УДК 004.75

СПОСОБ ЗАПУСКА И ОБРАБОТКИ В ГРИДЕ ЗАДАНИЙ, ПОДГОТОВЛЕННЫХ ДЛЯ РАЗЛИЧНЫХ СРЕД ИСПОЛНЕНИЯ

В. А. Ильин[1], **А. П. Крюков**[1], **Л. В. Шамардин**[1], **А. П. Демичев**[1], **И. Н. Горбунов**[1]

Предложен способ пакетной обработки в грид-среде вычислительных заданий, подготовленных для различных сред исполнения. Способ основан на технологии виртуализации рабочих узлов территориально распределенных ресурсных центров и позволяет существенно расширить класс прикладных задач, решаемых с помощью глобальных грид-инфраструктур. Работа выполнена при поддержке РФФИ (проект № 07–07–12023-офи) и Федерального агентства по науке и инновациям (государственный контракт № 02.514.11.4072).

1. Введение. Грид-среда позволяет объединить компьютерные ресурсы территориально распределенных ресурсных центров при помощи промежуточного программного обеспечения (ППО). Это ППО позволяет распределять задания по таким центрам, возвращать результаты пользователю, контролировать права пользователей на доступ к тем или иным ресурсам, а также осуществлять мониторинг ресурсов (см., например, [1]).

В настоящее время грид-системы используются для вычислений и обработки данных в самых разных прикладных областях, например в биомедицине, нанотехнологиях и материаловедении, космофизике и физике высоких энергий, а также в ряде промышленных и коммерческих областей. Однако одна из основных проблем, стоящих на пути широкого использования грид-систем, связана с тем, что, как правило, прикладные задания разрабатываются для исполнения во вполне конкретной вычислительной среде, характеризующейся типом и версией операционной системы (ОС), вспомогательным программным обеспечением (библиотеками), типом файловой системы, наличием или отсутствием аппаратных и программных средств для параллельного выполнения частей программы. С другой стороны, рабочие узлы ресурсных центров (где и выполняются задания) работают, как правило, под управлением определенной ОС и предоставляют некоторую фиксированную среду исполнения. Поэтому, если прикладные задачи изначально не разрабатываются для среды исполнения, предоставляемой рабочими узлами, то они не могут непосредственно выполняться в гриде.

Опыт применения научных грид-приложений, а также примеры использования научных грид-систем в промышленности, в финансовой и административной сферах показывают, что для широкого внедрения грид-технологий в этих сферах некоторые компоненты грид-инструментария либо недостаточно развиты, либо просто отсутствуют. Прежде всего это связано с тем, что крупные грид-системы (типа EGEE [2] и OSG [3]) до настоящего времени использовались в основном в рамках крупных научных проектов, например в области физики высоких энергий: для моделирования методом Монте-Карло и обработки данных с крупнейшего ускорителя элементарных частиц “Большой адронный коллайдер”, строящегося в Европейском центре ядерных исследований (ЦЕРН, Женева, Швейцария). Для таких крупных международных проектов является вполне допустимым, что все рабочие узлы грид-системы предоставляют стандартизованную среду исполнения, поскольку специальное прикладное программное обеспечение для решения уникальных задач таких проектов разрабатывается именно для использования в данной грид-среде.

Такая ситуация не является приемлемой при использовании грид-систем для решения различных задач инженерно-научных, промышленно-инновационных, административных, финансовых и коммерческих (особенно в случае малого и среднего бизнеса) проектов. Зачастую пользователи из указанных областей уже имеют программное обеспечение (ПО) для решения своих прикладных задач или хотят воспользоваться имеющимся на рынке готовым ПО и им лишь нужны вычислительные мощности (грид-ресурсы) для осуществления вычислений требуемого объема. При этом технологические, временны́е, лицензионные, финансовые и другие ограничения в большинстве случаев не позволяют адаптировать имеющееся прикладное ПО для среды исполнения, которую предоставляют рабочие узлы грида. Это обусловливает

[1] Научно-исследовательский институт ядерной физики им. Д. В. Скобельцына, Московский государственный университет им. М. В. Ломоносова (НИИЯФ МГУ), Ленинские горы, д. 1, стр. 2, 119991, Москва; e-mail: demichev@theory.sinp.msu.ru

актуальность разработки технологии предоставления сред исполнения заданий по запросам пользователей для широкого внедрения грид-технологий в различные сферы деятельности.

В рамках данной работы предложен простой подход, основанный на виртуализации рабочих узлов грида, который обеспечивает возможность запуска и пакетной обработки в грид-системе заданий, подготовленных для различных вычислительных сред. Это дает возможность выполнения приложений в гриде независимо от того, для какой вычислительной среды оно было изначально разработано. Например, приложения, разработанные для выполнения в среде широко распространенной операционной системы (ОС) Windows, могут выполняться в тех ресурсных центрах грид-системы, рабочие узлы которых функционируют под управлением ОС Linux.

Предлагаемый подход основан на использовании виртуальных машин (ВМ). В настоящее время ВМ (см., например, [4]) являются одним из основных средств оптимизации использования компьютерных и сетевых ресурсов. Дополнительное потребление вычислительных ресурсов, связанное с развертыванием собственно виртуальной машины, не превышает нескольких процентов. Время исполнения заданий также увеличивается незначительно по сравнению с их исполнением без использования ВМ. С другой стороны, качество услуг, предоставляемых пользователям грид-систем, существенно улучшается за счет расширения класса выполняемых задач и повышения уровня безопасности.

Предлагаемая технология виртуализации рабочих узлов в первую очередь предназначена для использования в глобальной грид-инфраструктуре EGEE [2] и в ее российском сегменте РДИГ [5]. Эти грид-системы основаны на ППО gLite [6]. Однако разработанные компоненты, позволяющие разворачивать на рабочих узлах виртуальные среды исполнения, фактически взаимодействуют только с грид-шлюзом ("Computing element" в терминах ППО gLite) к вычислительным ресурсам, основным компонентом которого является Grid Resource Allocation Manager (GRAM) в составе широко распространенного грид-инструментария Globus Toolkit (GT) [7]. Поэтому разработанный в рамках данной работы способ может использоваться практически в любой грид-системе, основанной на GT.

2. Существующие подходы к предоставлению различных сред исполнения в гриде. Возможность предоставления различных сред исполнения в грид-инфраструктурах может достигаться различными средствами. Одна из возможностей — это объединение в рамках единой грид-инфраструктуры рабочих узлов, работающих под управлением различных операционных систем, что обеспечивает некоторый набор возможных сред исполнения заданий. Реализация этого подхода основана на динамическом создании пула формальных локальных пользователей, причем различные группы таких пользователей имеют доступ к учетным записям с различными средами исполнения (в частности, такое направление развивается как "инкубаторное" [8] в рамках Globus Alliance). Большим недостатком такого подхода является малая гибкость системы (фиксированный и, как правило, небольшой набор сред), а также то, что задания могут исполняться только на части грид-ресурсов (с соответствующей средой исполнения). Кроме того, администрирование (обслуживание) такой неоднородной системы является весьма неудобным.

Существенно более предпочтительным является подход, основанный на использовании виртуальных машин (ВМ). Подход, основанный на ВМ, требует существенной адаптации существующего ППО, в частности, подсистем запуска заданий, мониторинга и учета использования ресурсов в грид-инфраструктурах. Тем не менее такой подход обеспечивает более сильную изоляцию среды от сред исполнения других пользователей грида, бóльшую гибкость (с точки зрения пользователя), возможность "замораживания" выполнения задания на промежуточном этапе и перемещения работы на другой ресурс, поддержку системы соглашений между пользователями и провайдерами по уровню обслуживания и ряд других функциональных преимуществ. Особенно привлекательным представляется использование так называемой паравиртуализации [4], которая гарантирует относительно небольшое дополнительное потребление компьютерных ресурсов при работе ВМ. Среди продуктов, обеспечивающих паравиртуализацию, наиболее зрелым продуктом является Xen [9]. Хотя немодифицированная ОС Windows не может работать в режиме паравиртуализации, благодаря быстро развивающейся технологии аппаратной поддержки виртуализации [10] с помощью Xen можно разворачивать и эту операционную систему.

Этот подход на основе ВМ в последнее время привлекает повышенное внимание различных исследовательских групп (см., например, [11, 12]). В частности, в рамках проекта In-VIGO [13] разрабатывалась распределенная грид-инфраструктура на основе ВМ в качестве ресурсов, в проекте COD [14] разрабатывался способ динамического развертывания рабочего пространства для авторизованных удаленных пользователей, в проектах Virtuoso [15] и Violin [16] исследовались сетевые проблемы, связанные с использованием виртуальных машин в грид-среде.

Наиболее разработанным среди таких проектов и наиболее близким к целям настоящей работы явля-

ется проект Virtual Workspaces [17], осуществляемый в рамках сообщества Globus Alliance. В рамках этого проекта создается специализированный грид-сервис Virtual Workspace (VW) для разворачивания и конфигурирования виртуальных машин на рабочих узлах грид-системы, а также для взаимодействия с ними авторизованных пользователей и администрирования со стороны локальных грид-администраторов. На текущий момент доступна тестовая версия Virtual Workspace 1.3.1 (Technology Preview). VW-сервис имеет WSRF-совместимый [18] внешний интерфейс для загрузки виртуальных машин на компьютеры кластера и управления процессом их работы. При этом управление может осуществляться как одной, так и целым набором ВМ (кластером ВМ). Для полномасштабной инсталляции сервиса требуется два выделенных компьютера и набор компьютеров, выполняющих роль рабочих узлов грид-системы (при инсталляции в тестовых целях набор компьютеров можно сократить вплоть до одного). Один выделенный компьютер служит как файловый сервер — хранилище (репозиторий) образов ВМ. На втором выделенном компьютере устанавливаются VW-сервер и базовые компоненты Globus Toolkit версии 4 (GT4). На компьютере пользователя должна быть установлена клиентская программа VWS. С ее помощью пользователь посылает запрос "фабрике" VW-сервисов на развертывание виртуальной машины с требуемой средой исполнения и на создание собственного экземпляра службы для управления этой ВМ и получения информации о ее состоянии. Вместе с запросом посылаются два набора конфигурационных данных: метаданные — общие параметры ВМ (не зависят от деталей конкретного запуска ВМ) и запрос на развертывание — детальные параметры текущего запуска ВМ.

Необходимо учитывать, что Virtual Workspace в основном предназначен для интерактивного режима работы пользователя с разворачиваемой виртуальной средой. В частности, он может использоваться как портал [19] для Amazon Elastic Compute Cloud [20]. С другой стороны, в больших грид-системах основным (а чаще всего — единственно возможным) режимом является пакетный запуск заданий. Это связано с тем, что только в таком режиме возможно эффективное планирование и оптимальное распределение грид-ресурсов. Другой вариант — интерактивное взаимодействие пользователей с ресурсами — приводит к большим проблемам при планировании выделения ресурсов, высоким требованиям к уровню обслуживания (Quality of Service, QoS), обеспечению резервирования ресурсов и др. Хотя грид-системы с интерактивным взаимодействием рассматриваются в литературе и исследуются в ряде проектов, такие грид-системы предполагаются сравнительно небольшими и специализированными для выполнения узкого круга задач, для которых интерактивность является обязательным условием.

В настоящей статье предлагается система запуска в пакетном режиме заданий, подготовленных для различных сред исполнения. Как уже указывалось, эта система предназначена, в первую очередь, для использования в глобальной грид-инфраструктуре EGEE [2] и в ее российском сегменте РДИГ [5] и поэтому представляет собой надстройку над ППО gLite [6].

3. Архитектура и алгоритм работы системы запуска в гриде заданий, подготовленных для различных сред исполнения. Система запуска заданий для различных сред исполнения (СЗЗ-РСИ) должна взаимодействовать с другими системами и модулями грид-инфраструктуры, в первую очередь с системой управления загрузкой грид-ресурсов (Workload Management System (WMS) в терминологии ППО gLite) и с ее центральным компонентом — менеджером загрузки (Workload Manager, WM). На рис. 1 схематически показана совместная архитектура подсистемы управления загрузкой и СЗЗ-РСИ (модули, содержащие компоненты СЗЗ-РСИ, указаны жирным шрифтом).

Описываемая архитектура СЗЗ-РСИ предполагает, что рабочим узлам грид-инфраструктуры доступен набор образов дисков с различными средами исполнения (они могут находиться на локальных дисках рабочих узлов или быть доступными по NFS либо по другим стандартным протоколам). Функциональные возможности менеджера загрузки ППО gLite достаточны для нахождения грид-ресурсов с возможностью запуска заданий в различных средах исполнения. Это связано с тем, что информацию о требуемой среде исполнения можно представить в рамках стандартной GLUE-схемы [21], используемой в ППО gLite. Для поиска необходимых ресурсов в описании задания на языке JDL [22] должна быть указана требуемая среда исполнения. Напротив, вычислительные элементы (Computing element, CE) ППО gLite (грид-шлюзы к рабочим узлам) и ППО собственно рабочих узлов (Working Node, WN) требуют модификации. Упрощенная архитектура этих компонентов представлена на рис. 2.

Общий сценарий использования системы запуска заданий, подготовленных для различных сред исполнения, выглядит так:

- пользователь в описании задания на языке JDL указывает ПО-тег, соответствующий развертыванию ВМ с необходимой средой исполнения;
- WMS с помощью информационной системы находит ресурсный центр с CE, имеющим такой ПО-тег, а также со свободными рабочими узлами и направляет туда задание;

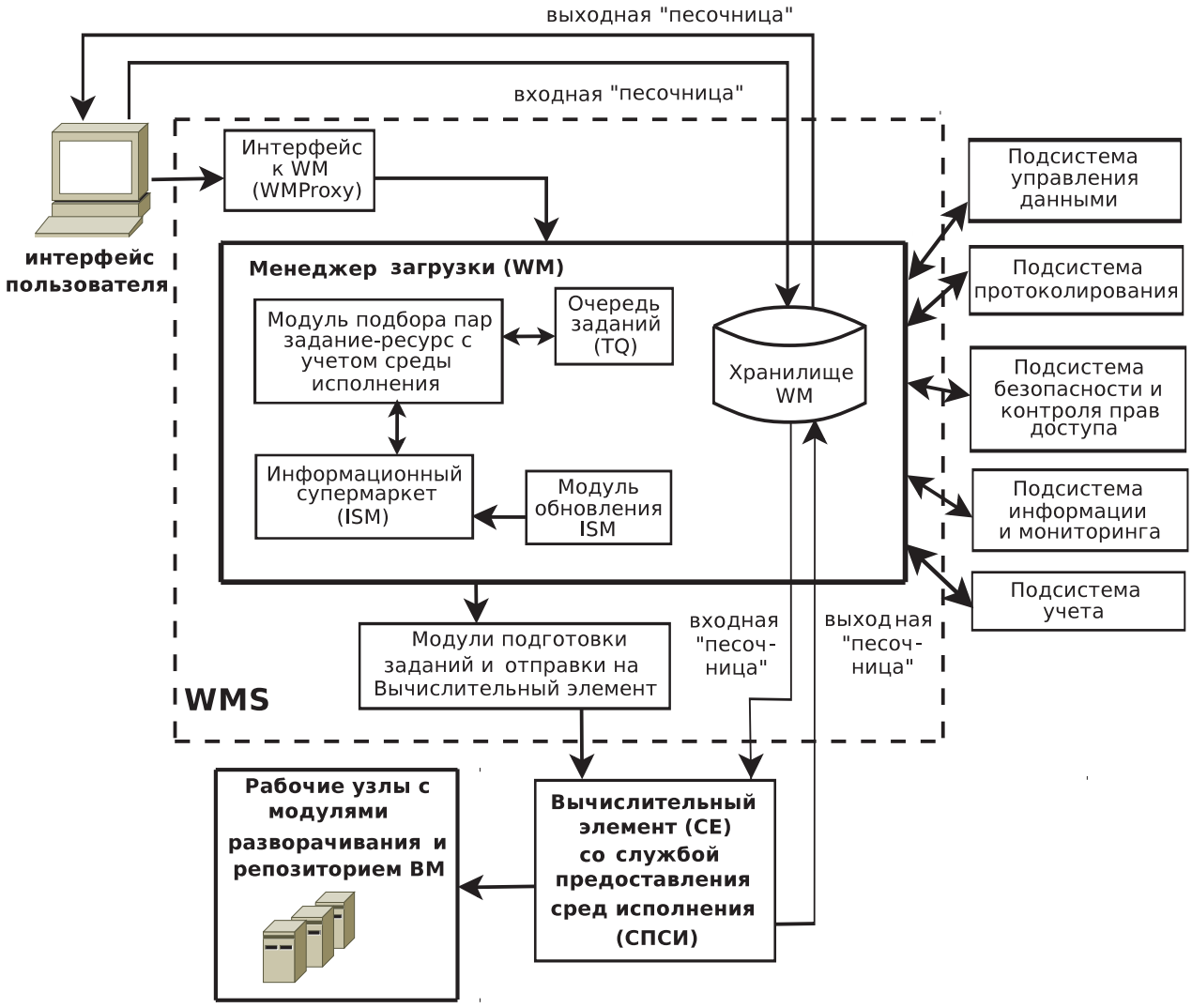

Рис. 1. Совместная архитектура подсистемы управления загрузкой и СЗЗ-РСИ

- на СЕ задание обрабатывается с участием службы предоставления сред исполнения (СПСИ) и "обертывается" специальным скриптом;
- при запуске на рабочем узле с установленным монитором виртуальных машин (МВМ) обертывающий скрипт:
  - монтирует копию образа требуемой среды к файловой системе рабочего узла (рабочие узлы ППО gLite функционируют под управлением ОС Scientific Linux);
  - записывает входную "песочницу" (sandbox) задания в файл образа;
  - обеспечивает запуск задания после разворачивания ВМ (например, для гостевой ОС Windows соответствующим образом дополняет файл autoexec.bat);
  - осуществляет размонтирование файла образа;
  - запускает ВМ и разворачивает файл образа требуемой среды исполнения;
  - после окончания выполнения задания останавливает ВМ;
  - вновь монтирует файл образа и копирует результаты в выходную "песочницу" на рабочем узле;
  - уничтожает использованную копию образа на локальном диске;
  - при необходимости получения или передачи данных в процессе выполнения задания используется клиентская программа GridFTP [23] (существует Java-версия этой программы [24], пригодная для использования в различных ОС);

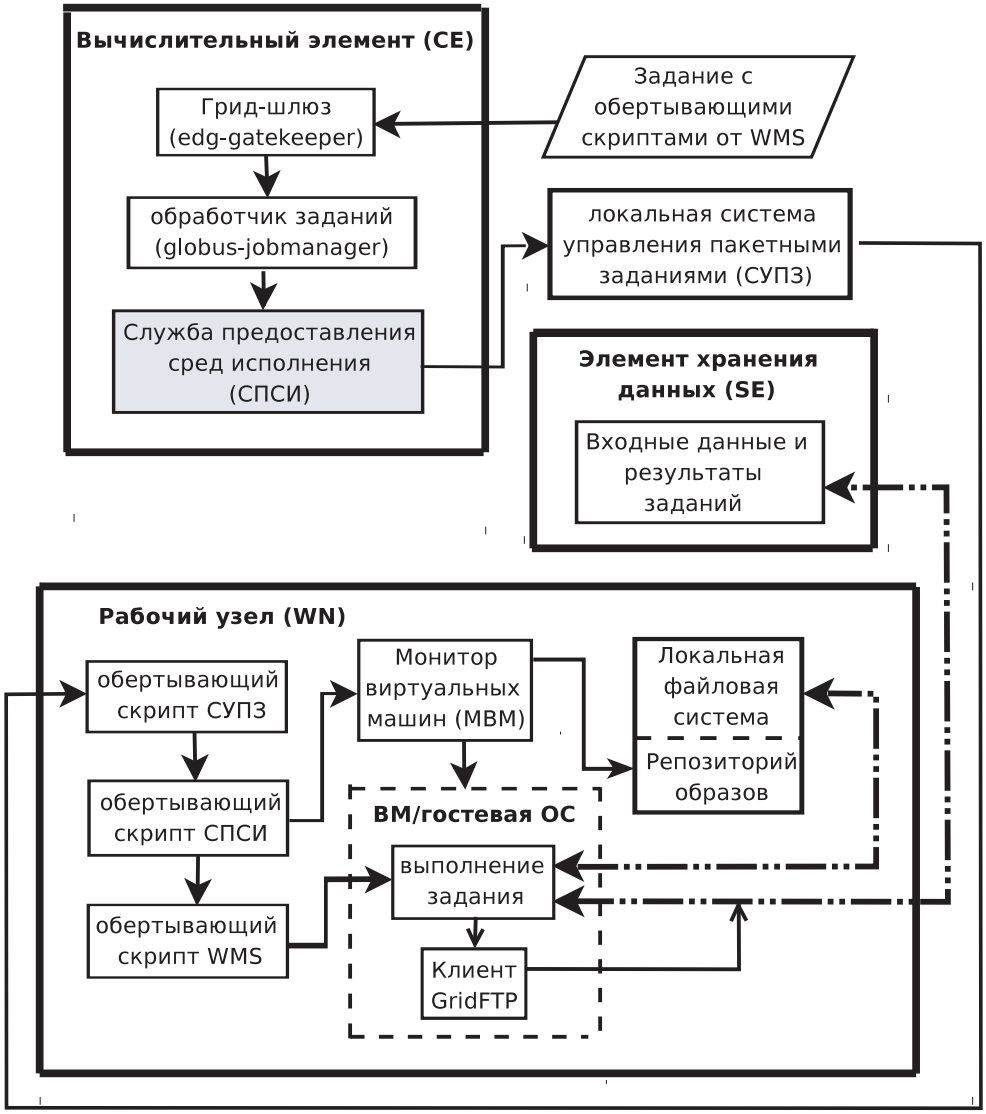

Рис. 2. Упрощенная архитектура модифицированных вычислительного элемента и рабочего узла для запуска в грид заданий, подготовленных для различных сред исполнения

- результаты задания (выходная "песочница") доставляются пользователю стандартными средствами gLite.

Кроме того, обертывающий скрипт следит за тем, чтобы развернутая ВМ корректно уничтожалась, если время выполнения задания превысило время, выделенное системой управления пакетными заданиями кластера рабочих узлов. Графически алгоритм работы этих модифицированных компонентов представлен на рис. 3.

4. Заключение. Рассмотренная в настоящей статье модифицированная система запуска заданий ППО gLite существенно расширит круг пользователей грид-систем, прикладные задачи которых разработаны для исполнения в средах, отличных от исходной среды на рабочих узлах ресурсных грид-узлов (под средой понимается конкретная ОС, библиотеки программ и др.). Разработка подобной технологии является критически важной при создании грид-инфраструктур, предоставляющих компьютерные услуги для решения вычислительных задач и задач по анализу и обработке данных в различных научно-инженерных и инновационно-промышленных исследованиях, таких как нанотехнологии, термоядерная энергетика, биомедицина, науки о Земле, дистанционное зондирование Земли и др. Широкий спектр требований, предъявляемых к предоставляемым вычислительным услугам со стороны пользователей, работающих в этих областях, может быть удовлетворен с помощью внедрения технологии виртуализации в современную инфраструктуру грид. Это явится важным шагом в развитии грид-технологии и выходом ее на уровень коммерческого применения.

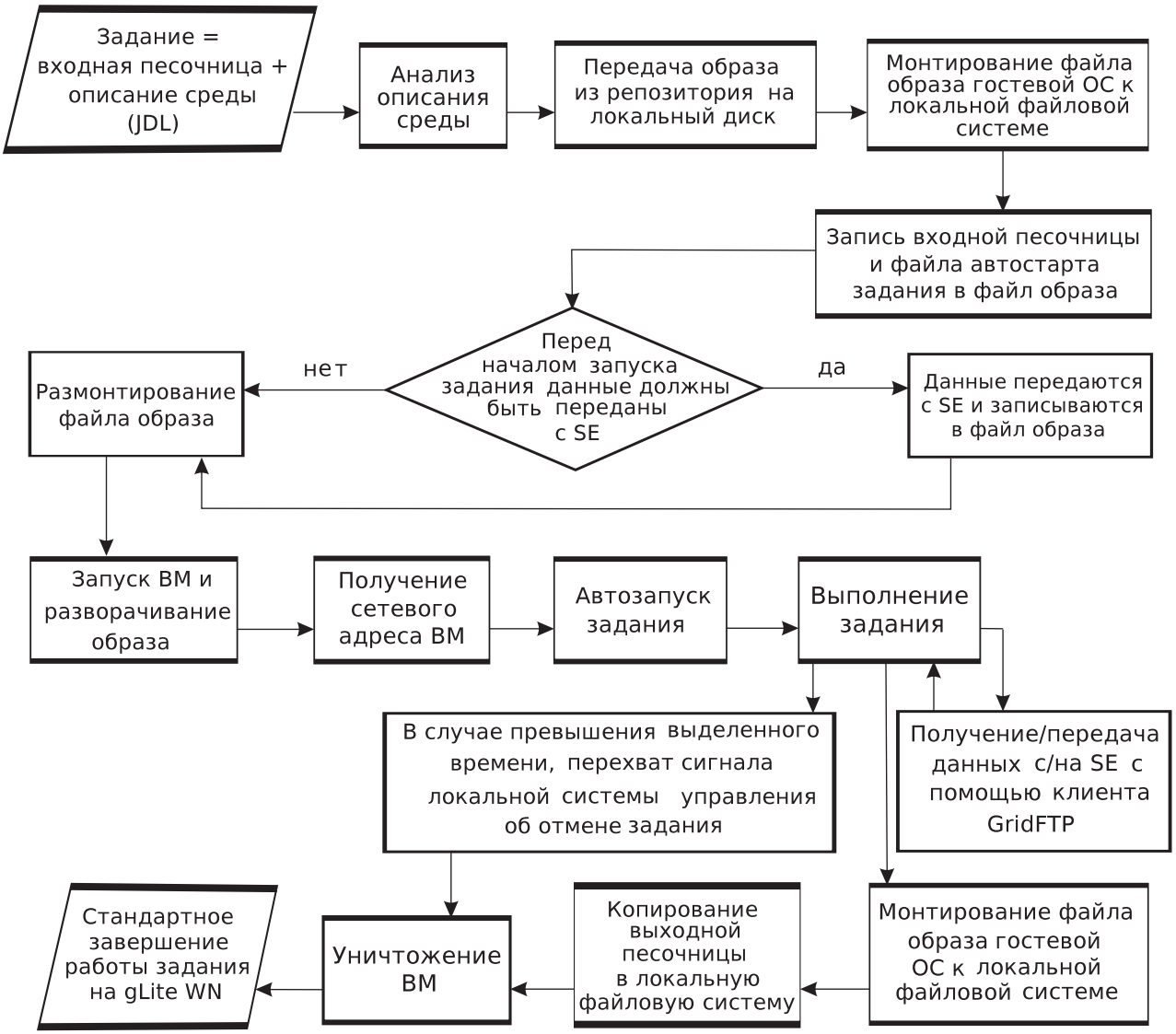

Рис. 3. Алгоритм выполнения задания в среде по запросу пользователя

Описание реализации предложенного способа запуска в грид заданий, подготовленных для различных сред исполнения, и результатов эксплуатации такой системы будет представлено в отдельной публикации.

СПИСОК ЛИТЕРАТУРЫ

1. *Демичев А.П., Ильин В.А., Крюков А.П.* Введение в грид-технологии. Препринт НИИЯФ МГУ № 2007–11/832. М., 2007 (http://dbserv.sinp.msu.ru:8080/sinp/files/pp-832.pdf).
2. Проект EGEE (http://www.eu-egee.org).
3. Проект OSG (http://www.opensciencegrid.org).
4. *Jones M.T.* Virtual Linux. An overview of virtualization methods, architectures, and implementations. 2006 (http://www-128.ibm.com/developerworks/linux/library/l-linuxvirt).
5. Проект RDIG (http://egee-rdig.ru).
6. Промежуточное программное обеспечение gLite (http://glite.web.cern.ch/glite).
7. Globus Alliance (http://www.globus.org).
8. Проект Incubator Dynamic Accounts (http://dev.globus.org/wiki/Incubator/Dynamic_Accounts).
9. Проект Xen (http://www.xen.org/).
10. *Александров А.* Спирали аппаратной виртуализации // Открытые системы. 2007. № 3. 12–19.
11. *Keahey K., Doering K., Foster I.* From sandbox to playground: dynamic virtual environments in the grid // Proc. of the 5th Int. Workshop in Grid Computing. Pittsburgh, 2004. pp. 190–199.
12. *Krsul I., Ganguly A., Zhang J., Fortes J., Figueiredo R.* VMPlants: providing and managing virtual machine ex-

ecution environments for grid computing // Proc. of the ACM/IEEE SC2004 Con. Pittsburgh, 2004. pp. 7–19 (http://ieeexplore.ieee.org/iel5/9595/30315/01392937.pdf).
13. Проект In-VIGO (http://invigo.acis.ufl.edu).
14. *Chase J., Irwin D.E., Grit L.E., Moore J.D., Sprenkle S.E.* Dynamic virtual clusters in a grid site manager // Proc. of the 12th IEEE Int. Symp. on High Performance Distributed Computing. Seattle, 2003. pp. 90–100.
15. *Sundararaj A., Dinda P.* Towards virtual networks for virtual machine grid computing // Proc. of the 3rd USENIX Conf. on Virtual Machine Technology. San Jose, 2004. pp. 14–29.
16. *Jiang X., Xu D.* VIOLIN: Virtual Internetworking on OverLay INfrastructure. Department of Computer Sciences Technical Report CSD TR 03-027. Lafayette: Purdue University, 2003.
17. Проект Incubator/Virtual Workspaces (http://workspace.globus.org; http://dev.globus.org/wiki/Incubator/Virtual_Workspaces).
18. Спецификации WSRF (http://www.globus.org/wsrf).
19. Virtual Workspace как портал для Amazon Elastic Compute Cloud (http://workspace.globus.org/vm/TP1.3.1/iaq.html).
20. Проект Amazon Elastic Compute Cloud (http://www.amazon.com/b/?node=201590011).
21. Схема GLUE (http://forge.ogf.org/sf/projects/glue-wg).
22. Job Description Language HowTo. DataGrid-01-TEN-0102-0_2 (http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-0_2-Document.doc).
23. Грид-сервис передачи данных GridFTP (http://www.globus.org/toolkit/data/gridftp).
24. Java-версия клиентской части GridFTP (http://www-unix.globus.org/cog/jftp/guide.html).

Поступила в редакцию
24.03.2008