

# Применение новых вычислительных технологий для повышения эффективности расчетов в грид-средах\*

А.В. Пивушков<sup>1</sup>, В.М. Волохов<sup>1</sup>, Д.А. Варламов<sup>1,2</sup>, А.В. Волохов<sup>1</sup>, Н.Ф. Сурков<sup>1</sup>

Институт проблем химической физики РАН<sup>1</sup>,  
Институт экспериментальной минералогии РАН<sup>2</sup>

В статье сделан обзор различных технологий, позволяющих существенно повысить эффективность проведения грид-расчетов (в том числе в области вычислительной химии). Описаны особенности применения технологий виртуализации грид-ресурсов и грид-сервисов, способов работы с большими пулами независимых грид-заданий на равномерных «сетках» данных или параметров, реализация в грид-средах версий прикладных пакетов с поддержкой CUDA. Сформулированы перспективы применения описанных технологий для проведения грид-вычислений.

## 1. Введение

После создания в России постоянно действующих грид-полигонов и интеграции в их состав значительных вычислительных ресурсов, перед администраторами ресурсных грид-сайтов и пользователями грид-ресурсов встали задачи по повышению эффективности проведения вычислений в распределенных средах (включая снижение расходов на эксплуатацию, понижение трудоемкости администрирования), увеличение степени совместимости прикладного ПО, повышения скорости проведения расчетов, разработка способов решения «объемных» задач на больших «сетках» равномерных данных или входных параметров.

В качестве предлагаемых технологий для решения подобных задач здесь рассмотрены следующие: технологии виртуализации грид-ресурсов, грид-сервисов и прикладных грид-приложений; способы работы в грид-среде с большими пулами независимых заданий, формируемых на равномерных сетках данных или входных параметров, работа в грид-средах с версиями прикладных пакетов, ориентированных на использование расчетов с поддержкой GPU (Graphical Processor Units) устройств.

Данная статья может представлять интерес для администраторов ресурсных грид-сайтов, заинтересованных в увеличении эффективности использования своих ресурсов, системных программистов, конечных пользователей-химиков, заинтересованных в интенсификации своих расчетов. Сразу заметим, что данные технологии могут быть применены для решения широкого круга и других задач, не связанных с химией.

## 2. Применение технологий виртуализации ресурсов и приложений

### 2.1 Виртуализация грид-ресурсов и грид-сервисов на основе виртуальных машин

В 2010-2011 годах в ИПХ РАН было проведено детальное изучение различных технологий виртуализации ресурсов для повышения функциональности ресурсных центров грид-полигонов, снижения трудоемкости и затрат на их настройку и поддержку [3,5]. Часть созданных виртуальных машин переведена в режим постоянной эксплуатации в качестве поставщиков грид-сервисов как для нужд самого узла, так и вовне, на используемых грид-полигонах.

Для изучения особенностей использования виртуальных ресурсов в грид-средах были протестированы свободно распространяемые гипервизоры виртуальных машин, среди которых среды Xen (<http://xen.org>, разработка XenSource, Inc.), VirtualBox (<http://www.virtualbox.org>, разработка Innotek, затем подразделение Sun Microsystems, далее Oracle Corporation), VMware (<http://www.vmware.com>, разработка VMware Inc. – в настоящее время предоставляется бесплатно распространяемая версия VMware Server), Linux-KVM (Kernel-based Virtual Machine,

\* Работа выполнена при поддержке гранта РФФИ № 11-07-00686-а

<http://www.linux-kvm.org>, фирма Qumranet, подразделение RedHat), oVirt (<http://ovirt.org>), а также эмуляторы виртуальных машин – QEMU (<http://bellard.org/qemu>).

Проведенное тестирование и исследование функционалов пакетов показали близость их характеристик для решения поставленных в проекте задач и возможность их применения в распределенных средах. Все указанные гипервизоры позволяют производить как аппаратную, так и паравиртуализацию для ОС семейств Linux и Windows (а так-же других типов ОС типа FreeBSD), осуществлять запуск нескольких изолированных виртуальных машин (ВМ) разного типа на одном физическом узле, адекватно воспринимать различные физические устройства host машины (сетевые карты, диски), проводить резервирование и миграцию ВМ между физическими узлами, высокую скорость запуска и т.д. Нами оценивалась прежде всего возможность их применения к работе в распределенных средах (удобство использования, простота администрирования гипервизора и ВМ, возможность работы на гетерогенных ресурсах и т.п.).

Опыт исследований и применения показал, что виртуальные машины в качестве серверных компонентов (ресурсы, сервисы) в первую очередь должны быть использованы тогда, когда на одном физическом узле желательно выполнение приложений и сервисов, не отличающихся высокой степенью загрузки физического ресурса и высоким сетевым трафиком, но требующих выполнения следующих условий: (а) наличия различных платформ (ОС) для выполнения специфического системного или прикладного ПО; (б) присутствия несовместимых между собой конфигураций одной и той же платформы; (в) изоляции сервисов друг от друга (невозможность выполнения их на одном узле – использование одних и тех же портов и сокетов, конфликт сетевых ресурсов и т.п.).

В качестве окончательного варианта используемого средства виртуализации ресурсов и сервисов был выбран гипервизор KVM (Kernel-based Virtual Machine, <http://www.linux-kvm.org>). Выбор основывается на простоте администрирования, устойчивости работы под нагрузкой, независимостью (в отличие от большинства прочих) от стороннего коммерческого разработчика, наибольшей ориентированностью на Linux архитектуру (на которой основано подавляющее большинство российских грид-ресурсов), интеграцией в Linux ядром (т.е. простотой использования), устойчивостью в работе, достаточно высокой эффективностью использования машинных ресурсов (низкая степень накладных «расходов»).

Для полномасштабного тестирования данного гипервизора и создаваемых им ВМ в условиях ресурсного центра грид-полигонов на 2-х управляющих машинах ресурсного сайта ИПХФ были установлены следующие сервисные виртуальные машины:

- для среды Globus Toolkit 4 (<http://www.globus.org>, полигон ГридННС): ОС CentOS 5.4, сервисы MDS, GRAM, GridFTP, RFT, User Interface
- для среды Unicore 6.2 (<http://www.unicore.eu>, СКИФ-Полигон) – ОС Ubuntu 9.10, сервисы: шлюз (Gateway); серверный контейнер (Unicore/X), интерфейс к целевой системе (TSI), авторизационный сервис и пользовательская база данных – XUUDBS\$, пользовательский интерфейс (UI);
- для Computing Element среды gLite (<http://glite.web.cern.ch>, российский сегмент EGI-RDIG) были размещены ОС (операционная система) ScientificLinux 4.5 и соответствующие сервисы типа lcg-ce, сервисов авторизации и мониторинга.

Выбор соответствующих ОС для управляющих узлов был обусловлен либо требованиями дистрибутивов распределенного ПО (как, например, для gLite), либо рекомендациями разработчиков, либо простотой администрирования. На все управляющие машины были установлены серверные компоненты свободно распространяемого пакета управления заданиями PBS/Torque (<http://www.clusterresources.com>), тогда как на расчетные узлы (под управлением ОС ScientificLinux 5.4 Boron) была установлена клиентская часть данного пакета. Поскольку все распределенные middleware требуют наличия своих собственных очередей PBS (Portable Batch System), было принято решение о настройке трех одновременно работающих экземпляров pbs\_tom на расчетных узлах с соответствующим набором очередей заданий. В таком варианте каждая управляющая машина связывается с расчетными узлами по уникальному порту и имеет дело только со своими заданиями.

Недостатком данного подхода является невозможность (пока) правильного учета ресурсов, используемых расчетным узлом, однако, для экспериментальных и исследовательских работ,

отладки прикладного ПО и проведения в меру ресурсоемких расчетов это вполне приемлемо. Явными преимуществами же являются:

- необходимость однократных установки и настройки ПО для решения входящих задач (особенно прикладных, поскольку часто установка прикладных пакетов оборачивается непредвиденными трудозатратами);
- простота администрирования расчетных узлов;
- экономия ресурсов (включая прежде всего электроэнергию);
- повышенный коэффициент загрузки за счет лучшей утилизации CPU.
- повышенная надежность ресурсного центра. В случае поломки физического узла копия виртуальной машины (размещенная на другом узле – файл-сервере емкостью в несколько терабайт) может быть запущена в считанные минуты (максимум в первые часы).

При наличии достаточного объема ресурсов (особенно оперативной памяти) все управляющие виртуальные машины могут быть размещены на одном физическом узле.

Отметим, что попутно на управляющих машинах можно также разместить дополнительные виртуальные машины, отвечающие за различные нересурсоемкие сервисы сети (например, web- или ftp-сервер, клиентские интерфейсы распределенных сетей, сервер баз данных и т.п.), поскольку их влияние на основные сервисы незначительно, при этом данные службы желательно изолировать от грид-узлов.

В качестве эксперимента на машине с 8 Гб оперативной памяти были размещены управляющие сервисы всех трех вышеуказанных ресурсных узла совместно с двумя машинами, обслуживающими внутрисетевые сервисы ИПХФ, что нисколько заметно не сказалась на качестве выполнения и доступности грид-сервисов, часть сервисов (для сред Globus Toolkit и Unicore) на базе ВМ продолжают эксплуатироваться в составе ресурсного центра ИПХФ.

Таким образом, в результате работ продемонстрирована возможность установки, запуска и работы виртуальных машин с поддержкой различных грид-сервисов на существующих физических узлах без вмешательства в рабочее пространство распределенных и/или параллельных сред ресурсных узлов. Это дает возможность разворачивать распределенные вычислительные полигоны на уже существующих вычислительных кластерах без кардинальной перенастройки их аппаратно-программной конфигурации, что особенно важно для постоянно работающих центров класса «production farm».

Наиболее важной проблемой использования виртуальных машин в качестве ресурсных становится правильный учет доступных для каждой грид среды ресурсов физического узла и мониторинг выполняемых ресурсным узлом задач (на всех входящих в него ВМ плюс хост-система). Учет ресурсов, востребованных ВМ, пока проводится на уровне гипервизора (т.е. только уровень загрузки ЦП) и не оценивается адекватно внешним мониторингом распределенной среды со стороны агентов мониторинга (собственных и внешних), что может вести к недоучету ресурсов и завышенным ожиданиям со стороны входящих задач. В планах развития большинства гипервизоров показана возможность решения этих проблем в дальнейшем, направленная на интеграцию аккаунтинга виртуальных машин для физического узла.

Таким образом, выбран и рекомендован к использованию свободно распространяемый гипервизор виртуальных машин – KVM, разработана методика одновременного размещения нескольких ресурсных сайтов разных распределенных полигонов (среды Globus Toolkit, gLite, Unicore) на едином физическом пространстве одного кластера. Подобная методика позволяет проводить обширные вычислительные эксперименты и разрабатывать сложные грид приложения для различных грид полигонов без вовлечения основных вычислительных ресурсов организации. Введение подобных вариантов использования кластеров с размещенными на нем виртуальными машинами позволяет участвовать в работе нескольких грид-полигонов, значительно повышает надежность управляющих узлов ресурсных сайтов, снижает трудоемкость администрирования сайтов, понижает расходы на эксплуатацию и поддержку грид-инфраструктуры.

## 2.2 Виртуализация грид-приложений

Другой стороной применения технологий виртуализации стала виртуализация исполняемых грид-заданий на основе прикладных программных пакетов, т.е. создание динамически формируемых образов исполняемых сред, или виртуальных «контейнеров», позволяющих про-

изводить запуски прикладных пакетов ПО вычислительной химии без предустановки и настройки на удаленных узлах распределенных сетей.

Данная технология была ранее успешно реализована авторами в грид-средах gLite и Unicore для бинарных приложений и квантово-химического прикладного программного пакета (ППП) GAMESS и детально описана в ряде статей [1,4,5], поэтому здесь приводится лишь краткое описание.

В 2009-2010 годах авторами был разработан метод создания виртуальных перемещаемых «контейнеров» (или, другими словами, – динамически формируемых исполняемых сред), для которых постулировано следующее содержание: «персональные» копии необходимых системных файлов и библиотек (в том числе, например, математических), скрипты по настройке операционной системы, необходимые файловые «деревья», собственно приложение, файлы данных и т.п.. В результате разработанной авторами оригинальной методики вычислений конечный грид-пользователь получил возможность формировать «виртуальное» (т.е. временное, создаваемое на время выполнения) приложение, которое в виде «виртуального контейнера» доставляется на ресурсный узел вместе со всеми конфигурационными настройками, относящимися к операционной системе, и поддержкой необходимых параллельных протоколов, не требуя процедуры предварительной установки и настройки. Далее «виртуальный контейнер» самостоятельно разворачивается на всех выделенных узлах грид-ресурса, подготавливая среду для исполнения параллельного приложения с последующим его запуском. При этом отсутствуют конфликты приложения с другими, уже установленными на узле программами и даже с другими экземплярами этого же приложения. Суть виртуализации приложения заключается в создании персональной копии необходимой части системных файлов и настроек операционной системы и доставке приложения совместно с этой информацией с последующим запуском в изолированном «контейнере». Проведенные авторами эксперименты в этой области показали, что так могут быть решены проблемы установки, настройки, несовместимости с операционной системой и другими программами, разрешаются конфликты одинаковых приложений.

В настоящее время этот метод применим для исполнения на узлах, поддерживающих ОС системы Linux, т.е. типичных кластерах, интегрированных в грид-среды.

В 2010 году программный пакет для работы с «виртуальными» контейнерами была адаптирован для работы с 64-битными вычислительными архитектурами, а также 64-битными версиями прикладных пакетов (для части которых разработчиками существенно была изменена схема параллелизации расчетов). В 2011 году было проведено успешное тестирование описанного способа виртуализации приложений в рамках среды Globus Toolkit на узлах полигона ГридННС (и ряда аналогичных по функциональности) с использованием «контейнера» для прикладного пакета GAMESS-US, что показало эффективность данного метода.

Серия первичных запусков (с использованием внутренних тестовых примеров собственно пакета GAMESS) вплоть до получения положительного результата тестов (равнозначность поведения сокетных и MPI вариантов) была проведена на ресурсных сайтах ИПХФ РАН для среды Globus Toolkit 4 (сайты использовались как удаленные, т.е. запуск задач шел через соответствующие инфраструктуры грид-полигона). Дальнейшее успешное тестирование было проведено на удаленных ресурсных узлах ГридННС в рамках ВО «NanoChem» (использовались узлы НИИЯФ МГУ и Курчатовского РНЦ). Были проведены успешные запуски пакета GAMESS-US с применением данной технологии. В качестве прикладной задачи с применением данной технологии были (по аналогии с предыдущими этапами) рассчитаны тестовые примеры молекулярных структур из дистрибутива GAMESS, например, серия ab initio расчетов по оптимизации геометрии в 15-атомной системе ( $P_3O_9H_3$ ) на уровне HF/6-31G), подтвердившие полную работоспособность разработанной технологии.

В настоящее время технология формирования «виртуальных контейнеров» для GAMESS-US интегрирована в GRID портал ИПХФ в высокоуровневый web-интерфейс по работе с GAMESS в распределенных средах Globus Tools, gLite, Unicore на пространстве всех доступных вычислительных полигонов, проведено успешное тестирование запусков «контейнеров» через web-портал на узлы ГридННС и аналогичных по функционалу грид-полигонах.

### 3. Технология работы с «пулами» (pools) независимых грид-заданий на равномерных «сетках» данных или параметров

В естественных науках (в том числе химии) существует огромное количество задач, которые представляют собой на самом деле совокупность независимых расчетных заданий, решение которых основано на переборе исходных данных или входных параметрах, а число зависит от количества параметров задачи или от «сетки» разбиения искомой области данных. Таковыми являются многопараметрические задачи вычислительной химии и химической физики, требующие последовательного перебора большого количества входных параметров. При этом полная задача разбивается на огромное количество независимых подзадач (каждая определяется группой значений совокупности параметров). Как вариант – решение задач на больших областях исходных данных, когда результат в каждой решаемой точке не зависит от «соседей». Задача автоматизации процесса разбиения полной задачи на фрагменты («нарезка») во многом аналогична таковой для многих кластерных задач и определяет детальность и полноту получаемого решения и удобство пользования системой. Типичный пример – туннельные реакции под воздействием электромагнитного излучения, где параметрами являются разные характеристики излучения. Задача имеет высокую вычислительную сложность, однако вычисления в каждой точке сетки в ней происходят независимо друг от друга, поэтому возможно разбить область вычислений на множество непересекающихся подобластей и каждую из них рассчитывать на разных узлах. Схожим примером могут служить траекторные расчеты химических реакций, например, реакция  $\text{H}_2 + \text{O}_2$ . Расчет ее представляет собой компьютерное моделирование с использованием классических траекторий элементарного акта столкновения. Для расчета полного сечения реакции для набора необходимой статистики следует рассчитать до десятков миллионов траекторий с учетом углов взаимной ориентации молекул, начальных колебательных и вращательных квантовых чисел, параметров столкновения и т.п. Как правило, расчет независимого задания в одном из узлов расчетной многомерной «сетки» занимает от нескольких минут до первых часов, однако, общее время расчета становится нереальным для одиночной вычислительной системы даже при высокой степени параллелизации выполнения задачи.

В последнее время помимо весьма простых заданий как вышеописанные, стало актуальным использование «тяжелых» прикладных пакетов типа GAMESS для расчета больших массивов точек (например, многомерных энергетических поверхностей), для которых нужно производить вычисления сотен и тысяч «точек» по «сетке» с закономерно изменяющимися параметрами.

Для решения подобных задач авторами ранее [2] была создана технология запуска «пучков» (или пулов) независимых заданий для использования всех доступных ресурсов распределенной среды с использованием грид технологий.

Технология предназначена для работы с большими заданиями на базе ППП или многопараметрических задач, которые могут быть представлены в виде объединения большого количества параллельно выполняемых независимых друг от друга задач. Созданный авторами программный комплекс обеспечивает разбиение первичной задачи по равномерным «сеткам» областей данных или параметров, формирование пула готовых к исполнению грид-заданий, их параллельного запуска в грид-среду, мониторинг и контроль выполнения участников пула, сборка результатов в конечный обобщающий результат.

Основные функции, выполняемые комплексом:

- разбиение первичной задачи на равномерных областях данных или входных параметров;
- автоматическое формирование пула независимых друг от друга грид-заданий вкуче с автоматически созданными файлами;
- параллельный запуск независимых заданий из пула в грид-среду;
- мониторинг и контроль выполнения заданий в рамках пула, включая останов и перезапуск по тайм-аутам;
- сборка результатов расчетов заданий с грид-ресурсов в конечный обобщающий результат.

Комплекс предоставляет средства одновременного запуска большого числа грид-заданий, в том числе с использованием адаптированных ППП (на примере пакета GAMESS) и широкого класса многопараметрических задач для проведения вычислений с использованием равномерных «сеток» данных или входных параметров ППП. Выполнение всех работ с пулами задания

должны происходить с использованием внутренней базы данных web-портала (MySQL или PostgreSQL) и внешних данных аккаунтинга и мониторинга, предоставляемых централизованными средствами грид-полигона.

Для разных распределенных сред разработана методика запуска подмножества независимых заданий (составляющих в совокупности вычислительную задачу) и получения результатов с удаленных ресурсных узлов. Первоначально метод был реализован в локальной гетерогенной вычислительной среде с использованием middleware Condor, что показало высокую эффективность метода. Затем метод был испытан на ряде задач в средах Nimrod и X-Com на географически распределенных вычислительных ресурсах. Далее, на языке Perl был написан комплекс скриптов для формирования «пулов» заданий, их запуска и получения результатов счета с использованием пользовательских интерфейсов (UI) сред gLite и Unicore. Для решения многопараметрических задач квантовой химии были разработаны методы формирования «пулов» независимых заданий с варьирующими параметрами – до  $10^4$ , в перспективе до  $10^7$  «атомарных» заданий на задачу. Для выбранных областей данных авторскими скриптами производится «нарезка» областей данных или входных параметров, формирование пулов независимых заданий, создание очередей запуска и отправки заданий на брокер ресурсов. После запуска периодически запускаемые средствами ОС (например, по cron) скрипты UI ведут мониторинг выполнения заданий (опрашивая брокер ресурсов или непосредственно удаленные ресурсные узлы), контроль тайм-аутов, перезапуск неудачных заданий и сбор результатов (с использованием базы данных и таблиц в ней, контролирующей состояние заданий – «ожидание», «запуск», «выполнение» и т.д.). По окончании расчетов проводится сборка «атомарных» результатов в единый выходной файл. Для части задач (требующих значительного числа параллельных независимых расчетов) в настоящее время создаются механизмы по разбиению областей данных (или расчетов) на большие независимые «подсетки» в форме независимых заданий, передачи всех их интерфейсам распределенных сред с последующим запуском на параллельных узлах и «сборки» финальных результатов из множества полученных независимых. Это позволяет достигнуть разумного баланса между собственно временем счета заданий и накладными расходами по передаче заданий по распределенной среде. Метод был протестирован на распределенных ресурсах ВО RGSTEST, СКИФ-полигона, ВО «Nanochem» полигона ГридННС в среде Globus.

Функционал комплекса работы с «пулами» заданий интегрирован также в Грид-портал ИПХФ РАН (<http://grid.icp.ac.ru>), пока только для стандартных многопараметрических задач.

#### **4. Реализация вычислительных грид-сервисов на основе прикладных пакетов с поддержкой GPU (Graphical Processor Units) устройств**

В последние годы произошел взрывной скачок интереса к параллельным расчетам с использованием высокоскоростных GPU (Graphical Processor Units) устройств, в том числе для прикладных пакетов в области квантовой химии и молекулярной динамики. Применение GPU вкуче со сменой парадигмы программирования прикладных пакетов во многих случаях ведет к ускорению расчетов от десятков процентов до первых порядков по времени. В настоящее время резко интенсифицировался перевод программного кода прикладных пакетов (в том числе – квантово-химических) на параллельные технологии с использованием технологий программирования CUDA<sup>tm</sup> и аналогичных. Это позволяет проводить высокоинтенсивные вычисления с применением встраиваемых в расчетные узлы кластеров высокопроизводительных графических процессоров (Nvidia, Tesla, AMD-ATI), что резко повышает эффективность использования параллельных методов расчетов и снижает как требования к расчетным центрам, так и операционную стоимость расчетов, особенно при использовании большого числа расчетных узлов. Во вновь создаваемых кластерах (в том числе ориентированных на использование в качестве грид-ресурсов) большинство расчетных узлов оснащается графическими адаптерами, что ставит задачу по использованию CUDA-ориентированных прикладных пакетов на грид-ресурсах весьма актуальной.

Использование массивно параллельных архитектур NVIDIA и Radeon GPU позволяет получать превосходные результаты при работе с приложениями в сфере квантовой химии и молекулярной динамики. По различным оценкам (приведена оценка Nvidia Group), выигрыш в во

времени расчетов для единичного узла составляет (указаны ППП, уже реализованные в ИПХФ в качестве грид-сервисов): GAMESS (метод самосогласованного поля) – до 50-60 раз, Gaussian (расчет потенциала Кулона) – 12-15 раз, VASP – 3-6 раз, NWChem – 3-8 раз, GROMACS (метод стице-ячеечной суммы Эвальда) – от 2 до 5 раз, LAMMPS (потенциалы Леннарда-Джонса, Гей-Берне) – в 6 раз, NAMD (расчет невалентных взаимодействий) – от 2 до 7 раз. Для прикладных пакетов же, которые создавались именно для CUDA вычислений (например, TeraChem и PetaChem - <http://www.petachem.com> или ‘Schrodinger Core Hopping’) выигрыш по времени может составлять 50-5000 раз.

В ИПХФ РАН начаты методические исследования возможности использования новых GPU-оптимизированных программных вариантов различных прикладных квантово-химических пакетов как на локальных кластерах ИПХФ, так и в рамках грид-сайтов различных грид-сред. Проводится тестирование и оптимизация имеющихся вариантов ППП с поддержкой CUDA вычислений с последующей адаптацией их к проведению расчетов в грид-средах. Проводится оценка повышения эффективности расчетов в условиях грид-ресурсов. Начата разработка методики запуска грид-заданий, ориентированных на поиск и использование ресурсов, поддерживающих использование CUDA технологий.

## 5. Заключение

Применение новых технологий применительно к грид-вычислениям с использованием прикладных программных пакетов в области квантовой химии и молекулярной динамики позволит существенно повысить эффективность использования грид-ресурсов (включая экономию затрат на эксплуатацию и трудоемкость обслуживания), повысить совместимость прикладного ПО с гетерогенными ресурсами грид-полигонов, заметно ускорить решение многих задач, вплоть до того, что возникает возможность ставить и решать вычислительные задачи фундаментального и прикладного характера в области химических наук, ранее не доступные из-за ограниченности возможностей вычислительных ресурсов. Основные научные области применения – химическая физика, квантовая химия, исследование наноструктур, молекулярная динамика, биохимия, фармацевтика, разработка топливных элементов и близкие отрасли наук.

## Литература

1. В.М. Волохов, Д.А. Варламов, А.В. Пивушков, Н.Ф. Сурков, А.В. Волохов Динамически формируемые параллельные среды в условиях грид-полигонов, проблемы и решения // «Вычислительные методы и программирование: Новые вычислительные технологии», М.: МГУ, 2011, т.12, № 1, с.39-45
2. Волохов В.М., Варламов Д.А., Пивушков А.В., Волохов А.В. Способы решения прикладных многопараметрических задач вычислительной химии на пета- и эксафлопных системах // Международная суперкомпьютерная конференция «Научный сервис в сети Интернет: эксафлопное будущее», 19-24 сентября 2011, Новороссийск – М.: Изд-во МГУ, 2011 с.382-384
3. Волохов В.М., Пивушков А.В., Волохов А.В., Варламов Д.А. Реализация нескольких независимых ресурсных грид-сайтов на едином физическом пространстве кластера // X международная конференция «Высокопроизводительные параллельные вычисления на кластерных системах» НРС - 2010, Пермь, ноябрь 2010; изд-во ПГТУ, том 1, стр. 119-124
4. Волохов В.М., Варламов Д.А., Сурков Н.Ф., Пивушков А.В. Виртуальные вычислительные среды: использование на GRID полигонах // Вестн. ЮУрГУ, серия «Математическое моделирование и программирование», 2009, № 17 (150), вып. 3, с.24-35
5. Варламов Д.А., Сурков Н.Ф., Волохов В.М., Пивушков А.В. Виртуализация вычислительной среды в ГРИД // «Параллельные вычислительные технологии 2010» ПаВТ-2010, (Уфа, март 2010), Челябинск, изд-во ЮУрГУ, с.63-70