

Раздел I. Распределенные вычислительные системы

УДК 004.75

DOI 10.18522/2311-3103-2021-4-6-21

А.И. Каляев

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ САМООРГАНИЗУЮЩИХСЯ ДИСПЕТЧЕРОВ РАСПРЕДЕЛЕННЫХ СИСТЕМ НА БАЗЕ МУЛЬТИАГЕНТНОГО СОЦИОИНСПИРИРОВАННОГО ПОДХОДА

Описаны новые принципы организации, методы и алгоритмы функционирования диспетчера Распределенной системы (РС), позволяющие распределять и перераспределять ресурсы с динамически изменяющимися параметрами между поступающими заданиями с целью минимизации времени их выполнения. Основная проблема, не позволяющая сегодня эффективно оценивать время выполнения заданий в гетерогенной РС, напрямую вытекает из распределенности системы: каждый из её элементов обладает частичной независимостью и может существенно отличаться от других, более того, в процессе работы его возможности могут изменяться, и все это существенно влияет на эффективность распределения задач между узлами РС и на время выполнения заданий. В статье предложен новый подход к организации диспетчера РС, базирующийся на применении теории мультиагентных систем и социоинспирированных (базирующихся на принятых в человеческом обществе) методов: пользователи РС размещают свои задания на специальных узлах – досках объявлений, на каждом узле РС размещается проактивный программный агент, осуществляющий постоянный мониторинг параметров своего узла и поиск на досках объявлений подходящих для решения задач. При этом агенты, участвующие в решении общего задания формируют сообщества, в которых осуществляют планирование процесса решения задания и распределение частей заданий для минимизации времени задержки их решения. В качестве критерия эффективности работы РС было решено принять значение среднего времени задержки выполнения функциональных заданий относительно требуемых моментов времени, соответственно агенты распределяют задания таким образом, чтобы минимизировать значение указанного критерия. Настоящая статья включает введение, формальную постановку задачи диспетчирования ресурсов РС, обзор существующих подходов к организации диспетчера РС, описание предлагаемого мультиагентного решения задачи диспетчирования ресурсов РС с использованием социоинспирированного подхода, алгоритм работы распределенной системы и ее элементов, описание применения социоинспирированного подхода применительно к процессу диспетчирования задач и заключение. К основным преимуществам предложенного подхода можно отнести: возможность использования достоверной и актуальной информации о специализации и текущей производительности ресурсов при диспетчировании; высокая отказоустойчивость, обусловленная отсутствием элементов РС, выход из строя которых приводит к полной потере работоспособности РС; возможность гибкого масштабирования РС (увеличения числа ресурсов), достигаемая за счет децентрализации процесса диспетчирования.

Распределенные системы; социоинспирированный подход; биоинспирированный подход; мультиагентная система; децентрализованная система.

A.I. Kalyaev

**THEORETICAL FOUNDATIONS OF CREATING SELF-ORGANIZING
DISPATCHERS OF DISTRIBUTED SYSTEMS BASED ON A MULTI-AGENT
SOCIO-INSPIRATIONAL APPROACH**

This article describes new principles of organization, methods and algorithms for the functioning of the Distributed System (DS) dispatcher, which allow allocating and reallocating resources with dynamically changing parameters between incoming tasks in order to minimize their execution time. The main problem that does not allow today to effectively estimate the execution time of tasks in a heterogeneous DS directly follows from the distribution of the system: each of its elements has partial independence and may differ significantly from others, moreover, in the process of operation, its capabilities may change, and all this is essential. affects the efficiency of distribution of tasks between DS nodes and the time it takes to complete tasks. The article proposes a new approach to organizing a DS dispatcher, based on the application of the theory of multi-agent systems and socio-inspirational (based on accepted in human society) methods: DS users place their tasks on special nodes – bulletin boards, a proactive software agent is placed on each DS node, which implements constant monitoring of the parameters of your site and search on message boards suitable for solving problems. At the same time, the agents participating in the solution of the common task form communities in which they plan the process of solving the task and the distribution of parts of the tasks to minimize the delay time for their solution. As a criterion for the effectiveness of the DS, it was decided to take the value of the average delay in the execution of functional tasks relative to the required points in time, respectively, the agents distribute tasks in such a way as to minimize the value of the specified criterion. This article includes an introduction, a formal statement of the task of scheduling DS resources, a review of existing approaches to organizing a DS dispatcher, a description of the proposed multi-agent solution to the task of scheduling DS resources using a socio-inspirational approach, an algorithm for the operation of a distributed system and its elements, a description of the application of a socio-inspirational approach in relation to the task scheduling process. and conclusion. The main advantages of the proposed approach include: the ability to use reliable and up-to-date information about the specialization and current performance of resources in dispatching; high fault tolerance due to the absence of DS elements, failure of which leads to a complete loss of DS performance; the possibility of flexible scaling of the DS (increasing the number of resources), achieved by decentralizing the dispatching process.

Distributed systems; socioinspired approach; bioinspired approach; multi-agent system; decentralized system.

Введение. Наблюдаемое в последние годы активное развитие систем передачи данных привело к возрастанию актуальности проблемы создания распределенных систем (РС), т.е. систем, состоящих из множества территориально распределённых подсистем (каждая из которых в общем случае выполняет некоторый набор индивидуальных функций) взаимодействующих друг с другом для выполнения общего функционального задания или некоторого множества заданий [1, 2]. К классу распределенных систем можно также отнести такие сложные технические объекты как атомные и тепловые станции, авиационные и корабельные комплексы, космические станции и т.п. Действительно, все они состоят из множества взаимосвязанных подсистем (ресурсов), каждая из которых имеет свою функциональную специализацию, но при все эти подсистемы используются для выполнения общего задания, выполняемого объектом. Кроме того, в настоящее время крайне актуальной является проблема группового взаимодействия роботов при решении общей задачи [3]. Группа роботов – это та же распределенная система, в которой каждый робот (ресурс) может иметь индивидуальную специализацию, но при этом их действия должны быть направлены на достижение общей групповой цели.

Основной функцией, выполняемой диспетчером РС, является распределение ресурсов, входящих в состав РС, между операциями функциональных заданий, в реальном времени их поступления [4]. В общем виде для каждого задания существует момент времени, к которому должен быть готов результат его выполнения. Основная задача диспетчера – обеспечить распределение ресурсов, позволяющее завершить выполнение всех поступающих задач за требуемое время [5], однако очевидно, что это достижимо лишь в случае наличия достаточных для этого ресурсов. В случае же нехватки ресурсов диспетчер должен делать выбор между заданиями.

Таким образом, актуальной задачей является создание диспетчера РС, позволяющего распределять и перераспределять ресурсы с динамически изменяющимися параметрами между поступающими заданиями с учетом моментов времени, к которым задания должны быть выполнены.

Исходя из сказанного выше, в качестве универсального критерия эффективности работы РС было решено принять значение среднего времени задержки выполнения функциональных заданий относительно требуемых моментов времени.

Целью настоящего исследования является разработка принципов организации, методов и алгоритмов функционирования диспетчера РС, обеспечивающего минимизацию среднего времени задержки выполнения функциональных заданий относительно требуемых моментов времени.

Основная проблема, не позволяющая сегодня эффективно оценивать время выполнения заданий в РС, напрямую вытекает из распределенности системы: каждый из её элементов обладает частичной независимостью и может существенно отличаться от других, более того, в процессе работы его возможности могут изменяться, и все это существенно влияет на эффективность распределения задач между узлами РС и на время выполнения заданий. При этом учет всех этих параметров и организация взаимодействия между узлами ложится на плечи диспетчера, а количество узлов в системе может исчисляться десятками тысяч и существенно изменяться со временем. История человечества показывает возможность взаимодействия сотен тысяч человек при выполнении общих заданий, главной особенностью, позволяющей совмещать работу такого числа исполнителей, является активность элементов: каждый человек в общем виде заинтересован в выполнении своей работы. Поэтому для достижения поставленной цели решено было в качестве основы для создаваемого диспетчера применить мультиагентный подход для организации прямого взаимодействия между элементами РС. При этом предполагается, что каждый ресурс, входящий в состав РС, должен обладать неким программным агентом, представляющим «его интересы» при реализации процедуры диспетчирования [5–10]. Агенты, представляющие различные ресурсы РС должны быть объединены с помощью некоторого информационного канала связи, посредством которого они могут взаимодействовать и «договариваться» друг с другом о выполнении тех или иных операций поступающих функциональных заданий на том или ином ресурсе РС.

Создание такого мультиагентного диспетчера требует разработку новой архитектуры диспетчера РС, а также новых методов и алгоритмов взаимодействия элементов РС.

Базируясь на приведенном выше можно дать формальную постановку задачи диспетчирования ресурсов РС.

Формальная постановка задачи диспетчирования ресурсов РС. Формально под распределенной системой (РС) в дальнейшем будем понимать систему R , включающую в свой состав N различных ресурсов R_1, R_2, \dots, R_N , каждый из которых может выполнять некоторое множество операций $O_i = \langle O_1^i, O_2^i, \dots, O_B^i \rangle$ ($i = 1, 2, \dots, N$), а множество операций O , выполняемых всей РС в целом опреде-

ляется как $O = \bigcup_{i=1}^N O_i$. При этом, будем считать, что операция O_j имеет трудоемкость $v(O_j)$, которая определяется некоторым числом элементарных действий, которые необходимо осуществить для ее выполнения, а ресурс R_i при выполнении операции O_j имеет производительность $D_i(O_j)$, определяемую количеством элементарных действий, выполняемым данным ресурсом в единицу времени. Под операцией O_j можно понимать, например, некоторую вычислительную операцию, если речь идет распределенной вычислительной системе, либо производственную операцию, если речь идет, например, о многостаночной производственной системе. Зная трудоемкость $v(O_j)$ операции O_j и производительность $D_i(O_j)$ ресурса R_i при ее выполнении, можно рассчитать время, затрачиваемое ресурсом R_i на выполнение операции O_j как

$$t_i(O_j) = \frac{v(O_j)}{D_i(O_j)}.$$

В зависимости от конкретных условий организации различные распределенные системы можно разбить на 4 класса.

1. К первому классу относятся РС, у которых все ресурсы R_1, R_2, \dots, R_N выполняют одинаковые множества операций, т.е. $O_i = O_j = O$, ($i = 1, 2, \dots, N, j = 1, 2, \dots, N$), причем производительность всех ресурсов R_i ($i = 1, 2, \dots, N$) при выполнении одной и той же операции $O_s \in O$ также одинакова, т.е. $D_i(O_s) = D_j(O_s)$. РС такого класса будем называть гомогенными распределенными системами первого типа.

2. Ко второму классу относятся РС, ресурсы R_1, R_2, \dots, R_N которых выполняют одинаковые множества операций, т.е. $O_i = O_j = O$ ($i = 1, 2, \dots, N, j = 1, 2, \dots, N$), но при этом производительность различных ресурсов $R_i \in R$ и $R_j \in R$ при выполнении одной и той же операции $O_s \in O$ различна, т.е. $D_i(O_s) \neq D_j(O_s)$. Системы данного класса будем называть гомогенными распределенными системами второго типа.

3. К третьему классу будем относить РС, у которых множества операций, выполняемых различными ресурсами, различны (т.е. каждый ресурс имеет свою функциональную специализацию) $O_i \neq O_j$ ($i = 1, 2, \dots, N, j = 1, 2, \dots, i-1, i+1, \dots, N$), хотя они могут и пересекаться, т.е. $O_i \cap O_j \neq \emptyset$. При этом производительность различных ресурсов $R_i \in R$ и $R_j \in R$ при выполнении одной и той же операции $O_s \in O$ одинаковая, т.е. $D_i(O_s) = D_j(O_s)$. Системы данного класса будем называть гетерогенными распределенными системами первого типа.

4. Наконец, к четвертому классу относятся РС, все ресурсы которых R_1, R_2, \dots, R_N выполняют различные множества операций, т.е. $O_i \neq O_j$ и $O_i \cap O_j \neq \emptyset$, причем производительность различных ресурсов $R_i \in R$ и $R_j \in R$ при выполнении одной и той же операции $O_s \in O_i$ также различна, т.е. $D_i(O_s) \neq D_j(O_s)$. РС данного класса будем называть гетерогенными распределенными системами второго типа.

Будем считать, что РС предназначена для выполнения некоторого множества функциональных заданий $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_L \rangle$, которые могут поступать от различных Заказчиков в произвольные моменты времени, причем в РС может одновременно выполняться сразу несколько различных заданий из множества $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_L \rangle$. При этом каждое отдельное задание $Z_l \in \mathbf{Z}$ представляется в виде некоторого ациклического графа операций $G_l(Q_l, X_l)$ (рис. 2), вершине $q_j \in Q_l$ которого приписана некоторая операция O_j , принадлежащая множеству $O = \bigcup_{i=1}^N O_i$, а дуга $x(q_j, q_{j+1})$ устанавливает взаимосвязь между операциями, приписанным вершинам q_j и q_{j+1} , т.е. определяет, что результат операции O_j , приписанной вершине q_j , необходим для выполнения следующей операции O_{j+1} , приписанной вершине q_{j+1} .

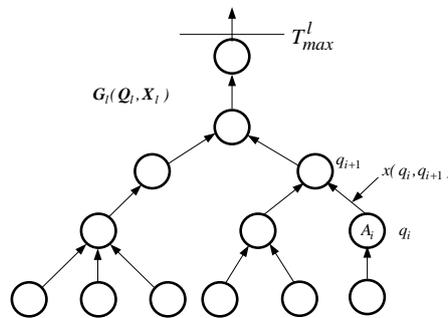


Рис. 2. Граф задания Z_l

Будем считать, что Заказчик задания $Z_l \in \mathbf{Z}$ задает момент времени T_{max}^l (см. рис. 2), к которому он желает получить результат выполнения своего задания, что момент времени T_{max}^l должен устанавливаться таким образом, чтобы выполнялось условие $T_{max}^l \geq T_{\Pi}^l + t_{min}^l$, где T_{Π}^l момент времени поступления задания в РС, t_{min}^l - минимально возможный период времени выполнения задания Z_l .

При этом значение t_{min}^l будет определяться временем выполнения наиболее трудоемкой ветви графа задания $G_l(Q_l, X_l)$, т.е.

$$t_{min}^l = \sum_{j=1}^k t(O_j) = \sum_{j=1}^k \frac{v(O_j)}{D_i(O_j)} \geq \sum_{j=1}^k \frac{v(O_j)}{D_{max}(O_j)}$$

где k – число операций в наиболее трудоемкой ветви графа-задания Z_l ;

$t(O_j)$ – время выполнения j -ой операции наиболее трудоемкой ветви графа

$G_l(Q_l, X_l)$;

$v(O_j)$ – трудоемкость операции O_j ;

$D_i(O_j)$ – производительность ресурса R_i выполняющего операцию O_j ;

$D_{max}(O_j)$ – максимальная производительность среди всех ресурсов множества

\mathbf{R} при выполнении операции O_j .

Очевидно, что реальный момент времени T_p^l выполнения задания $Z_l \in \mathbf{Z}$ может отличаться от желаемого T_{max}^l , причем если $T_p^l < T_{max}^l$ (т.е. задание выполняется раньше установленного Заказчиком времени), то это допустимо, а если $T_p^l > T_{max}^l$ (т.е. задание выполняется позже установленного заказчиком момента времени), то это нежелательно. Поэтому распределение операции задания Z_l между ресурсами R_1, R_2, \dots, R_N системы необходимо осуществлять таким образом, чтобы величина $\Delta T^l = T_p^l - T_{max}^l$ (при условии, что $T_p^l \geq T_{max}^l$) была минимальной. Если же на РС поступает поток задания $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_L \rangle$, то цель работы диспетчера должна заключаться в таком распределении операций, поступающих в произвольные моменты времени заданий, чтобы минимизировать среднее время задержки выполнения всех заданий потока, т.е. обеспечить минимум величины.

$$\Delta T = \frac{\sum_{l=1}^L \Delta T^l}{L} = \frac{\sum_{l=1}^L T_p^l - T_{max}^l}{L};$$

где L – число заданий, выполненных РС.

T_p^l – момент времени поступления задания Z_l .

T_{max}^l – требуемый (установленный Заказчиком) момент времени выполнения задания Z_l ;

T_p^l – реальный момент времени завершения выполнения задания Z_l на РС.

Исходя из приведенных выше соображений задачу, решаемую диспетчером РС, можно сформулировать в следующем виде: необходимо обеспечить такое распределение операций $O_j \subseteq \mathbf{O}$ заданий $Z_l \in \mathbf{Z}$, поступающих в произвольные моменты времени и описываемых графом $G_l(Q_l, X_l)$, между ресурсами множества $\mathbf{R} = \langle R_1, R_2, \dots, R_N \rangle$, которое бы минимизировало среднее время ΔT задержки выполнения всех заданий потока $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_L \rangle$.

При этом необходимо обратить внимание на следующее важное обстоятельство. Период времени t^l , необходимый для выполнения очередного поступающего на РС функционального задания $Z_l \in \mathbf{Z}$, будет складываться из двух составляющих

$$t^l = t_p^l + t_B^l,$$

где t_p^l – период времени, затрачиваемый диспетчером на распределение операций задания Z_l между ресурсами РС;

t_B^l – период времени, затрачиваемый ресурсами РС, между которыми распределены операции задания Z_l , непосредственно их выполнение.

Если время t_p^l распределения будет велико, то может оказаться, что к моменту $T_{п}^{l+1}$ поступления следующего функционального задания $Z_{l+1} \in \mathbf{Z}$ диспетчер еще не успеет решить задачу распределения операций предыдущего задания Z_l . В этом случае распределение, полученное диспетчером РС для задания Z_l становится уже неактуальным и процесс распределения нужно будет начинать заново уже с учетом сразу двух поступивших заданий Z_l и Z_{l+1} , что, очевидно, еще больше увеличит время решения задачи распределения. В результате, если время решения задачи распределения операций поступающих заданий будет большим, а период поступления заданий будет относительно малым, то выполнение потока заданий $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_M \rangle$ на РС становится практически невозможным. Отсюда следует вывод: для того, чтобы РС обеспечивала выполнение потока заданий в реальном времени их поступления, диспетчер должен успевать решать задачу распределения операций очередного поступающего задания Z_l до момента времени $T_{п}^{l+1}$ поступления следующего задания Z_{l+1} (рис. 3).

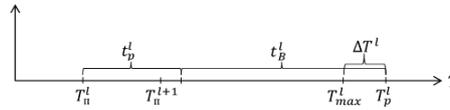


Рис. 3. Поступление очередного задания Z_{l+1} до момента завершения процедуры распределения операций предыдущего задания Z_l

Кроме того, очевидно, что от времени распределения t_p^l , зависит и время задержки ΔT^l выполнения задания Z_l , поскольку (см. рис. 3)

$$\Delta T^l = T_p^l - T_{max}^l = T_n^l + t_p^l + t_B^l - T_{max}^l,$$

где T_{max}^l – устанавливаемое Заказчиком (требуемое) время выполнения задания Z^l .

Т.е. чем больше время распределения t_p^l , тем вероятнее, что время задержки ΔT^l выполнения задания Z_l увеличится. Поэтому, для того чтобы минимизировать среднее время задержки ΔT выполнения заданий потока $\mathbf{Z} = \langle Z_1, Z_2, \dots, Z_l \rangle$, необходимо стараться минимизировать время решения задачи распределения операций очередного поступающего задания Z_l между ресурсами РС. Правда, попытка сокращения времени распределения может иметь и обратную сторону: чем меньше времени будет тратиться на решение задачи распределения операций задания Z_l , тем вероятнее, что получаемое при этом распределение будет менее оптимальным, что в свою очередь может приводить к увеличению времени выполнения задания t_B^l и, как следствие, к увеличению времени задержки ΔT^l . Поэтому необходимо обеспечить некоторый компромисс, т.е. диспетчер должен решать задачу распределения операций очередного задания Z_l за минимально возможное время при сохранении высокого уровня качества самого распределения [11–13].

Исходя из приведенных выше соображений рассмотрим более подробно различные подходы к построению диспетчера распределенной системы, включающей в свой состав N ресурсов R_1, R_2, \dots, R_N .

Существующие подходы к организации диспетчера РС. В простейшем случае диспетчер РС можно реализовывать по классической централизованной схеме. При этом каждый ресурс R_i ($i = 1, 2, \dots, N$), входящий в состав РС, должен обладать своим локальным устройством управления $УУ_i$, а координация их взаимодействия при выполнении функционального задания должна осуществляться с помощью центрального диспетчера. При такой организации функции центрального диспетчера заключаются в приеме очередного функционального задания Z_l , распределении его операций между отдельными ресурсами, входящими в состав РС, и организации их взаимодействия в процессе выполнения задания, в то время как локальные устройства управления отдельных ресурсов должны обеспечивать выполнение отдельных операций, назначенных им диспетчером.

К преимуществам такой схемы следует отнести простоту организации диспетчера, а также возможности использования классических методов и алгоритмов теории расписаний для решения задачи диспетчирования РС [14]. Однако, вычислительная трудоемкость задачи распределения будет экспоненциально зависеть от количества переменных (в данном случае – от числа ресурсов в РС, а также количества операций в поступающих заданиях), поэтому данная схема организации диспетчера возможна только в случае когда число N ресурсов, входящих в РС, относительно невелико, а новые функциональные задания поступают на РС достаточно редко. Кроме того, централизованная организация диспетчера РС приводит к значительному снижению отказоустойчивости системы, поскольку выход его из

строения приводит к катастрофическим последствиям для всей РС в целом. К недостаткам централизованной организации диспетчера РС следует также отнести трудности масштабирования (наращивания количества ресурсов) РС, поскольку при этом требуется кардинальная переделка алгоритмов работы диспетчера РС.

Частично указанные недостатки можно устранить за счет применения иерархической схемы организации диспетчера РС (рис. 4). При этом диспетчер РС строится в виде иерархического дерева, в котором диспетчеры нижнего уровня отвечают за организацию работы некоторого подмножества ресурсов РС, а функции диспетчера верхнего уровня заключаются в приеме функционального задания, разбиения его на подзадания. При этом, поскольку количество ресурсов, за которые «отвечает» каждый из диспетчеров нижнего уровня, а также число операций в подзадании, закрепляемым за ним диспетчером верхнего уровня, будет относительно невелико, то, соответственно, и временная сложность решаемой диспетчером нижнего уровня задачи диспетчерования также будет невелика. Это позволяет использовать при организации работы диспетчеров нижнего уровня классические методы и алгоритмы теории расписаний, реализация которых при небольшом количестве диспетчируемых ресурсов не потребуют больших вычислительных и временных затрат.

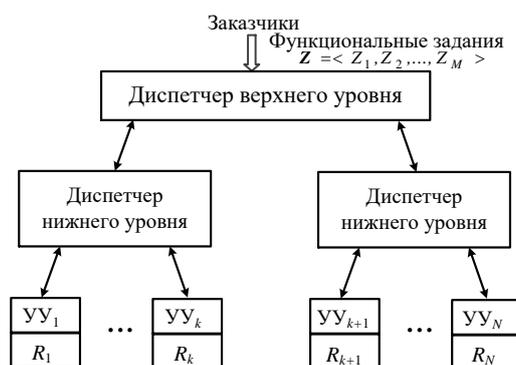
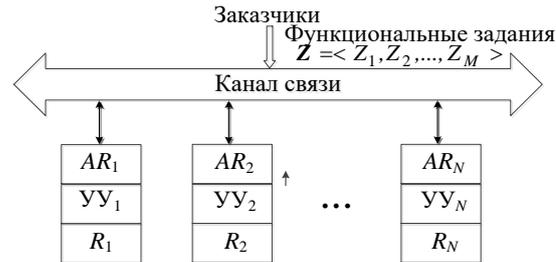


Рис. 4. РС с иерархическим диспетчером

Однако, с другой стороны, иерархическая схема организации диспетчера РС резко усложняет саму задачу диспетчерования, поскольку функциональное задание, поступающее на РС, необходимо предварительно каким-то образом разбить на подзадания, решаемые группами ресурсов, объединяемых диспетчерами нижнего уровня, а также организовать целенаправленное взаимодействие ресурсов как внутри групп, так и между группами. Также существенно усложняется проблема масштабирования количества ресурсов при модернизации РС, поскольку это потребует кардинальной переделки алгоритмов работы, как диспетчера верхнего уровня, так и диспетчеров нижнего уровня, обслуживающих эти дополнительные ресурсы. Кроме того, в системе все равно остается «узкое горло», а именно, диспетчер верхнего уровня, выход которого из строя будет приводить к катастрофическим последствиям для всей РС в целом.

Для устранения указанных недостатков предлагается воспользоваться мультиагентным подходом к диспетчерованию, предполагающим частичное решение задачи диспетчерования на всех входящих в РС узлах [15–18]. При этом возникает несколько проблем: как организовать диспетчер РС для возможности применения мультиагентного подхода и как организовать взаимодействие элементов распределенной системы.

Мультиагентное решение задачи диспетчирования ресурсов РС с использованием социоинспирированного подхода. Предположим, что каждый ресурс $R_i \in R$ обладает некоторым программным агентом AR_i , которые «умеют договариваться» друг с другом о распределении между собой операций поступающих функциональных заданий таким образом, чтобы минимизировать среднее время задержки ΔT их выполнения. Иными словами, задача диспетчирования в данном случае должна решаться в результате взаимодействия программных агентов друг с другом посредством информационного канала связи (рис. 5)



AR_i ($i = 1, 2, \dots, N$) - программный агент ресурса R_i

Рис. 5. Децентрализованная (мультиагентная) организация диспетчера РС

Однако, несмотря на очевидные преимущества, рассмотренная выше децентрализованная схема организации диспетчера практически не используется в современных РС. Это вызвано, в первую очередь, отсутствием теоретических основ мультиагентного диспетчирования ресурсов РС при выполнении потоков заданий, а также методов и алгоритмов работы программных агентов, обеспечивающих высокую скорость решения задачи диспетчирования при сохранении допустимого уровня качества распределения.

Для создания методов и алгоритмов, которые позволят применять подобный децентрализованный диспетчер предлагается воспользоваться социоинспирированным подходом – разновидностью биоинспирированного подхода, предполагающим использование методов взаимодействия, применяемый группами высоко развитых организмов при решении комплексных задач.

Действительно, в настоящее время децентрализованный подход широко распространен в социальных системах, например при выполнении группами людей сложных проектных или строительных работ, на сборочном производстве, при осуществлении боевых действий и т.п. [19] Во всех перечисленных выше социальных РС каждый ресурс системы (например станок в многостаночном производстве) обладает «своим» агентом – человеком, отвечающим за данный ресурс (станок), а распределение ресурсов при выполнении сложных заданий, включающих в своей состав множество взаимосвязанных операций, осуществляется путем взаимодействия (договоренности) агентов (людей) друг с другом о том, кто какую операцию и в какой последовательности выполняет. Такое взаимодействие в сообществе людей, направленное на выполнение сложной функционального задания, называется социальным поведением [20]. При этом, как показывает опыт эволюционного развития человеческого общества, именно навыки социального поведения дают возможность группам людей выполнять сложные задания в реальном времени их поступления, например при выполнении сложных боевых операций. Все это позволяет сделать вывод о том, что при разработке методов и алгоритмов мультиа-

гентного диспетчирования ресурсов в РС целесообразно использовать подходы, основанные на принципах социального поведения коллективов людей, задействованных в выполнении сложных функциональных заданий.

Приведем несколько примеров.

1. Игра в футбол. Футбольная команда – это пример гомогенной социальной РС с децентрализованным мультиагентным диспетчером, в которой каждый ресурс (игрок) самостоятельно принимает решение о своих текущих действиях на поле для достижения общекомандной цели.

2. Автомобильный трафик.

Автомобильный трафик – это «человеко-машинная» РС, включающая в свой состав тысячи и даже десятки тысяч ресурсов (автомобилей), каждым из которых управляет свой агент (шофер). В системе отсутствует какой-либо центральное устройство управления (если не считать дорожные знаки и светофоры), а взаимодействие агентов (шоферов) осуществляется посредством визуального информационного канала.

3. Боевые подразделения.

Боевые подразделения – это пример гетерогенных распределенных систем, включающих в свой состав множество ресурсов с различной специализацией, таких как отдельные бойцы-пехотинцы или боевые машины, управляемые экипажем. Общим функциональным заданием, стоящим перед такой РС, является нанесение максимального ущерба противнику при минимизации своих собственных потерь.

Во всех перечисленных выше социальных РС решение задачи диспетчирования ресурсов осуществляется с помощью множества агентов (людей), каждый из которых представляет свой ресурс (например, обрабатывающий станок) и решает относительно простую задачу загрузки своего ресурса операциями поступающих заданий, в то время как скоординированная работа всей РС в целом, направленная на выполнение общего функционального задания, осуществляется путем информационного взаимодействия агентов посредством некоторого канала связи. В общем случае стратегии социального поведения людей, участвующих в сообществе по выполнению некоторого общего функционального задания, могут быть различными. В идеале все агенты (люди), входящие в такое сообщество, должны четко и однозначно понимать стоящую перед сообществом функциональную задачу и стараться решить ее оптимальным образом с точки зрения всего сообщества в целом.

На базе приведенных утверждений предлагается применять следующий алгоритм работы распределенной системы и ее элементов.

Алгоритм работы распределенной системы и ее элементов. Итак, в качестве объекта дальнейшего исследования будем рассматривать распределенную систему R , состоящую из N ресурсов R_1, R_2, \dots, R_N , объединенных информационным каналом связи. При этом будем считать, что каждый ресурс R_i обладает своим локальным устройством управления $УУ_i$, обеспечивающем выполнение операций множества $O_i = \langle O_1^i, O_2^i, \dots, O_B^i \rangle$, а также некоторым программным агентом AR_i , представляющим его «интересы» в процессе диспетчирования. Кроме того, будем считать, что в произвольные моменты времени от различных Заказчиков в РС поступает поток функциональных заданий $Z = \langle Z_1, Z_2, \dots, Z_L \rangle$ (см. рис. 5), представляемых в виде графов $G_l(Q_l, X_l)$, причем для каждого задания Z_l Заказчик устанавливает требуемый момент времени его выполнения T_{max}^l .

Поскольку каждый ресурс, входящий в РС, обладает своим локальным устройством управления, то очевидно, задача выполнения некоторого задания $Z_l \in Z$ может быть решена в два этапа:

1. Сначала среди всего множества ресурсов R_1, R_2, \dots, R_N , входящих в РС, необходимо сформировать некоторое подмножество $R_l \subseteq R$ (в дальнейшем такое подмножество будем называть сообществом), обеспечивающее выполнение всего задания Z_l к установленному Заказчиком моменту времени T_{max}^l , и распределить операции, приписанные вершинам графа $G_l(Q_l, X_l)$ задания Z_l , между ними. При этом под распределением операций понимается закрепление операций, приписанных вершинам графа $G_l(Q_l, X_l)$ задания Z_l , за конкретными ресурсами, входящими в состав сообщества $R_l \subseteq R$, с привязкой их выполнения к определенным моментам времени (иначе говоря – построение временного графика выполнения операций задания Z_l различными ресурсами сообщества $R_l \subseteq R$).

2. После того как операции задания Z_l распределены между ресурсами сообщества $R_l \subseteq R$ последние, помощью своих локальных систем управления, обеспечивают выполнение закрепленных за ними операций по установленному временному графику.

Основную трудность при этом, очевидно, составляет задача первого этапа, а именно распределения операций графа $G_l(Q_l, X_l)$ задания Z_l между ресурсами РС с учетом их специализации и производительности. Проблеме распределения операций (сетевое планирование) посвящено достаточно большое число исследований с использованием, например, методов линейного или динамического программирования, теории расписаний и т.д. []. Поэтому, если графы $G_l(Q_l, X_l)$ ($l = 1, 2, \dots, M$) всех заданий множества Z и состав ресурсов распределенной системы R известны заранее, то с помощью известных методов, в принципе, такие расписания могут быть составлены для каждого из заданий множества $Z_l \in Z$ до начала их исполнения и просто храниться в памяти системы.

Однако, это идеализированный случай. В реальности графы операций выполняемых РС заданий $Z = \langle Z_1, Z_2, \dots, Z_L \rangle$ могут быть заранее неизвестны, а состав ресурсов, входящих в РС R , может динамически изменяться непредсказуемым образом (например, некоторые из них могут выходить из строя, либо изменять свою производительность). Кроме того, поскольку мы приняли, что функциональные задания от Заказчиков поступают в случайные моменты времени, то невозможно заранее предугадать, какие из заданий могут одновременно выполняться в РС. Поэтому возникает необходимость разработки таких методов и алгоритмов, которые бы обеспечивали автоматическое распределение операций потока заранее неизвестных заданий $Z_l \in Z$, поступающих в произвольные моменты времени среди множества ресурсов R_1, R_2, \dots, R_N распределенной системы R .

Выше было показано, что в качестве основы для разработки таких методов и алгоритмов целесообразно использовать принципы социального взаимодействия, используемые группой людей, участвующих в выполнении общего задания. Приведем формальное описание указанных принципов.

Соционспирированный подход к диспетчированию задач. Как мы приняли выше, за каждым ресурсом R_i ($i = 1, 2, \dots, N$), входящим в состав РС, закреплен отдельный агент AR_i , причем все агенты, представляющие различные ресурсы РС, могут общаться друг с другом посредством некоторого информационного канала связи. Будем считать, что все эти агенты заинтересованы в выполнении поставленного перед РС задания $Z_l \in Z$ к установленному Заказчиком моменту времени

T_{max}^l . Тогда каждый из агентов AR_i ($i = 1, 2, \dots, N$) в процессе распределения операций задания $Z_l \in \mathbf{Z}$ должен постараться взять «на себя» (т.е. на представляемый ими ресурс) исполнение такого подмножества вершин (ветви) графа $G_l(Q_l, X_l)$ задания $Z_l \in \mathbf{Z}$, операции которых он сможет выполнить с помощью «своего» ресурса R_i к установленному моменту времени. Осуществляя последовательный выбор различных ветвей графа задания $G_l(Q_l, X_l)$ для исполнения агенты различных ресурсов РС тем самым образуют сообщество R_l по его выполнению.

Здесь возникает один вопрос – где должен быть размещен граф $G_l(Q_l, X_l)$ задания Z_l и каким образом должен осуществляться доступ к нему агентов различных ресурсов R_1, R_2, \dots, R_N в процессе выбора для исполнения той или иной ветви графа. Очевидно, что это должно быть какое-то хранилище, к которому должны иметь доступ как агенты всех ресурсов РС, так и Заказчики заданий. В качестве такого общедоступного хранилища может быть использован некоторый пассивный узел, подключенный к информационному каналу и играющий роль «доски объявлений» (ДО) (рис. 6), на котором, с одной стороны, Заказчики могут размещать свои функциональные задания, представленные в виде графа $G_l(Q_l, X_l)$, а с другой стороны агенты всех ресурсов R_1, R_2, \dots, R_N могут иметь доступ к размещенным заданиям для участия в процедуре распределения их операций.

При этом дескриптор задания Z_l , размещаемого Заказчиком на ДО, должен содержать:

- ◆ граф $G_l(Q_l, X_l)$ задания Z_l , представленный, например, в виде матрицы инцидентности вершин;
- ◆ список вершин множества Q_l и приписанных им операций из множества O ;
- ◆ момент времени T_{max}^l , к которому Заказчик хочет получить результат выполнения задания Z_l ;
- ◆ список агентов, участвующих в сообществе R_l по выполнению задания Z_l (заметим, что в момент размещения задания на ДО этот список пуст и должен заполняться в процессе распределения).

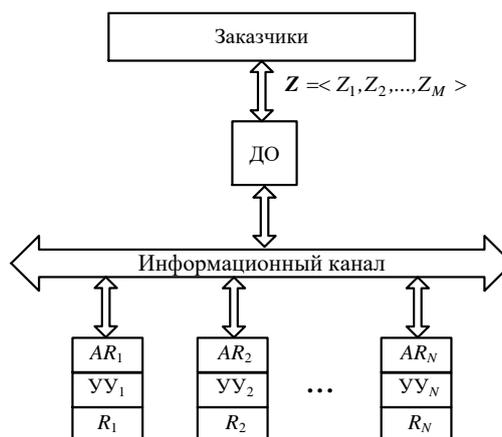


Рис. 6. Схема взаимодействия Заказчиков и агентов ресурсов РС посредством «доски объявлений»

Учитывая приведенные выше соображения работу децентрализованной РС с мультиагентным диспетчером в общем виде можно представить следующим образом:

1. Заказчик формирует свое функциональное задание $Z_l \in \mathbf{Z}$ в виде графа $G_l(Q_l, X_l)$, устанавливает момент времени T_{max}^l , к которому он желает получить результат его исполнения, и размещает дескриптор задания Z_l на ДО.

2. Агенты ресурсов R_1, R_2, \dots, R_N , входящих в состав РС, периодически опрашивают ДО в поисках работы для «своих» ресурсов. В случае обнаружения на ДО некоторого задания $Z_l \in \mathbf{Z}$ агент ресурса R_1 делает попытку войти в состав сообщества R_l по его выполнению. Для этого агент AR_1 выделяет в графе $G_l(Q_l, X_l)$ задания $Z_l \in \mathbf{Z}$ некоторую ветвь $H_1 = \langle q_1^1, q_2^1, \dots, q_k^1 \rangle$ (т.е. некоторую последовательность входящих в него операций, не закрепленных ранее за другими агентами), операции которой он может выполнить с минимальной задержкой относительно требуемого момента времени T_{k+1}^1 , приписанного ее конечной вершине q_k^1 , с помощью «своего» ресурса R_1 . Здесь следует отметить, что при размещении Заказчиком функционального задания Z_l на ДО, требуемый момент времени исполнения будет задан установлен только для конечной вершины q_k всего графа $G_l(Q_l, X_l)$, который определяется установленным Заказчиком моментом времени T_{max}^l выполнения всего задания. Если агент AR_1 обнаруживает ветвь H_1 графа $G_l(Q_l, X_l)$, удовлетворяющую данным условиям, то он вступает в сообщество R_l по выполнению задания $Z_l \in \mathbf{Z}$.

4. При этом агент AR_1 осуществляет следующие модификации графа $G_l(Q_l, X_l)$ на ДО: выбранная им для исполнения ветвь H_1 удаляется из графа задания $G_l(Q_l, X_l)$, т.е. формируется новый граф $G_l^1(Q_l^1, X_l^1) = G_l(Q_l, X_l) / H_1$, а всем вершинам этого нового графа, инцидентным вершинам удаленной ветви H_1 , агент AR_1 приписывает моменты времени, когда соответствующие им операции должны быть выполнены с тем, чтобы «вписаться» в график выполнения ветви H_1 , закрепленной за его ресурсом R_1 .

5. Далее аналогичный выбор делает агент AR_2 , представляющий ресурс R_2 – в графе $G_l^1(Q_l^1, X_l^1)$, оставшемся после модификации агентом AR_1 , он выбирает такую ветвь $H_2 = \langle q_1^2, q_2^2, \dots, q_r^2 \rangle$, которую с помощью «своего» ресурса R_2 он может выполнить к моменту времени, приписанному ее конечной вершине q_r^2 с минимальной задержкой (в результате модификации, произведенной агентом AR_1 , в графе $G_l^1(Q_l^1, X_l^1)$ может уже иметься некоторое множество вершин, которым приписаны требуемые моменты времени их исполнения). Далее в процесс распределения вступает агент AR_1 , представляющий ресурс R_3 и т.д., до тех пор, пока не будут разобраны все ветви графа задания $G_l(Q_l, X_l)$, т.е. пока он окажется, что после очередной модификации граф задания на ДО становится пустым.

6. После того, как агент AR_i выбрал для исполнения ветвь H_i он инициирует процедуру выполнения операций, приписанных вершинам этой ветви, с помощью локального устройства управления «своего» узла R_i .

Заключение. Предложенная в статье мультиагентная организация диспетчера РС имеет очевидные преимущества:

- ◆ во-первых, программный агент AR_i ресурса R_i будет иметь достоверную и полную информацию о специализации и текущей производительности «своего» ресурса при выполнении тех или иных операций, что позволяет повысить качество решения задачи диспетчирования;

- ◆ во-вторых, в процессе диспетчирования вычислительная и, соответственно, временная трудность задачи, решаемой программным агентом, существенно снижается, поскольку агент «отвечает» только за один «свой» ресурс, входящий в состав РС, что позволяет существенно снизить время решения задачи диспетчирования;

- ◆ во-третьих, система будет обладать максимальной отказоустойчивостью, поскольку в ней отсутствуют части, выход из строя которых приводит к полной потере работоспособности РС в целом;

- ◆ в-четвертых, для масштабирования РС (увеличения числа ресурсов) нет необходимости менять какие-либо алгоритмы работы диспетчеров, а достаточно просто подключить новый ресурс (обладающий соответствующим программным агентом) к общему каналу информационного обмена.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шамакина А.В. Обзор технологий распределенных вычислений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2014. – № 3. – URL: <https://cyberleninka.ru/article/n/obzor-tehnologiy-raspredeleennyh-vychisleniy> (дата обращения: 18.08.2021).
2. Смелянский Р.Л. Модель функционирования распределенных вычислительных систем // Вестник Московского университета. – 1990. – Т. 15. – Р. 3-21.
3. Карпов В.Э. Коллективное поведение роботов. Желаемое и действительное // Современная мехатроника: Сб. научн. трудов Всероссийской научной школы (г. Орехово-Зуево, 22-23 сентября 2011). Орехово-Зуево, 2011. – Р. С132.
4. Данеев А.В., Басыров А.Г., Мастин А.Б. Диспетчер энергосберегающего параллельного вычислительного процесса // Современные технологии. Системный анализ. Моделирование. – 2010. – № 3. – URL: <https://cyberleninka.ru/article/n/dispatcher-energoberegayuschego-parallelnogo-vychislitelnogo-protsesta> (дата обращения: 18.08.2021).
5. Kalyaev A.I., Kalyaev I.A. Method of multiagent scheduling of resources in cloud computing environments // Journal of Computer and Systems Sciences International. – 2016. – Vol. 55, No. 2. – P. 211-221. – DOI: 10.1134/S1064230716010081.
6. Иващенко А.В. и др. Мультиагентные технологии для разработки сетцентрических систем управления // Известия ЮФУ. Технические науки. – 2011. – № 3 (116).
7. Городецкий В.И., Бухвалов О.Л., Скобелев П.О. Современное состояние и перспективы промышленных применений многоагентных систем // Управление большими системами: Сб. трудов. – 2017. – № 66.
8. Dechesne F., Ghorbani A., Yorke-Smith N. Introduction to the special issue on agent-based modelling for policy engineering // AI Soc. – Springer, 2015. – Vol. 30, No. 3. – P. 311-313.
9. Кулинич А.А. Модель командного поведения агентов в качественной семиотической среде. Ч. 1. Качественная среда функционирования. Основные определения и постановка задачи // Искусственный интеллект и принятие решений. – 2017. – № 3. – P. 95-105.
10. Карпов В.Э. Модели социального поведения в групповой робототехнике // Управление большими системами: сборник трудов. – 2016. – № 59.
11. Kalyaev A. et al. An effective algorithm for multiagent dispatching of resources in heterogeneous cloud environments // 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). – IEEE, 2016. – P. 1140-1142.
12. Kalyaev I.A., Kalyaev A.I., Korovin I.S. Multiagent Resource Dispatching in a Heterogeneous Cloud Environment // 2019 International Conference on Electronical, Mechanical and Materials Engineering. – Atlantis Press, 2019. – P. 79-85.

13. Kalyaev I.A., Kalyaev A.I., Korovin I.S. A Modified Method of Multiagent Resource Dispatching in a Heterogeneous Cloud Environment // 2019 International Conference on Electrical, Mechanical and Materials Engineering (ICE2ME 2019). – Atlantis Press, 2019. – P. 72-78. – DOI: 10.2991/ice2me-19.2019.17.
14. Герасимов Б.Н., Морозов В.В., Яковлева Н.Г. Системы управления: понятие, структура, исследование. – Самара СГАУ, 2002.
15. Городецкий В.И. Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации // Известия Российской академии наук. Теория и системы управления. – 2012. – № 2. – P. 92.
16. Легович Ю.С., Максимов Д.Ю. Выбор исполнителя в группе интеллектуальных агентов // Управление большими системами: сборник трудов. – 2015. – № 56.
17. Истомин В.В. Прогнозирование поведения групп автономных интеллектуальных агентов на основе теории многоагентных систем // Инженерный вестник Дона. – 2011. – Т. 18, № 4.
18. Campbell A., Wu A.S. Multi-agent role allocation: issues, approaches, and multiple perspectives // Auton. Agent. Multi. Agent. Syst. – Springer, 2011. – Vol. 22, No. 2. – P. 317-355.
19. Гаазе-Ранопорт М.Г., Поселов Д.А. От амебы до робота: модели поведения. – Рипол Классик, 1987. – 150 с.
20. Conte R. et al. Sociology and social theory in agent based social simulation: A symposium // Comput. Math. Organ // Theory. – Springer, 2001. – Vol. 7, No. 3. – P. 183-205.

REFERENCES

1. Shamakina A.V. Obzor tekhnologiy raspredelennykh vychisleniy [Overview of distributed computing technologies], *Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika* [Bulletin of SUSU. Series: Computational Mathematics and Computer Science], 2014, No. 3. Available at: <https://cyberleninka.ru/article/n/obzor-tehnologiy-raspredelennykh-vychisleniy> (accessed 18 August 2021).
2. Smelyanskiy R.L. Model' funktsionirovaniya raspredelennykh vychislitel'nykh sistem [Model of functioning of distributed computing systems], *Vestnik Moskovskogo universiteta* [Bulletin of the Moscow University], 1990, Vol. 15, pp. 3-21.
3. Karpov V.E. Kollektivnoe povedenie robotov. Zhelaemoe i deystvitel'noe [Collective behavior of robots. Wishful and valid], *Sovremennaya mekhatronika: Sb. nauchn. trudov Vserossiyskoy nauchnoy shkoly (g. Orekhovo-Zuevo, 22-23 sentyabrya 2011)* [Modern mechatronics: Collection of scientific papers of the All-Russian Scientific School (Orekhovo-Zuevo, September 22-23, 2011)]. Orekhovo-Zuevo, 2011, pp. C132.
4. Daneev A.V., Basyrov A.G., Mastin A.B. Dispatcher energosberegayushchego parallel'nogo vychislitel'nogo protsessa [Dispatcher of an energy-saving parallel computing process], *Sovremennye tekhnologii. Sistemyy analiz. Modelirovaniye* [Modern technologies. System analysis. Modeling], 2010, No. 3. Available at: <https://cyberleninka.ru/article/n/dispatcher-energoberegayushchego-parallelnogo-vychislitel'nogo-protsess> (accessed 18 August 2021).
5. Kalyaev A.I., Kalyaev I.A. Method of multiagent scheduling of resources in cloud computing environments, *Journal of Computer and Systems Sciences International*, 2016, Vol. 55, No. 2, pp. 211-221. DOI: 10.1134/S1064230716010081.
6. Ivashchenko A.V. i dr. Multiagentnye tekhnologii dlya razrabotki setentsricheskikh sistem upravleniya [Multi-agent technologies for the development of network-centric control systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 3 (116).
7. Gorodetskiy V.I., Bukhvalov O.L., Skobelev P.O. Sovremennoe sostoyanie i perspektivy industrial'nykh primeneniy mnogoagentnykh sistem [The current state and prospects of industrial applications of multi-agent systems], *Upravlenie bol'shimi sistemami: Sb. trudov* [Managing large systems: Proceedings], 2017, No. 66.
8. Dechesne F., Ghorbani A., Yorke-Smith N. Introduction to the special issue on agent-based modelling for policy engineering, *AI Soc.* Springer, 2015, Vol. 30, No. 3, pp. 311-313.
9. Kulimich A.A. Model' komandnogo povedeniya agentov v kachestvennoy semioticheskoy srede. Ch. 1. Kachestvennaya sreda funktsionirovaniya. Osnovnye opredeleniya i postanovka zadachi [The model of team behavior of agents in a qualitative semiotic environment. Part 1. Qualitative functioning environment. Basic definitions and problem statement], *Iskusstvennyy intellekt i prinyatie resheniy* [Artificial intelligence and decision-making], 2017, No. 3, pp. 95-105.

10. Karpov V.E. Modeli sotsial'nogo povedeniya v gruppovoy robototekhnike [Models of social behavior in group robotics], *Upravlenie bol'shimi sistemami: sbornik trudov* [Management of large systems: proceedings], 2016, No. 59.
11. Kalyaev A. et al. An effective algorithm for multiagent dispatching of resources in heterogeneous cloud environments, *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, 2016, pp. 1140-1142.
12. Kalyaev I.A., Kalyaev A.I., Korovin I.S. Multiagent Resource Dispatching in a Heterogeneous Cloud Environment, *2019 International Conference on Electronical, Mechanical and Materials Engineering*. Atlantis Press, 2019, pp. 79-85.
13. Kalyaev I.A., Kalyaev A.I., Korovin I.S. A Modified Method of Multiagent Resource Dispatching in a Heterogeneous Cloud Environment, *2019 International Conference on Electronical, Mechanical and Materials Engineering (ICE2ME 2019)*. Atlantis Press, 2019, pp. 72-78. DOI: 10.2991/ice2me-19.2019.17.
14. Gerasimov B.N., Morozov V.V., yakovleva N.G. Sistemy upravleniya: ponyatie, struktura, issledovanie [Management systems: concept, structure, research]. Samara SGAU, 2002.
15. Gorodetskiy V.I. Samoorganizatsiya i mnogoagentnye sistemy. I. Modeli mnogoagentnoy samoorganizatsii [Self-organization and multi-agent systems. I. Models of multi-agent self-organization], *Izvestiya Rossiyskoy akademii nauk. Teoriya i sistemy upravleniya* [Proceedings of the Russian Academy of Sciences. Theory and control systems], 2012, No. 2, pp. 92.
16. Legovich Yu.S., Maksimov D.Yu. Vybor ispolnitelya v gruppe intellektual'nykh agentov [Choosing a performer in a group of intelligent agents], *Upravlenie bol'shimi sistemami: Sb. trudov* [Managing large systems: a collection of works], 2015, No. 56.
17. Istomin V.V. Prognozirovaniye povedeniya grupp avtonomnykh intellektual'nykh agentov na osnove teorii mnogoagentnykh sistem [Predicting the behavior of groups of autonomous intelligent agents based on the theory of multi-agent systems], *Inzhenernyy vestnik Dona* [Engineering Bulletin of the Don], 2011, Vol. 18, No. 4.
18. Campbell A., Wu A.S. Multi-agent role allocation: issues, approaches, and multiple perspectives, *Auton. Agent. Multi. Agent. Syst.* Springer, 2011, Vol. 22, No. 2, pp. 317-355.
19. Gaaze-Rapoport M.G., Pospelov D.A. Ot ameby do robota: modeli povedeniya [From amoeba to robot: behavioral models]. Ripol Klassik, 1987, 150 p.
20. Conte R. et al. Sociology and social theory in agent based social simulation: A symposium // *Comput. Math. Organ. Theory*. Springer, 2001, Vol. 7, No. 3, pp. 183-205.

Статью рекомендовал к опубликованию д.т.н. Э.В. Мельник.

Калыев Анатолий Игоревич – Южный федеральный университет; e-mail: anatoly@kalyaev.net; г. Таганрог, Россия; тел.: +79525666699; к.т.н.; зав. лабораторией.

Kalyaev Anatoly Igorevich – South Federal University; e-mail: anatoly@kalyaev.net; Taganrog, Russia; phone: +79525666699; cand. of eng. sc.; head of the laboratory.

УДК 004.056.5

DOI 10.18522/2311-3103-2021-4-21-31

В.А. Литвиненко, С.А. Ховансков, В.С. Хованскова

ОЦЕНКА ВЕРОЯТНОСТИ ОБНАРУЖЕНИЯ ЛОЖНОГО РЕЗУЛЬТАТА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ВЫПОЛНЯЕМЫХ ЦЕНТРАЛИЗОВАННОЙ МНОГОАГЕНТНОЙ СИСТЕМОЙ

Рассматриваются вопросы защиты распределённых вычислений организованных на основе многоагентной системе для решения задач многовариантного моделирования. При моделировании выбор одного из многих вариантов может потребовать перебора огромного множества параметров недоступного для быстродействующей ЭВМ. Для сокращения времени решения таких задач используют распределенные вычисления. Существует множество различных подходов для организации распределенных вычислений в компьютерной сети – технология grid, metacomputing (BOINC, PVM и другие). Все они предназначены для создания централизованных систем распределенных вычислений. Распределенные вычисления органи-