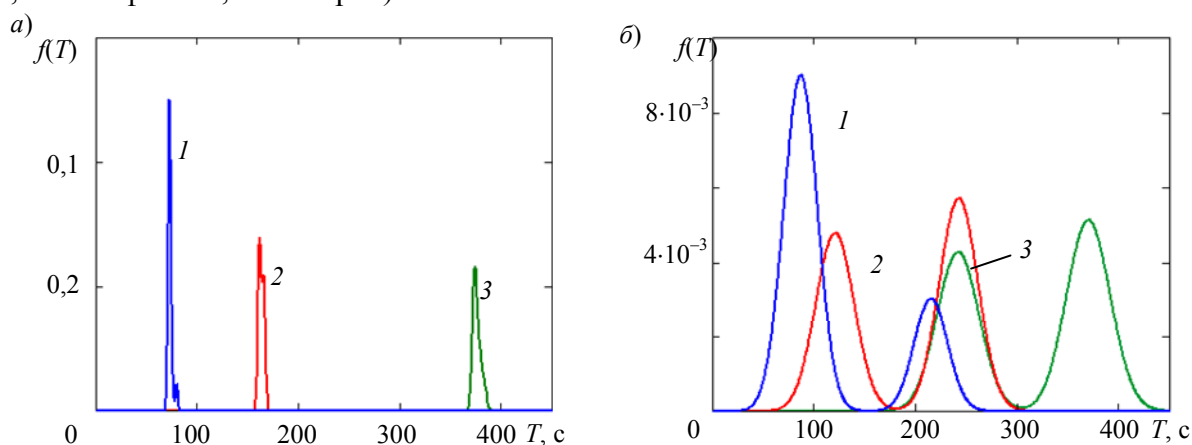


Рис. 1

На графике распределения времени исполнения в ГридННС хорошо заметны два отдельных пика, связанных с возникновением циклических процессов обработки очередей заданий в Грид. При этом накладные расходы сопоставимы с собственным временем исполнения пакета.

**Расширение состава прикладных сервисов HPC-NASIS.** Отдельным направлением исследований и разработок является адаптация технологии HPC-NASIS для решения новых классов задач как в рамках заданной предметной области (моделирование наноразмерных атомно-молекулярных систем и комплексов), так и в других предметных областях (пилотные решения). В частности, в рамках данного проекта разработан экспериментальный образец молекулярного редактора в форме сервиса в комплексе HPC-NASIS. Основная задача сервиса заключается в предоставлении возможностей визуализации и редактировании молекул и их систем. Молекулярный редактор в составе HPC-NASIS делает подготовку данных для расчетов более удобной, а также улучшает восприятие решения самой задачи расчета молекулярной структуры.

На рис. 2 представлена молекулярная поверхность большой молекулы белка, изображение построено с помощью разработанного для HPC-NASIS молекулярного редактора.

— Обоснованы возможности комплекса HPC-NASIS в области интеграции пакетов инженерного проектирования и анализа, разработано программное решение, которое позволяет строить вычислительные цепочки из нескольких сервисов, реализующих процесс исследования наноматериалов и сред с переходом от микро- к макроуровням. Для этого в состав HPC-NASIS встроен сервис, обеспечивающий доступ к возможностям свободно распространяемого пакета OpenFOAM.

— Разработан и реализован в качестве прикладного сервиса HPC-NASIS сопоставимый с лучшими мировыми аналогами параллельный алгоритм геометрической декомпозиции тетраэдральных сеток. Преимущества технологии HPC-NASIS для решения данной задачи заключаются в возможности составления экспериментальной базы разбиений различных сеток для различных вычислительных задач с использованием экспертной системы, помогающей в настройке параметров под конкретную задачу. Интеграция в комплекс HPC-NASIS различных пакетов решения вычислительных задач наравне с пакетом параллельной декомпозиции сеток позволит создавать сценарии цепочек запуска различных пакетов для решения комплексных ресурсоемких задач.

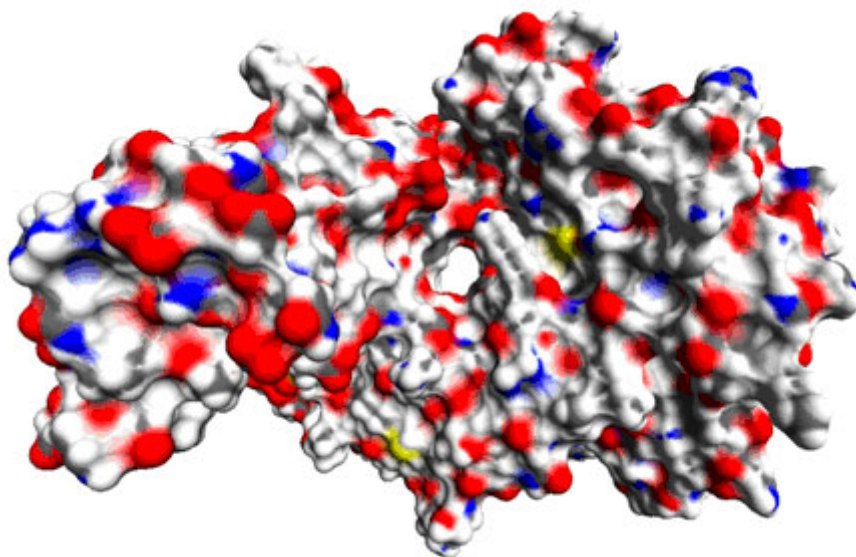


Рис. 2

— Разработан и программно реализован в качестве прикладного сервиса HPC-NASIS параллельный алгоритм оптимизации многошаговых управляющих воздействий в наносистемах. В качестве примера расчетов была выбрана задача распределения ресурсов в целях оптимизации эффективности нанотехнологического процесса, которая показала хорошую масштабируемость алгоритма.

— Рассмотрен вариант решения проблемы поиска сферических кодов с использованием ресурсов комплекса HPC-NASIS и математической среды MathCloud, выявлена возможность интеграции указанных сред в рамках решения поставленной задачи, обеспечивающей унифицированный доступ к широкому кругу доступных вычислительных платформ.

**Заключение.** В рамках проекта „Развитие технологии проблемно-ориентированных распределенных инструментальных сред нового поколения для компьютерного моделирования сложных систем на основе высокопроизводительного программного комплекса HPC-NASIS“ в 2010 г. работы по развитию технологии HPC-NASIS выполнялись с привлечением двенадцати молодых приглашенных специалистов (преподавателей, аспирантов и научных сотрудников). Так, исследования в области механизмов, методов и технологий адаптации сторонних программных модулей выполнены специалистами ЮУрГУ и МФТИ и ЮФУ. Работы по развитию дополнительных системных инструментов проведены специалистами ННГУ, ЮФУ и НИВЦ МГУ. Разработки в области адаптации технологии HPC-NASIS к решению новых классов прикладных задач связаны с деятельностью специалистов ИММ РАН, НИВЦ МГУ, ВГУ и ИСА РАН.

Результаты проведенных исследований в целом продемонстрировали масштабируемость и востребованность технологии HPC-NASIS для создания проблемно-ориентированных платформ компьютерного моделирования в различных предметных областях; при этом

научно-технический уровень достигнутых результатов соответствует современным мировым образцам программного инструментария eScience.

Работа выполнена при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“. Результаты проекта использованы в рамках выполнения государственного контракта № 16.647.12.2025 „Создание функционирующего в режиме удаленного доступа интерактивного учебно-методического комплекса для выполнения работ в области моделирования наноразмерных атомно-молекулярных структур, наноматериалов, процессов и устройств на их основе, в распределенной вычислительной среде“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бухановский А. В., Васильев В. Н. Современные программные комплексы компьютерного моделирования e-Science // Изв. вузов. Приборостроение. 2010. Т. 53, № 3. С. 60—64.
2. Ядро высокопроизводительного программного комплекса для квантово-механических расчетов и моделирования наноразмерных атомно-молекулярных систем и комплексов „HPC-NASIS“. Свид-во о гос. регистрации программы для ЭВМ № 20010610161 от 11.01.2010 г.
3. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.
4. Гуськов А. А., Дунаев А. В., Подтелкин Ф. В., Бухановский А. В. Интеллектуальные технологии поддержки пользователей в области компьютерного моделирования наноразмерных структур и наноматериалов // Науч.-техн. вестн. СПбГУ ИТМО. 2010. Вып. 66. С. 120.

#### Сведения об авторах

- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор; E-mail: avb\_mail@mail.ru
- Сергей Владимирович Марьин** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: sm.niinkt@gmail.com
- Константин Валерьевич Князьков** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник; E-mail: constantinvk@gmail.com
- Алексей Александрович Сиднев** — Нижегородский государственный университет им. Н. И. Лобачевского, кафедра математического обеспечения ЭВМ; ассистент; E-mail: sidnev.a.a@gmail.com
- Сергей Николаевич Жабин** — Московский физико-технический институт, кафедра общей физики; ассистент; E-mail: delafrag@gmail.com
- Антон Павлович Баглий** — аспирант; Южный федеральный университет, факультет механики, математики и компьютерных наук, Ростов-на-Дону; E-mail: taccessviolation@yandex.ru
- Роман Борисович Штейнберг** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; научный сотрудник; E-mail: romanofficial@yandex.ru
- Анастасия Валерьевна Шамакина** — Южно-Уральский государственный университет, отдел распределенных вычислений Лаборатории суперкомпьютерного моделирования, Челябинск; начальник; E-mail: sham2004@bk.ru
- Вадим Владимирович Воеводин** — НИВЦ МГУ им. М. В. Ломоносова, лаборатория параллельных информационных технологий, Москва; младший научный сотрудник; E-mail: vadim@parallel.ru
- Евдокия Николаевна Головченко** — Институт прикладной математики им. М. В. Келдыша РАН, отдел № 16, Москва; младший научный сотрудник; E-mail: ge03@imamod.ru

- Роман Тимофеевич Фалалеев* — Московский физико-технический институт, кафедра информатики; зав. кафедрой; E-mail: roman@phystech.edu
- Алексей Валентинович Духанов* — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; E-mail: avd\_mail@mail.ru
- Алексей Александрович Тарасов* — канд. физ.-мат. наук; Московский физико-технический институт, кафедра распределительных вычислений; доцент; E-mail: tarasov@isa.ru
- Лев Витальевич Шамардин* — Научно-исследовательский институт ядерной физики им. Д. В. Скобельцына, Москва; старший научный сотрудник; E-mail: shamardin@theory.sinp.msu.ru
- Алексей Игоревич Моисеенко* — ЗАО „Фирма АйТи. Информационные технологии“, Департамент технологий информационного менеджмента, Москва; директор; E-mail: AMoiseenko@it.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

---

---

## КРАТКИЕ СООБЩЕНИЯ

---

---

УДК 681.3

Б. Я. ШТЕЙНБЕРГ, Е. Н. КРАВЧЕНКО, Р. И. МОРЫЛЕВ, З. Я. НИС, В. В. ПЕТРЕНКО,  
И. С. СКИБА, В. Н. ШАПОВАЛОВ, О. Б. ШТЕЙНБЕРГ, Р. Б. ШТЕЙНБЕРГ

### ОСОБЕННОСТИ РЕАЛИЗАЦИИ РАСПАРАЛЛЕЛИВАЮЩИХ ПРЕОБРАЗОВАНИЙ ПРОГРАММ В СИСТЕМЕ ДВОР

Обсуждаются особенности функционирования библиотеки преобразований программ в диалоговом высокоуровневом оптимизирующем распараллеливателе.

*Ключевые слова:* распараллеливание, преобразование программ, оптимизация.

Диалоговый высокоуровневый оптимизирующий распараллеливатель программ (ДВОР) является развитием Открытой распараллеливающей системы [1]; он позволяет работать с различными языками программирования, поддерживает общую библиотеку и возможность диалоговой компиляции. Для реализации этих функций ДВОР использует высокоуровневое внутреннее представление программы, в котором проводятся оптимизирующие (распараллеливающие) преобразования, что выгодно отличает его от существующих аналогов [2, 3]. Представление является иерархическим, с разной детализацией программы на каждом уровне. Верхний уровень для каждого входного языка индивидуален и ориентирован на удобство разбора программы. Второй уровень, общий для всех входных языков, ориентирован на преобразования программ. На этом уровне разобранная программа хранится в виде четырех деревьев: идентификаторов, типов, операторов и выражений, что удобно для диалоговой формы преобразований.

Диалоговое распараллеливание является компромиссом между ручным и автоматическим. Программисту предоставляются автоматизированные инструменты анализа и преобразования программ, ускоряющие распараллеливание по сравнению с ручной обработкой. Использование автоматизированных инструментов повышает надежность кода и снижает требования к квалификации и опыту программиста, поскольку в диалоговом режиме компиляции возможно запросить недостающую информацию для эффективного распараллеливания программы. Например, для определения информационных зависимостей в программе могут быть полезны диапазоны значений переменных, размеры массивов данных, соотношения между различными переменными, которые в ряде случаев в программе явно не определены.

В целях распараллеливания ДВОР позволяет распознавать характерные фрагменты кода, выполнять над ними соответствующие преобразования, а также осуществлять проверку сохранения синтаксиса и семантики, эквивалентности преобразования данных. Также оцениваются целесообразность преобразования, а при наличии нескольких вариантов — выбор оптимального способа преобразования для заданного фрагмента. Проверки могут выполняться автоматически и в режиме диалога. Для контроля эквивалентности преобразований кроме традиционного графа информационных связей используется решетчатый граф программы [4].

ДВОР поддерживает такие преобразования, как линеаризация выражений, гнездование и разрезание циклов, разрезание условных операторов. При линеаризации выражений в качестве

базовых переменных кроме счетчиков циклов могут выступать и внешние переменные (которые не изменяют своего значения в преобразуемом фрагменте), и имена массивов с различными индексами (при распараллеливании рекуррентных циклов). При линеаризации на этапе компиляции приводятся подобные *члены* с учетом типов данных. Гнездование циклов используется при распараллеливании цикла на параллельной архитектуре класса MIMD. ДВОР поддерживает процедуру проверки структуры преобразуемого фрагмента на возможность применения преобразования гнездования. При данной проверке система получает фрагмент программы и список допустимых в нем единиц языка (список допустимых типов операторов и т.д.) и проверяет, содержатся ли в данном фрагменте только единицы языка из списка. При такой форме проверки в случае последующего расширения внутреннего представления новыми операторами система останется работоспособной без изменения преобразования.

Разрезание циклов в системе ДВОР использует граф информационных связей, по которому строится фактор-граф. В полученном фактор-графе строится правильная нумерация. Далее проводятся анализ применения вспомогательных преобразований и непосредственно разрезание на части, следуемые в порядке правильной нумерации. Преобразование разрезания цикла может иметь разные вариации (например, разрезание в указанном месте или на максимальное количество частей). Разрезание условного оператора заменяет один условный оператор двумя подряд идущими условными операторами, при этом блок ветки *then* исходного условного оператора разбивается на части и каждая из частей становится блоком одного из результирующих условных операторов.

Дополнительно в ДВОР поддерживается расширение скалярной SSA-формы (Static Single Assignment Form, форма со статически однократным присваиванием) [5] на код с массивами. SSA-форма для массивов строится с помощью нечеткого анализа потока данных (Fuzzy Array Dataflow Analysis) [6]. В SSA-форме для массивов некоторые скалярные преобразования расширяют область применения: удаление мертвого кода, протягивание констант, использование общих подвыражений, распределение регистров и др.

Работа выполнена в рамках ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Открытая распараллеливающая система [Электронный ресурс]: <<http://ops.rsu.ru/>>.
2. [Электронный ресурс]: <<http://www.csrd.uiuc.edu/promis/>>.
3. [Электронный ресурс]: <<http://www-suif.stanford.edu/research/>>.
4. Воеводин В. В. Воеводин Вл. В. Параллельные вычисления. СПб: БХВ-Петербург, 2002. 599 с.
5. Muchnick S. S. Advanced Compiler-Design and Implementation. Morgan Kaufmann Publishers, 1997. 860 p.
6. Collard J.-F., Barthou D., Feautrier P. Fuzzy Array Dataflow Analysis // ACM SIGPLAN Notices. 1995. Vol. 30, Iss. 8.

#### Сведения об авторах

- Борис Яковлевич Штейнберг** — д-р техн. наук; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; зав. кафедрой; E-mail: borsteinb@mail.ru
- Евгений Николаевич Кравченко** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: peon\_sxe@mail.ru
- Роман Игоревич Морылев** — аспирант; Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; E-mail: frg10@yandex.ru
- Зиновий Яковлевич Нис** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: irishrover@mail.ru



- Виктор Владимирович Петренко** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; научный сотрудник;  
E-mail: vpetrenko@gmail.com
- Иван Сергеевич Скиба** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; программист; E-mail: aidan@pisem.net
- Василий Николаевич Шаповалов** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; младший научный сотрудник;  
E-mail: Vasily.Shapovalov@gmail.com
- Роман Борисович Штейнберг** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; научный сотрудник;  
E-mail: romanofficial@yandex.ru
- Олег Борисович Штейнберг** — Южный федеральный университет, кафедра алгебры и дискретной математики, Ростов-на-Дону; ассистент;  
E-mail: romanofficial@yandex.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 004.021, 004.252, 004.75

Д. А. ФАДЕЕВ

## ОСОБЕННОСТИ ЧИСЛЕННОГО РЕШЕНИЯ ЭВОЛЮЦИОННЫХ ЗАДАЧ РАСПРОСТРАНЕНИЯ КОРОТКИХ ЛАЗЕРНЫХ ИМПУЛЬСОВ НА СИСТЕМАХ АРХИТЕКТУРЫ NUMA

Рассмотрены приемы адаптации алгоритмов решения эволюционных задач распространения коротких лазерных импульсов к особенностям вычислительной архитектуры NUMA.

**Ключевые слова:** NUMA, прогонка, волновое уравнение, сверхкороткий импульс.

Архитектура NUMA (Non Uniform Memory Access) определяет класс параллельных вычислительных систем с логически общей, но физически раздельной памятью [1]. Существует достаточно много вариантов реализации такой архитектуры. В частности, компания Advanced Micro Devices ввела подход, при котором чипы RAM подключаются непосредственно к чипу CPU, содержащему встроенный контроллер памяти. В случае использования MultiCPU-платформ каждый CPU подключается к своему набору планок RAM. При этом память остается логически общей благодаря наличию шины HyperTransport, имеющей существенно больший пропускной канал, чем связка CPU-RAM. Более того, серверные процессоры AMD имеют  $N-1$  HyperTransport каналов ( $N$  — число процессоров (чипов) в одном вычислительном узле (материнской плате)) и каждый чип CPU соединен с каждым. Таким образом, достигается реализация логически общей памяти. При этом скорость доступа данного CPU к памяти любого нода (связки CPU-RAM) одинакова, если требуемая память не занята другим запросом. Таким образом, в NUMA ширина канала увеличивается во столько же раз, сколько нодов имеется на вычислительном узле. В настоящее время архитектура NUMA поддерживается и на персональных компьютерах, в частности, использующих платформу Nehalem [2].

В настоящей статье рассматриваются особенности реализации задачи эволюционного (по координате  $z$ ) численного моделирования распространения лазерного импульса  $A(z, \tau, r)$  с широким спектром:

$$\frac{2}{c} \frac{\partial^2 A}{\partial \tau \partial z} + \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial A}{\partial r} \right) = 0. \quad (1)$$

Такое соотношение может быть получено из уравнений Максвелла для условий малоуглового распространения короткого импульса вдоль оси  $z$ , где  $\tau = t - z/c$  — сопутствующее время. В фурье-пространстве уравнение (1) перейдет в следующее:

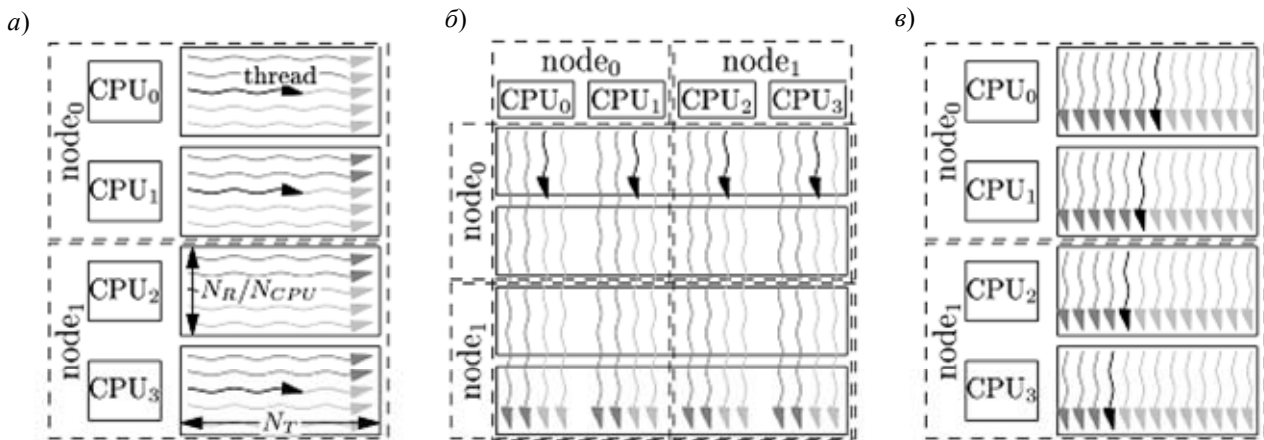
$$-\frac{2}{c} i\omega \frac{\partial B}{\partial z} + \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial B}{\partial r} \right) = 0, \quad (2)$$

которое, в свою очередь, аппроксимируется посредством  $N_T$  независимых конечно-разностных уравнений:

$$-i\omega_n \frac{\partial B}{\partial z} + \hat{L}B = 0, \quad n \in [0, N_T - 1], \quad (3)$$

где оператор  $\hat{L}$  аппроксимирует дифференциальный оператор в соотношении (2). Данный оператор может быть записан в трехдиагональном виде как обобщение обычного двумерного пятиточечного лапласиана. Уравнение (3) решается численно методом Крэнка—Никольсона для каждой гармоники  $\omega_n$ .

С точки зрения эффективного использования архитектуры NUMA элементы массива  $\mathbf{A}(r_i, \tau_j)$  целесообразно расположить так, чтобы соседние элементы по второй (временной) координате оказались соседними в памяти. Это удобно для выполнения преобразования Фурье, которое при таком размещении данных может выполняться построчно. Пусть далее в системе есть  $N_{\text{NUMA}}$  нодов, тогда все массивы  $\mathbf{A}$ ,  $\mathbf{B}$  размером  $N_R \times N_T$  разбиваются на  $N_{\text{NUMA}}$  отдельных частей размером  $(N_R/N_{\text{NUMA}}) \times N_T$ . На рисунке, *a* представлено параллельное выполнение преобразования Фурье; *b* — первая часть прогонки (прямая прогонка) с конфликтом памяти; *в* — прямая прогонка без конфликта памяти. Для выполнения преобразования Фурье по строкам достаточно запустить  $N_{\text{NUMA}}$  независимых нитей — thread (*a*). В соответствии с предложенным расположением массивов в памяти соседние по  $r$  элементы оказываются удалены на  $N_T$  элементов в памяти. Более того, элементы из разных подмассивов оказываются на разных нодах, поэтому простейший цикл по столбцам матрицы  $\mathbf{B}$ , реализуемый при выполнении прогонки, оказывается неоптимальным, так как при этом физические процессоры будут синхронно обращаться к памяти одного нода, как это представлено на рисунке, *б*.



Выходом в данном случае может быть „лестничная“ процедура, представленная на рисунке, *в*. Процесс прямой прогонки (обнуление одной из боковых лент матрицы обрабатываемого оператора) можно начинать выполнять на одном ядре (CPU<sub>0</sub>), после того как элементы

$\mathbf{V}(r_0, \tau_0), \dots, \mathbf{V}(r_{N_R/N_{CPU}}, \tau_0)$  обработаны, можно переходить к обработке элементов  $\mathbf{V}(r_0, \tau_1), \dots, \mathbf{V}(r_{N_R/N_{CPU}}, \tau_1)$ , а CPU<sub>1</sub> может начать обработку элементов  $\mathbf{V}(r_{N_R/N_{CPU}}, \tau_0), \dots, \mathbf{V}(r_{2N_R/N_{CPU}-1}, \tau_0)$ . Таким образом, все члены CPU будут задействованы с некоторой задержкой. Однако большую часть данных ядра будут обрабатывать одновременно, не обращаясь к данным чужого нода. Если окажется что блок памяти, привязанный к данному ядру, помещается в собственный кэш ядра то конфликт между CPU<sub>0</sub>, CPU<sub>1</sub> и CPU<sub>2</sub>, CPU<sub>3</sub> будет также разрешен. Вторая часть (обратная прогонка) выполняется так же, только процесс начинается с последнего вычислительного ядра.

Для практической реализации предложенной выше процедуры необходима некоторая модификация. Так как обращение к любому элементу памяти, отсутствующему в кэше, вызывает считывание целой кэш-линии первого уровня (L1 cache line), следует выполнять модифицированную прогонку, реализующую один шаг с номером  $i$  сразу для 16 элементов ( $\mathbf{A}(r_i, \tau_j), \dots, \mathbf{A}(r_i, \tau_{j+16})$ ). При этом время ожидания в начале предложенного „лестничного“ процесса несколько возрастает, однако если выполнять прогонку для каждого столбца отдельно, неоптимальное обращение к памяти приведет к еще большей потере производительности.

Дополнительной особенностью моделирования распространения лазерного импульса является необходимость оптимизации распределения нагрузки, поскольку в области слабых полей вычисления можно не производить. Но в таком случае возникает простой отдельных вычислительных узлов на каждом эволюционном шаге. Одним из способов создания равномерной нагрузки может быть варьирование размеров подмассивов, рассчитываемых на отдельных CPU. При этом размер массива  $N_T$  по координате  $\tau$  остается прежним, изменяется только количество строк  $N_{Rk}$  в подмассиве (исходно  $N_{Rk} = N_R / N_{CPU}, \forall k$ ). Так как при каждом изменении размера подмассива выделяются и высвобождаются большие объемы памяти, данную операцию следует производить через некоторое, достаточно большое, число эволюционных шагов.

Рассмотренные подходы являются достаточно перспективными благодаря разработке NUMA-систем. На данный момент на рынке широко представлены системы с максимальным числом нодов 2—4. В условиях увеличения числа ядер на один NUMA-нод (до 12 в последних предложениях от AMD) эти системы остаются, на наш взгляд, по-прежнему несбалансированными по соотношению вычислительная производительность—пропускная способность памяти. Эксперименты с вычислительной станцией на базе Intel Nehalem 2 CPU X5550 показали, что увеличение каналов памяти (3 канала в Nehalem) не оправдывает заявленной производительности и при этом не допускает контроля со стороны программиста. В этом смысле использование особенностей NUMA более эффективно, чем применение в рамках модели глобальной общей памяти, разделяемой между ядрами.

#### СПИСОК ЛИТЕРАТУРЫ

1. A NUMA API for LINUX\* 2005 [Электронный ресурс]: <developer.amd.com/assets/LibNUMA-WP-fv1.pdf>.
2. [Электронный ресурс]: <http://download.intel.com/pressroom/pdf/nehalem-ex.pdf>.

#### Сведения об авторе

**Даниил Александрович Фадеев**

— Институт прикладной физики РАН, отдел нелинейной электродинамики, Нижний Новгород; младший научный сотрудник;  
E-mail: fadey@appl.sci-nnov.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

Д. К. БОГОЛЕПОВ, Д. П. СОПИН, Д. Я. УЛЬЯНОВ, В. Е. ТУРЛАПОВ

## ПОСТРОЕНИЕ SAH BVH ДЕРЕВЬЕВ ДЛЯ ТРАССИРОВКИ ЛУЧЕЙ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

Разработан алгоритм построения SAH BVH деревьев на графических процессорах в режиме реального времени. Алгоритм позволяет более чем на порядок увеличить производительность визуализации по сравнению с аналогами.

**Ключевые слова:** трассировка лучей, ускоряющие структуры, SAH, BVH, динамические сцены, графический процессор, OpenCL.

Метод трассировки лучей традиционно используется в компьютерной графике для синтеза изображений высокого качества. Первые реализации этого метода на графических процессорных устройствах (ГПУ) допускали обработку сцен только со *статической* геометрией (см., например, [1]).

В настоящей работе рассматривается подход к визуализации *динамических* сцен с использованием ускоряющей структуры в форме деревьев BVH. Данная структура обеспечивает наиболее „плотную“ аппроксимацию геометрии сцены при минимальном числе узлов; в процессе построения дерева используются только *центроиды* треугольников, что исключает случай пересечения треугольником плоскости разбиения. Производительность процесса трассировки лучей напрямую зависит от *качества* построения дерева, наилучшим критерием которого является *эвристика площадей поверхности* (SAH) [2]. На данный момент, по-видимому, отсутствуют эффективные реализации алгоритмов построения SAH BVH деревьев на ГПУ, которые обеспечили бы возможность визуализации произвольных динамических сцен [3].

Целью настоящей статьи является разработка эффективного алгоритма построения SAH BVH деревьев на ГПУ в режиме *реального* времени. Предполагается, что все объекты сцены представлены набором треугольников (*triangle soup*), при этом допускаются произвольные изменения геометрии во время визуализации. Приведем общую схему алгоритма построения SAH BVH дерева.

1. На каждом шаге выполняются все доступные задания. В результате возможно наиболее полно использовать ресурсы ГПУ, а также снизить накладные расходы, связанные с передачей заданий с центрального процессора.

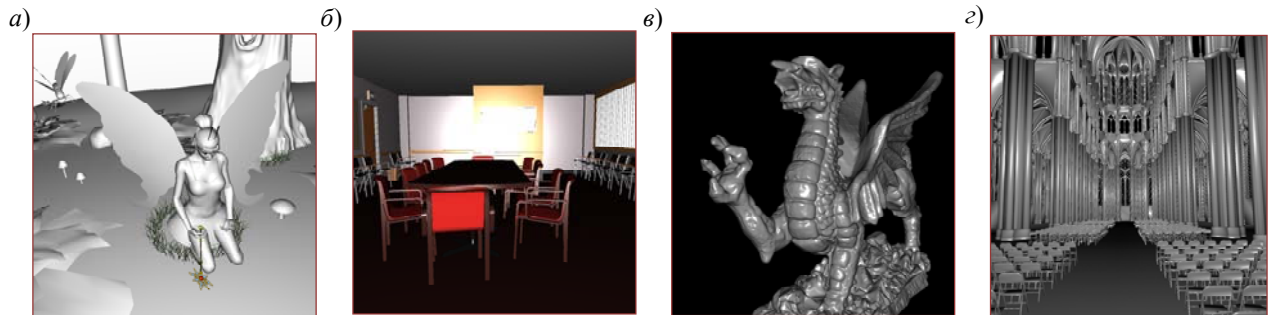
2. Для обработки узлов с достаточно большим числом геометрических примитивов (в данном случае — более 256) используются несколько групп заданий на узел (в данной работе —  $N/512$ , где  $N$  — число геометрических примитивов в узле). В сочетании с оптимизацией из пункта (1) такой подход обеспечивает использование ресурсов, близкое к оптимальному.

3. Построение дерева включает 3 этапа: обработка больших узлов (более 32К примитивов), обработка средних узлов (от 256 до 32К примитивов) и обработка малых узлов (менее 256 примитивов). Такой подход позволяет использовать специальные модификации алгоритма на каждом уровне, в результате чего удается сократить накладные расходы и оптимизировать использование ресурсов ГПУ.

В целом на центральном процессорном устройстве происходит только отправка существующих заданий и генерация новых, все этапы выполнения алгоритма реализуются на ГПУ.

Экспериментальные исследования производительности алгоритма были проведены на примере тестовых сцен (см. рисунок), характеризующихся различной вычислительной сложностью обработки изображения (содержат от 10К до 3М треугольников). Поскольку основная цель исследований связана с ускоряющими структурами, для задач визуализации в вычислительных экспериментах использовалась классическая трассировка лучей Уиттеда. Для обхода

дерева BVH использовался классический алгоритм на основе полного стека (в частной памяти каждого потока), который позволяет оптимизировать вычисления за счет обработки дерева в порядке от ближних узлов к дальним.



Исследования проводились на компьютере с ГПУ NVIDIA GeForce GTX 480 под управлением операционной системы Ubuntu Linux 11.04 (с версией видеодрайвера 270.41.06). В таблице полученные результаты сравниваются с данными других работ.

Результаты экспериментальных исследований  
производительности алгоритма построения SAH BVH деревьев на ГПУ

Тестовая сцена		Lauterbach [3]		Wald [4]		Разработанный алгоритм	
Fairy Forest (a)	174K	488 мс	21,7 Флоп	31 мс	29 Флоп	40 мс	25 Флоп
Conference (б)	284K	477 мс	24,5 Флоп	42 мс	46 Флоп	68 мс	36 Флоп
Welsh-dragon (в)	2,2M	—	—	—	—	362 мс	19,6 Флоп
Cathedral (г)	3,2M	—	—	—	—	697 мс	14 Флоп

Из таблицы видно, что предложенный алгоритм позволил улучшить результаты более чем в 10 раз, по сравнению с лучшей известной реализацией [3] (с учетом разницы в использованной аппаратуре — в 5 раз). Более того, полученные временные оценки сопоставимы с оценками для „компромиссных“ структур (обеспечивающих быстрое построение, но менее эффективных при трассировке — Hybrid BVH, Two-level Grids), полученными в последних работах [3, 5]. Текущая реализация позволяет выполнять визуализацию произвольных динамических сцен до 800К треугольников и статических сцен до 5М треугольников.

Работа выполнена в рамках ФЦП „Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2012 гг.“ и „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

СПИСОК ЛИТЕРАТУРЫ

1. Foley T., Sugeran J. KD-Tree Acceleration Structures for a GPU Raytracer // Proc. of the ACM SIGGRAPH/EUROGRAPHICS Conf. on Graphics Hardware (HWWS'05). Los Angeles, CA, USA, 2005. P. 15—22.
2. Goldsmith J., Salmon J. Automatic Creation of Object Hierarchies for Ray Tracing // IEEE Comput. Graph. Appl. 1987. N 7. P. 14—20.
3. Lauterbach C., Garland M. Fast BVH Construction on GPUs // Computer Graphics Forum. 2009. N 28. P. 375—384.
4. Wald I. Fast Construction of SAH BVHs on the Intel Many Integrated Core (MIC) Architecture [Электронный ресурс]: <[http://techresearch.intel.com/spaw2/uploads/files/TR\\_MIC\\_SAH\\_BVHsBuild.pdf](http://techresearch.intel.com/spaw2/uploads/files/TR_MIC_SAH_BVHsBuild.pdf)>.
5. Kalojanov J., Billeter M., Slusallek P. Two-level Grids for Ray Tracing on GPUs // Computer Graphics Forum. 2011 (to be published).

Сведения об авторах

Денис Константинович Боголепов — аспирант; Нижегородский государственный университет им. Н. И. Лобачевского, кафедра математического обеспечения ЭВМ; E-mail: denisbogol@gmail.com

- Дмитрий Петрович Сопин* — Нижегородский государственный университет им. Н. И. Лобачевского, кафедра математического обеспечения ЭВМ; ассистент;  
E-mail: sopindm@gmail.com
- Данила Ярославич Ульянов* — студент; Нижегородский государственный технический университет им. Р. Е. Алексеева, кафедра вычислительных систем и технологий;  
E-mail: danila-ulyanov@yandex.ru
- Вадим Евгеньевич Турлапов* — д-р техн. наук, доцент; Нижегородский государственный университет им. Н. И. Лобачевского, кафедра математического обеспечения ЭВМ;  
E-mail: vadim.turlapov@gmail.com, vadim.turlapov@cs.vmk.unn.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 004.925.3

Д. А. НАСОНОВ, С. В. ИВАНОВ, А. В. БУХАНОВСКИЙ

**РАСПРЕДЕЛЕННАЯ ПРОГРАММНАЯ ПЛАТФОРМА  
ДЛЯ ПРОТИПИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ  
УПРАВЛЕНИЯ СЛОЖНЫМИ ДИНАМИЧЕСКИМИ ОБЪЕКТАМИ**

Рассмотрены принципы построения инструментальной среды для прототипирования и исследования бортовых систем поддержки оператора сложных динамических объектов в рамках концепции облачных вычислений.

*Ключевые слова:* бортовая система, поддержка оператора, распределенная среда, Грид, облачные вычисления, прикладные сервисы, прототипирование.

Усложнение принципов управления сложными динамическими объектами требует разработки и внедрения бортовых систем поддержки оператора (СПО) в экстремальных условиях эксплуатации. Однако технология разработки таких систем в настоящее время не унифицирована в должной мере. В частности, отсутствуют эффективные средства прототипирования, которые могли бы обеспечить всестороннее экспериментальное исследование характеристик разрабатываемых систем на макетах в составе виртуального испытательного стенда.

В рамках настоящей работы рассматриваются принципы построения распределенной программной платформы (РПП) для прототипирования интеллектуальных систем управления сложными динамическими объектами, функционирующей в рамках концепции облачных вычислений.

РПП представляет собой открытую iPSE-систему [1], функционирующую в распределенной среде. Она включает в себя выделенные вычислительные системы, в том числе обеспечивающие интерфейс с устройствами сбора данных в режиме реального времени, вычислители с графическими процессорами (GPU) и суперкомпьютерные системы, необходимые для решения ресурсоемких задач численного моделирования. Кроме того, в состав среды могут входить уже существующие ресурсы Грид-систем первого поколения. При этом доступ ко всем ресурсам обеспечивается в рамках модели облачных вычислений [2], реализуемой на основе многофункциональной инструментально-технологической платформы CLAVIRE (CLoud Applications VIRTual Environment).

Встраивание компонентов в РПП выполняется на высшем уровне изоляции — в форме прикладных пакетов, взаимодействующих друг с другом через файлы входных и выходных данных. Это обеспечивает возможность выбора и использования уже готовых прикладных пакетов без модификации исходных кодов. Содержательной основой РПП является репози-

торий проблемно-ориентированных сервисов, отвечающих за процесс поддержки принятия решений. В состав репозитория входят прикладные сервисы:

- сбора данных (обращение к датчикам в режиме реального времени через IP или к программам-симуляторам, а также технологические сервисы для эмуляции режимов реального времени);

- обработки и анализа данных (фильтрация и контроль ошибок, статистический анализ, data mining и пр.);

- организации баз знаний (поддержка описания знаний в разных структурах, средства формализации знаний, средства проверки качества баз знаний);

- машин логического вывода (во многом привязаны к структуре баз знаний — вывод по правилам, учет неопределенности, анализ альтернатив, прецедентные технологии и пр.);

- технологий машинного обучения (нейросетевые модели, эволюционное программирование);

- человеко-компьютерного взаимодействия (формирование прикладных интерфейсов системы, визуализации и когнитивной графики);

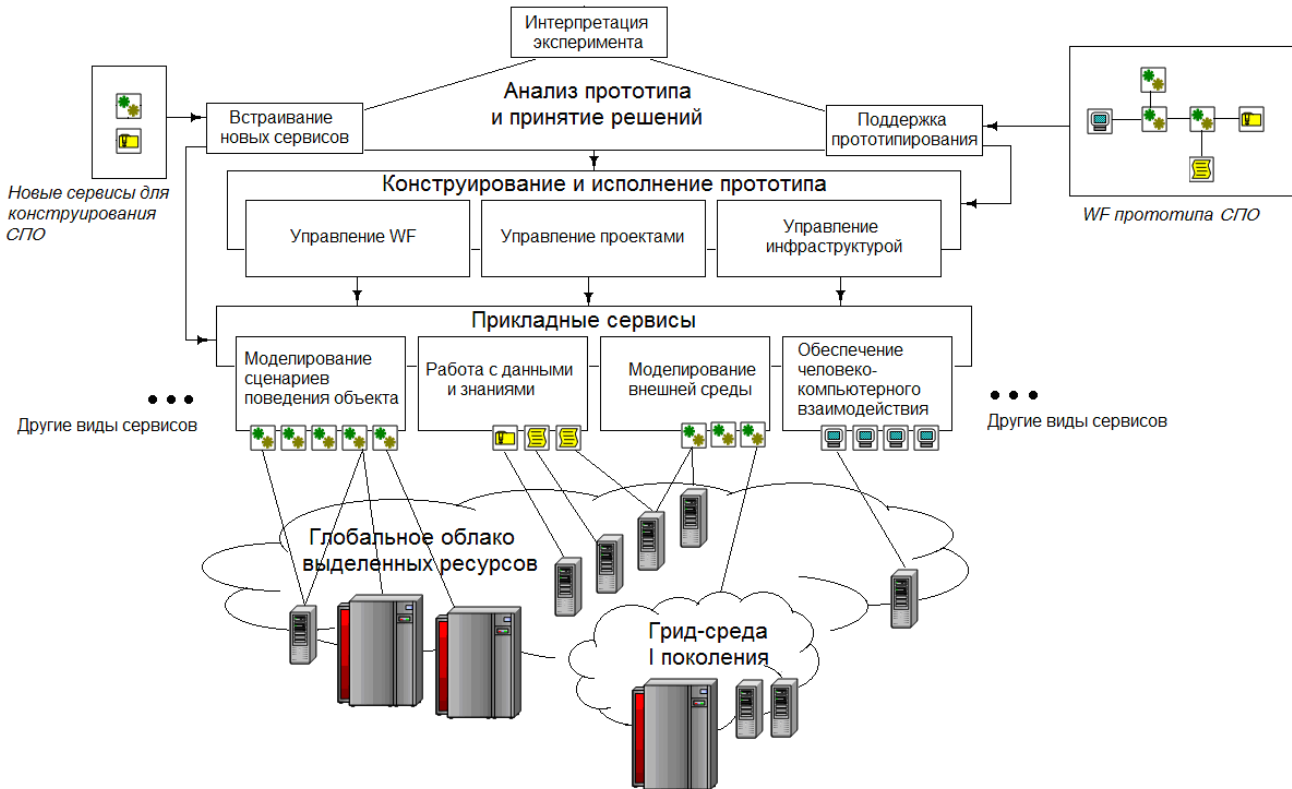
- компьютерного моделирования объекта и среды (по видам динамических объектов).

В целом проблемно-ориентированные сервисы рассматриваются в связке с вычислительными системами соответствующей архитектуры, управляемыми посредством РПП. Это позволяет пользователю в ходе протипирования экспериментировать не только с функциональными элементами программного обеспечения СПО, но и с аппаратной составляющей. Пользователь (разработчик конкретной бортовой СПО) взаимодействует с РПП посредством сети Интернет через web-интерфейс. Он имеет возможность создавать распределенные образы-прототипы собственных СПО, описывая их в форме потока заданий (workflow, WF) на языке EasyFlow [3]. Это позволяет в ходе протипирования исследовать отдельные этапы разработки СПО, постепенно наращивая функционал, заменяя компоненты и пр., в процессе отработки окончательного решения. В том случае, когда доступные в РПП сервисы не могут в полной мере удовлетворить пользователя, он имеет возможность встраивать в WF собственные компоненты; при этом процесс исполнения WF будет замыкаться только на пользовательскую вычислительную систему. Все результаты работы пользователя (создаваемые WF, структуры данных и знаний и пр.) размещаются в соответствующем хранилище, что позволяет географически распределенным группам пользователей работать над проектом поэтапно.

В целом РПП является средством эмуляции работы бортовых СПО с ограничением на реализацию технологий реального времени вследствие существенной инерционности распределенной среды и неопределенности временных характеристик исполнения WF. Дополнительным ограничением является привязка сервисов к конкретным вычислительным ресурсам (например, суперкомпьютерам), которые *a priori* не могут быть использованы в бортовых системах. Вместе с тем РПП позволяет получать информацию о временных характеристиках по отдельным элементам WF и операциям их взаимодействия, а также предоставляет данные о сложности реализуемых алгоритмов, что дает возможность оценить время исполнения WF при изменении архитектуры (без изменения алгоритма). Для моделирования процессов реального времени в состав WF необходимо включать дополнительные компоненты „моделирования времени“, обеспечивающие, в частности, синхронизацию и ранжирование передачи данных между элементами WF.

На рисунке приведена принципиальная архитектура РПП. На схеме отмечены четыре основных уровня. Верхний уровень (анализ прототипа и принятие решений) связан с процессами интеллектуальной поддержки и взаимодействия с пользователем. Следующий уровень относится к автоматическим средствам конструирования и исполнения прототипа. Третий уровень составляют собственно прикладные сервисы (их декларативные описания в репозитории). На четвертом уровне находится облако вычислительных ресурсов, включающих как

выделенные системы, так и существующие распределенные Грид-среды первого поколения. На вычислительных ресурсах установлены пакеты, входящие в состав проблемно-ориентированных сервисов.



Таким образом, посредством РПП разработчик получает возможность в процессе прототипирования сосредоточиться на отработке уникальных частей СПО; для обеспечения функционирования макета в целом используются подходящие сервисы, предоставляемые РПП. Это дает возможность эксперту осуществлять выбор способа реализации различных компонентов СПО путем сравнения работы различных реализаций сервисов в рамках решения отдельных тестовых задач. В целом это позволяет создавать виртуальное окружение для тестирования и обучения уже разработанных бортовых систем; в данном случае сама СПО функционирует на вычислительной системе пользователя, а РПП является лишь эмулятором источников входных данных, характеризующих внешнюю среду. Дополнительно данный подход позволяет минимизировать затраты пользователей на содержание собственной аппаратной инфраструктуры, особенно в области систем ограниченной доступности (например, бортовых вычислителей или суперкомпьютеров нестандартной архитектуры).

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52, № 10. С. 5—24.
2. Foster I., Zhao Y., Raicu I., Lu S. Cloud Computing and Grid Computing 360-Degree Compared // eprint arXiv:0901.0131. 2008 [Электронный ресурс]: <<http://arxiv.org/ftp/arxiv/papers/0901/0901.0131.pdf>>.
3. Князьков К. В., Ларченко А. В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Изв. вузов. Приборостроение. 2011. Т. 54, № 10. С. 36—43.



- Денис Александрович Насонов** — *Сведения об авторах*  
 НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
 E-mail: denis.nasonov@gmail.com
- Сергей Владимирович Иванов** — канд. техн. наук; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник; E-mail: Sergey.v.ivanov@rambler.ru
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
 E-mail: avb\_mail@mail.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
 15.05.11 г.

УДК 681.3.069, 681.324

К. В. КНЯЗЬКОВ

## ОСОБЕННОСТИ РАБОТЫ С ПОТОКАМИ ЗАДАЧ ДЛИТЕЛЬНОГО ИСПОЛНЕНИЯ В РАМКАХ КОНЦЕПЦИИ iPSE

Рассмотрены особенности применения модели потоков заданий длительного исполнения для повышения эффективности вычислительных экспериментов с динамически изменяющимися параметрами в распределенной среде.

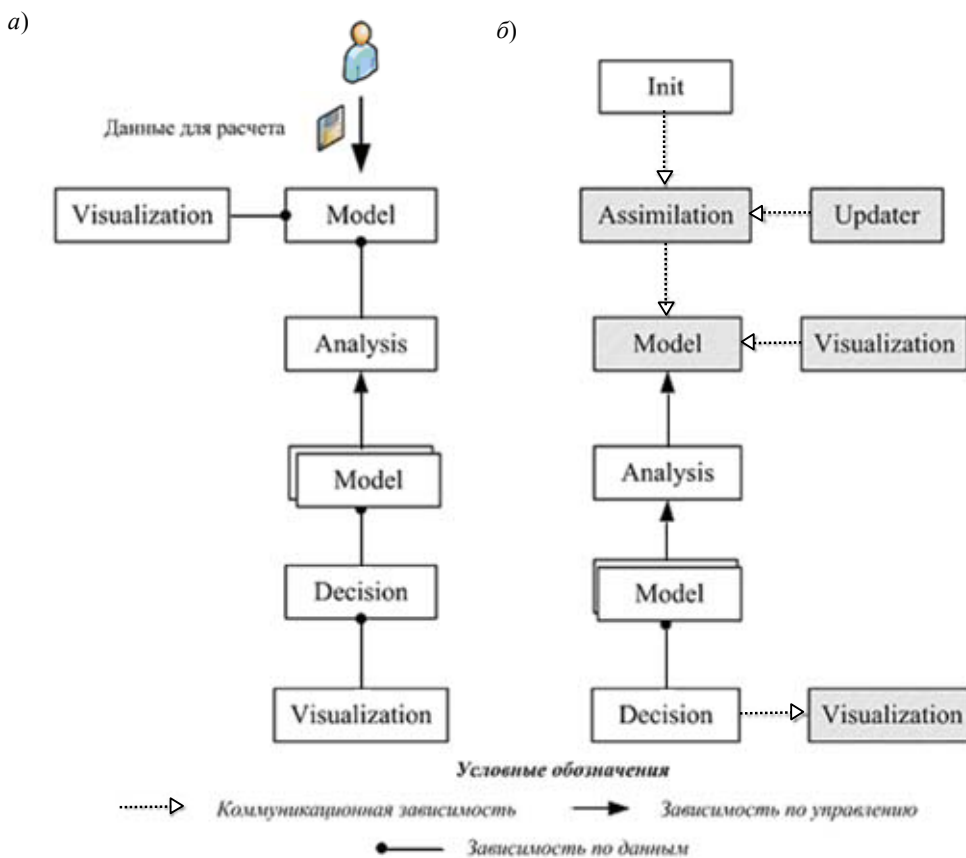
**Ключевые слова:** потоки заданий, задания длительного исполнения, методы коммуникаций, динамическое управление исполнением, усвоение данных.

Перспективы развития технологий решения мультидисциплинарных задач компьютерного моделирования в распределенных средах определяются возможностями абстрактного описания структуры композитных приложений в форме, не зависящей от низкоуровневой вычислительной архитектуры. Для этого применяется формализм потоков заданий (workflow, WF), описывающий сценарий вычислительного эксперимента в виде направленного ациклического графа, узлами которого являются задания, а ребра представляют зависимости (по данным и по управлению) между заданиями. В большинстве случаев система исполнения WF основывается на модели запуска в пакетном режиме, в соответствии с которой WF и его задания после запуска переходят в режим непрерывного исполнения без возможности их модификации и взаимодействия с ними [1]. Однако существует ряд задач, которые сложно адаптировать к пакетному режиму работы без снижения эффективности использования ресурсов. В основном они характеризуются непрерывным поступлением данных в распределенную среду через внешние источники, при котором условия эксперимента могут динамически изменяться в ходе функционирования WF без его перезапуска (см., например, [2]). В настоящей работе рассматриваются возможности использования расширенной модели WF длительного исполнения (long running WF, LRWF) для динамического управления процессом вычислений в облачной среде на основе многофункциональной инструментально-технологической платформы (МИТП) CLAVIRE (CLOUD Applications VIRTUAL Environment), разрабатываемой в рамках концепции iPSE [3].

Модель LRWF имеет следующие характерные особенности:

- поддержка заданий длительного исполнения, время жизни которых ограничено не условиями расчетов внутри WF, а временем работы самой распределенной среды;
- поддержка коммуникации между заданиями одного WF во время исполнения;
- возможность изменения WF во время исполнения за счет сценария WF, а также за счет внешнего управления.

На рисунке приведены схемы WF для задачи поддержки принятия решений в области предотвращения наводненческой ситуации [4] с периодическим усвоением данных наблюдений. Слева (а) представлена схема WF пакетного исполнения, справа (б) — WF длительного исполнения. Рассмотрим подробно второй вариант (серые блоки на рисунке обозначают задания длительного исполнения). Задание *Init* отвечает за передачу в модель начальных условий; *Updater* отвечает за получение извне актуальных данных и передачу их в блок ассимиляции; *Assimilation* подготавливает данные для усвоения в вычислительной модели; *Model* — процесс моделирования динамики уровня и течений, результатом которого является прогноз; *Visualization* призван визуализировать актуальное состояние модели. Группа заданий *Analysis—Model—Decision* начинает свою работу по условию генерации сигнала, который может быть сформирован в задании *Model* вследствие изменения контрольного параметра расчета (достижения прогнозируемым уровнем значения, критического для наступления наводнения). Сценарий работы при этом следующий: запускается задача анализа (*Analysis*) текущей ситуации по данным модели, определяются варианты воздействий на моделируемую систему (которые могут нормализовать ситуацию), далее производится запуск нескольких заданий прорасчета модели путем варьирования параметров и в итоге вызывается модуль оценки и отображения возможных решений (*Decision*). Из схемы видно, что первым запустится задание длительного исполнения *Model*.



Модель LRWF позволяет не только упростить представление WF для задач с динамически изменяемыми условиями расчета, но и более эффективно использовать имеющееся оборудование за счет сокращения накладных расходов при запуске однотипных задач в

различных WF. Такой подход дает возможность производить автоматическую репликацию и балансировку нагрузки на ресурсы на уровне системы исполнения.

Для оценки эффективности модели LRWF в терминах временных издержек расчета в распределенной среде в качестве примера рассмотрим сценарий, когда существуют источник последовательно поступающих данных и задание обработки этих данных, которое сохраняет результаты обработки в хранилище (при этом данные должны быть обработаны строго последовательно). Данные поступают частями объемом  $b$ , с постоянным периодом времени  $t$  (что соответствует задаче усвоения, см. рисунок) непрерывно; общее время передачи данных источником —  $T$ . В этом случае мера эффективности LRWF (по сравнению с пакетным WF) определяется нижней оценкой временных издержек:

$$\delta = \left[ \frac{T}{bt} - 1 \right] (T_p + T_f),$$

где  $T_p$  — время на подготовку к запуску одного задания (выбор ресурса, работа адаптеров ресурсов и пакетов),  $T_f$  — время на завершение задания (обработка результатов, очистка служебных ресурсов и пр.). Параметры в соотношении определяются по результатам измерения времени работы МИП CLAVIRE. Так, для задачи предотвращения наводненческой ситуации в среднем  $\delta = 50$ — $80$  с, что ощутимо влияет на реактивность распределенной среды в целом.

Модель LRWF, помимо повышения эффективности, в ряде аспектов упрощает технологическую реализацию основных положений концепции iPSE. В частности, она позволяет модифицировать описание прикладных пакетов на языке EasyPackage за счет формализации методов коммуникации [5]. Возможности по управлению заданиями определяются в терминах сигналов и слотов (модель издатель—подписчик) путем описания поддерживаемых слотов и их соотношения с механизмами управления пакетом. В свою очередь, это расширяет язык EasyFlow за счет добавления команд генерации сигнала с параметрами и возможностей: объявления коммуникационной зависимости и обработчиков слотов, параметризации зависимости по управлению. Вместе с тем использование модели LRWF существенно усложняет процедуру планирования, поскольку в этом случае производительность сильно зависит от характеристик каналов связи и способов передачи данных.

Для проведения экспериментальных исследований применимости модели LRWF реализован прототип альтернативной системы исполнения для МИП CLAVIRE (язык программирования — Erlang, платформа — OTP). Экспериментальные исследования в целом подтвердили качественные оценки эффективности модели и целесообразность ее практического применения.

Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Deelman E., Gannon D., Shields M., Taylor I. Workflows and e-science: An overview of workflow system features and capabilities // *Future Generation Computer Systems*. 2009. Vol. 25, N 5. P. 528—540.
2. Lawenda M., Okoń M., Oleksiak A., Ludwiczak B., Piontek T., Pukacki J., Meyer N., Nabrzyski J., Stroiński M. Running Interactive Jobs in the Grid Environment // *Parallel Proc. and Appl. Math. Lecture Notes in Computer Sci.* 2006. Vol. 3911/2006. P. 758—765.
3. Бухановский А. В., Ковальчук С. В., Марьин С. В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // *Изв. вузов. Приборостроение*. 2009. Т. 52, № 10. С. 5—24.
4. Бухановский А. В., Житников А. Н., Петросян С. Г., Слоот П. М. А. Высокопроизводительные технологии экстренных вычислений для предотвращения угрозы наводнений // *Там же*. 2011. Т. 54, № 10. С. 14—20.

5. Князьков К. В., Ларченко А. В. Предметно-ориентированные технологии разработки приложений в распределенных средах // Там же. С. 36—43.

#### Сведения об авторе

**Константин Валерьевич Князьков** — НИИ Наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: constantinvk@gmail.com

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 519.853.4

А. В. СЫСОЕВ

## О ПОСТРОЕНИИ СЕМЕЙСТВА МНОЖЕСТВЕННЫХ РАЗВЕРТОК НА ОСНОВЕ КРИВЫХ ПЕАНО ДЛЯ ПАРАЛЛЕЛЬНОГО РЕШЕНИЯ ЗАДАЧ ГЛОБАЛЬНО-ОПТИМАЛЬНОГО ПОИСКА

Предложена схема построения модифицированной множественной развертки на основе кривых Пеано. Схема позволяет многократно увеличить число используемых процессоров для параллельного решения задач глобально-оптимального поиска.

**Ключевые слова:** кривые Пеано, множественная развертка, параллельные вычисления, задачи глобальной оптимизации, модифицированная множественная развертка.

Отображения, называемые *кривыми (развертками) Пеано*, сопоставляют любой липшицевой в гиперкубе  $P = \{v \in R^N : -2^{-1} \leq v_i \leq 2^{-1}, 1 \leq i \leq N\}$  функции  $\varphi(y)$  одномерную функцию  $\varphi(y(x))$ , удовлетворяющую на отрезке  $[0, 1]$  равномерному условию Гельдера с показателем  $1/N$ .

Алгоритм вычисления кривой Пеано с любой заданной точностью подробно описан, например, в работе [1]. Требуемая точность указывается целым числом  $M$  (*номер разбиения*), которое определяет допустимую максимальную погрешность оценки каждой координаты кривой  $y(x)$  для любого заданного значения аргумента  $x$ , равную  $2^{-(M+1)}$ . Значения точек кривой сопоставляют равномерной с шагом  $2^{-NM}$  сетке на отрезке  $[0, 1]$  равномерную с шагом  $2^{-M}$  сетку в гиперкубе  $P$ .

Редукция многомерных задач глобальной оптимизации к одномерным с помощью разверток сохраняет непрерывность и равномерную ограниченность разностей при ограниченной вариации аргумента, однако теряет часть информации о близости точек в многомерном пространстве, поскольку у точки  $x$  на отрезке  $[0, 1]$  имеются две соседние точки, тогда как у соответствующей ей точки  $y(x) \in P$  — соседние по  $2^N$  направлениям.

Возможен следующий способ учета этой информации [2]. Вводится гиперкуб

$$P_0 = \{v \in R^N : -2^{-1} \leq v_i \leq 3 \cdot 2^{-1}, 1 \leq i \leq N\} \quad (1)$$

с длиной ребра, равной двум, и семейство гиперкубов

$$P_l = \{v \in R^N : -2^{-1} \leq v_i + 2^{-l} \leq 3 \cdot 2^{-1}, 1 \leq i \leq N\}, 1 \leq l \leq L, \quad (2)$$

где гиперкуб  $P_{l+1}$  получается путем сдвига гиперкуба  $P_l$  вдоль главной диагонали на шаг  $-2^{-l}$  по каждой координате, и для каждого гиперкуба  $P_l$ ,  $0 \leq l \leq L$ , вводится своя развертка  $y_l(x)$

типа кривой Пеано, отображающая отрезок  $[0, 1]$  на этот гиперкуб. Приближенное построение развертки  $y(x)$  для точности, соответствующей разбиению с номером  $M + 1$ , порождает в гиперкубе  $P_l$  равномерную сетку с шагом  $2^{-M}$  по каждой координате.

Построенная множественная развертка позволяет естественным образом организовать параллельную схему поиска глобального оптимума, используя для расчетов с каждой из  $L$  разверток отдельный процессор. Однако масштабируемость в этом случае ограничена условием, которое накладывает приближенное построение каждой отдельной развертки,  $-L + 1 \leq M$ . Это означает, что при точности построения развертки  $M = 10$  возможно использование только десяти разверток, а значит, только десяти процессоров. Это ограничение существенно сужает возможности применения параллельных вычислений. Для его преодоления в настоящей работе предложена модификация исходной схемы, позволяющая строить семейство множественных разверток.

Рассмотрим двумерный случай. Левый нижний угол гиперкуба  $P_0$  совпадает с левым нижним углом гиперкуба  $P$ . Построим еще три множественные развертки, в каждой из которых положение базового гиперкуба  $P_0$  выбирается так, чтобы у него с гиперкубом  $P$  совпадал один из трех оставшихся углов. Нетрудно показать, что все четыре множественные развертки будут обладать одной, и только одной, общей разверткой, задаваемой в каждой из них гиперкубом

$$P_1 = \{v \in R^N : -1 \leq v_i \leq 1, 1 \leq i \leq N\}. \quad (3)$$

Дадим общее описание правила построения каждой множественной развертки для произвольного  $N$ . Введем двоичную нумерацию множественных разверток, состоящую из  $N$  разрядов:

$$\alpha = (\alpha_0, \dots, \alpha_{N-1}), \quad \alpha_i = 0, 1. \quad (4)$$

Пусть исходная множественная развертка имеет двоичный номер, в котором все разряды равны нулю. Для построения каждой множественной развертки достаточно задать описание базового гиперкуба  $P_0$  и правила сдвига гиперкубов.

Базовый гиперкуб  $P_0$  для множественной развертки с двоичным номером  $\alpha$  будет задаваться как

$$P_0 = \{v \in R^N : -2^{-1} - \alpha_i \leq v_i + 2^{-l} \leq 3 \cdot 2^{-1} - \alpha_i, 1 \leq i \leq N\}, \quad (5)$$

т.е. по тем разрядам, в которых в двоичном номере множественной развертки стоит единица, базовый гиперкуб будет иметь по соответствующим переменным границы  $(-3-1) \cdot 2^{-1}$ , а по разрядам, равным нулю, границы по переменным будут  $(-1-3) \cdot 2^{-1}$ .

Смещение остальных гиперкубов  $P_1, \dots, P_L$  в множественной развертке с номером  $\alpha$  будет описываться вектором

$$\beta = (-2^{-l} \cdot (-1)^{\alpha_0}, \dots, -2^{-l} \cdot (-1)^{\alpha_{N-1}}), \quad (6)$$

т.е. по переменным, которым в двоичном номере множественной развертки соответствует единица, сдвиг будет выполняться на  $2^{-l}$ , а по остальным на  $-2^{-l}$ .

Предложенная в работе модифицированная множественная развертка значительно повышает потенциал использования параллелизма в процессе поиска глобального оптимума. Так, уже для трехмерных задач при точности построения приближения развертки  $M = 10$  общее число доступных разверток в семействе множественных разверток составит  $M \cdot 2^3 - 1 = 79$ . В общем случае для задач размерности  $N$  при точности приближенного построения кривых Пеано  $M$  максимальное число разверток в семействе множественных разверток равно  $M \cdot 2^N - 1$ .

Работа выполнена в рамках ФЦП „Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2012 гг.“ и „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

## СПИСОК ЛИТЕРАТУРЫ

1. Стронгин Р. Г. Поиск глобального оптимума. М.: Знание, 1990.
2. Стронгин Р. Г. Параллельная многоэкстремальная оптимизация с использованием множества разверток // Ж. вычисл. матем. и матем. физ. 1991. Т. 31, № 8. С. 1173—1185.

*Сведения об авторе*

*Александр Владимирович Сысоев* — Нижегородский государственный университет им. Н. И. Лобачевского, кафедра математического обеспечения ЭВМ; ассистент;  
E-mail: sysoyev@vmk.unn.ru

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.

УДК 681.3.069, 681.324

С. В. МОСТАМАНДИ, Д. А. НАСОНОВ, А. А. КАЛЮЖНАЯ, А. В. БУХАНОВСКИЙ

**АНСАМБЛЕВЫЕ ПРОГНОЗЫ ЭКСТРЕМАЛЬНЫХ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИХ ЯВЛЕНИЙ  
В РАСПРЕДЕЛЕННОЙ СРЕДЕ CLAVIRE**

Рассмотрены особенности применения среды CLAVIRE в задачах ансамблевого прогноза экстремальных гидрометеорологических явлений применительно к наводнениям в Санкт-Петербурге.

*Ключевые слова:* ансамблевый прогноз, ансамблевый фильтр Калмана, чувствительность прогнозов, распределенные вычисления, наводнения.

Оперативное прогнозирование гидрометеорологических процессов связано с проблемой учета неопределенности и неполноты исходных данных, что, в свою очередь, отрицательно сказывается на точности прогнозов и их заблаговременности. Применительно к экстремальным явлениям проблема усугубляется вследствие их редкой повторяемости, что не позволяет в полной мере воспользоваться ретроспективными данными для настройки прогностических моделей. Отчасти снизить влияние неопределенности можно с помощью процедуры оперативного усвоения данных наблюдений в моделях, однако качество усвоения зависит от того, насколько полно в данной процедуре используются априорные знания о поведении модели в различных ситуациях. Для получения этих знаний могут быть использованы разного рода адаптивные механизмы, основанные на уточнении процедуры усвоения по ходу поступления новой информации. Однако при прогнозировании экстремальных явлений (например, наводнений в Санкт-Петербурге [1]) для такой настройки требуется несколько лет оперативной эксплуатации. Другой подход заключается в использовании результатов непрерывных гидродинамических расчетов за несколько десятков лет. Его эффективность зависит от выбора климатического интервала; в частности, он применим только при стационарной межгодовой изменчивости моделируемого гидрометеорологического явления.

Альтернативой рассмотренным подходам является использование ансамблевых методов, в рамках которых влияние неопределенности на точность прогнозов исследуется непосредственно в оперативном режиме путем искусственного внесения стохастической изменчивости в исходные данные, граничные условия и параметры модели. Существуют различные варианты применения ансамблевых методов прогноза:

— интервальное оценивание прогнозов с заданной заблаговременностью для определения меры риска при принятии решений по прогностическим данным;

— определение ковариационных характеристик в процедуре усвоения на основе ансамблевого фильтра Калмана [2];

— изучение чувствительности прогностических моделей к изменению настроечных параметров при диагностических расчетах.

Практическая реализация ансамблевых методов прогноза связана с выполнением большого количества одинаковых расчетов при различных входных данных; при этом каждый расчет, требующий численного интегрирования системы уравнений аэрогидродинамики, достаточно ресурсоемок. При этом в ходе выполнения отдельных расчетов не требуется синхронизация; незавершение какого-либо расчета не является критическим для выполнения всей процедуры. Поэтому для такого рода задач целесообразно использовать не традиционные суперкомпьютерные системы высокой производительности, а проблемно-ориентированные среды распределенных вычислений, автоматизирующие построение композитных приложений, распределение заданий, балансировку нагрузки и сбор результатов.

В настоящей работе обсуждается опыт выполнения ансамблевых прогнозов с использованием среды CLAVIRE (CLOUD Applications VIRTUAL Environment) применительно к наводнениям в Санкт-Петербурге. Композитное приложение для выполнения прогнозов включает следующие прикладные сервисы:

— гидродинамическая спектральная модель морского волнения SWAN, необходимая для определения коэффициента трения водной поверхности, отражающего интенсивность передачи энергии от атмосферы водным массам;

— гидродинамическая баротропная модель течений и уровня Балтийского моря BSM, по которой выполняется прогноз уровня в Санкт-Петербурге;

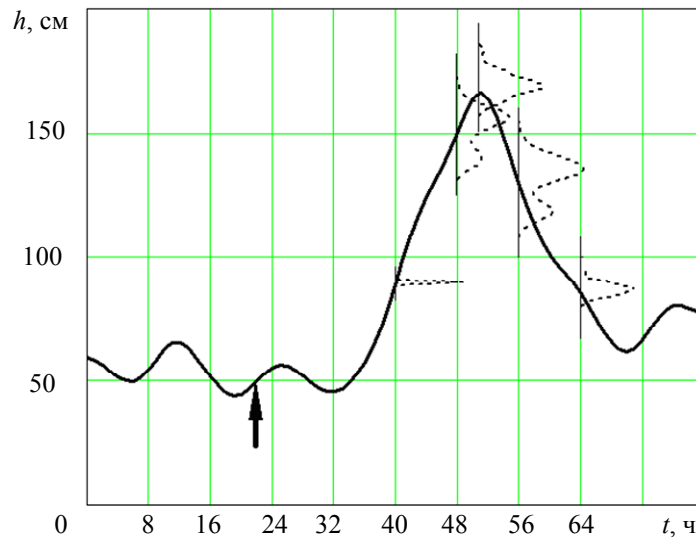
— модуль усвоения данных измерений уровня моря в модели BSM, основанный на редуцированном фильтре Калмана.

Выполнение композитного приложения требует параллельного запуска модели BSM с усвоением и без усвоения данных измерений; на основе сопоставления результатов расчетов выполняется корректировка ошибки прогноза на последующем временном шаге. В качестве входных данных используется метеорологический прогноз HIRLAM [3], граничные условия учитывают водообмен через Датские проливы и сток реки Нева.

При выполнении ансамблевого прогноза наводнения ситуации необходимо использовать временную предысторию расчетов; в зависимости от источника неопределенности (ветер, речной сток, водообмен и пр.) ансамбль строится или по отдельным сервисам из списка выше, или непосредственно по всей цепочке. При этом в среде CLAVIRE формируется очередь из независимых заданий, которые распределяются по целевым системам по мере освобождения ресурсов. Для этого на каждой из целевых систем должны быть установлены все или некоторые из перечисленных выше моделей; они регистрируются в соответствующей базе ресурсов и сервисов распределенной среды. Наличие одного и того же набора сервисов на всех целевых системах не является обязательным; это влияет только на общее время выполнения расчетов, поскольку обмен данными между шагами композитного приложения выполняется через централизованное хранилище.

В качестве иллюстрации на рисунке приведен ансамблевый прогноз наводнения 14 ноября 2008 г. (за 30 часов до наступления пика) в пункте Горный институт, выполненный по ретроспективным данным. Начало ансамблевого прогноза отмечено стрелкой. При построении ансамбля предполагалось, что прогностические данные по ветру в восточной части Финского залива имеют ошибку со среднеквадратичным отклонением 2 м/с. На рисунке приведен детерминированный прогноз на 48 часов вперед, а также ядерные оценки распределений хода уровня для различной заблаговременности. Видно, что ошибки в прогнозе ветра на локальной акватории могут приводить к вариациям уровня на 30—40 см; при этом наибольший разброс наблюдается на подъеме и спаде уровня; т.е. более чувствителен к неопределенности в

исходных данных прогноз скорости наступления наводнения, чем его пика. Экспериментальные исследования производительности композитного приложения, используемого для получения результатов на рисунке, показали ускорение вычислений (за счет распараллеливания в распределенной среде и в рамках общей памяти отдельных целевых систем) в 45 раз. Накладные расходы, вносимые CLAVIRE, не превышают 0,3 %.



Работа выполнена в рамках проектов по реализации Постановлений № 218 и 220 Правительства Российской Федерации при поддержке ФЦП „Научные и научно-педагогические кадры инновационной России на 2009—2013 гг.“.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бухановский А. В., Рожков В. А. Невские наводнения редкой повторяемости // Изв. РГО. 1999. Т. 131, вып. 3. С. 42—48.
2. Mourre B. et al. Assimilation of sea level data over continental shelves: an ensemble method for the exploration of model errors due to uncertainties in bathymetry // Dynamics of Atmospheres and Oceans. 2004. Vol. 38, Iss. 2. P. 93—121.
3. HIRLAM homepage [Электронный ресурс]: <<http://hirlam.org/>>.

#### Сведения об авторах

- Мохаммад Сулейман Мостаманди** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; старший научный сотрудник;  
E-mail: [suleiman.mostamandy@gmail.com](mailto:suleiman.mostamandy@gmail.com)
- Денис Александрович Насонов** — НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; младший научный сотрудник;  
E-mail: [denis.nasonov@gmail.com](mailto:denis.nasonov@gmail.com)
- Анна Владимировна Калюжная** — студентка; Санкт-Петербургский Государственный университет;  
E-mail: [kalyuzhnaya.ann@gmail.com](mailto:kalyuzhnaya.ann@gmail.com)
- Александр Валерьевич Бухановский** — д-р техн. наук, профессор; НИИ Научно-технических компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики; директор;  
E-mail: [avb\\_mail@mail.ru](mailto:avb_mail@mail.ru)

Рекомендована НИИ НКТ

Поступила в редакцию  
15.05.11 г.



## SUMMARY

P. 7—14.

### CLAVIRE: PERSPECTIVE TECHNOLOGY FOR SECOND GENERATION CLOUD COMPUTING

Features of technology for 2-nd generation cloud computing based on SaaS/AaaS model are concerned. Multifunctional technology platform CLAVIRE for building and execution of composite application using distributed cloud services is described.

**Keywords:** cloud computing, services, resources, virtualization, composite application.

#### *Data on authors*

- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru
- Vladimir N. Vasiliev* — Dr. Techn. Sci., Professor; St. Petersburg State University of Information Technologies, Mechanics and Optics; Rector;
- Valeriy N. Vinogradov* — I.T. Co., Project Management Center, Moscow; Head;  
E-mail: VVinogradov@it.ru
- Dmitry Yu. Smirnov* — I.T. Co., Department for Government and Educational Projects, Moscow; Head;  
E-mail: DSmirnov@it.ru
- Sergey A. Sukhorukov* — I.T. Co., Department for Government and Educational Projects, Moscow; Deputy Head; E-mail: SSukhorukov@it.ru
- Tagir G. Yapparov* — I.T. Co., Department for Government and Educational Projects, Moscow; Chairman of Board of Directors; E-mail: TYapparov@it.ru

P. 14—20.

### HIGH-PERFORMANCE TECHNOLOGIES OF URGENT COMPUTING FOR FLOOD HAZARD PREVENTION

Application of urgent computing technologies to development and organization of flood hazard prevention systems is considered using St. Petersburg case as an example.

**Keywords:** urgent computing, decision making support, supercomputers, Grid, forecast of water level and currents, data assimilation.

#### *Data on authors*

- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru
- Andrey N. Zhitnikov* — BCC Co., Department of Consulting and Software Projects, St. Petersburg; Deputy Head; E-mail: azhitnikov@bcc.ru
- Sergey G. Petrosyan* — BCC Co., Department of Consulting and Software Projects, St. Petersburg; Head of Project; E-mail: spetrosyan@bcc.ru
- Petrus Maria Arnoldus Sloot* — PhD; University of Amsterdam; E-mail: p.m.a.sloot@uva.nl

**P. 21—29.****SERVICE-ORIENTED PLATFORM FOR EXECUTION OF COMPOSITE APPLICATIONS IN DISTRIBUTED ENVIRONMENT**

An intellectual platform is designed for control over execution of composite applications in a distributed computing environment on the base of service-oriented architecture. The platform is adapted for incorporation into program complex HPC-NASIS aimed at multi-scale modeling in nanotechnologies.

**Keywords:** intellectual control, planning, parametric model, heuristics, service-oriented architecture.

*Data on authors*

- Sergey V. Maryin* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: sm.niinkt@gmail.com
- Sergey V. Kovalchuk* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: kovalchuk@mail.ifmo.ru

**P. 29—36.****INTELLECTUAL SUPPORT OF COMPOSITE APPLICATION DESIGN IN DISTRIBUTED PROBLEM-ORIENTED ENVIRONMENT**

Methods of organization of intellectual support of a user designing a composite application on the base of ontological approach are considered. A system is developed for intellectual support of users in computer modeling of quantum chemistry problem.

**Keywords:** intellectual system, expert knowledge, ontology, logical inference.

*Data on authors*

- Sergey V. Kovalchuk* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: kovalchuk@mail.ifmo.ru
- Vladimir G. Maslov* — Dr. Phys.-Math. Sci.; St. Petersburg State University of Information Technologies, Mechanics and Optics, Department of Optical Physics and Modern Natural Science;  
E-mail: maslov04@bk.ru

**P. 36—43.****DOMAIN-SPECIFIC TECHNOLOGIES FOR APPLICATION DESIGN IN DISTRIBUTED ENVIRONMENTS**

An approach is proposed for design and realization of composite applications in distributed computing environment on the base of domain-specific description. Domain-specific languages EasyPackage and EasyFlow are developed for realization of the approach. Technologies based on the languages are proposed for creation of domain-specific interfaces, design and execution of composite applications realized on CLAVIRE platform.

**Keywords:** job flow control, application package, composite application, domain-specific language, domain-specific interface.

*Data on authors*

- Konstantin V. Knyazkov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: konstantinvk@gmail.com
- Aleksey V. Larchenko* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: aleksey.larchenko@gmail.com

P. 44—50.

**SCIENTIFIC-TRAINING COMPLEX “COMPUTER MODELING IN NANOTECHNOLOGIES” ON THE BASE OF A GRID SYSTEM**

A scientific-training complex is developed for performance of tasks in modeling of nano-sized atomic and molecular structures, nano-materials, processes and devices based on such structures. The complex is operated in remote access mode and uses resources of distributed computing environment of the National nano-technological grid system.

**Keywords:** nanotechnologies, service, distributed environment, remote access, portal, nano-sized systems, training methods.

*Data on authors*

- Dmitry M. Spelnikov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: pilule@ya.ru
- Alexander A. Guskov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: aleksander.guskov@gmail.com
- Vladimir G. Maslov* — Dr. Phys.-Math. Sci.; St. Petersburg State University of Information Technologies, Mechanics and Optics, Department of Optical Physics and Modern Natural Science; E-mail: maslov04@bk.ru
- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head; E-mail: avb\_mail@mail.ru

P. 51—57

**FEATURES OF PROBABILISTIC ANALYSIS OF EFFICIENCY AND RELIABILITY OF PROBLEM-ORIENTED ENVIRONMENTS FOR CLOUD COMPUTING**

A probabilistic model of cloud computing environment performance is proposed. Approaches to estimation of efficiency and reliability of service providing are considered on the base of the proposed model.

**Keywords:** cloud computing, services, distributed environment, efficiency, error-free running time.

*Data on authors*

- Timofey N. Churov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: tchurovtim@gmail.com
- Denis A. Nasonov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: denis.nasonov@gmail.com
- Boris Ya. Shteinberg* — Dr. Techn. Sci., Professor; Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Head of Department; E-mail: borsteinb@mail.ru
- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head; E-mail: avb\_mail@mail.ru

**P. 58—64.****VIRTUAL RANGE FOR RESEARCH DESIGN OF MARITIME OBJECTS AND CONSTRUCTIONS**

The problem of organization of virtual range for research design in shipbuilding is considered. A method is developed for formation of dynamic scenes of maritime object behavior under 3D roughness in extreme service conditions. The method is adapted to employment in wide-screen virtual reality systems.

**Keywords:** virtual range, research design, extreme situation, 3D roughness.

*Data on authors*

- Aleksey A. Bezgodov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: demiurghg@gmail.com
- Sergey V. Ivanov* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: Sergey.v.ivanov@rambler.ru
- Sergey S. Kosukhin* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: skosukhin@gmail.com
- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru

**P. 65—72.****WEB-ORIENTED RESEARCH-PRODUCTION CENTER “SOCIODYNAMICS”**

Concept and principal architecture of a problem-oriented cloud-computing environment is described. The environment under consideration ensures the functioning of research-production center for investigation into social dynamics and its applications in the frames of web 2.0 paradigm.

**Keywords:** sociodynamics, sociometry, cloud computing, applied services, social network, web crawler, spreading of rumours.

*Data on authors*

- Sergey V. Ivanov* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: Sergey.v.ivanov@rambler.ru
- Ekatherina V. Bolgova* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: katerina.bolgova@gmail.com
- Viktor V. Kashirin* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: kashirin.viktor@gmail.com
- Andrey V. Yakushev* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: yaja30@gmail.com
- Andrey V. Chugunov* — Cand. Politic. Sci., St. Petersburg State University of Information Technologies, Mechanics and Optics, eGovernment Center; Head;  
E-mail: chugunov@egov-center.ru
- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru

P. 72—79.

### PARALLEL ALGORITHMS OF DYNAMIC PROCESS MODELING IN COMPLEX NETWORK

A formal description of complex networks is presented. Code parallelization schemes in modeling of dynamic processes in complex networks are considered.

**Keywords:** complex network, dynamic process, random graph, code parallelization.

#### *Data on authors*

- Ekatherina V. Bolgova* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: katerina.bolgova@gmail.com
- Sergey V. Ivanov* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher;  
E-mail: Sergey.v.ivanov@rambler.ru
- Ekatherina A. Grinina* — Cand. Phys.-Math. Sci.; St. Petersburg State University of Information Technologies, Mechanics and Optics; E-mail: grinina@tprs.ru
- Petrus Maria Arnoldus Sloot* — PhD; University of Amsterdam; E-mail: p.m.a.sloot@uva.nl
- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru

P. 80—86.

### RESULTS OF PROJECT “YOUNG RESEARCHERS MOBILITY” REALIZATION IN 2010: DEVELOPMENT OF FUNCTIONAL ELEMENTS OF IPSE TECHNOLOGY AND APPLIED SERVICE WIDENING

New generation of problem-oriented distributed environments for computer modeling of complex systems are considered. Results of joint study in application and development of corresponding technologies are presented. The investigations are carried out in the frames of “Young Researchers Mobility” activity in 2010 as a part of the Federal Target Program “Professional Community of Innovative Russia for 2007—2012”.

**Keywords:** distributed computing, computer modeling, complex system.

#### *Data on authors*

- Alexander V. Boukhanovsky* — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head;  
E-mail: avb\_mail@mail.ru
- Sergey V. Maryin* — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: sm.niinkt@gmail.com
- Konstantin V. Knyazkov* — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher;  
E-mail: constantinvk@gmail.com
- Aleksey A. Sidnev* — Lobachevsky State University, Department of Computer Software, Nizhni Novgorod; Assistant; E-mail: sidnev.a.a@gmail.com
- Sergey N. Zhabin* — Moscow Institute of Physics and Technology, Department of General Physics; Assistant; E-mail: delafrog@gmail.com
- Anton P. Bagly* — Post-Graduate Student; Southern Federal University, Department of Mechanics, Mathematics, and Computing Science, Rostov-on-Don;  
E-mail: taccessviolation@yandex.ru
- Roman B. Shteinberg* — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Researcher; E-mail: romanofficial@yandex.ru
- Anastasia V. Shamakina* — South Ural State University, Laboratory of Supercomputer Modeling, Department of Distributed Computing, Chelyabinsk; Head of Department;  
E-mail: sham2004@bk.ru

- Vadim V. Voevodin** — Lomonosov Moscow State University, Research Computing Center, Laboratory of Parallel Information Technologies; Junior Researcher; E-mail: vadim@parallel.ru
- Evdokiya N. Golovchenko** — Keldysh Institute of Applied Mathematics of RAS, Section #16, Moscow; Junior Researcher; E-mail: ge03@imamod.ru
- Roman T. Phalaleev** — Moscow Institute of Physics and Technology; Informatization Department, Head; E-mail: roman@phystech.edu
- Aleksey V. Dukhanov** — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher; E-mail: avd\_mail@mail.ru
- Aleksey A. Tarasov** — Cand. Phys.-Math. Sci.; Moscow Institute of Physics and Technology, Department of Distributive Computing; E-mail: tarasov@isa.ru
- Lev V. Shamardin** — D.V.Skobeltsyn Scientific Research Institute of Nuclear Physics, Moscow; Senior Researcher; E-mail: shamardin@theory.sinp.msu.ru
- Aleksey I. Moiseenko** — I.T. Co., Department of Information Management Technology, Moscow; Head of Department; E-mail: AMoiseenko@it.ru

**P. 87—89.**

**FEATURES OF REALIZATION OF PARALLELIZING TRANSFORMATIONS IN INTERACTIVE HIGH-LEVEL OPTIMIZATING PARALLELIZER OF CODES**

Performance peculiarities of transformations library in interactive high-level optimizing parallelizer are discussed.

**Keywords:** parallelizing, code transformation, optimization.

**Data on authors**

- Boris Ya. Shteinberg** — Dr. Techn. Sci., Professor; Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Head of Department; E-mail: borsteinb@mail.ru
- Eugeny N. Kravchenko** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Programmer; E-mail: peon\_sxe@mail.ru
- Roman I. Morylev** — Post-Graduate Student; Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; E-mail: frg10@yandex.ru
- Zinovy Ya. Nis** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Programmer; E-mail: irishrover@mail.ru
- Viktor V. Petrenko** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Researcher; E-mail: vpetrenko@gmail.com
- Ivan S. Skiba** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Programmer; E-mail: aidan@pisem.net
- Vasily N. Shapovalov** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Junior Researcher; E-mail: Vasily.Shapovalov@gmail.com
- Roman B. Shteinberg** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Researcher; E-mail: romanofficial@yandex.ru
- Oleg B. Shteinberg** — Southern Federal University, Department of Algebra and Discrete Mathematics, Rostov-on-Don; Assistant; E-mail: romanofficial@yandex.ru

P. 89—91.

### PECULIARITIES OF DIGITAL SOLUTION FOR EVOLUTIONAL PROBLEMS OF SHORT LASER PULSE PROPAGATION WITH NUMA ARCHITECTURE SYSTEMS

Methods of adaptation of algorithms of short laser pulse propagation problems solution for computing systems of NUMA architecture are considered.

**Keywords:** NUMA, program run, wave equation, extra short pulse.

#### *Data on author*

**Daniil A. Fadeev** — Institute of Applied Physics of RAS, Department of Nonlinear Electrodynamics, Nizhni Novgorod; Junior Researcher; E-mail: fadev@appl.sci-nnov.ru

P. 92—94.

### CONSTRUCTION OF SAH BVH TREES FOR RAY TRACING WITH THE USE OF GRAPHIC PROCESSORS

A real-time algorithm of SAH BVH trees construction on graphic processors is developed. The algorithm allows for increase the visualization effectiveness for an order of magnitude as compared with analogues.

**Keywords:** ray tracing, accelerating structure, SAH, BVH, dynamic scenes, graphic processor, OpenCL.

#### *Data on authors*

**Denis K. Bogolepov** — Post-Graduate Student; Lobachevsky State University, Department of Computer Software, Nizhni Novgorod; E-mail: denisbogol@gmail.com

**Dmitry P. Sopin** — Lobachevsky State University, Department of Computer Software, Nizhni Novgorod; Assistant; E-mail: sopindm@gmail.com

**Danila Ya. Ulyanov** — Student; Alekseev State Technical University, Department of Computer Software, Nizhni Novgorod; E-mail: danila-ulyanov@yandex.ru

**Vadim E. Turlapov** — Dr. Techn. Sci., Professor; Lobachevsky State University, Department of Computer Software, Nizhni Novgorod; E-mail: vadim.turlapov@gmail.com, vadim.turlapov@cs.vmk.unn.ru

P. 94—97.

### DISTRIBUTED PROGRAM PLATFORM FOR PROTOTYPING OF INTELLECTUAL SYSTEMS FOR CONTROL OVER COMPLEX DYNAMIC OBJECTS

Principles of instrument environment construction for prototyping and analysis of on-board support systems for complex dynamic objects operator are considered in the frame of cloud computing concept.

**Keywords:** on-board system, operator support, distributed environment, Grid, cloud computing, applied services, prototyping.

#### *Data on authors*

**Denis A. Nasonov** — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: denis.nasonov@gmail.com

**Sergey V. Ivanov** — Cand. Techn. Sci.; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher; E-mail: Sergey.v.ivanov@rambler.ru

**Alexander V. Boukhanovsky** — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head; E-mail: avb\_mail@mail.ru

**P. 97—100.****PECULIARITIES OF LONG-TERM EXECUTION PROGRAMS FLOWS IN THE FRAMES OF IPSE CONCEPT**

Peculiarities of application of the model of long-term execution task flow for enhancement of efficiency of calculation experiments performance with dynamically varying parameters in distributed environment.

**Keywords:** task flow, long-term execution task, communication method, dynamic control over execution, data assimilation.

**Data on author**

**Konstantin V. Knyazkov** — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: konstantinvk@gmail.com

**P. 100—102.****ON FAMILY OF MULTIPLE SCANS BASED ON PEANO CURVES CONSTRUCTION FOR GLOBAL OPTIMIZATION TASKS PARALLEL SOLVING**

Scheme for building modified multiple scans based on Peano curves is proposed. The scheme allows increasing manyfold quantity of CPUs used for global optimization tasks parallel solving.

**Keywords:** Peano curves, multiple scan, parallel computing, global optimization tasks, modified multiple scans.

**Data on author**

**Alexander V. Sysoev** — Lobachevsky State University, Department of Computer Software, Nizhni Novgorod; Assistant; E-mail: sysoev@vmk.unn.ru

**P. 102—104.****ENSEMBLE FORECASTS OF THE EXTREME HYDROMETEOROLOGICAL PHENOMENA IN DISTRIBUTED ENVIRONMENT CLAVIRE**

Features of CLAVIRE environment application for extreme metocean events ensemble forecasting tasks concerning flooding of Saint-Petersburg area is described.

**Keywords:** ensemble forecast, ensemble Kalman filter, forecast sensitivity, distributed computing, flooding.

**Data on authors**

**Suleiman Mohamad Mostamandi** — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Senior Researcher; E-mail: suleiman.mostamandy@gmail.com

**Denis A. Nasonov** — eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Junior Researcher; E-mail: denis.nasonov@gmail.com

**Anna V. Kaluzhnaya** — Student; St. Petersburg State University, E-mail: kalyuzhnaya.ann@gmail.com

**Alexander V. Boukhanovsky** — Dr. Techn. Sci., Professor; eScience Research Institute, St. Petersburg State University of Information Technologies, Mechanics and Optics; Head; E-mail: avb\_mail@mail.ru