



Math-Net.Ru

Общероссийский математический портал

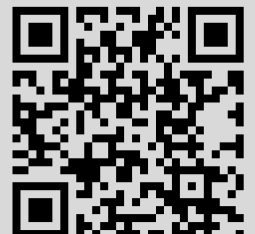
И. В. Бычков, Г. А. Опарин, А. Г. Феоктистов, В. Г. Богданова, А. А. Пашинин,
Сервис-ориентированное мультиагентное управление распределенными вычислениями, *Автомат. и телемех.*, 2015, выпуск 11, 118–131

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.142.255.225

1 ноября 2024 г., 20:19:38



Интеллектуальные системы управления

© 2015 г. **И.В. БЫЧКОВ**, д-р техн. наук (idstu@icc.ru),
Г.А. ОПАРИН, д-р техн. наук (oparin@icc.ru),
А.Г. ФЕОКТИСТОВ, канд. техн. наук (agf@icc.ru),
В.Г. БОГДАНОВА, канд. техн. наук (bvg@icc.ru),
А.А. ПАШИНИН (apcrol@gmail.com)

(Институт динамики систем и теории управления им. В.М. Матросова СО РАН,
Иркутск)

СЕРВИС-ОРИЕНТИРОВАННОЕ МУЛЬТИАГЕНТНОЕ УПРАВЛЕНИЕ РАСПРЕДЕЛЕННЫМИ ВЫЧИСЛЕНИЯМИ¹

Рассматриваются мультиагентные методы и инструментальные средства для эффективного управления потоками заданий, порождаемых сервис-ориентированными приложениями. Эти разработки интегрированы в рамках единой технологии поддержки автоматизации процесса решения больших научных задач в современной кластерной Grid, вычислительные узлы которой (кластеры) могут иметь сложную гибридную структуру. Новизна и практическая значимость представленных в статье методов и средств заключаются в существенном расширении функциональных возможностей системы управления вычислениями кластерной Grid (по сравнению с известными) по распределению и разделению ресурсов Grid на разных уровнях выполнения заданий, в наличии возможности встраивания интеллектуальных средств управления вычислениями в проблемно-ориентированные приложения.

1. Введение

Анализ тенденций развития технологий построения высокопроизводительных распределенных систем как в России, так и за рубежом позволяет сделать вывод о необходимости интеграции методов и средств повышения эффективности загрузки вычислительных ресурсов при решении больших научных задач, всестороннего учета специфики предметных областей в процессе решения этих задач с помощью пользовательских приложений и обеспечения возможности гибкого использования приложений на основе сервис-ориентированного подхода

К настоящему времени разработан широкий спектр инструментов для построения сервис-ориентированных распределенных вычислительных сред. К их числу относятся, например, средства общего назначения Amazon Elastic Compute Cloud, Google AppEngine Microsoft Windows Azure и Manjrasoft Aneka [1], специализированные системы Globus Toolkit [2] и Unicore [3], а

¹ Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект № 15-29-07955-офи_м).

также язык программирования Ора [4]. Из отечественных средств построения сервис-ориентированных сред следует отметить систему MathCloud [5] и системы, реализуемые на основе концепции Intelligent Problem Solving Environment (iPSE) [6]. Основной упор в системах для построения сервис-ориентированных распределенных вычислительных сред, как правило, делается на упрощение процесса создания сервисов. В подобных системах управление вычислениями реализуется либо с помощью традиционных метапланировщиков таких как GridWay [7] или Workload Management System [8], либо с помощью специализированных системных средств

Традиционные метапланировщики недостаточно полно учитывают специфические требования пользователей к ресурсам при распределении этих ресурсов. Поэтому пользователям и администраторам ресурсов распределенной вычислительной среды приходится “вручную” (соответственно путем спецификации пользователем каждого своего задания и настройки администратором конфигурационных параметров метапланировщика для определения политик в отношении ресурсов и заданий пользователей) решать задачи выбора и предоставления ресурсов, которые бывают весьма противоречивыми.

В связи со сложностью решения перечисленных задач возникает проблема автоматизации и интеллектуализации процессов их решения. Широко используемым на практике подходом к решению этой проблемы является применение мультиагентных систем (МАС) для управления вычислениями [9]. Повышение качества управленческих решений в МАС зачастую достигается путем использования экономических механизмов регулирования спроса и предложения ресурсов распределенной вычислительной среды [1]. Можно выделить два основных подхода к мультиагентному управлению вычислениями [11]: взаимодействие МАС с локальными менеджерами ресурсов узлов среды с целью оптимизации использования ресурсов и интеграция пользовательского приложения с мультиагентной системой выбора ресурсов с целью повышения эффективности решения задач приложением.

В первом случае, как правило, использование МАС предполагает замену традиционных метапланировщиков специальными агентами управления вычислениями. Вследствие такой замены каждый пользователь независимо от его желания становится глобальным пользователем, осуществляющим взаимодействие с ресурсами распределенной вычислительной среды только с помощью МАС. Таким образом, ограничиваются возможности широкого круга локальных пользователей, желающих решать свои задачи в конкретных узлах среды без использования промежуточного “посредника”. Во втором случае, при наличии большого числа приложений различных пользователей, эффективность систем управления вычислениями может быть существенно снижена вследствие конкуренции агентов приложений за общие разделяемые ресурсы.

В статье рассматривается МАС, обеспечивающая интеграцию двух рассмотренных подходов к управлению вычислениями для кластерной Grid-системы вычислительного типа, узлы (кластеры) которой имеют сложную гибридную структуру. Гибридный кластер включает вычислительные модули (аппаратные компоненты), поддерживающие различные технологии па-

раллельного программирования и различающиеся своими вычислительными характеристиками.

2. Управление вычислениями на уровне Grid-системы

Управление вычислениями на уровне Grid-системы реализуется МАС с заданной организационной структурой. Координация действий агентов осуществляется с помощью общих правил группового поведения. Агенты функционируют в соответствии с заданными ролями и для каждой роли определены свои правила поведения в виртуальном сообществе агентов. МАС включает агентов распределения ресурсов и управляющего агента. Агенты распределения ресурсов могут объединяться в виртуальные сообщества (ВС). В разных ВС, возникающих в МАС, агенты могут координировать свои действия путем кооперации или соперничества.

Задачей МАС на уровне Grid-системы является получение такого распределения поступающих в систему потоков заданий, которое сохраняет показатели качества функционирования этой системы в заданных администратором Grid-системы пределах. Задание представляет собой спецификацию процесса решения задач, содержащую информацию о требуемых вычислительных ресурсах, исполняемых прикладных программах, входных/выходных данных, а также другие необходимые сведения. Все задания разбиваются по классам в соответствии со своими вычислительными характеристиками [12]. В число показателей качества функционирования Grid-системы входят: коэффициенты полезного использования узлов Grid-системы и самой системы в целом; средние показатели времени нахождения в очередях, коэффициенты успешного завершения и средняя стоимость выполнения заданий различных классов в узлах Grid-системы и в самой системе в целом. Структурная схема системы управления вычислениями на уровне Grid-системы представлена на рис. 1.

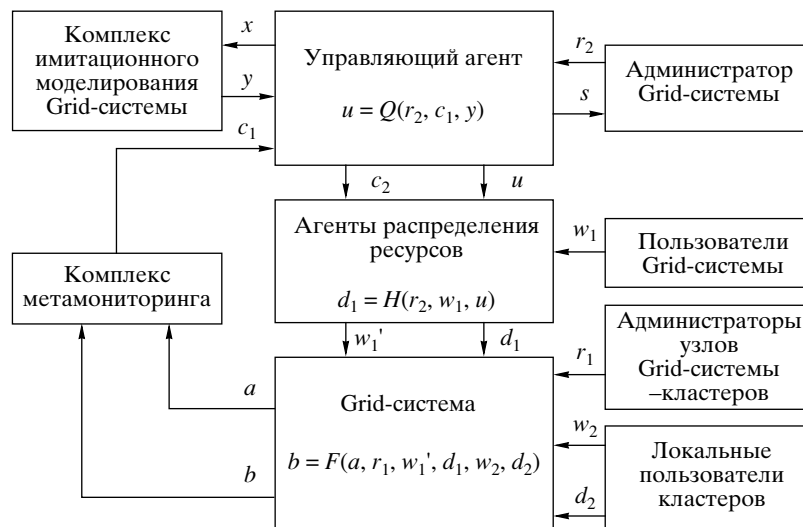


Рис. 1. Структурная схема системы управления вычислениями.

На этой схеме в качестве объекта управления выступает Grid-система, узлы которой представлены разнородными вычислительными кластерами (ВК), в том числе и гибридными. Внешними возмущениями для объекта управления являются поток w_1 заданий пользователей Grid-системы и поток w_2 локальных пользователей ВК. Результаты распределения d_1 и d_2 потоков w_1 и w_2 по ВК являются соответственно управляющими воздействиями МАС и локальных пользователей ВК на объект управления. Задающим воздействием для объекта управления является вектор параметров административных политик ВК r_1 . Агенты распределения ресурсов осуществляют перехват заданий потока w_1 с целью более детальной настройки требований к вычислительной системе, содержащихся в заданиях. Таким образом, поток w_1 модифицируется в поток w'_1 . Распределение d_1 потока w'_1 производится агентами распределения ресурсов на основе экономических механизмов регулирования спроса и предложения этих ресурсов [13]. Распределение d_2 потока w_2 задается локальными пользователями ВК. Потоки заданий w_1 и w_2 характеризуются следующими свойствами: динамичностью (мощность и состав потока заданий изменяется во времени); стохастичностью (поток заданий предполагает возникновение случайных событий); неоднородностью (задания соответствуют разным классам задач и отличаются друг от друга своей спецификой); отсутствием обратной связи (число заданий, поступивших за один промежуток времени, не зависит от числа заданий, поступивших за другой промежуток времени); неординарностью (возможно поступление двух и более заданий в один и тот же момент времени); стационарностью (число событий, поступивших за определенный промежуток времени, зависит от длины этого промежутка и не зависит от момента его начала).

Сведения о вычислительных характеристиках узлов Grid-системы собираются комплексом метамониторинга [14] с помощью контрольно-измерительных приборов в виде файловой структуры данных a . Сведения о текущих показателях объемов вычислительных работ в узлах Grid-системы (выполняющихся и стоящих в очередях) также собираются комплексом метамониторинга в виде файловой структуры данных b . Предполагается, что между компонентами структуры b с одной стороны и вычислительными характеристиками узлов a , задающим воздействием r_1 для объекта управления, потоками заданий w'_1 и w_2 , распределениями d_1 и d_2 с другой стороны существует некоторая абстрактная связь $b = F(a, r_1, w'_1, d_1, w_2, d_2)$. Для различных компонентов структуры b эта связь может быть представлена функциональным, статистическим, неоднозначным или иным отображением. В частности, объем одной и той же вычислительной работы для разнородных узлов определяется с учетом отличий их вычислительных характеристик от соответствующих характеристик виртуального эталонного узла Grid-системы.

Собранные сведения передаются управляющему агенту в виде вектора s_1 агрегированных показателей функционирования объекта управления по его запросу. Запросы управляющего агента к системе мониторинга посылаются с некоторым периодом дискретности T_1 . Величина T_1 выбирается таким образом, чтобы не перегружать коммуникационную среду Grid-системы сбором информации и в то же время с необходимой точностью фиксировать моменты приближения показателей функционирования объекта управления к их

предельным значениям. Часть сведений, представленных вектором c_1 и актуальных для агентов распределения ресурсов, немедленно передается этим агентам в виде вектора c_2 .

Задающим воздействием для управляющего агента является вектор r_2 параметров административных политик Grid-системы. На основе информации, представленной векторами c_1 и r_2 , в заданные моменты времени управляющий агент прогнозирует на определенный промежуток времени динамику показателей качества функционирования объекта управления с помощью комплекса имитационного моделирования Grid-системы. Результаты моделирования используются для формирования вектора управляющих воздействий u на алгоритмы работы агентов распределения ресурсов ВС путем параметрической настройки алгоритмов. Элементами вектора u являются следующие параметры алгоритма работы агента распределения ресурсов: размеры штрафов для “жадных” или “ленивых” агентов относительно среднего объема вычислительных работ в узлах, представленных агентами ВС, и допустимое отклонение от этого объема. Процесс имитационного моделирования инициируется управляющим агентом с некоторым периодом дискретности $T_2 > T_1$. После того как вектор u управляющих воздействий сформирован, он передается каждому ВС. Параметрическая настройка алгоритмов работы агентов распределения ресурсов конкретного ВС осуществляется путем применения полученных управляющих воздействий с учетом весовых коэффициентов, отражающих вычислительные особенности узлов этого ВС.

Итак, пусть x и y – векторы входных и наблюдаемых переменных имитационной модели Grid-системы. Наблюдаемые переменные представляют показатели качества функционирования Grid-системы. Элементы векторов x_i , $i = \overline{1, n_x}$ и y_j , $j = \overline{1, n_y}$ имеют соответственно области X_i и Y_j допустимых значений. Предполагается, что эффекты влияния входных переменных на наблюдаемые переменные исследованы с помощью факторного анализа [15] заранее – при построении и испытании имитационной модели Grid-системы. Также предполагается, что для каждого j -го элемента вектора y администратором Grid-системы задан критерий вычисления оценки \hat{y}_j качества значения этого элемента (стремление значения к минимуму или максимуму на множестве сравниваемых значений) и его предельные значения y_j^{\min} и $y_j^{\max} \in Y_j$. Ряд элементов вектора x играют роль варьируемых переменных, образуют подмножество X^* и отождествляются с элементами вектора u : $u_q \equiv x_i$, $q = \overline{1, n_u}$, $i \in \overline{1, n_x}$, $1 \leq n_u < n_x$. Как правило, при решении практических задач управления вычислениями в Grid-системе в процессе имитационного моделирования целесообразно использовать $q \leq 8$. Число значений варьируемых переменных определяется, исходя из соотношения

$$\left(t_m \times \prod_{q=1}^{n_u} z_q \right) / n_c \leq T_2,$$

где t_m – среднее время прогона имитационной модели, определяемое комплексом метамониторинга на основе вычислительной истории прогонов модели, $z_q > 0$ – число варьируемых значений q -й переменной, n_c – число ядер узла, в котором размещен управляющий агент, $n_u < n_c$. В качестве начальных зна-

чений варьируемых переменных используются базовые значения, соответствующие принятым по умолчанию значениям конфигурационных параметров действующей системы управления вычислениями (например, металланировщика) Grid-системы. Остальные значения варьируемых переменных выбираются из соответствующих областей допустимых значений с учетом эффектов влияния варьируемых переменных на наблюдаемые переменные. Значения неварьируемых входных переменных, являющихся элементами вектора x , задаются на основе соответствующей числовой информации, представленной векторами r_2 и c_1 .

В процессе моделирования выполняется имитация процесса функционирования Grid-системы путем проведения параллельных многовариантных расчетов (прогона модели для каждого сочетания значений варьируемых переменных) и формируется множество V вариантов значений наблюдаемых переменных: значение $y_{jk} \in Y_j$ является элементом k -го варианта $v_k \in V$ для переменной y_j $j = \overline{1, n_y}$, $k = \overline{1, n_v}$. Выбор из множества V подмножества $V^* \subseteq V$ вариантов значений наблюдаемых переменных с целью дальнейшего определения значений элементов вектора u является многокритериальным. Выбор вариантов для подмножества V^* осуществляется управляющим агентом либо на основе лексикографического метода, если администратор Grid-системы может упорядочить наблюдаемые переменные по значимости, либо в противном случае на основе мажоритарного метода.

Лексикографический метод отбора вариантов значений наблюдаемых переменных использует следующее правило многокритериального выбора, приведенное в [16]:

$$V^* = \{v_k \in V : (\forall v_l \in V \exists p \in \overline{1, n_y - 1} : (\hat{y}_{1k} = \hat{y}_{1l}) \wedge \dots \wedge (\hat{y}_{pk} = \hat{y}_{pl}) \wedge (\hat{y}_{(p+1)k} > \hat{y}_{(p+1)l}))\},$$

где

$$y_j^{\min} \leq y_{jk} \leq y_j^{\max}, \quad j = \overline{1, n_y}, \quad k \in \overline{1, n_v}, \quad l \in \overline{1, n_v}, \quad k \neq l.$$

Мажоритарный метод отбора вариантов значений наблюдаемых переменных использует следующее правило многокритериального выбора, приведенное в [16]:

$$V^* = \left\{ v_k \in V : \left(\neg \exists v_l \in V : \sum_{j=1}^{n_y} \text{sign}(\hat{y}_{jl} - \hat{y}_{jk}) > 0 \right) \right\},$$

где $\text{sign}(0) = 0$, $y_j^{\min} \leq y_{jk} \leq y_j^{\max}$, $k \in \overline{1, n_v}$, $l \in \overline{1, n_v}$, $k \neq l$.

Применение рассмотренных выше методов многокритериального выбора обусловлено тем, что они обладают наименьшей сложностью с вычислительной точки зрения по сравнению с другими известными методами решения подобной задачи, просты в реализации и управляющему агенту требуется минимальная дополнительная информация от администратора Grid-системы.

Пусть в результате формирования множества V^* получен единственный k -й вариант v_k значений наблюдаемых переменных. Этому вектору однозначно соответствует k -й вариант значений варьируемых переменных вектора x .

Выбрав из них значения x_{ik} такие, что $x_i \in X^*$, получим значения элементов вектора u . В том случае, когда полученное подмножество V^* содержит более одного варианта значений наблюдаемых переменных, окончательный выбор единственного варианта v_l осуществляется случайным образом. При $V^* = \emptyset$ управляющий агент порождает сигнал s , требующий от администратора Grid-системы новых задающих воздействий, заключающихся в корректировке текущих административных политик системы Grid-системы.

Таким образом, агенты распределения ресурсов осуществляют управляющие воздействия $d_1 = H(c_2, w_1, u)$ на Grid-систему, где управляющее воздействие $u = Q(r_2, c_1, y)$ предназначено для повышения качества решений, принимаемых агентами распределения ресурсов, путем влияния на степень намерений агентов выполнять задания различных классов. Связи H и Q имеют ту же природу, что и рассмотренная связь F . Следует отметить, что при выходе из строя любого агента МАС, включая управляющего агента, функционирование объекта управления продолжится. При этом возможно снизятся лишь показатели качества его работы.

При управлении потоками заданий агентами распределения ресурсов на уровне Grid-системы время выполнения отдельных приложений может увеличиваться, поскольку этим агентам не удастся учесть ряд важных особенностей процесса решения задачи и пользовательских предпочтений, касающихся ресурсов. В разделе 3 статьи представим новые дополнительные агентные средства планирования вычислений и распределения ресурсов на уровне приложений, позволяющие в определенной степени решать эту проблему.

3. Управление вычислениями на уровне приложения

Дополнительные средства управления вычислениями на уровне приложений представляют собой виртуальное сообщество приложения (ВСП), создаваемое для организации параллельного выполнения приложения локального пользователя в Grid-системе. Основное назначение ВСП – обеспечение выбора наименее загруженных ВК запуск параллельного приложения, мониторинг и передача пользователю результатов вычислений. ВСП включает пользовательского агента, агентов классификации и планирования, агента-менеджера и динамически меняющийся набор локальных агентов. Первые три агента осуществляют прием и классификацию [12] запроса пользователя, построение плана вычислений и формирование нового (относительно схемы системы управления вычислениями, рассмотренной ранее) потока w_3 заданий, порождаемого на основе приложения пользователя. Поток w_3 передается агенту-менеджеру, который осуществляет его распределение d_3 по локальным агентам. Необходимую для распределения информацию о вычислительных характеристиках узлов и текущих показателях объемов вычислительных работ ВК, представленную в виде вектора c_2 , агент-менеджер получает от управляющего агента. Агент-менеджер отвечает также за автоматический перезапуск задач с новыми параметрами (для определенного класса задач) и за мониторинг процесса решения задачи пользователя. Локальные агенты отвечают за отправку заданий локальной системе управления ВК, анализ текущего состояния ВК и за передачу результатов выполнения заданий агенту-менеджеру.

Для распределения задач по локальным агентам применяется модель тендера, где в качестве лотов выступают вычислительные работы, а в качестве участников – представители вычислительных ресурсов, претендующие на выполнение работ. Использование аукциона Викри для реализации модели тендера, рассмотренное подробно в [13], позволяет достичь согласованного устойчивого состояния участников аукциона по окончании торгов, а также обеспечить более сбалансированное распределение заданий локальных пользователей кластеров (появление свободных ресурсов Grid-системы в процессе решения пользовательских задач влечет перераспределение в узлах этой системы заданий, управляемых МАС).

Приложение пользователя оформляется в виде Grid-сервиса. Используемый в статье метод создания Grid-сервисов приложений базируется на сочетании технологий Web Services Resource Framework (WSRF) [17] и использовании шаблонов [18] взаимодействия с локальными менеджерами ресурсов ВК. Для создания ВСП используется инструментальная среда High-performance computing Service-oriented Multi-agent System (HpcSoMaS) Framework, разработанная авторами на основе перечисленных технологий. Инструментальная среда включает: библиотеку классов и утилиты создания агентов; библиотеку шаблонов создания сервисов на базе архитектуры Representational State Transfer (REST) [19] и протокола Simple Object Access Protocol (SOAP) [2]; готовые сервисы, разработанные на основе указанных библиотек; спецификации настроек сервисов в формате XML; шаблоны для компонентов обмена XML-сообщениями и взаимодействия с локальными менеджерами ресурсов; инструмент генерации описания Grid-сервиса на языке Web Services Description Language (WSDL) [21]. Для запуска приложения в Grid-системе формируется задание одному из локальных менеджеров ресурсов ВК, используемых в системе. В качестве такого менеджера могут быть использованы как традиционные метапланировщики, так и локальные системы управления ВК, например Portable batch system (PBS) [22] или Condor [23]. Таким образом, в рамках предложенного метода требуется решить две основные задачи: 1) получить описание сервиса для приложения на языке WSDL; 2) конвертировать пользовательский запрос к сервису в вычислительное задание для Grid-системы.

В целом сервис обеспечивает пользователю следующие возможности: формулировку постановок задач, конфигурационную настройку и ввод исходных данных для приложения, получение результатов расчетов, просмотр текущей загруженности ВК и получение информационных уведомлений как почтовых, так и средствами веб-интерфейса. Наряду с перечисленными функциями системного характера в создаваемом ВСП можно учесть возможности, обусловленные спецификой предметной области приложения пользователя, примеры таких ВСП приводятся далее.

4. Результаты исследований

В качестве первого примера применения ВСП был реализован сервис для решения задачи построения области устойчивости в пространстве двух выбранных параметров K и T регулятора замкнутой системы управления, описываемой дифференциальным уравнением $\frac{dx}{dt} = Ax$, где элементы матрицы A

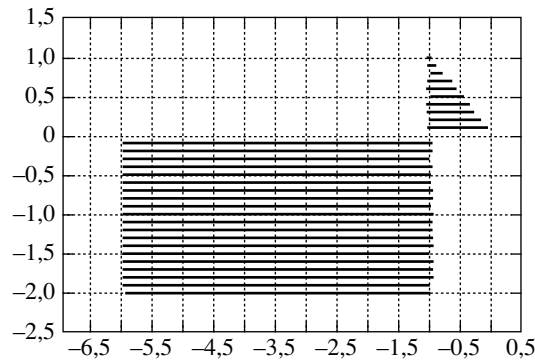


Рис. 2. Пример графического изображения сервисом области устойчивости.

зависят от параметров K и T . Эта задача сводится к решению множества независимых подзадач (проведению многовариантных расчетов) по определению устойчивости матрицы A при изменении значений параметров K и T в заданных диапазонах $K_{\min} \leq K \leq K_{\max}$ и $T_{\min} \leq T \leq T_{\max}$ с шагом ΔK и ΔT соответственно. Путем варьирования значений параметров K и T строится числовая сетка, на основе которой формируется множество подзадач. Необходимым условием решения исходной задачи является выполнение задания для решения каждой подзадачи. Вычисление собственных значений произвольной плотной матрицы выполняется с использованием алгоритмов, представленных в [24]. Результат работы сервиса представляет собой табличные данные и графическое изображение области устойчивости. Например, область устойчивости для системы

$$\begin{aligned} \frac{dx}{dt} &= Kx + Ty, & \frac{dy}{dt} &= x - z, & \frac{dz}{dt} &= -x + y \\ -2 \leq T \leq 1, & & -6 \leq K \leq 0 & & & \end{aligned}$$

изображена на рис. 2. Более детальное описание работы пользователя с этим сервисом приведено в работе [25].

В качестве второго примера реализован сервис для решения SAT-задач (задач булевой выполнимости) путем проведения многовариантных расчетов на основе декомпозиции исходной задачи. Декомпозиция SAT-задачи на произвольное число независимых подзадач, объединение решений которых дает решение исходной задачи, выполняется на основе метода расщепления из [26]. Каждая независимая подзадача может решаться в своем узле Gridсистемы. В этом случае многовариантные расчеты могут выполняться в режиме динамического выбора ресурсов. В вычислительных экспериментах использовались как существующие SAT-решатели с открытой лицензией, так и разработанные авторами решатели [27, 28].

Пользователь с помощью веб-интерфейса (пользовательского агента) формулирует постановку задачи: передает файл с исходной булевой функцией, заданной в виде конъюнктивной нормальной формы (КНФ), затем инициирует процесс решения задачи. На основе исходных данных пользовательский

агент формирует множество подзадач и передает их агенту планирования вычислений, который для каждой подзадачи формирует задание. Далее информация о пуле заданий направляется агенту-менеджеру, который распределяет задания между локальными агентами. Для выполнения задания локальный агент запускает один SAT-решатель. Агенты мониторинга отслеживают состояние каждого задания. В случае появления дополнительных свободных вычислительных ресурсов в кластерной Grid-системе и при наличии пула заданий у агента-менеджера процесс вычислений приостанавливается, одна из активных подзадач снимается с решения и для нее производится дальнейшая декомпозиция, формируются дополнительные задания и процесс вычислений возобновляется. В случае нахождения решения одним из заданий остальные задания в зависимости от статуса выполнения либо снимаются с решения, либо удаляются из очереди. Если после выполнения всех заданий решение не найдено, то пользователь получает уведомление о завершении процесса решения поставленной задачи с ответом “unsat”.

Для эффективного использования ресурсов Grid-системы в процессе решения SAT-задачи были разработаны гибридные методы ее решения, применяющие сразу несколько параллельных технологий программирования: технологии MPI для организации обменов данными между узлами ВК; технологии OpenMP (или библиотеки Pthread) для организации нескольких потоков процессора в рамках одного узла ВК; технологии CUDA для организации вычислений на графических ускорителях. Также были разработаны методы динамического перераспределения потоков заданий для центральных и графических процессоров.

В табл. 1 представлены результаты решения известной задачи Эйлера о ходе шахматного коня с указанием среднего и минимального времени для серии из 100 запусков. В табл. 2 представлены результаты решения ряда sat-задач с помощью сервиса гибридного решателя hpcsat, разработанного в ИДСТУ СО РАН. Результаты приведены для задач, не решенных за 5000 секунд последовательными и параллельными sat-решателями, принимавшими участие в соревнованиях SAT Competition13. Результаты вычислительных экспериментов позволяют сделать следующие выводы. В первом примере сервис приложения предназначен для реализации многовариантных расчетов с использованием статического выбора ресурсов (для решения задачи необходимо выполнение всех заданий, сформированных агентом планирования вычислений до начала вычислительного процесса). Очевидно, что для задач такого вида преимущество ВСП по сравнению с локальными менеджерами ресурсов ВК незначительно. Во втором же примере сервис приложения предназначен для реализации многовариантных расчетов в режиме динамического выбора ресурсов (в процессе решения задачи выполняющиеся задания могут приостанавливаться и далее удаляться, а задания, находящиеся в очереди узла Grid-системы, могут перемещаться в другие узлы этой среды, кроме того, могут формироваться новые задания). Для решения подзадач используются как последовательные, так и параллельные решатели. В этом случае применение ВСП позволяет существенно сократить время решения задачи по сравнению с локальными менеджерами ресурсов ВК за счет выбора наиболее оптимальных решателей и возможности миграции заданий по узлам Grid-системы.

Таблица 1. Сравнение времени решения sat-задачи при распределении потока заданий приложения пользователем (d_2) и агентом-менеджером ВСП (d_3)

КНФ	Число переменных/дизъюнктов	Среднее время решения sat-задачи, с		Минимальное время решение sat-задачи, с	
		d_2	d_3	d_2	d_3
knight8	4096/491024	183,0	132,0	61,0	0,6
knight9	6561/1007603	499,0	359,0	282,0	199,0
knight10	10000/1913276	3599,0	2464,0	651,0	276,0

Таблица 2. Время решения sat-задач, полученное с помощью hpcsat

КНФ	Переменные/дизъюнкты	Время на 256 потоках, с	Время на 1024 потоках, с
rbsat-v1150c84314gyes7.cnf	1150/84314	1316	514
toughsat_factoring_inf.cnf	2878/15516	2147	553
gss-25-s100.cnf	31931/96111	1985	126
b04_s_2_unknown_pre.cnf	123133/801488	2988	1640

В обоих примерах предоставляющие сервисы ВСП реализованы с помощью инструментальных средств, рассмотренных в разделе 3. В первом примере специфика решаемой задачи требует включения в сервис функций графической визуализации результата. Во втором примере был рассмотрен частный случай решения систем булевых уравнений, sat-задача, для которой использовалась построенная ранее булева модель. Однако в общем случае учет специфики предметной области, подробно описанной в [27], осуществляется, во-первых, за счет дополнения перечисленных в разделе 3 функций ВСП функциями построения, анализа и преобразования к разным форматам булевой модели; во-вторых, путем указания режима проведения вычислительного эксперимента (решение задачи, тестирование решателя или тестирование модели), в зависимости от которого сервис требует заполнения пользователем различных наборов параметров; в-третьих, с помощью формулировки постановки задачи (поиска первого решения, k решений или всех решений задачи), соответствующей предметной области решаемой задачи. В первых двух случаях при достижении результата незавершенные задания могут быть удалены, что и используется во втором примере для сокращения времени решения задачи и повышения степени рациональности использования вычислительных ресурсов. Кроме того, в зависимости от предпочтений пользователя, анализа структуры булевой функции и наличия свободных вычислительных ресурсов агентами может осуществляться выбор sat-решателя.

Вычислительные эксперименты проводились в разнородной кластерной Grid-системе ИДСТУ СО РАН, включающей: однородный ВК “Blackford”, имеющий 20 вычислительных узлов, в каждом из которых установлены по два четырехядерных процессора Intel Xeon 5345 EM64T (“Clovertown”) 2.33 GHz; неоднородный ВК “Академик В.М. Матросов”, имеющий 110 двух-процессорных узлов с 16-ядерными процессорами AMD Opteron 6276 2,3 GHz (“Interlagos”) на основе x86_64-микроархитектуры “Bulldozer” и узел с графическими процессорами NVidia C2070 (“Fermi”); неоднородный ВК с GPU

NVidia “Tesla”, в состав которого входят четыре четырехъядерных процессора Intel Xeon X5570 (“Nehalem”) и 8 GPU NVidia “Tesla” C1060 с общим числом ядер 1920.

5. Заключение

В статье рассмотрены сервис-ориентированные методы и средства управления проблемно-ориентированными распределенными вычислениями в кластерной Grid-системе, интегрированные с традиционными метапланировщиками и локальными менеджерами ресурсов узлов Grid-системы, в том числе оригинальные методы и средства конвертирования пользовательских запросов к сервис-ориентированной среде в вычислительные задания, классификации заданий и декомпозиции ресурсов среды в соответствии с классами заданий, разнообразные мультиагентные средства управления вычислениями, новые высокоуровневые инструментальные средства генерации спецификаций и построения интерфейсов сервис-ориентированных приложений.

Перечисленные методы и средства обладают рядом отличительных особенностей. Во-первых, они обеспечивают возможность разработки и выполнения сервисов приложений в разных режимах: управление вычислениями как на уровне отдельных приложений, так и на уровне потоков заданий; распределение ресурсов, необходимых для выполнения операций сервисов, специальными агентами или традиционными локальными менеджерами ресурсов, применение статического или динамического планирования вычислений. Во-вторых, создание ВСП агентов, функционирующего на уровне приложений, позволяет представить в виде Grid-сервисов ряд системных функций приложений, в частности планирование вычислений, распределение и мониторинг ресурсов, определение статуса заданий, учет специфики проблемной области решения задачи в процессе управления вычислениями.

СПИСОК ЛИТЕРАТУРЫ

1. *Rajkumar Buyya R., Vecchiola C., Selvi S.T.* Mastering Cloud Computing. Burlington, Massachusetts, USA: Morgan Kaufmann, 2013.
2. *Foster I.* Globus Toolkit Version 4: Software for Service-Oriented Systems // IFIP Int. Conf. on Network and Parallel Computing. Springer, 2006. P. 2–13.
3. *Streit A., Bala P., Beck-Ratzka A. et al.* UNICORE 6 – Recent and Future Advancements. Forschungszentrum Jülich Zentralbibliothek, 2010. <http://hdl.handle.net/2128/3695>.
4. *Binszok H., Koprowski A., Swarczewskaja I.* Оpa: Up and Running. O’Reilly Media, Inc., 2013.
5. *Астафьев А.С., Афанасьев А.П., Лазарев И.В. и др.* Научная сервис-ориентированная среда на основе технологий Web и распределенных вычислений // Научн. сервис в сети Интернет: масштабируемость, параллельность, эффективность. Тр. Всерос. суперкомпьютер. конф. М.: Изд-во МГУ, 2009. С. 463–467.
6. *Бухановский А.В., Ковальчук С.В., Марьин С.В.* Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. вузов. Приборостроение. 2009. Т. 52. № 10. С. 5–24.

7. *Herrera J., Huedo E., Montero R., et al.* Porting of Scientific Applications to Grid Computing on GridWay // Scientific Programming. 2005. V. 13. No. 4. P. 317–331.
8. *Коваленко В.Н.* Комплексное программное обеспечение грида вычислительного типа. Препринт. М. ИПМ им. М.В. Келдыша РАН, 2007.
9. *Durfee E.H.* Distributed Problem Solving and Planning // Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence / Ed. G. Weiss. MIT Press, 1999. P. 121–164.
10. Market-Oriented Grid and Utility Computing / Eds. R. Buyya, K. Bubendorfer. Wiley & Sons, 2010.
11. *Топорков В.В.* Модели распределенных вычислений. М.: Физматлит, 2004.
12. *Бычков И.В., Опарин Г.А., Феохтистов А.Г. и др.* Распределение заданий в интегрированной кластерной системе на основе их классификации // Вычислительные технологии. 2013. Т. 18. № 2. С. 25–32.
13. *Богданова В.Г., Бычков И.В., Корсуков А.С. и др.* Мультиагентный подход к управлению распределенными вычислениями в кластерной Grid-системе // Изв. РАН. Теория и системы управления. 2014. № 5. С. 95–105.
14. *Опарин Г.А., Новопашин А.П., Сидоров И.А. и др.* Система метамониторинга распределительных вычислительных сред // Программные продукты и системы. 2014. № 2. С. 45–48.
15. *Боев В.Д.* Моделирование систем. Инструментальные средства GPSS World. СПб.: БХВ-Петербург, 2004.
16. *Шоломов Л.А.* Логические методы исследования дискретных моделей выбора. М.: Наука, 1989.
17. *Czajkowski K., Ferguson D.F., Foster I., et al.* The WS-Resource Framework. Version 1.0. <http://www.globus.org/wsrf/specs/ws-wsrf.pdf>.
18. *Бычков И.В., Опарин Г.А., Феохтистов А.Г. и др.* Сервис-ориентированный подход к организации распределенных вычислений с помощью инструментального комплекса DISCENT // Информационные технологии и вычислительные системы. 2014. № 2. С. 7–15.
19. *Kundu P., Das D., Ratha B.* WSDL Specification of Services for Service Oriented Architecture (SOA) Based Implementation of a CRM Process // Int. J. Sci. Engin. Res. 2012. V. 3. No. 10. P. 1–24.
20. *Fielding R.T.* Architectural Styles and the Design of Network-based Software Architectures. Dissertation. Irvine, University of California, 2000.
21. *Walsh A.* UDDI, SOAP and WSDL: The Web Services Specification Reference Book. Pearson Education, 2002.
22. *Henderson R.* Job scheduling under the portable batch system / Job scheduling strategies for parallel processing. Springer, 1995. P. 279–294.
23. *Litzkow M., Livny M., Mutka M.* Condor – A Hunter of Idle Workstations // 8th Int. Conf. of Distributed Computing Systems (ICDCS). IEEE CS Press, Los Alamitos, CA, USA, 1988. P. 104–111.
24. *Wilkinson J.X., Reinsch C.* Handbook for Automatic Computation. V. II: Linear Algebra. N.Y.: Shpringer-Verlag, 1971.
25. *Бычков И.В., Опарин Г.А., Феохтистов А.Г. и др.* Сервис-ориентированный подход к мультиагентному управлению распределенными вычислениями // XII Всеросс. совещ. по проблемам управления. Тр. М.: Изд-во ИПУ РАН, 2014. С. 8942–8953.

26. *Oparin G.A., Feoktistov A.G., Bogdanova V.G., et al.* The Solution of Boolean Equations of High Dimensionality in the Distributed Computing Environment // Distributed Computing and Grid-Technologies in Science and Education. Book of the Abstr. of the Int. Conf. Dubna, Russia, June 29–July 2, 2004. JINR, 2004. D11-2004-82. P. 65.
27. *Опарин Г.А., Богданова В.Г.* РЕБУС – интеллектуальный решатель комбинаторных задач в булевых ограничениях // Вестн. НГУ. Серия: Информац. технологии. 2008. Т. 6. Вып. 1. С. 60–68.
28. *Опарин Г.А., Богданова В.Г.* Инструментальные средства автоматизации параллельного решения булевых уравнений на многоядерных процессорах // Программные продукты и системы. 2012. № 1. С. 10–14.

Статья представлена к публикации членом редколлегии С.Н. Васильевым.

Поступила в редакцию 19.11.2014