

УДК 004.75

А.И. Каляев**ДЕЦЕНТРАЛИЗОВАННАЯ ОРГАНИЗАЦИЯ ДИСПЕТЧЕРА GRID НА БАЗЕ СООБЩЕСТВ АГЕНТОВ***

В настоящее время человечество нуждается в сложных вычислениях. Одним из самых удобных способов реализовать такие вычисления являются GRID. В данной работе предлагается подход, позволяющий существенно уменьшить стоимость GRID-вычислений за счет переноса функций диспетчирования со специальных выделенных служебных серверов на сообщество исполнительных вычислительных узлов. Разрабатывается алгоритм работы децентрализованной GRID, основанный на мультиагентной технологии. Работоспособность алгоритмов демонстрируется на программной модели.

GRID; распределенные вычисления; мультиагентные системы; децентрализация; распределенное диспетчирование; сообщество; коллективное принятие решений.

A.I. Kalyaev**DECENTRALISED ORGANIZATION OF DISPATCHER OF GRID BASED ON AGENTS COMMUNITIES**

Nowadays many control systems need complex computations. One of the most convenient ways to perform such computations is GRID. This work presents the approach, allowing reducing cost of the distributed calculations in GRID by placing dispatching functions of system from special servers to collective of executive computing nodes. The algorithm of work of decentralized GRID based on multiagent technology is developing. Workability of algorithms is demonstrated by program model.

GRID; distributed computations; multiagent systems; decentralization; distributed dispatching; community; collective decision-making.

Введение. Основной идеей GRID-компьютинга является объединение и предоставление заинтересованным пользователям самых разнообразных вычислительных, коммуникационных и информационных ресурсов путем удаленного доступа [1]. За последний десяток лет GRID-системы обрели большую популярность [2,3,4]. Однако, несмотря на достаточно большое разнообразие существующих GRID-систем, способы их организации остаются практически неизменными: в системе, как правило, имеется ряд выделенных серверов (служебных вычислительных узлов (СВУ)), организующих и диспетчирующих работу всей GRID-системы и выполняющих роль «посредника» между пользователями GRID и исполнителями, выполняющими всю полезную работу по решению пользовательских задач (исполнительные вычислительные узлы (ИВУ)) [5]. Очевидно, что наличие такого «посредника», который должен постоянно находиться в рабочем состоянии, значительно повышает стоимость GRID-вычислений.

Влияние данного недостатка может быть существенно снижено, если минимизировать участие «посредника» (СВУ) в работе GRID-системы и переложить функции организации и диспетчирования вычислений непосредственно на ИВУ, входящие в её состав. При этом данные функции должны быть каким-то образом распределены между множеством ИВУ, т.е. децентрализованы.

В настоящей работе предлагаются принципы организации и функционирования подобной децентрализованной GRID-системы на базе мультиагентного подхода, работоспособность которых подтверждается результатами моделирования.

* Работа выполнена при финансовой поддержке РФФИ (грант № 11-07-00963а).

Обобщенная структура современных GRID-систем. В результате анализа современных GRID-систем можно сделать вывод, что их организация, так или иначе, сводится к схеме, представленной на рис. 1, и содержит три логических уровнях:

- ◆ **пользовательский уровень** – программно-аппаратные средства, на которых формируется и компилируется задача, которая должна быть решена в GRID;
- ◆ **служебный уровень** – программные и аппаратные средства, обеспечивающие функционирование GRID и выполняющие роль «посредника» между пользователями и исполнителями;
- ◆ **исполнительный уровень** – программно-аппаратные средства, используемые для решения пользовательских задач в GRID.

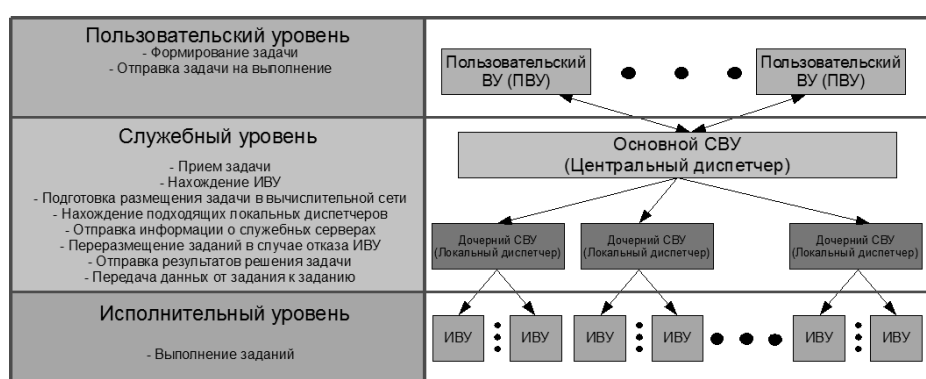


Рис. 1. Распределение функций в иерархической GRID-системе

Основную роль в такой системе играет служебный уровень, на который возлагаются функции, связанные с организацией и диспетчированием работы всей системы в целом (см. левую часть рис. 1). При этом в современных GRID-системах, как правило, служебный уровень состоит из некоторого количества иерархически связанных серверов. При такой организации основному СВУ (центральному диспетчеру) подчиняется несколько дочерних СВУ (локальных диспетчеров), ответственных за реализацию некоторого набора специальных функций или за диспетчирование работы некоторого подмножества ИВУ исполнительного уровня.

Подобной схеме организации GRID присущи некоторые недостатки:

- ◆ Во-первых, на служебный уровень системы возлагается целый ряд достаточно сложных функций (см. рис. 1), реализация которых требует наличия мощных вычислительных средств. Поэтому, как правило, служебный уровень системы строится на основе высокопроизводительных серверов.
- ◆ Во-вторых, вычислительная сложность задач диспетчирования, возлагаемых на служебный уровень, заметно возрастает при увеличении числа ИВУ, задействованных в системе. Решение этих задач за ограниченный промежуток времени опять-таки требует наличия мощных вычислительных устройств, реализующих служебный уровень, либо они могут быть решены в заданный срок путем увеличения количества локальных диспетчеров.
- ◆ В-третьих, указанная система обладает невысокой надежностью, поскольку выход из строя одного из СВУ приводит к её частичной или даже полной неработоспособности. Повышение надежности системы путем дублирования диспетчеров потребует двукратного увеличения количества СВУ в системе, что в большинстве случаев является неоправданным шагом.

Все приведенные выше недостатки приводят к значительному повышению стоимости содержания и обеспечения работоспособности GRID, что, в свою очередь, ведет к удорожанию GRID-вычислений для конечного пользователя. Указанные недостатки могут быть устранены или, по крайней мере, существенно снижены, если минимизировать функции, возлагаемые на СВУ. Этого можно достичь, если переложить часть функций служебного уровня, реализуемых в современных GRID на СВУ, на компьютеры исполнительного уровня (ИБУ).

При этом нельзя просто перенести все эти функции на один из ИБУ, поскольку мы снова получим тот же самый набор проблем. Переносимые функции должны быть распределены между множеством ИБУ, задействованных в составе GRID. Иными словами, функции служебного уровня должны быть децентрализованы между компьютерами исполнительного уровня таким образом, чтобы, во-первых, отключение любого из ИБУ не приводило к сбою системы, а во-вторых, была обеспечена возможность неограниченного масштабирования исполнительного уровня GRID путем подключения дополнительных ИБУ.

Принципы децентрализованной организации GRID. Основной идеей предлагаемого подхода к построению GRID-систем является минимизация функций служебных узлов, ответственных за организацию (диспетчирование) процесса вычислений [8]. Это позволит добиться увеличения отказоустойчивости и повышения возможностей масштабирования, однако использование данного подхода требует применения новых методов и алгоритмов децентрализованной организации и диспетчирования процессов вычисления в GRID. В качестве основы для разработки таких методов и алгоритмов целесообразно использовать мультиагентные технологии. Это позволит решать задачу диспетчирования GRID-системы в целом путем решения множества локальных задач, распределенных по отдельным ИБУ, задействованными в системе.

На основе этих соображений, основные принципы работы децентрализованной GRID с мультиагентным диспетчером можно представить следующим образом. Основными взаимодействующими элементами системы целесообразно сделать агентов, которые выполняют в GRID функции как исполнителя, так и диспетчера. Агенты физически реализуются на базе ИБУ, при этом нет централизованного управления агентами: они являются проактивными, самостоятельно взаимодействуют и решают совместно задачи, поступающие в систему. При этом целесообразно положить, что основной мотивацией, которая будет толкать независимых агентов на участие в работе GRID, будет получение некоторой виртуальной или материальной выгоды: за решение каждой задачи ее заказчик (пользователь) предлагает некоторое количество виртуальных очков. Эти очки могут в реальности быть чем угодно: деньгами, различными привилегиями, просто показателем проделанной определенным ИБУ работы. Таким образом, агент заинтересован в решении как можно большего количества задач за как можно меньшее время, при этом для него выгодно решать задачи с наибольшей ценой и наименьшим количеством работы.

Однако, в связи с различной сложностью задач, одному независимому агенту зачастую не удастся решить задачу в течение заданного времени. Поэтому для решения каждой полученной системой задачи необходимо сформировать некоторое множество (сообщество) независимых агентов, способное решить данную задачу за требуемое время. Под сообществом в дальнейшем будем понимать некоторый сформированный коллектив агентов, объединившихся для решения определенной общей задачи. Основной целью сообщества является решение своей задачи в указанный срок, при этом состав сообщества может динамически изменяться в процессе решения задачи: отказы оборудования ИБУ, снижение скорости передачи

данных и другие изменения вычислительной сети, приводящие к снижению скорости решения задачи ниже необходимой для своевременного решения, могут вызывать изменение состава сообщества.

В то же время подобная децентрализованная организация системы накладывает определенное ограничение на способы взаимодействия пользователей и исполнителей GRID-системы. Если при иерархической организации GRID такое взаимодействие осуществляется посредством СВУ служебного уровня, то в данном случае пользователь должен каким-то образом передавать задачи исполнителям (ИБУ), адреса и состояния которых могут динамически изменяться.

Данная проблема может быть решена путем организации взаимодействия пользователей и исполнителей с помощью пассивных служебных серверов, реализующих минимум функций. За счет пассивности наличие таких узлов в системе не приведет к централизации функций, а если функциональная нагрузка таких узлов будет сведена к минимуму, то для их реализации не потребуется высокопроизводительных компьютеров. Такие серверы фактически будут играть роль «досок объявлений», с помощью которых пользователи и ИБУ смогут осуществлять обмен информацией. Обобщенная схема организации подобной децентрализованной GRID-системы представлена на рис. 2.

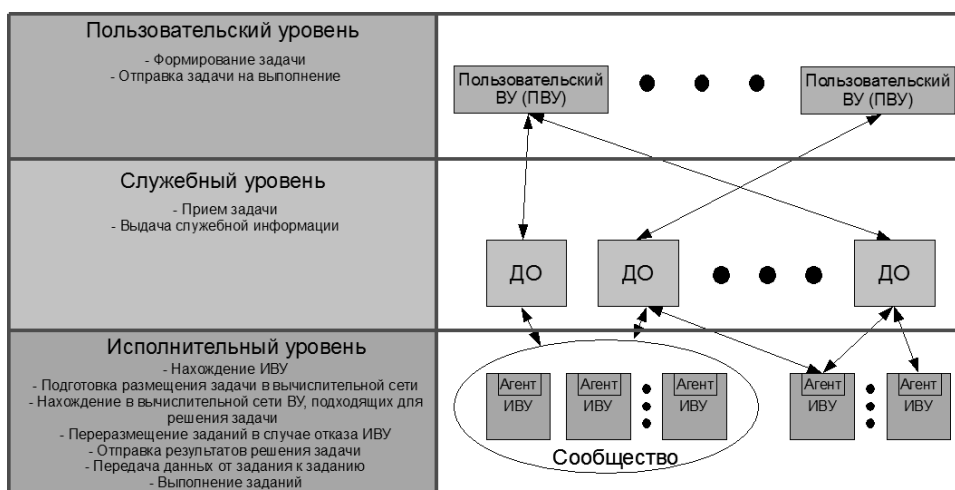


Рис. 2. Распределение функций в децентрализованной GRID на базе сообществ агентов

Здесь доске объявлений отданы только функции приема и хранения задач, а также выдачи служебной информации о задачах и состоянии системы, в то время, как остальные функции диспетчирования реализуются на уровне множества ИБУ. Количество досок объявлений, используемых в системе, ничем не ограничено, они могут быть выделены как пользователями GRID-системы для своих собственных нужд, так и быть предоставленными третьей стороной, заинтересованной в получении выгоды от GRID-системы. Это фактически делает систему полностью децентрализованной как на исполнительном, так и на служебном уровнях.

Алгоритм решения задачи в децентрализованной GRID. На базе описанного выше подхода можно представить следующую укрупненную схему решения задачи в децентрализованной GRID (рис. 3).

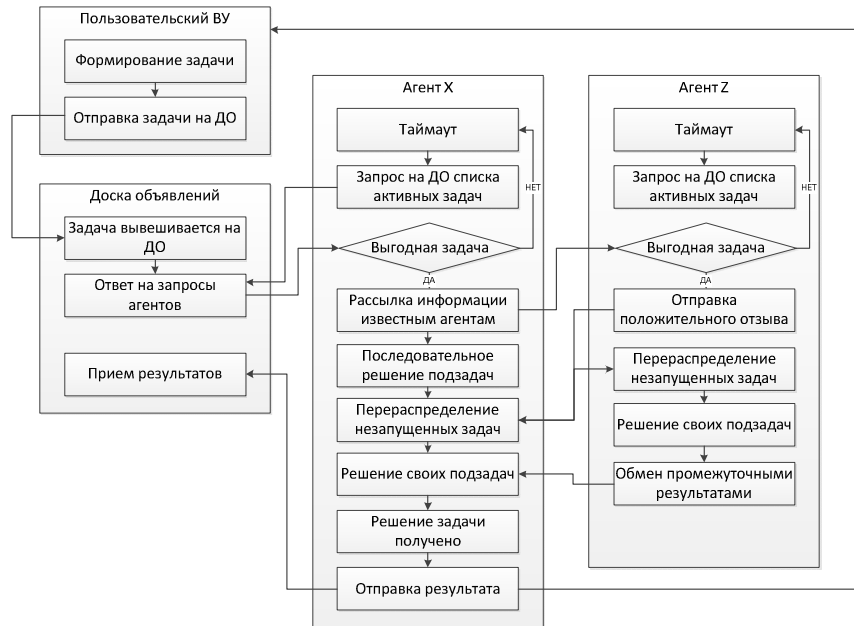


Рис. 3. Схема решения задачи в децентрализованной GRID

При этом во время работы децентрализованная GRID-система руководствуется следующим алгоритмом:

1. Пользователь формирует задачу в виде набора взаимосвязанных подзадач, каждая из которых характеризуется набором ее параметров.
2. Пользователь отправляет информацию о задаче на доску объявлений.
3. Свободные агенты с определенным интервалом опрашивают доски объявлений, откуда им отправляется информация об имеющихся задачах.
4. Агент просматривает список доступных задач и выбирает из них ту, которую наиболее выгодно решать.
5. Агент переходит в режим создания сообщества для решения выбранной задачи: рассылает всем известным по предыдущей работе агентам предложение участвовать в решении задачи, а затем начинает решение самостоятельно.
6. Каждый из свободных агентов, получивших информацию о задаче и согласных участвовать в ее решении, отправляет положительный отзыв, затем незапущенные подзадачи перераспределяются в обновленном сообществе.
7. Как только необходимое для решения задачи в указанный срок количество агентов набрано, дальнейший набор агентов прекращается.
8. Если в процессе решения задачи какой-либо ИВУ сообщества отказывает, в сообществе инициируется поиск дополнительных ИВУ для решения.
9. Как только сообщество агентов получает окончательно решение задачи, оно отправляется на доску объявлений и заинтересованному пользователю.

Данному алгоритму отвечает схема организации вычислительного процесса в децентрализованной GRID, представленная на рис. 3.

Программная модель децентрализованной GRID. Для проверки работоспособности предложенного подхода была разработана имитационная модель распределенной GRID-системы, позволяющая моделировать работу децентрализо-

ванной GRID, описанной выше. Интерфейс имитационной модели представлен на рис.4 и 5. Здесь таблица в правой части окна – это набор пользовательских задач, решаемых в системе, моменты времени их поступления в систему, их статус и параметры. По центру окна расположено визуальное отображение множества ИВУ системы: каждая из окружностей представляет собой отдельного агента (и соответственно ИВУ), на котором он запущен, цвет окружности соответствует цвету задачи, которая выполняется на данном ИВУ (рис. 5). В левой части окна расположены элементы управления экспериментом и статистика.

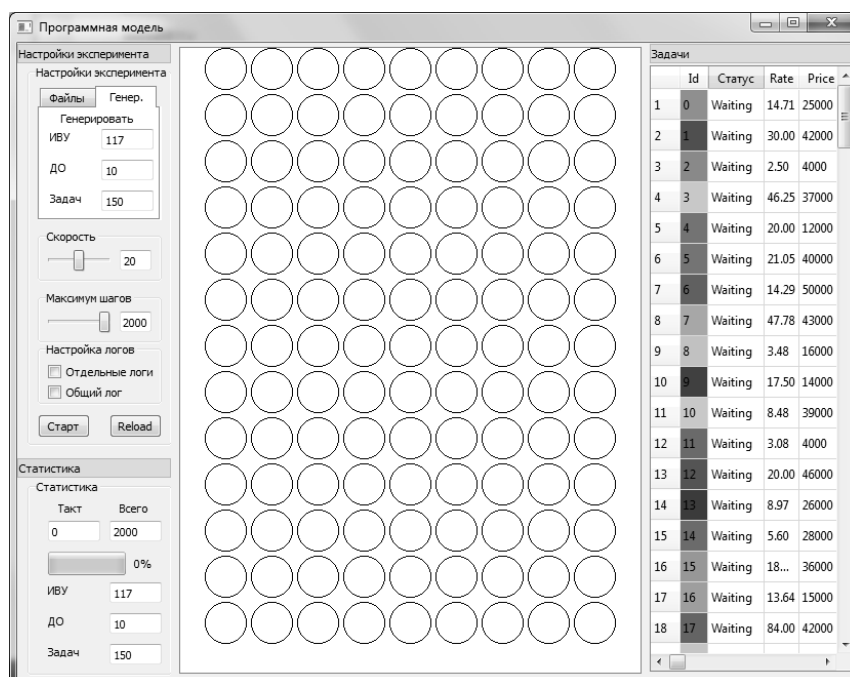


Рис. 4. Интерфейс имитационной модели децентрализованной GRID в начале работы

В общем виде процесс моделирования может быть описан следующим образом. Изначально в имитационной модели формируется список задач, которые должны быть решены в системе, при этом каждая задача может состоять из нескольких подзадач [9], каждая из которых может решаться на одном из ИВУ. Каждый из агентов в определенной последовательности начинает поиск задач для решения. Для этого он посылает сообщения с запросом на каждую из досок объявлений, существующую в системе. Изначально ДО не содержит задач, но в определенные заранее моменты времени задачи поступают на ДО (в интерфейсе задача помечается как Active), и тогда ДО в ответ на запросы отправляет агентам список имеющихся в наличии задач со всеми необходимыми параметрами. Агент, получая сообщение со списком, выбирает для себя наиболее выгодную для решения задачу (по соотношению цена / количество вычислений), проверяет, достаточно ли количества ИВУ, уже работающих над задачей, для ее решения к заданному моменту времени. Если количества ИВУ достаточно, агент ищет следующую задачу, если не хватает, то отправляет ДО и другим агентам сообщения об участии в решении задачи. При этом в интерфейсе программной модели окружность агента заполняется цветом задачи, которую он выбрал для решения. Затем агент выбирает сво-

бодную подзадачу и начинается её решение (которое моделируется путем задержки работы агента на срок, определяемый с помощью значений вычислительной мощности агента и сложности подзадачи). Как только все подзадачи общей задачи решены с помощью ИВУ сообщества, считается, что задача решена в целом, и ее статус в интерфейсе программы меняется на Solved. После этого все агенты, участвовавшие в решении данной задачи, начинают процесс поиска очередной задачи на ДО. Этот процесс происходит в цикле до тех пор, пока не будут решены все задачи, имеющиеся в системе, или не закончится время моделирования.

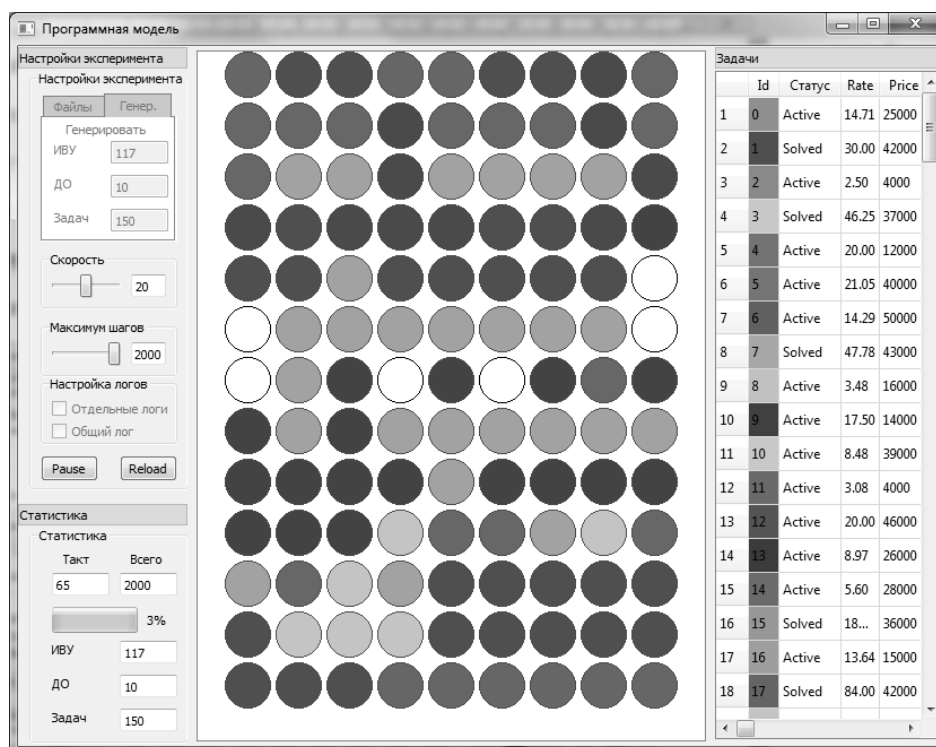


Рис. 5. Интерфейс имитационной модели децентрализованной GRID в процессе моделирования

Для проведения исследований было составлено несколько серий экспериментов с различными параметрами системы и задач, при этом основной целью исследований были экспериментальное доказательство работоспособности разработанных алгоритмов функционирования децентрализованной GRID, а также расчет относительных потерь времени работы ИВУ на организацию процедур децентрализованного диспетчирования [10]. Каждый эксперимент проходил следующим образом: моделировалась децентрализованная GRID, в зависимости от эксперимента, содержащая от 1 до 10000 ИВУ и от 1 до 100 ДО. Также создавался набор задач (от 1 до 1000) различной сложности и с различным количеством подзадач, для каждой задачи указывался или генерировался определенный момент поступления в систему. Кроме того, составлялся график отказов, который позволял моделировать отказ любого ИВУ или любой ДО в заданный момент времени в процессе функционирования GRID. После настройки необходимых параметров начинался процесс моделирования: в заданные моменты времени на ДО системы подавались определенные задачи из списка задач программной модели, агенты ИВУ об-

рашались к ДО и решали поступающие задачи, при этом велся подсчет времени, потраченного ИВУ непосредственно на диспетчирование, и общего времени работы для решения всех задач. Кроме того, фиксировались моменты успешного завершения задач. Проведенные исследования показали, что все задачи, поступившие в систему, были успешно решены, при этом вычислительное время, потраченное на полезную работу ИВУ по решению пользовательских задач, составило от 82 до 94 % от общего времени работы в зависимости от параметров моделирования и от потока задач.

Заключение. Предложенный в работе подход к построению децентрализованных GRID-систем обладает рядом преимуществ.

Во-первых, снижается стоимость вычислений в GRID-системах, поскольку на служебный уровень системы возлагается ряд простейших пассивных функций доски объявлений, реализация которых не требует наличия дорогостоящего оборудования и больших затрат на его обслуживание.

Во-вторых, при децентрализованной организации GRID исчезает проблема масштабируемости. Это происходит потому, что количество исполнительных узлов в вычислительной сети практически не влияет на вычислительную загрузку каждого из них, так как функции диспетчирования распределены равномерно.

В-третьих, децентрализованная организация GRID подразумевает более высокую отказоустойчивость в связи с вынесением большинства функций управления с центрального диспетчера на узлы вычислительной сети, которые являются независимыми друг от друга.

Все это позволяет сделать вывод о том, что предложенная децентрализованная организации диспетчера GRID на базе сообществ агентов даст возможность создавать гибкие GRID различного масштаба и целевого назначения, а это, в свою очередь, позволит расширить сферу применения GRID-компьютинга.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Foster I., Kesselman C. and Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of High Performance Computing Applications. – 2001. – № 15 (3). – P. 200-222.
2. Инструментарий Globus Toolkit – <http://www.globus.org>.
3. Система Condor – <http://www.cs.wisc.edu/condor>.
4. Филамофитский М.П. Система поддержки метакомпьютерных расчетов X-Com: архитектура и технология работы // Вычислительные методы и программирование. – 2004. – Т. 5. – С. 123-137.
5. Foster C., Kesselman J., Nick S. Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration., Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002Metropolis N. et al. Equation of state calculations by fast computing. Machines // J.Chem.Phys. – 1953. – Vol. 21.
6. Каляев И.А., Гайдук А.Р., Капустян С.Г. Проблемы интеллектуального управления мультиробототехническими системами // Мехатроника, автоматизация, управление (МАУ-2009): Материалы Междунар. науч.-техн. конф. Т. 2. – Таганрог: Изд-во ТТИ ЮФУ, 2009. – С. 190-194.
7. Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов. – М.: Физматлит, 2009. – 279 с.
8. Каляев А.И. Разработка и исследование новых децентрализованных методов организации вычислительного процесса в информационно-управляющих системах // Тезисы докладов Четвертой ежегодной научной конференции студентов и аспирантов базовых кафедр ЮНЦ РАН. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2008. – С. 101-102.
9. Каляев А.И. Разработка алгоритма взаимодействия между подзадачами в отказоустойчивых распределенных информационно-управляющих системах // Материалы Междунар. науч.-техн. конференции “Высокопроизводительные вычислительные системы – 2008” (ВПВС-2008). – Таганрог: Изд-во ТТИ ЮФУ, 2008. – С. 408-409.

10. Горелова Г.В., Каляев А.И., Мельник Э.В., Радченко С.А. Планирование эксперимента при исследовании новых методов и алгоритмов организации распределенных вычислений // Вестник информационных и компьютерных технологий. – 2007. – № 10. – С. 49-56.

Статью рекомендовал к опубликованию д.т.н. С.Г. Капустян.

Каляев Анатолий Игоревич

Научно-исследовательский институт многопроцессорных вычислительных систем им. акад. А.В. Каляева Южного федерального университета, г. Таганрог.

E-mail: anatoly.kaliaev@gmail.com.

477922, г. Таганрог, ул. Р. Люксембург, 8-А.

Тел.: +79043497677.

Старший программист.

Kalyaev Anatoly Igorevich

Research Institute of Multiprocessor Computer Systems Southern Federal University.

E-mail: anatoly.kaliaev@gmail.com.

8-A, R. Luxemburg Street, Taganrog, 477922, Russia.

Phone: +79043497677.

Programmer.