

УДК 004.942

О. Е. ФЕДОРОВИЧ, Е. М. ПАХНИНА*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина*

АГЕНТНАЯ ИМИТАЦИОННАЯ МОДЕЛЬ УПРАВЛЕНИЯ РЕСУРСАМИ В РАСПРЕДЕЛЕННЫХ ДИНАМИЧЕСКИХ СРЕДАХ

Рассмотрены основные особенности анализа процессов управления ресурсами в распределенных динамических средах. Предложена мультиагентная имитационная модель управления ресурсами в сложных динамических средах, с возможностями экономической самоорганизации и коллективной адаптации элементов модели с собственными интересами и процессами принятия решений, которая позволяет осуществлять децентрализованное планирование с гибким управлением ресурсами и нагрузкой в условиях неопределенности и динамического окружения. Описаны основные особенности поведения и взаимодействия агентов при моделировании процессов управления ресурсами.

Ключевые слова: агентная имитационная модель, распределенные системы, управление ресурсами, планирование, Grid.

Введение

Сегодня все чаще при организации вычислений применяется разделение работ, данных и процессорных мощностей, а также иные режимы взаимодействия, предусматривающие использование распределенных ресурсов, особенно в рамках grid- и cloud computing.

Ключевым вопросом при этом является управление ресурсами, нацеленное на эффективное планирование выполнения заданий различной трудоемкости на имеющихся ресурсах гетерогенной среды и оптимизацию отношений между владельцами ресурсов и пользователями в соответствии с определенными стратегиями.

К сложностям в управлении ресурсами в распределенных динамических средах приводит: большое количество участников, требующее многоуровневого планирования ресурсов; конфликт интересов пользователей, владельцев ресурсов и администраторов системы; общая нехватка информации о системе, а также несогласованность и непредсказуемость вычислительных мощностей доступных ресурсов; наличие собственных политик у большинства ресурсов, которые также должны быть учтены и др.

Анализ последних исследований и публикаций

Эффективность эксплуатации распределенных систем напрямую зависит от методов планирования, включающих последовательное нахождение связанных расписаний выполнения заданий. В общем случае сложность задачи оптимального отображения заданий на ресурсы является нелинейной функцией от размерности системы, то есть от количества ее узлов и относится к классу NP-полных задач [1].

Для решения задач планирования возможно

применение аналитических методов, таких как методы сетевого планирования и управления; линейное и динамическое программирование и др. Следует отметить, что при этом имеет место высокая сложность методов точного решения задач планирования, неоправданно большие затраты времени, кроме того, такие методы не позволяют получить гарантированные оценки времени решения.

Поэтому на практике используют различные эвристические и вероятностные алгоритмы планирования, предоставляющие решение, близкое к оптимальному. Здесь можно выделить несколько классов алгоритмов планирования по используемым стратегиям: списочные (простые эвристики, такие как FCFS, EDF и др.), алгоритм обратного заполнения (Backfill), переборные эвристики, на основе моделей (генетические алгоритмы, муравьиных алгоритмов оптимизации, метод роящихся частиц, алгоритм имитации отжига, табуированный поиск).

Наиболее перспективным представляется экономический подход, который основан на формализации процесса распределения ресурсов с помощью модели рынка: на рынке ресурсов Grid в качестве продавцов выступают владельцы ресурсов, а в качестве покупателей – пользователи [2].

Традиционно план представляется в виде множества временных слотов [3], где каждый слот соответствует началу и концу некоторого кластерного задания. Планирование заключается в подборе слотов, удовлетворяющих ресурсному запросу задания с учётом их стоимости. В рамках экономического подхода рассматриваются: модели рынка товаров (реализовано в Nimrod/G, Mungi), модели ценообразования (Nimrod/G), модели переговоров (CATNETS, OCEAN), теория игр, модели аукционов (SPAWN, Bellagio, G-Commerce) и другие.

В работе [4] предлагается модель системы планирования заданий на основе условно-стоимостного исчисления и способ дифференциации заданий, основанный на пользовательской оценке значимости заданий, характеризуемых случайным временем выполнения и невозрастающей функцией актуальности (полезности).

Результаты анализа показывают, что экономический подход позволяет успешно управлять децентрализованными и гетерогенными ресурсами так, как это происходит в реальной экономике. Экономические системы управления ресурсами Grid динамически определяют наилучшие ресурсы, учитывая их цену и производительность, и распределяют задачи на этих ресурсах так, чтобы удовлетворить потребности пользователей.

Принцип «виртуальной» торговли за ресурсы используется и в агентном подходе [5], в результате каждому агенту-заказу выделяется часть рабочего времени агента-ресурса. Такой способ позволяет осуществлять планирование в реальном времени, т.е. учитывать внезапные отказы или возникновения новых ресурсов и заказов добавлением, удалением или изменением параметров агентов модели.

Агентный подход является наиболее перспективным направлением повышения эффективности управления ресурсами в Grid с учетом его динамики и неопределенности. Тем более, если речь идет о координировании коллекций ресурсов – это задача высшего уровня, и большинство процессов (мониторинг, диагностика, планирование) требует интеллектуальных, автономных, социальных способностей. Использование агентных систем для планирования задач в Grid дает возможность решить две основные проблемы – масштабируемости и адаптивности [6].

В работе [7] подчеркивается, что агенты дают возможность получить автономное и адаптивное поведение компонентов Grid и приводится пример с задачей выявления аномальной деятельности пользователей Grid-систем. Описание поведения пользователей в Grid на основе агентного подхода также приводится в [8]. В работе [9] рассматривается агентный подход к планированию в Grid, где используются два типа агентов супер-агент и агент задачи. Супер-агент определяет тип задачи и направляет ее агенту задачи, который координирует ресурсы и возвращает результат ее решения. Вопросы улучшения производительности Grid приложений и использования ресурсов с помощью агентного подхода рассматриваются в [10]. Агенты взаимодействуют друг с другом, чтобы сбалансировать нагрузку в распределенной сети, используя рекламный сервис и механизмы обнаружения. В работе [11] используется агентная модель, где в качестве аген-

тов рассматриваются пользовательские приложения, ресурсы, а также сервис обнаружения ресурсов. Возможности ресурсов и требования задач описываются с помощью объектной модели данных, что позволяет определять новые типы устройств или новые функции в существующих устройствах. В работе [12] описываются алгоритмы глобального и локального планирования на основе мультиагентного подхода. Для этого в каждом узле размещаются автономные агенты, которые взаимодействуют с агентами других узлов для балансировки нагрузки и выполняют задачи с целью достижения глобального решения. Поведение агентов в предложенной модели описывается в виде простых сценариев и условий, однако в работе подчеркивается, что развитием подхода может быть использование сложных алгоритмов обучения и других методов искусственного интеллекта для прогнозирования состояния динамики Grid. В работе [13] предложена агентно-ориентированная система управления ресурсами в Grid (ARMS), которая использует методы прогнозирования эффективности инструментария PACE (Performance Analysis and Characterise Environment) для обеспечения однородных агентов, организованных в иерархию, числовыми данными о производительности сложных приложений, работающих на локальных ресурсах Grid. В работе [14] в качестве агентов управления ресурсами в Grid рассматриваются их два типа – агент представления автономии и агент мониторинга состояния узла. Оба этих типа агентов организуются в иерархическое двухуровневое дерево для более эффективного решения задачи планирования и балансировки.

Постановка задачи исследования

Проведенный анализ показал, что мультиагентные системы с учетом предоставляемых ими преимуществ – автономности, децентрализованности, индивидуального поведения с возможностью обучения и адаптации для составляющих их агентов – автоматизируют полный цикл управления ресурсами в реальном времени, включая согласование интересов, скоординированное взаимодействие, динамическое планирование и адаптивное перепланирование заказов/ресурсов.

В рамках данного исследования предлагается мультиагентная имитационная модель управления ресурсами в сложных динамических средах, с возможностями экономической самоорганизации и коллективной адаптации элементов модели с собственными интересами и процессами принятия решений, которая позволяет осуществлять децентрализованное планирование с гибким управлением ресурсами и нагрузкой в условиях неопределенности и динамического окружения.

Агентная имитационная модель управления ресурсами

Множество ресурсов и заказов/заданий моделируются в виде агентов (рис. 1).

Таким образом, в составе мультиагентной модели управления ресурсами каждый агент *TaskAgent* представляет собой конкретный заказ/задачу и характеризуется своими целями, приоритетом, сроками, стоимостью. Целью для каждого агента *TaskAgent* является поиск контрактов с ресурсами, которые могут выполнить необходимые работы с минимальными затратами.

Ресурсы являются элементом с индивидуальным поведением, поэтому моделируются агентом *ResourceAgent*. Его цель состоит в том, чтобы увеличить уровень загрузки ресурсов с точки зрения экономического подхода и доходов.

Рассмотрим основные составляющие модели.

Множество ресурсов $R = \{R_1, R_2, \dots, R_M\}$.

Формализовано представить ресурс R_m ($m = \overline{1..M}$) можно в виде совокупности следующих компонентов:

$$R_m = \langle P_m^R, H_m^R, E_m^R, PV_m^R, S_m^R \rangle,$$

где P_m^R – вектор начальных характеристик m -го ресурса; H_m^R – множество компетенций ресурса; E_m^R – вектор экономических характеристик ресурса; PV_m^R – вектор индивидуальных предпочтений ресурса; S_m^R – вектор, описывающий состояние ресурса.

Отличительной особенностью предложенной модели является введение в составе характеристик

ресурсов множества компетенций, которыми располагает тот или иной ресурс, а в составе агенто-заданий набора компетенций, которые необходимы для их выполнения.

В условиях интеграции кластерных/облачных технологий и сервисно-ориентированной архитектуры SOA можно провести аналогию между компетенцией и сервисом.

Множество компетенций ресурса H_m^R состоит из набора пар для каждой i -й компетенции (h_{mi}^R, e_{mi}^R) , где h_{mi}^R – характеризует доступный уровень по каждой компетенции, e_{mi}^R , представляет собой эффективность (способность) ресурса при выполнении компетенции так, что $0 \leq e_{mi}^R \leq 1$.

Для вычислительных ресурсов, которые составляют большинство, в компетенции включаются вычислительная скорость ресурса c_m (задается в млн. инструкций в секунду – MIPS) и базовые требования со стороны заданий – статические параметры, которые описывают конфигурацию узла – тип операционной системы $type_m^{OS}$, версия операционной системы ver_m^{OS} , архитектура A_m , число ядер C_m^A , частота F_m^A , объем кэш памяти M_m^C , объем оперативной памяти M_m^R , объем дискового накопителя M_m^D , пропускная способность сети на получение F_m^{NR} и передачу данных F_m^{NS} .

Кроме того компетенциями для Grid-ресурсов являются: поддержка набора специализированных библиотек, прикладных пакетов и всевозможных вспомогательных программных средств, например,

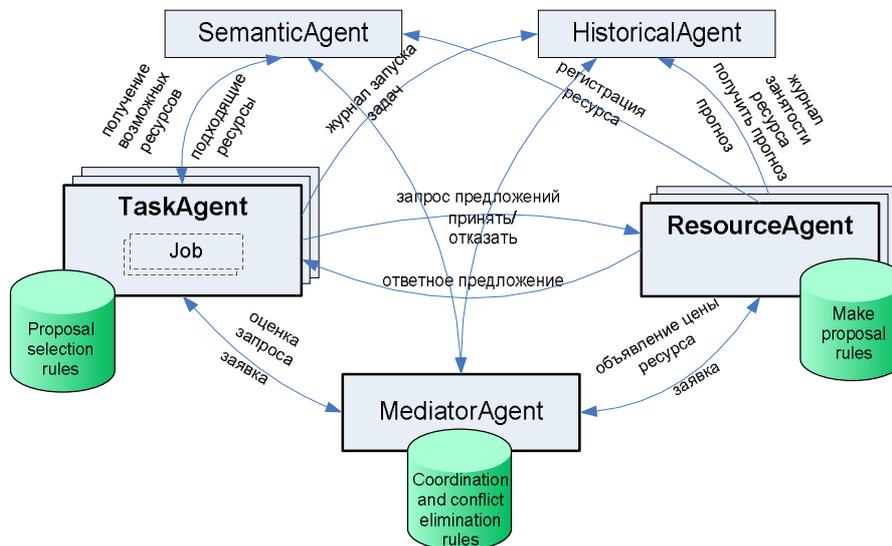


Рис. 1. Мультиагентная модель управления ресурсами

поддержка технологии параллельных вычислений (Message Passing Interface, MPI), производительность вычислений чисел с плавающей точкой, доступность, надежность и другие.

Вектор экономических характеристик ресурса включает:

- нижнюю цену s_m , которая выражается в условных денежных Grid-единицах G\$ за MIPS (G\$/MIPS);

- максимальную цену ms_m , которая назначаетса самим ресурсом и может быть изменена в ходе моделирования;

- заданную себестоимость в расчете на единицу времени cr_m , которая не изменяется, причем, обычно $s_m > cr_m$;

- прибыль ресурса

$$SV(R_m) = \sum_{i=1} (s_m(O_i) - cr_m(O_i)) \text{ рассматривается}$$

как сумма прибылей от выполнения каждого задания, где $s_m(O_i)$ - стоимость ресурса для задания O_i , а $cr_m(O_i)$ - себестоимость выполнения этого задания.

Вектор индивидуальных предпочтений $PV(R_m)$ - это предпочтения ресурса по времени выполнения заказа (краткосрочные и долгосрочные задания), сложность и размерность заданий, приоритеты удовлетворения требований и т.д.

Вектор, описывающий состояние ресурса S_m^R включает динамические параметры, которые характеризуют его состояние в произвольный момент времени: загруженность вычислительных ядер процессоров узла L_m (означает, что требуется L_m секунд для выполнения заданий, которые уже приняты ресурсом или другими словами - это время начала выполнения новой принятой работы), объем доступной оперативной памяти m_m^R , объем доступной дисковой памяти m_m^D .

Целью каждого агента ресурса является максимизация прибыли.

Для этого они пытаются продать свои ресурсы по более высоким ценам и конкурируют друг с другом за получение большего числа работ.

Множество заданий $O = \{O_1, O_2, \dots, O_N\}$. Каждое задание O_n ($n = \overline{1..N}$) можно представить в виде совокупности следующих компонентов:

$$O_n = \langle P_n^O, Y_n^O, H_n^O, PV_n^O, S_n^O \rangle,$$

где P_n^O - вектор начальных характеристик n -го задания; Y_n^O - комплекс работ; H_n^O - множество

компетенций, необходимых для выполнения задания; E_n^O - вектор экономических характеристик задания; PV_n^O - вектор индивидуальных предпочтений задания; S_n^O - состояние выполнения задания.

Вектор начальных характеристик задания включает вложенность (означает, что выполнение задания предполагает комплекс работ), имя пользователя или владельца, имена файлов исполняемой программой и данных.

Комплекс работ

$$Y_n^O = \{y_{nj}\},$$

где $j = \overline{1..p_n}$, p_n - количество работ n -го задания. Каждая работа задания характеризуется

$$y_{nj} = (l_{nj}, s_{nj}, t_{nj}^k),$$

где l_{nj} - размерность j -й работы (требуемый объем ресурса, определяется для Grid-систем в млн. инструкций - MI), s_{nj} - объем финансовых ресурсов (бюджет), выделенный для выполнения работы, т.е. максимальная стоимость, которую агент готов заплатить за выполнение работы, t_{nj}^k - желаемый срок завершения работы.

Максимально сколько может стоить MI определяется

$$\varphi = \frac{s_{nj}}{l_{nj}}.$$

Подмножество $H_n^O = (h_{n1}^O, h_{n2}^O, \dots, h_{nk}^O)$ - набор компетенций, которые необходимы для выполнения задания, h_{ni}^O - характеризует требуемый уровень по каждой компетенции.

Задание характеризуется базовым набором компетенций, релевантным уже объявленным выше компетенциям ресурсов, а именно типы ресурсов, архитектура аппаратного обеспечения, количество необходимых вычислительных ядер (число создаваемых процессов), объемы оперативной и дисковой памяти, набор необходимых для исполнения библиотек и прикладных пакетов и др.

Вектор экономических характеристик задания включает:

- объем средств r_n , которые задание может потратить на выполнение работ $r_n = \sum_j s_{nj}$;

- прибыль задания

$$SV(O_n) = \sum_{i=1} (r_n(R_i) - s_i)$$

определяется как сумма денежных средств от использования ресурсов для выполнения задания, т.е.

разница между объемом средств $r_n(R_i)$, имеющихся на покупку i -го ресурса и его текущей ценой s_i , принятой заданием.

Вектор индивидуальных предпочтений $PV(O_m)$ - это предпочтения задания относительно желательного и предельно допустимого срока завершения, объема, штрафа за невыполнение задания и т.д.

Целью каждого агента задания является выполнение своих работ в срок и с минимальными затратами.

Особенности взаимодействия агентов модели управления ресурсами

Наибольшую сложность в теоретических исследованиях и практических реализациях современных мультиагентных систем представляют вопросы, связанные с процессами взаимодействия агентов при коллективном решении задач реальной практической сложности и значимости, поскольку каждый агент, решающий конкретную подзадачу, имеет лишь частичное представление об общей задаче и должен постоянно взаимодействовать с другими агентами.

В каждую единицу времени, агенты-задания и агенты-ресурсы подают заявки и запросы аукционисту (*MediatorAgent*) в виде G\$/MIPS. Аукционист ведет список текущих предложений и запросов и контрактует два предложения, когда цена лота превышает или равна самому низкому запросу. Определение значений ставки и запроса для агентов-заданий и агентов-ресурсов можно сделать автономным и на основе их целей. В работе мы рассматриваем два способа принятия решений для определения значений ставок и запросов.

Рассмотрим поведение агента *TaskAgent*.

Агент задания определяет значение заявки в каждую единицу времени на основе двух параметров: среднее оставшееся время для проведения торгов и оставшиеся ресурсы для заявки.

1) *Определение значения заявки, основанное на количестве оставшихся ресурсов на участие в аукционе.*

В этом варианте, в каждую единицу времени, агент задания определяет значение ставки в зависимости от количества оставшихся ресурсов, которые могут предложить цену за их использование. Задание может претендовать на ресурс, если он может выполнять работу в пределах срока и цена резервирования ресурса (G\$/MIPS) меньше или равна максимальному значению, которое агент может заплатить за работу.

Формально работа i задания n может торго-

ваться за ресурс j

$$t_{ni}^k - L_j - \frac{l_{ni}}{c_j} \geq 0,$$

где l_{ni}/c_j - время выполнения работы i задания n на ресурсе j .

Число оставшихся ресурсов уменьшается (из-за принятия новых работ ресурсами на выполнение).

В каждый момент времени, значение ставки для работы i задания n основывается на оставшихся ресурсах и определяется

$$V_{\pi} = \left[\frac{s_{\min}}{\varphi_i} + \left(1 - \frac{s_{\min}}{\varphi_i} \right) \left(1 - \frac{M^t - 1}{M} \right)^{\frac{1}{\alpha}} \right] \cdot \varphi_i$$

где s_{\min} - минимальная цена резервирования среди оставшихся ресурсов; M^t - число оставшихся ресурсов в момент времени t для работы i задания n ; α - полиномиальный коэффициент ($0,01 \leq \alpha \leq 100$).

Когда $\alpha < 1$ агент задания поддерживает низкую стоимость покупки, пока число оставшихся ресурсов не станет близким к нулю. С другой стороны, когда $\alpha > 1$ агент задания начинает со значения ставки близким к φ_i , т.е. максимальным значением заявки за M . Когда число оставшихся ресурсов уменьшается, значение заявки увеличивается.

2) *Определение значения заявки, основанное на среднем оставшемся времени для проведения торгов.*

Здесь в каждый момент времени, агент задания определяет значение ставки на основе среднего оставшегося времени для проведения торгов с ресурсами. Пусть заявка работы i задания n подается в момент времени t . Оставшееся время для агента задания, когда он может сделать заявку ресурсу j , определяется

$$rt_{nij} = \left(t_{ni}^k - L_j - \frac{l_{ni}}{c_j} \right) \cdot x, \text{ где } x = \begin{cases} 1, & \text{если } \varphi_i \geq s_j \\ 0, & \text{если } \varphi_i < s_j \end{cases}$$

Если $rt_{ij} < 0$, то это означает, что ресурс j не может выполнить работу i задания n в желаемый срок.

Среднее оставшееся время для подачи заявки может быть получено следующим образом

$$rt_{ni} = \frac{\sum_j (rt_{nij} \cdot y)}{M},$$

$$\text{где } y = \begin{cases} 1, & \text{если } rt_{nij} > 0 \\ 0, & \text{если } rt_{nij} \leq 0 \end{cases}$$

На момент подачи работы, среднее оставшееся время для подачи заявки имеет максимальное значение rt_{ni}^{\max} .

Значение заявки на основе среднего оставшегося времени для торгов может быть получено следующим образом

$$V_{rt} = \left[\frac{s_{\min}}{\varphi_i} + \left(1 - \frac{s_{\min}}{\varphi_i} \right) \left(1 - \frac{r_{ni}}{rt_{ni}^{\max}} \right)^{\beta} \right] \cdot \varphi_i,$$

где $0,01 \leq \beta \leq 100$. Здесь параметр β аналогичен α и используется для управления степенью выпуклости кривой.

После определения значений заявки для каждого из ограничений, упомянутых выше, агент задания объединяет их для расчета окончательной суммы заявки

$$V_b = \lambda \cdot V_{rt} + (1 - \lambda) \cdot V_{rt},$$

где $0 \leq \lambda \leq 1$ – используется для регулирования эффективности параметров, используемых в этом уравнении. Если $\lambda = 1$, то это означает, что только ограничение на оставшиеся ресурсы рассматривается в окончательной стоимости тендерного предложения, а $\lambda = 0$ означает, что учитывается только ограничение на оставшееся время. Значение $\lambda \in (0,1)$ означает, что учитываются оба параметра.

Рассмотрим поведение агента *ResourceAgent*.

Агент ресурса направлен на получение большей прибыли. Для этого он пытается продать свой ресурс по более высокой цене и конкурирует с другими ресурсами для получения новых работ на выполнение.

Будем считать, что в момент появления ресурса в Grid, его загруженность L_m равна нулю и ресурс устанавливает цену на свое резервирование. После закрепления работы он обновляет свою рабочую нагрузку (т.е. по сути время начала новой работы) и устанавливает свою цену по максимальному значению ms_m . Постепенно загруженность ресурса L_m уменьшается и приближается к нулю.

С уменьшением загруженности агент ресурса уменьшает свою стоимость и в случае, когда загруженность равна нулю, он устанавливает цену резервирования s_m . Максимальная цена может быть определена либо владельцем ресурса или агентом через сотрудничество с другими агентами. Агент ресурса определяет свою цену следующим образом

$$V_r = \left[\frac{s_j}{ms_j} + \left(1 - \frac{s_j}{ms_j} \right) \left(\frac{L_j}{L_j^t} \right)^{\frac{1}{\gamma}} \right] \cdot ms_j,$$

где L_j^t – загруженность ресурса j после последнего размещения работы; L_j – текущая загруженность или время начала новой работы. Здесь параметр γ аналогично используется для управления степенью выпуклости кривой.

В каждый момент времени агенты заданий и агенты ресурсов определяют свои заявки и запросы и отправляют их аукционисту (*MediatorAgent*). Аукционист сортирует значения заявок в порядке возрастания, а значения запросов в порядке убывания. Если заявка больше или равна самому низкому запросу, то сделка происходит по следующей цене

$$V_C = \frac{(V_b^{\max} + V_r^{\min})}{2}.$$

Для анализа эффективности мультиагентной системы управления ресурсами подготовлена экспериментальная среда (рис. 2). В качестве среды для разработки модели был выбран программный продукт AnyLogic.

Целью экспериментов являлось:

- определение среднего времени планирования одного задания и построение зависимости среднего времени планирования от количества заданий;
- определение показателей эффективности алгоритмов планирования: среднее время ожидания, время счёта заданий, коэффициент использования ресурсов, успешность выполнения и коэффициент «справедливости» рынка;
- сравнение предложенной модели планирования с другими алгоритмами.

Для примера в данной работе продемонстрированы результаты сравнения с алгоритмом планирования «сначала самый ранний крайний срок» (Earliest Deadline First, EDF), который является централизованной политикой по умолчанию для многих систем планирования.

Если есть несколько ресурсов, способных выполнять работу в пределах отведенного срока, планировщик может использовать ряд политик для отбора ресурса. Например, выбор ресурса, который может выполнить работу ближе всего к предельному сроку (стратегия планирования называется EDF_CTD), или ресурс с минимальным временем завершения (EDF_MCT) или случайный выбор (EDF_RND).

Для эксперимента были выбраны три типа ресурсов, отличающихся вычислительной скоростью ресурса, MIPS (I – 10, II – 20, III – 30).

На рисунке 3 представлені сравнительные результати по коефіцієнту використання ресурсів. Як видно по результатам дослідження запропонований підхід має більше, ніж в порівнюваних алгоритмах планування, значення коефіцієнта використання ресурсів для більшості типів ресурсів.

Заключення

В роботі запропонована агентна імітаційна модель управління ресурсами в складних динамічних середовищах, з можливостями економічної самоорганізації і колективної адаптації елементів моделі з власними інтересами і процесами прийняття рішень, що дозволяє вирішувати задачі

децентрализованного планування з гнучким управлінням ресурсами і навантажкою в умовах неопределенности і динамічного оточення.

Проведені експериментальні дослідження процесів управління ресурсами в Grid. Створено іспытательный моделюющий стенд, где проводились експериментальні дослідження різних стратегій планування по економічним принципам самоорганізації агентів.

Литература

1. Методы и модели планирования ресурсов в GRID-системах [Текст] : монография / В. С. Пономаренко, С.В. Листровой, С.В. Минухин, С.В. Знахур. – ИД «ИНЖЕК», 2008. – 408 с.

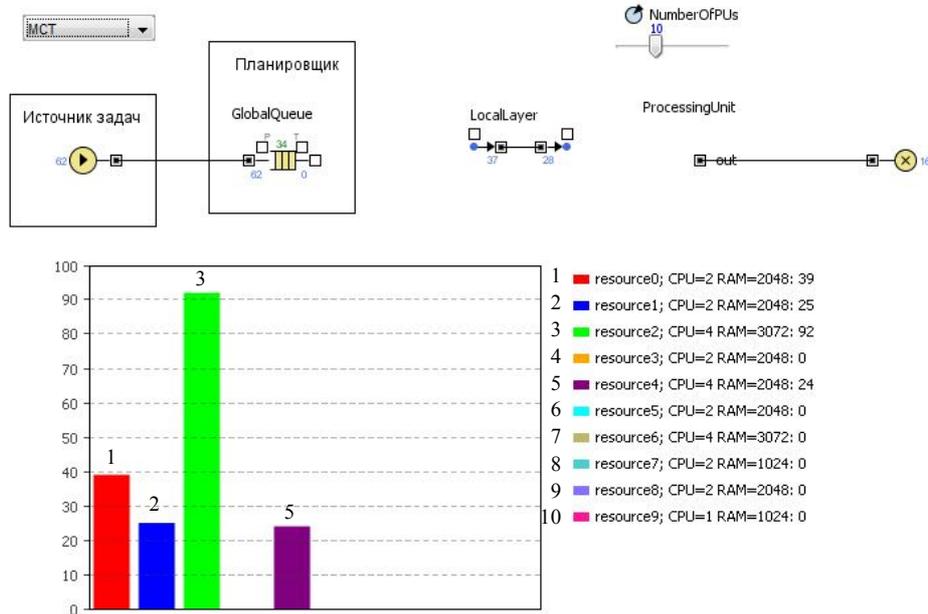


Рис. 2. Імітаційна модель аналізу процесів управління ресурсами в Grid

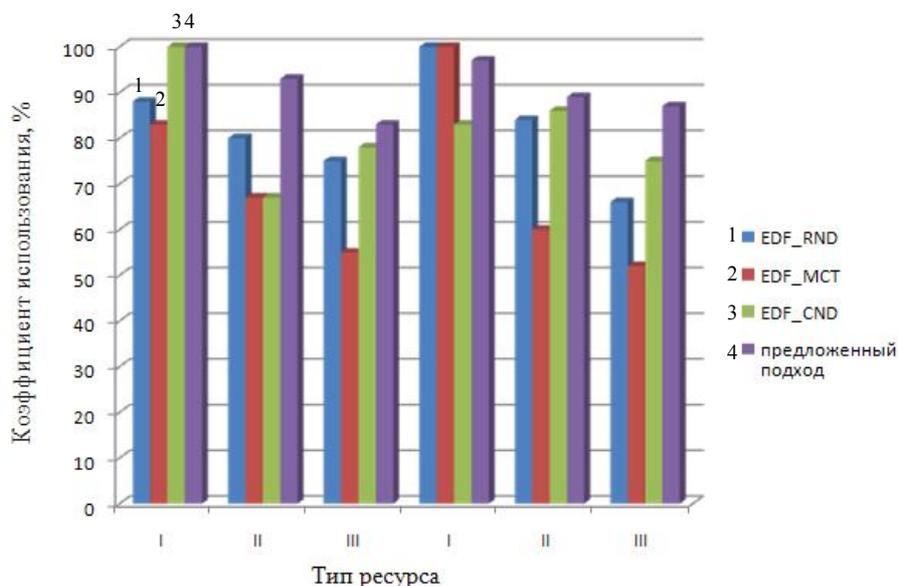


Рис. 3. Результати моделювання по коефіцієнту використання ресурсів

2. Метод опережающего планирования для грид [Электронный ресурс] / В. Н. Коваленко, Е. И. Коваленко, Д. А. Корягин, Э. З. Любимский // Препринт ИПМ. – 2005. – № 112. – Режим доступа: http://www.keldysh.ru/papers/2005/prep112/prep2005_112.html. – 17.03.2014.

3. Economic Models for Resource Management and Scheduling in Grid Computing [Text] / R. Buyya, D. Abramson, J. Giddy, H. Stockinger // The Journal of Concurrency and Computation. – 2002. Vol.14. – P. 1507-1542.

4. Голосов, П. Е. Планирование заданий с временной функцией потери ценности решения в сетевой среде распределенных вычислений [Текст] : автореф. дис. ... канд. техн. наук : 05.13.01 / Голосов Павел Евгеньевич ; Моск. гос. ин-т электроники и математики. – М., 2010. – 19 с.

5. Молев, А. А. Агентные модели динамического планирования и распределения сетевых ресурсов системы Grid [Электронный ресурс] / А. А. Молев, И. В. Зайцев // Режим доступа: <http://www.gpss.ru/immod05/s1/kazakov/index.html>. – 17.03.2014.

6. Гороховський, С. С. Технологія сітки (GRID) і використання агентних платформ для задач планування [Текст] / С. С. Гороховський, В. К. Римарчук // Наукові записки НаУКМА. Комп'ютерні науки. – 2005. – С. 26-32.

7. Куссуль, Н. Н. Grid-системы для задач исследования Земли / Н. Н. Куссуль, А. Ю. Шелестов. – К.: Наукова думка. 2008. – 452 с.

8. Шелестов, А. Ю. Агентный подход к реализации модели поведения пользователей Grid-систем [Текст] / А. Ю. Шелестов, С. В. Скакун, О. М. Куссуль // Информатика, кибернетика та обчислювальна

техніка : зб. наук. праць / Національний технічний університет України «КПІ». – Київ, 2008. – С.8-14.

9. Chandak, A. Agent based Task Scheduling in Grid [Text] / A. Chandak, B. Sahoo, A.K. Turuk // International Journal of Computer Applications and Technology. – 2012. – Vol. 1, Issue 2. – P. 44-48.

10. Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling [Text] / J. Cao, D.P. Spooner, S.A. Jarvis, S. Saini, G.R. Nudd // In Proceedings of 17th IEEE International Parallel & Distributed Processing Symposium. – 2003. – P. 49-60.

11. Resource discovery for dynamic clusters in computational grids [Text] / O. F. Rana, D. Bunford-Jones, D. W. Walker, M. Addis, M. Surridge, K. Hawick // In Proceedings of 10th IEEE Heterogeneous Computing Workshop. – 2001. – P. 759-767.

12. A Multiagent-Based Approach to the Grid-Scheduling Problem [Electronic resource] / J. Rojas, M. Solar, M. Mendoza, R. Monge, V. Parada // CLEI electronic journal. – 2012. – Vol.15 – Mode of access: www.elsevier.nl/locate/entcs. – 17.03.2014.

13. Cao, J. ARMS: an agent-base resource management infrastructure for grid computing [Text] / J. Cao, D. J. Kerbyson, G. R. Nudd // Proc. 1st IEEE International Symposium on Cluster Computing, Scientific Programming, Special Issue on Grid Computing. – 2002. – P. 135-148.

14. Research on Novel Dynamic Resource Management and Job Scheduling in Grid Computing [Text] / L. Fufang, O. Devu, Z. Limin, Z. Xianguang, Z. Zhili // Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences. – 2006. – Vol.1. – P. 709-713.

Поступила в редакцію 17.03.2014, рассмотрена на редколлегии 11.06.2014

Рецензент: д-р техн. наук, проф., зав. каф. информационных технологий проектирования ЛА Е. А. Дружинин, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков.

АГЕНТНА ІМІТАЦІЙНА МОДЕЛЬ УПРАВЛІННЯ РЕСУРСАМИ У РОЗПОДІЛЕНИХ ДИНАМІЧНИХ СЕРЕДОВИЩАХ

О. Є. Федорович, О. М. Пахніна

Розглянуто основні особливості аналізу процесів управління ресурсами в розподілених динамічних середовищах. Запропоновано мультиагентну імітаційну модель управління ресурсами в складних динамічних середовищах, з можливостями економічної самоорганізації та колективної адаптації елементів моделі з власними інтересами і процесами прийняття рішень, яка дозволяє здійснювати децентралізоване планування з гнучким управлінням ресурсами і навантаженням в умовах невизначеності і динамічного оточення. Описано основні особливості поведінки та взаємодії агентів при моделюванні процесів управління ресурсами.

Ключові слова: агентна імітаційна модель, розподілені системи, управління ресурсами, планування, Grid.

AGENT-BASED SIMULATION MODEL OF RESOURCE MANAGEMENT IN DISTRIBUTED DYNAMIC SYSTEMS

O. Ye. Fedorovich, E. M. Pakhnina

The main features of the resource management in distributed dynamic systems are considered. We propose a multi-agent simulation model of resource management in complex dynamic systems with the possibility of economic self-organization and collective adaptation model elements. It allows implement decentralized planning with flexible management of resources and its utilization under conditions of uncertainty and dynamic environment. We describe the detail of behavior and agents interaction in the simulation process of resource management.

Key words: agent-based simulation model, distributed systems, resource management, scheduling, Grid.

Федорович Олег Евгеньевич – д-р техн. наук, проф., зав. каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина, e-mail: lesch@khai.edu.

Пахніна Елена Михайловна – науч. сотр. каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.