

О.В. Сухорослов**ОРГАНИЗАЦИЯ ВЫЧИСЛЕНИЙ В ГЕТЕРОГЕННЫХ
РАСПРЕДЕЛЕННЫХ СРЕДАХ***

Современные научные исследования неразрывно связаны с проведением сложных расчетов и использованием высокопроизводительных вычислительных ресурсов. В настоящее время в распоряжении исследователей имеется целый спектр подобных вычислительных ресурсов, включая отдельные сервера и рабочие станции, кластеры и суперкомпьютеры, грид-системы и облачные инфраструктуры. Каждый из данных типов ресурсов имеет свои особенности, преимущества и недостатки. Совместное использование вычислительного потенциала данных ресурсов в рамках гетерогенных распределенных вычислительных сред (ГРВС) позволило бы многократно повысить производительность исследований и эффективность использования ресурсов. В данной статье делается попытка в максимально широкой постановке описать проблематику организации вычислений в ГРВС, включая основные типы ресурсов и их характеристики, разновидности и сценарии использования ГРВС, классы приложений и предъявляемые пользователями требования. Под ГРВС подразумеваются произвольные сочетания распределенных вычислительных ресурсов различного типа. Данные ресурсы могут существенно отличаться друг от друга по своим характеристикам, механизмам доступа, административной принадлежности, кругу обслуживаемых пользователей, периодами доступности, стоимости, политиками и уровнями качества обслуживания. При этом сами ГРВС могут иметь временный (под решение одной задачи) или постоянный характер, а также отличаться по числу пользователей (персональные или многопользовательские среды) и кругу поставщиков ресурсов. За последние десятилетия существенно расширился спектр и усложнилась структура ГРВС. Появились новые типы ресурсов, сами по себе представляющие ГРВС, и позволяющие строить новые среды на более высоком уровне абстракции. При этом возникли новые требования к методам и средствам управления вычислениями в ГРВС, такие как, например, оптимизация расходов на облачные ресурсы при соблюдении заданных сроков на проведение вычислений. Высокая степень неоднородности и динамичный характер ГРВС, массовое распространение облачных сервисов, разнообразие сценариев использования и видов приложений требуют развития имеющихся и создания новых подходов к решению задач, связанных с организацией вычислений в подобных средах.

Распределенные вычисления; гетерогенные вычислительные среды; интеграция вычислительных ресурсов; грид; облачные вычисления.

O.V. Sukhoroslov**ORGANIZATION OF COMPUTATIONS IN HETEROGENOUS DISTRIBUTED ENVIRONMENTS**

Modern scientific research is inextricably linked to the carrying out of complex calculations and the use of high performance computing resources. Currently available to researchers, there are a variety of such computing resources, including servers and the individual workstations, clusters and supercomputers, grid systems and cloud infrastructures. Each of these types of resources has its own characteristics, advantages and disadvantages. Sharing computing capacity of these resources within the heterogeneous distributed computing environments (HDCE) would significantly increase research productivity and resource efficiency. This article is an attempt to describe in the most general formulation the problems of organization of computations in HDCE, including the main types of resources and their characteristics, types of environments and their use, classes of applications and user requirements. HDCE refers to arbitrary combinations of distributed computing resources of various types. These resources can significantly differ from each other in their

* Работа выполнена при поддержке РФФИ (проект № 15-29-07068).

characteristics, access mechanisms, administrative affiliation, groups of served users, availability periods, costs, policies and quality of service levels. HDCE can have temporary (for solution of a certain problem) or permanent nature, and vary according to the number of users (personal or multiuser environments) and the range of resource providers. Over the past decade, the range of HDCE is significantly expanded while their structure has become more complex. There are new types of resources being HDCE themselves and allowing building new environments at a higher level of abstraction. At the same time there are new requirements to the methods and tools used for management of computations in HDCE, such as optimization of expenses for renting cloud computing resources while meeting the specified deadline to carry out the computations. The high degree of heterogeneity and the dynamic nature of HDCE, the proliferation of cloud services, a variety of usage scenarios and types of applications require the improvement of existing and development of new approaches to solving problems related to the organization of computations in such environments.

Distributed computing; heterogeneous computing environments; integration of computing resources; grid; cloud computing.

Введение. Под *гетерогенной распределенной вычислительной средой (ГРВС)* будем понимать совокупность независимо функционирующих вычислительных ресурсов, существенно отличающихся друг от друга по своим характеристикам или типу. В состав ГРВС также обычно включают управляющие и служебные компоненты для организации вычислений в рамках среды как единого виртуального ресурса. Основными целями использования ГРВС являются увеличение доступной производительности для решения больших задач, повышение эффективности использования имеющихся ресурсов и интеграция ресурсов участников совместного проекта.

Наиболее хорошо изученным примером ГРВС являются простые системы типа *кластера*, состоящие из нескольких машин различной производительности, как правило находящихся в одной локальной сети. Развитие глобальных сетей привело к появлению в 1990-х годах концепции *метакомпьютинга* [1–3], основанной на идее объединения по сети ресурсов суперкомпьютерных центров для проведения крупномасштабных расчетов. В конце 1990-х годов возникли системы *добровольных вычислений* [4–5], объединяющие простаивающие ресурсы большого числа персональных компьютеров в рамках глобальной сети. Дальнейшим развитием идей метакомпьютинга стала концепция *грид-вычислений* [6–8], приведшая к появлению в 2000-х годах грид-инфраструктур, объединяющих ресурсы географически распределенных вычислительных центров. Сложность подобных инфраструктур послужила толчком к созданию новых высокоуровневых инструментариев и проблемно-ориентированных сред, поддерживающих проведение расчетов на распределенных ресурсах [9–11]. Конец 2000-х годов ознаменовался появлением облачных вычислительных сервисов, предоставляющих удобный доступ к практически неограниченным и гибко настраиваемым ресурсам в рамках концепции *облачных вычислений* [12–13]. В дальнейшем данная концепция была перенесена внутрь отдельных организаций в виде так называемых *частных облаков*. Также органичным образом появилась концепция *cloud bursting* [14] как способа комбинированного использования внутренних ресурсов и ресурсов облака. Модели облачных вычислений и лежащий в их основе сервисно-ориентированный подход нашли широкое применение при реализации нового поколения систем, ориентированных на поддержку решения различных классов научных задач на гетерогенных распределенных вычислительных ресурсах [15–20].

Таким образом, за последние десятилетия существенно расширился спектр и усложнилась структура ГРВС. Появились новые типы ресурсов, сами по себе представляющие ГРВС, и позволяющие строить новые среды на более высоком уровне абстракции. Помимо фиксированной производительности ресурсов, при

организации вычислений в современных ГРВС необходимо учитывать целый ряд дополнительных характеристик, таких как доступность, надежность, стоимость, механизмы доступа, качество обслуживания и т.д., в том числе изменяющихся во времени. При этом возникли новые требования к методам и средствам управления вычислениями в ГРВС, такие как, например, оптимизация расходов на облачные ресурсы при соблюдении заданных сроков на проведение вычислений.

В данной статье делается попытка систематизированно и в максимально широкой постановке описать проблематику организации вычислений в ГРВС, включая основные типы ресурсов и их характеристики, разновидности и сценарии использования ГРВС, классы приложений и предъявляемые пользователями требования.

1. Типы вычислительных ресурсов. Можно выделить два основных типа *элементарных ресурсов* - серверы и персональные компьютеры. Сюда также можно отнести мобильные и встраиваемые устройства, не рассматриваемые в данной работе.

Элементарные ресурсы могут быть объединены в вычислительные инфраструктуры (кластеры, гриды, облака), рассматриваемые как отдельные типы *сложных ресурсов*. При этом сложные ресурсы одного типа (например, кластеры) могут являться элементами сложного ресурса другого типа (например, грида).

По степени контроля ГРВС над ресурсом, вне зависимости от его типа, важно различать *выделенные* и *разделяемые* ресурсы. В первом случае среда имеет монопольный доступ к ресурсу. Во втором случае одновременно с ГРВС и ее пользователями ресурс может использоваться другими системами и пользователями. При этом данная внешняя по отношению к ГРВС нагрузка часто носит динамический нерегулярный характер и может приводить к частичной или полной недоступности ресурса в рамках среды.

Под *сервером* будем понимать непрерывно функционирующий компьютер, выделенный для проведения вычислений. Данный тип ресурсов наиболее часто встречается в рамках гетерогенных сред, в том числе неявно в составе сложных ресурсов (кластеры, гриды, облака).

Под *персональным компьютером* будем понимать компьютер, который может проводить вычисления в рамках ГРВС только в моменты, когда его ресурсы не используются владельцем. Сюда также можно отнести персональные мобильные устройства (смартфоны), приблизившиеся по своим характеристикам к классическим компьютерам.

В сравнении с сервером данный тип ресурса характеризуется невысокими аппаратными и сетевыми характеристиками, динамической доступностью (объем доступных вычислительных ресурсов варьируется со временем), меньшими надежностью и степенью доверия.

Основными факторами, обуславливающими использование персональных компьютеров в ГРВС, являются многочисленность ресурсов данного типа и бесплатность их использования (в рамках проектов добровольных вычислений), что позволяет получить совокупную вычислительную мощность среды на уровне суперкомпьютера с минимальными затратами.

Под *кластером* будем понимать сложный ресурс, образованный из множества серверов. Кластеры часто представляют собой простейший случай ГРВС, поскольку вследствие обновления и расширения ресурсов кластера аппаратные характеристики отдельных узлов могут отличаться. Под кластерами будем подразумевать также суперкомпьютеры, вне зависимости от их архитектуры (впрочем, большинство сейчас имеет именно кластерную архитектуру).

Отличием кластера от просто группы серверов является наличие специального ПО (менеджера ресурсов или системы пакетной обработки), выполняющего роль посредника между пользователями и ресурсами кластера. *Менеджер ресурсов*

осуществляет управление очередями заданий, включая распределение заданий между узлами кластера (планирование) в соответствии с требованиями заданий и политиками, определяемыми администратором кластера. При этом прямой доступ к узлам кластера, минуя менеджера ресурсов, как правило, отсутствует. Менеджер ресурсов также предоставляет пользователям информацию о состоянии узлов кластера, очередей и заданий.

Под *грид-инфраструктурой* или *гридом (grid)* [6–8] будем понимать сложный ресурс, образованный из множества кластеров. Как правило, входящие в грид кластеры являются автономными ресурсами, распределенными географически и административно. Автономность в данном случае означает, что ресурс может функционировать в отрыве от инфраструктуры, а владелец ресурса сохраняет полный контроль над ним. Группу ресурсов грида, управляемую одним владельцем, часто называют ресурсным центром. Наиболее масштабными примерами таких систем являются исследовательские инфраструктуры Worldwide LHC Computing Grid (WLCG), European Grid Infrastructure (EGI), Open Science Grid (OSG), Distributed European Infrastructure for Supercomputing Applications (DEISA).

Доступ к ресурсам грида обеспечивается при помощи специального промежуточного ПО (*grid middleware*) [21–23], примерами которого являются технологии Globus Toolkit и gLite. В состав данного ПО входит ряд управляющих и служебных сервисов, в том числе брокеры ресурсов и информационные сервисы. *Брокер ресурсов* играет роль аналога менеджера ресурсов кластера, но уже на более высоком уровне, распределяя задания пользователей между ресурсными центрами грида, где они передаются на обработку локальному менеджеру ресурсов. Отсюда другое часто используемое название брокера ресурсов - *метапланировщик*. Информационные сервисы собирают и предоставляют службам и пользователям грида информацию о состоянии входящих в грид ресурсов.

Отличительными особенностями грид-инфраструктур являются высокая неоднородность аппаратных и сетевых характеристик входящих ресурсов, специфический механизм доступа на основе цифровых сертификатов, относительно высокие надежность и доступность при отсутствии гарантий качества обслуживания (при запуске заданий возможны длительные задержки).

Под *грид-системой из персональных компьютеров (ГСПК)* [5, 24] будем понимать сложный ресурс, образованный из множества персональных компьютеров, которые используются в моменты простоя. По аналогии с грид-инфраструктурами входящие в ГСПК компьютеры обычно являются автономными ресурсами, распределенными географически и управляемыми различными лицами.

По своему масштабу различают два класса ГСПК. *Корпоративные ГСПК* [25] объединяют ресурсы внутри некоторой организации и используются для решения внутренних задач. *Глобальные ГСПК* объединяют ресурсы различных владельцев в масштабах глобальной сети.

По соотношению между поставщиками и пользователями ресурсов выделяют два класса ГСПК. В *проектах добровольных вычислений* [26–27] ресурсами среды пользуется небольшая группа пользователей, в то время как поставщики ресурсов (добровольцы) не могут использовать ресурсы среды. В *пиринговых (peer-to-peer) ГСПК* [28] каждый поставщик ресурсов может использовать ресурсы среды для проведения своих вычислений.

При потенциально большом числе ресурсов (сотни тысяч компьютеров) ГСПК характеризуются еще более высокой неоднородностью аппаратных и сетевых характеристик входящих ресурсов, разнообразием операционных систем, динамическими надежностью и доступностью отдельных ресурсов при общем низком уровне доверия.

Под *облачной инфраструктурой* или *облаком* будем понимать пул конфигурируемых вычислительных ресурсов, доступ к которому осуществляется в соответствии с концепцией облачных вычислений. Данная концепция обладает рядом характерных особенностей, таких как самообслуживание по требованию, универсальный доступ по сети, объединение ресурсов, эластичность и учет потребления.

Различают следующие модели развертывания облаков. *Публичное облако* (*public cloud*) подразумевает свободное использование широкой публикой и находится в юрисдикции некоторой организации поставщика ресурсов. *Частное облако* (*private cloud*) предназначено для использования только в целях некоторой организации, а *общественное облако* (*community cloud*) - в целях некоторого сообщества потребителей из организаций, имеющих общие задачи. При этом как частные, так и общественные облака допускают передачу управления облаком и его физическое размещение у внешнего поставщика ресурсов. Также различают гибридные облака, рассматриваемые ниже.

Облака могут реализовывать различные модели обслуживания, основанные на сервис-ориентированном подходе: инфраструктура как сервис (IaaS), платформа как сервис (PaaS), программное обеспечение как сервис (SaaS). В данной работе будем рассматривать только облака, реализующие модель *инфраструктура как сервис (IaaS)*, которая подразумевает непосредственный доступ к вычислительным ресурсам, как правило виртуализованным, с возможностью развертывания на них произвольного ПО.

Под *гибридным облаком* или *федерацией облаков* [28] будем понимать комбинацию из нескольких облачных инфраструктур простого типа (публичных, частных или общественных), сохраняющих свою автономность и находящихся в ведении различных поставщиков. В некотором роде это аналог грид-систем для облачных инфраструктур.

Типичным примером гибридного облака является комбинация ресурсов частного облака организации с публичным облаком, используемым для выполнения определенных задач или для получения дополнительных ресурсов в моменты высокой нагрузки, когда наблюдается дефицит локальных ресурсов (см. далее). Также возможны варианты использования в рамках гибридного облака нескольких публичных облаков, а также обмена ресурсами между несколькими дружественными облаками в рамках федерации.

2. Характеристики ресурсов. Рассмотрим основные характеристики вычислительных ресурсов, имеющих значение для целей организации вычислений в ГРВС.

Будем отдавать предпочтение характеристикам с числовыми значениями. В зависимости от того, может ли изменяться значение характеристики со временем, будем различать *статические* и *динамические* характеристики. В некоторых случаях характеристика может относиться к тому или иному виду в зависимости от типа ресурса.

Объем физических ресурсов. Данные характеристики описывают емкость ресурса и способность выполнять задачи с определенными ресурсными требованиями. Примерами являются число процессорных ядер, объем оперативной памяти, объем дискового пространства и т.п. Данные характеристики являются статическими для элементарных ресурсов, но могут быть динамическими у сложных ресурсов.

Аппаратные и программные платформы, программное обеспечение. Данные характеристики описывают аппаратные и программные платформы, поддерживаемые ресурсом, а также состав установленного на ресурсе программного обеспечения, включая системные библиотеки и прикладные пакеты. Примерами являются архитектура процессора, порядок байтов, семейство и версия операционной системы, версия библиотеки и т.д.

Производительность. Данный класс характеристик описывает производительность ресурса в общем или относительно выполнения определенного типа задач. К общим характеристикам производительности обычно относят тактовую частоту процессора, число операций с плавающей точкой в секунду (FLOPS) и т.п. Характеристики производительности для конкретного типа задач (приложения) обычно задаются в виде относительной скорости выполнения задачи и определяются на основе статистики выполнения задач или моделей производительности приложений, в которые могут входить общие характеристики производительности и нагрузки.

Нагрузка. Данные характеристики описывают текущий уровень использования ресурса, то есть всегда являются динамическими. Примерами являются процентная загрузка процессора, число используемых ядер, объем используемой оперативной памяти, дискового пространства и т.п.

Отдельно стоит отметить характеристики нагрузки, описывающие уровень конкуренции за доступ к ресурсу, такие как среднее количество процессов в очереди на исполнение за определенный временной интервал (load average), длина очереди на кластере и т.п.

Характеристики сетевых каналов. Выделим отдельно характеристики, описывающие производительность и нагрузку сетевых каналов, связывающих ресурсы с клиентами ГРВС. Примерами характеристик производительности являются максимальная пропускная способность (также можно отнести к объему ресурсов) и минимальная латентность. В случае наличия нескольких сетевых каналов на пути между клиентом и ресурсом эффективная пропускная способность равна минимальной пропускной способности среди данных каналов. Примерами характеристик нагрузки являются текущая загрузка канала (сетевой трафик) и частота потери пакетов. В ГРВС со сложной структурой сетевых каналов вводят дополнительные характеристики, описывающие топологию и связность сети.

Доступность. Будет говорить, что ресурс доступен в те моменты времени, когда ресурс может быть использован для вычислений в рамках ГРВС. Ресурс может становиться недоступен вследствие отказов или политики доступа, ограничивающей использование ресурса при определенных условиях. Например, персональный компьютер в рамках ГСПК доступен только в моменты, когда он включен, подключен к сети и простаивает (не используется владельцем). *Доступность* может быть определена как мгновенная величина, характеризующая присутствие ресурса в ГРВС в текущий момент, так и как вероятность присутствия ресурса в течение заданного промежутка времени, определяемая обычно на основе накопленной статистики.

Надежность ресурса обычно определяется как вероятность функционирования без отказов в течение заданного промежутка времени. Другими характеристиками надежности являются средняя наработка на отказ (mean time between failures, MTBF), доля неудачно завершившихся из-за отказа задач (определяется на основе статистики выполнения задач) и т.д.

Уровень доверия характеризует степень доверия пользователя ГРВС заданному ресурсу. Доверие может выражаться, например, по отношению к результатам вычислений, выполненным ресурсом, или защищенности передаваемых на ресурс данных. Аналогично надежности уровень доверия можно определить, например, как вероятность получения правильного результата (не измененного намеренно) и вычислять на основе статистики выполнения задач. В отличие от надежности при определении уровня доверия учитываются не обычные отказы, а злонамеренное поведение.

Стоимость. Плата за использование ресурса характерна для публичных облачных инфраструктур. Обычно стоимость определяется исходя из конфигурации виртуального ресурса (instance type) и цены за единичное время аренды ресурса (час или минута). В некоторых случаях стоимость использования ресурса также может определяться объемом потребленных физических ресурсов (процессорного времени, дискового пространства, сетевого трафика и т.п.) и ценами за единицу соответствующего типа физических ресурсов.

Механизмы доступа. Ресурсы различного типа значительно отличаются по используемым механизмам удаленного доступа и интерфейсам запуска вычислительных заданий. Так доступ к кластерам обычно осуществляется по протоколу SSH, а запуск заданий осуществляется через набор команд системы очередей. Доступ к грид-инфраструктурам осуществляется через протоколы на основе веб-сервисов с использованием цифровых сертификатов. Доступ к облачным инфраструктурам осуществляется через проприетарные или открытые интерфейсы на основе REST-сервисов, причем обычно в совершенно других терминах (не заданий, а виртуальных машин). Доступ к персональным компьютерам в рамках ГСПК осуществляется через специально реализованных клиентов, выполняемых на компьютерах.

Используемые механизмы доступа в свою очередь могут отличаться по своим характеристикам, таким как накладные расходы, скорость обработки запросов и т.д.

Владелец ресурса. В рамках ГРВС ресурсы могут принадлежать различным владельцам, то есть находиться в ведении различных лиц и организаций. Владелец ресурса может учитываться при выборе ресурса для выполнения задания. Например, пользователь в требованиях задания может указать, что предпочтительным является выполнение задания на ресурсе его организации.

Уровень контроля. По степени контроля ГРВС над ресурсом, вне зависимости от его типа, будем различать *выделенные* и *разделяемые* ресурсы. В первом случае среда имеет монопольный доступ к ресурсу. Во втором случае одновременно с ГРВС и ее пользователями ресурс может использоваться другими системами и пользователями. При этом данная внешняя по отношению к ГРВС нагрузка часто носит динамический нерегулярный характер и может приводить к частичной или полной недоступности ресурса в рамках среды.

Политики доступа. Владелец ресурса может определять политики доступа к ресурсу для отдельных пользователей или ГРВС в целом. Данные политики обычно задаются в виде *приоритетов* (включая самый низкий – только при простое ресурса) и различного рода *ограничений* (на объем потребляемых ресурсов, число заданий, время суток и т.д.).

Качество обслуживания. Уровень качества обслуживания, реализуемый ресурсом, может выражаться в виде гарантий определенных уровней производительности, доступности и надежности ресурса, максимального времени выделения ресурсов (ожидания в очереди), автоматического перезапуска заданий при отказах и т.д.

3. Типовые сценарии использования ГРВС. Основными целями использования ГРВС являются увеличение доступной производительности для решения больших задач, повышение эффективности использования имеющихся ресурсов и интеграция ресурсов участников совместного проекта.

Гетерогенный ресурс. В данном сценарии ресурс в ходе его эксплуатации естественным образом превращается в ГРВС. Например, кластер после добавления новых узлов, отличающихся по своим характеристикам от прежних. Чтобы избежать гетерогенности, администраторы кластеров часто выделяют узлы с одинаковыми характеристиками в отдельные очереди или группы в менеджере ресурсов.

Однако это затрудняет одновременное использование всех узлов и может привести к простоям отдельных групп. Подобные ГРВС характеризуются централизованным управлением, едиными механизмами доступа, максимальной сетевой связностью и высоким уровнем доверия.

Интеграция ресурсов внутри организации. В данном сценарии ГРВС образуется путем объединения ресурсов, принадлежащих одной организации. Основной целью при этом является консолидация ресурсов для их более эффективного использования для задач данной организации. Подобные среды характеризуются высокой степенью контроля над ресурсами, высоким уровнем доверия и высокой скоростью сетевых обменов. Наиболее частыми примерами являются объединение нескольких кластеров и/или серверов внутри организации, а также использование персональных компьютеров сотрудников в нерабочее время.

Сочетание ресурсов организации с внешними ресурсами. Данный сценарий является логичным продолжением предыдущего сценария, когда помимо ресурсов организации в состав ГРВС входят также внешние ресурсы. Основной целью при этом является наращивание вычислительных ресурсов для решения задач данной организации, в том числе в моменты повышенной потребности в ресурсах. Типичным примером является динамическое распределение вычислительной нагрузки с внутренних ресурсов в публичную облачную инфраструктуру (т.н. cloud bursting).

Подобные ГРВС по-прежнему характеризуются единым потребителем ресурсов, однако, в сравнении с предыдущими сценариями, степень контроля над ресурсами и уровень доверия уменьшаются. Кроме того, вследствие географического распределения ресурсов уменьшаются степень связности и скорость сетевых обменов.

Разделяемое использование ресурсов. В данном сценарии ГРВС образуется путем объединения ресурсов нескольких владельцев для совместного использования данных ресурсов. Как правило, целью создания таких сред является наращивание вычислительных ресурсов для решения общих задач, стоящих перед участниками, например при выполнении совместного проекта или более широкой кооперации. В качестве примера можно привести виртуальные организации в рамках научных грид-инфраструктур. Данные среды характеризуются наличием нескольких (в грид – большого числа) потребителей ресурсов, географическим и административным распределением ресурсов, общими правилами участия и использования ресурсов.

Возможен также вариант, когда участники ГРВС могут входить в нее без какой-либо предварительной договоренности и использовать ресурсы среды для решения своих специфичных задач. Такой подход используется в рамках глобальных пиринговых ГСПК (peer-to-peer desktop grids), таких как OurGrid [28]. Однако он мало распространен в силу проблем обеспечения безопасности, возникающих из-за изначально низкой степени доверия к ресурсам и пользователям в подобных системах.

Добровольные вычисления. В рамках концепции добровольных вычислений ГРВС образуется путем объединения ресурсов простаивающих персональных компьютеров в масштабах глобальной сети. Целью создания подобных сред является получение вычислительных мощностей, необходимых для поддержки исследовательских проектов [26–27], как правило, имеющих важное общественное значение или представляющих интерес для большого числа людей.

Характерной чертой данных сред является наличие небольшой группы потребителей ресурсов (участников научного проекта) при большом числе поставщиков ресурсов (добровольцев), число которых может достигать сотен тысяч. Данные среды также характеризуются высокой степенью гетерогенности плат-

форм и характеристик ресурсов, низком приоритетом заданий, динамической доступностью и невысокой надежностью отдельных ресурсов при общем низком уровне доверия.

Индивидуальные инфраструктуры. В данном сценарии ГРВС образуется из ресурсов, доступных одному лицу или группе, и используется в целях увеличения производительности для решения задач данного лица или группы. В настоящее время многие исследователи имеют доступ к нескольким ресурсам различных типов, таким как сервер лаборатории, кластер организации, региональный суперкомпьютерный центр, грид-инфраструктура, облако и т.д. При этом проблема эффективного использования вычислительного потенциала данных ресурсов не может быть решена в рамках существующих коллективных инфраструктур (например, грид), поскольку ни одна из них не охватывает все возможные типы ресурсов.

Характерными особенностями данных ГРВС являются узкий круг потребителей ресурсов, разнообразие типов ресурсов, географическое и административное распределение ресурсов, отсутствие контроля над ресурсами. В отличие от добровольных вычислений здесь рассматриваются в первую очередь ресурсы, отличные от персональных компьютеров, хотя их и ГСПК также можно рассматривать как часть индивидуальной инфраструктуры.

4. Классы ГРВС. На основе описанных выше типовых сценариев использования ГРВС и различных признаков можно выделить следующие классы данных сред.

По типам используемых ресурсов будем различать *однородные* (используются ресурсы одного типа) и *неоднородные* (используются ресурсы различных типов) ГРВС.

В зависимости от того, включает ли ГРВС в себя другие ГРВС, будем различать *простые* и *сложные* среды.

По своему масштабу ГРВС делятся на *локальные*, использующие ресурсы только в рамках локальной сети, и *глобальные*, использующие географически распределенные ресурсы через глобальную сеть.

По кругу потребителей ресурсов ГРВС делятся на *индивидуальные* (одно лицо), *корпоративные* (одна организация), *коллективные* (несколько лиц или организаций) и *публичные* (состав потребителей не ограничен).

По кругу поставщиков ресурсов ГРВС делятся на *внутренние* (один поставщик) и *распределенные* (несколько поставщиков).

В зависимости от того, могут ли поставщики ресурсов выступать потребителями ресурсов, ГРВС делятся на *симметричные* (могут) и *асимметричные* (не могут).

По своему времени жизни можно выделить *временные* (под решение одной задачи) и *постоянные* (функционируют непрерывно) среды.

5. Классы вычислительных приложений. Помимо характеристик ГРВС для дальнейших исследований требуется также иметь представление о классах приложений, которые могут выполняться в данных средах. Рассмотрим основные классы вычислительных приложений, их особенности и приспособленность для выполнения в ГРВС.

Здесь и далее под задачей будем подразумевать элементарную вычислительную единицу приложения, которая может быть запущена полностью на одном физическом ресурсе (процессоре, компьютере).

Пакет задач (bag of tasks). Данный класс включает приложения, состоящие из множества независимых задач [30]. Сами задачи при этом являются однотипными и предъявляют одинаковые требования к ресурсам. Подобные приложения лучше всего приспособлены для выполнения в распределенной среде, поскольку не требуют сложной координации между задачами, и задачи могут выполняться в

произвольной последовательности. Данный класс приложений широко используется на практике и подробно изучен в литературе. Примерами таких приложений являются многовариантные расчеты и алгоритмы, параллельные по задачам или данным.

Итеративные приложения. Данный класс включает реализации итеративных методов, в которых каждая итерация распараллелена и представлена в виде множества независимых вычислительных задач (пакета задач). Таким образом, данные приложения представляют собой композицию из нескольких запусков пакетов задач, между которыми могут производиться дополнительные действия (анализ результатов итерации, проверка условия сходимости, генерация новых задач и т.д.). Примерами таких приложений являются генетические алгоритмы, алгоритмы на графах и т.п.

Поток работ (workflow, композитные приложения). Приложения данного класса состоят из нескольких задач с произвольными зависимостями между ними и обычно описываются в виде ориентированного ациклического графа, вершинами которого являются задачи [31–32]. Задачи здесь могут быть разнотипными и предъявлять различные требования к ресурсам. Ребра графа могут задавать потоки данных между задачами (data flow) или явный порядок выполнения задачи (control flow). Обычно считается, что задача не может начать выполнение до тех пор, пока не завершатся все задачи, от которых она зависит (вариант, когда зависимые задачи выполняются одновременно рассмотрим отдельно ниже). Выполнение данных приложений в распределенной среде усложняется необходимостью учета специфичных требований отдельных задач и зависимостей между ними.

Приложения с взаимодействиями между задачами (связанные). Приложения данного типа характеризуются наличием обменов данными между выполняющимися задачами. Задачи, между которыми необходимо производить обмены, должны выполняться одновременно, что усложняет организацию запуска задач в распределенной среде. Кроме того, отказ одной из задач может повлиять на выполнение других задач, что усложняет обработку отказов.

Типичными примерами данного класса приложений являются параллельные программы, использующие парадигмы обмена сообщениями (например, на основе стандарта MPI) или распределенной общей памяти (например, на основе модели PGAS). Приложения, реализующие потоковую обработку данных (stream processing), устроены по принципу графа аналогично workflow-приложениям, однако при этом требуют одновременной работы всех задач графа. Похожая ситуация наблюдается при распределенном имитационном или многомасштабном моделировании.

Отдельно отметим приложения в которых требуется организовать рассылку данных между всеми задачами. Например, при реализации метода ветвей и границ (branch and bound) задачам необходимо обмениваться друг с другом найденными значениями рекордов. При улучшении рекорда задача рассылает это значение всем остальным задачам, которые обновляют свои локальные значения.

В зависимости от степени интенсивности взаимодействий между задачами можно выделить *сильносвязанные* и *слабосвязанные* приложения. Первые предъявляют высокие требования и чувствительны к характеристикам сетевых каналов, в то время как последние лучше приспособлены для выполнения в глобальной сети.

Приложения, интенсивные по данным. Будем относить сюда приложения любых классов, задачи которых характеризуются работой с большими объемами данных [33–34]. Например, задачи в приложениях типа пакет задач могут обрабатывать или генерировать, а задачи в workflow-приложениях также передавать между собой большие объемы данных. Это усложняет реализацию выполнения дан-

ных приложений в распределенной среде, так как требует эффективного распределения данных и оптимизации накладных расходов на передачу данных по сети, например, путем назначения задачи на узел, хранящий требуемые для задачи данные.

Интерактивные приложения. Данные приложения характеризуются активным участием пользователей в управлении процессом вычислений (т.н. computational steering). Это может выражаться в контроле промежуточных результатов, изменении параметров и состава задач по ходу выполнения приложения, остановке и перезапуске вычислений. Примерами являются системы совместного анализа данных и поддержки принятия решений, в том числе в экстренных ситуациях (см. далее).

Приложения реального времени. Данные приложения характеризуются наличием ограничений на сроки получения результатов вычислений, по истечении которых данные результаты быстро теряют свою ценность и актуальность. Типичным примером являются вычислительные приложения, ориентированные на поддержку принятия решений в экстренных ситуациях (т.н. urgent computing) [35–36]. Для экстренных вычислений также характерны элементы интерактивного управления процессом вычислений, а также автоматической реконфигурации приложения по ходу вычислений в целях обеспечения установленных ограничений на время расчетов.

6. Анализ требований к организации вычислений в ГРВС. Рассмотрим типичные требования, предъявляемые пользователями к ГРВС и выполняемым в них приложениям.

Обеспечение *максимальной производительности* подразумевает минимизацию времени выполнения приложения в ГРВС за счет эффективного использования доступных ресурсов.

Обеспечение *заданного уровня производительности* подразумевает наличие гарантий на объем доступных ресурсов.

Обеспечение *эластичности* подразумевает поддержку автоматического наращивания ресурсов или их освобождения в соответствии с текущей нагрузкой.

Обеспечение *отказоустойчивости* подразумевает автоматическую обработку некритичных отказов при выполнении приложения без нарушения его работы.

Обеспечение *справедливости (fairness)* характерно для коллективных сред и подразумевает равномерное распределение ресурсов между пользователями среды и между их приложениями, возможно с учетом установленных приоритетов.

Обеспечение *надежности результатов* характерно для сред с низким уровнем доверия к ресурсам (например, ГСПК) и подразумевает наличие механизмов валидации результатов задач и защиты от преднамеренных фальсификаций.

Обеспечение *выполнения приложения в срок* подразумевает указание пользователем времени, к которому необходимо получить результаты вычислений. Данное ограничение может носить желательный или обязательный характер. В свою очередь среда может предоставлять гарантии различного уровня качества обслуживания, от обслуживания по возможности (best effort) до строгого соблюдения указанного срока.

Обеспечение *эффективного расходования средств* характерно для сред с платными ресурсами и подразумевает наличие механизмов, минимизирующих расход средств на оплату ресурсов при заданных пользователем ограничениях на время выполнения (или минимизирующих время выполнения при заданном ограничении на бюджет).

Заключение. Проведенный анализ разновидностей и сценариев использования ГРВС показывает, что данный класс вычислительных систем сейчас широко востребован, в том числе для проведения крупномасштабных научных расчетов.

Вместе с тем выявлено большое разнообразие данных сред – по сути любая крупная вычислительная система сейчас является распределенной и гетерогенной. В рамках дальнейших исследований основное внимание планируется уделить гетерогенным средам, включающим географически распределенные ресурсы различных типов, а также индивидуальным инфраструктурам, как наименее развитым и изученным разновидностям ГРВС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Catlett C.E.* Metacomputing // Communications of the ACM. – 1992. – Vol. 35, No. 6. – P. 44-53.
2. *Коваленко В., Корягин Д.* Вычислительная инфраструктура будущего // Открытые системы. – 1999. – № 11-12. – С. 45-52.
3. *Воеводин В.В., Жолудев Ю.А., Соболев С.И., Стефанов К.С.* Эволюция системы метакомпьютинга X-Com // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2009. – № 4.
4. *Anderson D.P., Fedak G.* The computational and storage potential of volunteer computing // Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on. – IEEE, 2006. – Vol. 1. – P. 73-80.
5. *Lovas R., Афанасьев А.П., Волошинов В.В., Посыпкин М.А., Сухорослов О.В., Храпов Н.П.* Увеличение вычислительной мощности распределенных систем с помощью грид-систем из персональных компьютеров // Параллельные вычислительные технологии (ПАВТ'2011): Труды Международной научной конференции. – Челябинск: ЮУрГУ, 2011. – С. 6-14.
6. *Foster I., Kesselman C. (ed.).* The Grid 2: Blueprint for a new computing infrastructure. – Elsevier, 2003.
7. *Афанасьев А.П., Волошинов В.В., Рогов С.В., Сухорослов О.В.* Развитие концепции распределенных вычислительных сред // Проблемы вычислений в распределенной среде: организация вычислений в глобальных сетях: Труды ИСА РАН. – М.: РОХОС, 2004. – С. 6-105.
8. *Коваленко В.Н., Корягин Д.А.* Грид: истоки, принципы и перспективы развития // Информационные технологии и вычислительные системы. – 2008. – № 4. – С. 38-50.
9. *Емельянов С.В., Афанасьев А.П., Волошинов В.В., Гринберг Я.Р., Кривцов В.Е., Сухорослов О.В.* Реализация Grid-вычислений в среде IARnet // Информационные технологии и вычислительные системы. – 2005. – № 2. – С. 61-75.
10. *Радченко Г.И.* Технология построения проблемно-ориентированных иерархических оболочек над инженерными пакетами в грид-средах // Системы управления и информационные технологии. – 2008. – № 4. – С. 57-61.
11. *Бухановский А.В., Ковальчук С.В., Марьин С.В.* Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Известия вузов. Приборостроение. – 2009. – Т. 52, № 10. – С. 5-24.
12. *Armbrust M. et al.* A view of cloud computing // Communications of the ACM. – 2010. – Vol. 53, No. 4. – P. 50-58.
13. *Vuuya R. et al.* Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility // Future Generation computer systems. – 2009. – Т. 25, No. 6. – P. 599-616.
14. *Nair S. K. et al.* Towards secure cloud bursting, brokerage and aggregation // Web services (ecows), 2010 IEEE 8th European conference on. – IEEE, 2010. – P. 189-196.
15. *Астафьев А.С., Афанасьев А.П., Лазарев И.В., Сухорослов О.В., Тарасов А.С.* Научная сервис-ориентированная среда на основе технологий Web и распределенных вычислений // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции. – М.: МГУ, 2009. – С. 463-467.
16. *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 // Известия высших учебных заведений. Приборостроение. – 2011. – Т. 54, № 10.

17. *Самоваров О.И., Гайсарян С.С.* Архитектура и особенности реализации платформы UniHUB в модели облачных вычислений на базе открытого пакета OpenStack // Труды Института системного программирования РАН. – 2014. – Т. 26, № 1.
18. *Сухорослов О.В.* Интеграция вычислительных приложений и распределенных ресурсов на базе облачной программной платформы // Программные системы: теория и приложения. – 2014. – Т. 5, № 4 (22). – С. 171-182.
19. *Поляков С.В., Выродов А.В., Пурырьков Д.В., Якобовский М.В.* Облачный сервис для решения многомасштабных задач нанотехнологии на суперкомпьютерных системах // Труды Института системного программирования РАН. – 2015. – Т. 27, № 6.
20. *Велихов В.Е., Климентов А.А., Машишистов Р.Ю., Пойда А.А., Рябинкин Е.А.* Интеграция гетерогенных вычислительных мощностей НИЦ «Курчатовский институт» для проведения масштабных научных вычислений // Известия ЮФУ. Технические науки. – 2016. – № 11 (184). – С. 88-99.
21. *Foster I.* Globus toolkit version 4: Software for service-oriented systems // IFIP international conference on network and parallel computing. – Springer Berlin Heidelberg, 2005. – P. 2-13.
22. *Laure E. et al.* Programming the Grid with gLite // Computational methods in science and technology. – 2006. – Vol. 12, No. 1. – P. 33-45.
23. *Демичев А.П., Крюков А.П., Шамардин Л.В.* Принципы построения грид с использованием RESTful-веб-сервисов // Программные продукты и системы. – 2009. – № 4.
24. *Kondo D. et al.* Characterizing and evaluating desktop grids: An empirical study // Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International. – IEEE, 2004. – P. 26.
25. *Ивашко Е.Е.* Desktop Grid корпоративного уровня // Программные системы: теория и приложения. – 2014. – Т. 5, № 1. – С. 183-190.
26. *Anderson D.P. et al.* SETI@home: an experiment in public-resource computing // Communications of the ACM. – 2002. – Vol. 45, No. 11. – P. 56-61.
27. *Заикин О.С., Посыпкин М.А., Семёнов А.А., Храпов Н.П.* Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2012. – № 5-2.
28. *Andrade N. et al.* OurGrid: An approach to easily assemble grids with equitable resource sharing // Workshop on Job Scheduling Strategies for Parallel Processing. – Springer Berlin Heidelberg, 2003. – P. 61-86.
29. *Kurze T. et al.* Cloud federation // CLOUD COMPUTING. – 2011. – Vol. 2011. – P. 32-38.
30. *Cirne W. et al.* Grid computing for bag of tasks applications // In Proc. of the 3rd IFIP Conference on E-Commerce, E-Business and EGovernment. – 2003.
31. *Taylor I.J. et al.* Workflows for e-Science: scientific workflows for grids. – Springer Publishing Company, Incorporated, 2014.
32. *Лазарев И.В., Сухорослов О.В.* Использование workflow-методологии для описания процесса распределенных вычислений // Проблемы вычислений в распределенной среде: Модели обработки и представления данных. Динамические системы: Труды ИСА РАН. – 2005. – Т. 14. – С. 254-255.
33. *Сухорослов О.В.* Новые технологии распределенного хранения и обработки больших массивов данных. – М.: Институт системного анализа РАН, 2008. – 40 с.
34. *Chen C.L.P., Zhang C.Y.* Data-intensive applications, challenges, techniques and technologies: A survey on Big Data // Information Sciences. – 2014. – Vol. 275. – P. 314-347.
35. *Beckman P. et al.* SPRUCE: A system for supporting urgent high-performance computing // Grid-Based Problem Solving Environments. – Springer US, 2007. – P. 295-311.
36. *Бухановский А.В., Житников А.Н., Петросян С.Г.* Высокопроизводительные технологии экстренных вычислений для предотвращения угрозы наводнений // Известия высших учебных заведений. Приборостроение. – 2011. – Т. 54, № 10.

REFERENCES

1. *Catlett C.E.* Metacomputing, *Communications of the ACM*, 1992, Vol. 35, No. 6, pp. 44-53.
2. *Kovalenko V., Koryagin D.* Vychislitel'naya infrastruktura budushchego [Future computing infrastructure], *Otkrytye sistemy* [Open systems], 1999, No. 11-12, pp. 45-52.
3. *Voevodin V.V., Zholudev Yu.A., Sobolev S.I., Stefanov K.S.* Evolyutsiya sistemy metakomp'yutinga X-Com [Evolution of X-Com metacomputing system], *Vestnik Nizhegorodskogo universiteta im. N.I. Lobachevskogo* [Vestnik of Lobachevsky University of Nizhni Novgorod], 2009, No. 4.

4. Anderson D.P., Fedak G. The computational and storage potential of volunteer computing, *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*. IEEE, 2006, Vol. 1, pp. 73-80.
5. Lovas R., Afanas'ev A.P., Voloshinov V.V., Posypkin M.A., Sukhoroslov O.V., Khrapov N.P. Uvelichenie vychislitel'noy moshchnosti raspredelennykh sistem s pomoshch'yu grid-sistem iz personal'nykh komp'yutеров [Increasing the computing performance of distributed systems with the help of desktop grids], *Parallel'nye vychislitel'nye tekhnologii (PaVT'2011): Trudy Mezhdunarodnoy nauchnoy konferentsii* [Parallel computational technologies (PaVT'2011): Proceedings of the International scientific confer]. Chelyabinsk: YuUrGU, 2011, pp. 6-14.
6. Foster I., Kesselman C. (ed.). *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2003.
7. Afanas'ev A.P., Voloshinov V.V., Rogov S.V., Sukhoroslov O.V. Razvitie kontseptsii raspredelennykh vychislitel'nykh sred [The development of distributed computing environment concept], *Problemy vychisleniy v raspredelennoy srede: organizatsiya vychisleniy v global'nykh setyakh: Trudy ISA RAN* [Problems computing in a distributed environment: the organization of computing in global networks: Proceedings of ISA RAS]. Moscow: ROKhOS, 2004, pp. 6-105.
8. Kovalenko V.N., Koryagin D.A. Grid: istoki, printsipy i perspektivy razvitiya [Grid: origins, principles and prospects], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information technology and computer systems], 2008, No. 4, pp. 38-50.
9. Emel'yanov S.V., Afanas'ev A.P., Voloshinov V.V., Grinberg Ya.R., Krivtsov V.E., Sukhoroslov O.V. Realizatsiya Grid-vychisleniy v srede IARnet [Implementation of grid computing in IARnet environment], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information technology and computer systems], 2005, No. 2, pp. 61-75.
10. Radchenko G.I. Tekhnologiya postroeniya problemno-orientirovannykh ierarkhicheskikh obolochek nad inzhenernymi paketami v grid-sredakh [Technology for building problem-oriented hierarchical wrappers of engineering packages in grid environments], *Sistemy upravleniya i informatsionnye tekhnologii* [Control systems and information technology], 2008, No. 4, pp. 57-61.
11. Bukhanovskiy A.V., Koval'chuk S.V., Mar'in S.V. Intellektual'nye vysokoproizvodi-tel'nye programnye komplekсы modelirovaniya slozhnykh sistem: kontseptsiya, arkhitektura i primery realizatsii [Intellectual high-performance software packages for modeling complex systems: concept, architecture and examples of implementation], *Izvestiya vuzov. Priborostroenie* [Journal of Instrument Engineering], 2009, Vol. 52, No. 10, pp. 5-24.
12. Armbrust M. et al. A view of cloud computing, *Communications of the ACM*, 2010, Vol. 53, No. 4, pp. 50-58.
13. Buyya R. et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation computer systems*, 2009, Vol. 25, No. 6, pp. 599-616.
14. Nair S. K. et al. Towards secure cloud bursting, brokerage and aggregation, *Web services (ecows), 2010 IEEE 8th European conference on*. IEEE, 2010, pp. 189-196.
15. Astaf'ev A.S., Afanas'ev A.P., Lazarev I.V., Sukhoroslov O.V., Tarasov A.S. Nauchnaya servis-orientirovannaya sreda na osnove tekhnologiy Web i raspredelennykh vychisleniy [Academic service-oriented environment based on web technologies and distributed computing], *Nauchnyy servis v seti Internet: masshtabiruemost', parallel'nost', effektivnost': Trudy Vserossiyskoy superkomp'yuternoy konferentsii* [Scientific service in the Internet: scalability, parallelism, efficiency: Proceedings of all-Russian supercomputer conference]. Moscow: MGU, 2009, pp. 463-467.
16. Bukhanovskiy A.V., Vasil'ev V.N., Vinogradov V.N., Smirnov D.Yu., Sukhorukov S.A., Yapparov T.G. CLAVIRE: perspektivnaya tekhnologiya oblachnykh vychisleniy vtorogo pokoleniya [CLAVIRE: perspective second generation cloud computing technology], *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie* [Journal of Instrument Engineering], 2011, Vol. 54, No. 10.
17. Samovarov O.I., Gaysaryan S.S. Arkhitektura i osobennosti realizatsii platformy UniHUB v modeli oblachnykh vychisleniy na baze otkrytogo paketa OpenStack [Architecture and details of implementation of UniHUB platform in cloud computing model on the base of OpenStack package], *Trudy Instituta sistemnogo programirovaniya RAN* [Proceedings of Institute for system programming of Russian Academy of Sciences], 2014, Vol. 26, No. 1.

18. Sukhoroslov O.V. Integratsiya vychislitel'nykh prilozheniy i raspredelennykh resursov na baze oblachnoy programmnoy platformy [Integration of computing applications and distributed resources on the base of cloud software platform], *Programmnye sistemy: teoriya i prilozheniya* [Program systems: theory and applications], 2014, Vol. 5, No. 4 (22), pp. 171-182.
19. Polyakov S.V., Vyrodov A.V., Puryr'kov D.V., Yakobovskiy M.V. Oblachnyy servis dlya resheniya mnogomasshtabnykh zadach nanotekhnologii na superkomp'yuternykh sistemakh [Cloud service for solution of multi-scale nanotechnology problems on supercomputer systems], *Trudy Instituta sistemnogo programmirovaniya RAN* [Proceedings of Institute for system programming of Russian Academy of Sciences], 2015, Vol. 27, No. 6.
20. Velikhov V.E., Klimentov A.A., Mashinistov R.Yu., Poyda A.A., Ryabinkin E.A. Integratsiya geterogennykh vychislitel'nykh moshchnostey NITs «Kurchatovskiy institut» dlya provedeniya masshtabnykh nauchnykh vychisleniy [Integration of heterogeneous computing resources at nrc «Kurchatov institute» for large-scale scientific computations], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 11 (184), pp. 88-99.
21. Foster I. Globus toolkit version 4: Software for service-oriented systems, *IFIP international conference on network and parallel computing*. Springer Berlin Heidelberg, 2005, pp. 2-13.
22. Laure E. et al. Programming the Grid with gLite, *Computational methods in science and technology*, 2006, Vol. 12, No. 1, pp. 33-45.
23. Demichev A.P., Kryukov A.P., Shamardin L.V. Printsipy postroeniya grid s ispol'zovaniem RESTful-veb-servisov [Principles of building grid using RESTful web services], *Programmnye produkty i sistemy* [Software products and systems], 2009, No. 4.
24. Kondo D. et al. Characterizing and evaluating desktop grids: An empirical study, *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, pp. 26.
25. Ivashko E.E. Desktop Grid korporativnogo urovnya [Enterprise desktop grids], *Programmnye sistemy: teoriya i prilozheniya* [Program systems: theory and applications], 2014, Vol. 5, No. 1, pp. 183-190.
26. Anderson D.P. et al. SETI@home: an experiment in public-resource computing, *Communications of the ACM*, 2002, Vol. 45, No. 11, pp. 56-61.
27. Zaikin O.S., Posypkin M.A., Semenov A.A., Khrapov N.P. Opyt organizatsii dobrovol'nykh vychisleniy na primere proektov OPTIMA@home i SAT@home [Experience in volunteer computing on the example of OPTIMA@home and SAT@home projects], *Vestnik Nizhegorodskogo universiteta im. N.I. Lobachevskogo* [Vestnik of Lobachevsky University of Nizhni Novgorod], 2012, No. 5-2.
28. Andrade N. et al. OurGrid: An approach to easily assemble grids with equitable resource sharing, *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer Berlin Heidelberg, 2003, pp. 61-86.
29. Kurze T. et al. Cloud federation, *CLOUD COMPUTING*, 2011, Vol. 2011, pp. 32-38.
30. Cirne W. et al. Grid computing for bag of tasks applications, *In Proc. of the 3rd IFIP Conference on E-Commerce, E-Business and EGovernment*, 2003.
31. Taylor I.J. et al. Workflows for e-Science: scientific workflows for grids. Springer Publishing Company, Incorporated, 2014.
32. Lazarev I.V., Sukhoroslov O.V. Ispol'zovanie workflow-metodologii dlya opisaniya protsessa raspredelennykh vychisleniy [Using workflow methodology for description of the process of distributed computations], *Problemy vychisleniy v raspredelennoy srede: Modeli obrabotki i predstavleniya dannykh. Dinamicheskie sistemy: Trudy ISA RAN* [Computing problems in a distributed environment: models of processing and presentation of data. Dynamical systems: Proceedings of ISA RAS], 2005, Vol. 14, pp. 254-255.
33. Sukhoroslov O.V. Novye tekhnologii raspredelennogo khraneniya i obrabotki bol'shikh massivov dannykh [New technologies for distributed storage and processing of big datasets]. Moscow: Institut sistemnogo analiza RAN, 2008, 40 p.
34. Chen C.L.P., Zhang C.Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, *Information Sciences*, 2014, Vol. 275, pp. 314-347.
35. Beckman P. et al. SPRUCE: A system for supporting urgent high-performance computing, *Grid-Based Problem Solving Environments*. Springer US, 2007, pp. 295-311.

36. *Bukhanovskiy A.V., Zhitnikov A.N., Petrosyan S.G. Vysokoproizvoditel'nye tekhnologii ekstrennykh vychisleniy dlya predotvrashcheniya ugrozy navodneniy [High-performance technologies of urgent computing for preventing the threat of flooding], Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie [Journal of Instrument Engineering], 2011, Vol. 54, No. 10.*

Статью рекомендовал к опубликованию д.ф.-м.н., профессор М.В. Якововский.

Сухорослов Олег Викторович – Институт проблем передачи информации им. А.А. Харкевича Российской академии наук; e-mail: sukhoroslov@iitp.ru; 117218, Москва, Нахимовский проспект, 36, корпус 1; тел.: +74957189631; лаборатория распределенных вычислительных систем; с.н.с.; к.т.н.

Sukhoroslov Oleg Viktorovich – Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute); e-mail: sukhoroslov@iitp.ru; 36/1, Nakhimovskiy ave., Moscow, 117218, Russia; phone: +74957189631; Laboratory of Distributed Computing Systems; senior researcher; cand. of eng. sc.

УДК 004.052.34, 004.75

DOI 10.18522/2311-3103-2016-12-130140

И.В. Богомолов, А.В. Алексиянц, О.Д. Борисенко, А.И. Аветисян

ПРОБЛЕМЫ МАСШТАБИРУЕМОСТИ ОБЛАЧНЫХ СРЕД И ПОИСК ПРИЧИН ДЕГРАДАЦИИ ЦЕНТРАЛЬНОГО СЕРВИСА ИДЕНТИФИКАЦИИ OPENSTACK KEYSTONE*

Облачные среды приобретают все большую популярность, и они являются одним из наиболее удобных средств организации вычислений для большого спектра задач. Устройство систем разделения прав доступа и предоставления доступа для клиентских приложений, работающих с облаками напрямую, в открытых облачных средах отличается от промышленных проприетарных облачных сред. Атомарной сущностью в открытых средах является не индивидуальный пользователь, а группа пользователей. Данный подход накладывает ограничения на построение системы аутентификации и авторизации (идентификации) пользователей. Кроме того, одна и та же система отвечает как за работу с пользователями системы, так и за работу с остальными сервисами облака: это необходимо для защиты облака от добавления фальшивых узлов. Системы идентификации в открытых облачных платформах опираются на два общих компонента: реляционные СУБД для хранения данных и на алгоритмы шифрования в применении к хранимым данным. Разработчики открытых облачных платформ не ставят на первое место производительность системы идентификации и предлагают лишь подходы, обеспечивающие отказоустойчивость системы идентификации с точки зрения сохранности данных. В данной работе мы показываем важность масштабирования системы идентификации с точки зрения производительности на примере Openstack Keystone. Приводятся эксперименты, доказывающие, что за счет централизации системы хранения данных и сложности алгоритмов шифрования, число аутентификационных запросов в секунду, которое Openstack Keystone может обработать без ошибок, растет не линейно с увеличением доступных ресурсов. Как следствие, система идентификации не позволяет наращивать вычислительные мощности облачных систем на базе Openstack как относительно числа пользователей системы, так и относительно физических узлов облачной системы. В данной работе мы описываем новый подход, позволяющий избежать проблем масштабирования за счет замены РСУБД для хранения данных на решения класса In-Memory Data Grid (IMDG). Реализован прототип на базе IMDG Tarantool.

Сервис идентификации и авторизации; Openstack Keystone; облачные вычисления; Denial of Service; масштабирование облачных сред.

* РФФИ 15-29-07111 офи_м Исследование методов обеспечения масштабируемости систем в облачных средах и разработка высокопроизводительного отказоустойчивого центрального сервиса идентификации.