19. *Kutateladze S.S., Leont'ev A.I.* Teplomassoobmen i trenie v turbulentnom pogranichnom sloe [Heat and mass transfer and friction in a turbulent boundary layer]. Moscow: Energoatomizdat, 1985, 319 p.
20. *Repik E.U., Sosedko Yu.P.* Turbulentnyy pogranichnyy sloy. Metodika i rezul'taty turbulentnykh issledovaniy [A turbulent boundary layer. The methodology and results of turbulent research]. Moscow: Fizmatlit, 2007, 306 p.

Статью рекомендовал к опубликованию д.т.н., профессор Г.С. Панатов.

Палий Александр Викторович – Южный федеральный университет; e-mail: a.v._paliy@mail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371603; к.т.н.; кафедра конструирования электронных средств; доцент.

Paliy Alexander Viktorovich – Southern Federal University; e-mail: a.v._paliy@mail.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371603; the department of electronic apparatuses design; lecturer.

УДК 004.75 DOI 10.18522/2311-3103-2016-10-146158

С.А. Хованскоv, В.А. Литвиненко, В.С. Хованскова

АЛГОРИТМ ОРГАНИЗАЦИИ БЕЗОПАСНЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ МНОГОАГЕНТНОЙ СИСТЕМЫ*

В настоящее время разработке методов организации распределенных вычислений уделяется много внимания. Одним из методов создания распределенных вычислений является использование многоагентной системы. При организации распределенных вычислений на основе обычных сетевых компьютеров могут возникать угрозы безопасности выполняемых вычислительных процессов. Авторами разработан единый алгоритм агентов системы управления работой вычислительных узлов сети. В качестве вычислительных узлов используются персональные компьютеры, объединенные в сеть. Предлагаемый алгоритм управления агентом системы позволяет организовать использование агентами вычислительных мощностей своих компьютеров сети для решения большеобъемных задач путём создания безопасных распределенных вычислений. Агенты, управляющие сетевыми компьютерами, могут: организовать распределенную вычислительную систему, распределить вычислительную нагрузку между компьютерами управляемыми агентами, выполнить оптимизацию вычислительной нагрузки в зависимости от вычислительной мощности компьютеров. В ходе выполнения вычислений количество компьютеров, включенных в многоагентную систему, может увеличиваться без участия внешней управляющей системы. Это происходит за счет подключения новых компьютеров к вычислительной системе путем передачи им агентов, что приводит к увеличению вычислительной мощности системы. Благодаря разработанному алгоритму агента, организуемые безопасные распределенные вычисления позволяют сократить общее время решения большеобъемных задач. Организуемое алгоритмом взаимодействие агентов повышает отказоустойчивость (живучесть) вычислительных процессов в условиях нестабильности внутренней и внешней вычислительной среды (изменение количество работающих сетевых компьютеров). При отсутствии угрозы безопасности вычислениям многоагентная система способна выполнять перечисленные функции без управляющего центра. В случае наличия угрозы в децентрализованную многоагентную систему для повышения степени ее безопасности добавляется несколько «центральных» агентов, алгоритм работы которых усложнен по сравнению с другими агентами. Под управлением разработанного алгоритма агенты выполняют обнаружение случаев фальсификации результатов работы распределенной системы, которые могут привести к принятию неправильных решений.

Распределенные вычисления; многоагентная система; безопасность вычислений; защита работоспособности.

* Работа выполнена при поддержке РФФИ (проект № 14-01-00665).

S.A. Khovanskov, V.A. Litvinenko, V.S. Khovanskova

ALGORITHM FOR ORGANIZATION OF SAFE DISTRIBUTED COMPUTING ON THE BASIS OF MULTIAGENT SYSTEM

Nowadays the developing methods for distributed computing attract much attention. One of the methods for distributed computing is the use of multi-agent systems. The organization of distributed computing based on the conventional network computers can experience security threats performed by computational processes. Authors have developed the unified agent algorithm of control system of computing network nodes operation. Network PCs are used as computing nodes. The proposed multi-agent control system for the implementation of distributed computing allows in a short time to organize using of the processing power of computers in the network for solving large-tasks by creating a distributed computing. Agents who control a computer network can: configure a distributed computing system; distribute the computational load among computers operated by the agents; perform an optimization of computing load according to the computing power of computers in the network. The number of computers connected to the network can be increased by connecting computers to the new computer system, which leads to an increase in overall processing power. Adding multi-agent system in the central agent increases the security of distributed computing. This organization of the distributed computing system reduces the problem solving time and increase fault tolerance (vitality) of computing processes in a changing computing environment (dynamic change of the number of computers on the network). In the absence of security threats computing multi-agent system is able to perform the enumerated functions without a control center. In case of a threat several "central" agents which operation algorithms are complicated compared to other agents are added in a decentralized multi-agent system to enhance its degree of security. Under control of the developed algorithm agents will detect cases of falsification of the distributed system operation results, which may lead to wrong decisions.

Distributed computing; multi-agent system; security of computing; efficiency protection.

Введение. Использование распределенных вычислений позволяет значительно сократить время выполнения сложных задач.

Одним из методов создания распределенных вычислений является использование многоагентной системы [1–5].

Такой метод позволяет за небольшой промежуток времени организовывать многоагентной системой распределенные вычисления в обычной компьютерной сети. В качестве вычислительных узлов могут использоваться любые вычислительные системы, например персональные компьютеры.

При организации распределенных вычислений на основе обычных сетевых компьютеров могут возникать угрозы безопасности выполняемых вычислительных процессов. В качестве угроз безопасности рассматриваются, как отказ оборудования, так и несанкционированные действия злоумышленников.

Основные действия злоумышленников направлены на блокирование узлов вычислительной системы или дискредитацию результатов работы распределенного вычислительного процесса.

Обзор литературы. В работах по многоагентным системам предлагаются различные пути обеспечение безопасности вычислительных процессов под управлением многоагентной системы.

В работах [6–10] для организации распределенных вычислений и обеспечения их безопасности предлагается использовать централизованную многоагентную систему. Все узлы вычислительной системы работают под управлением централизованной системы. Сама система управления представляет собой несколько компьютеров. На них устанавливается специализированное программное обеспечение. Защиту процессов распределенных вычислений эти управляющие компьютеры обеспечивают путем постоянного контроля поведения всех компьютеров управляемых многоагентной системой, целостности вычислительных процессов и правильности получаемых результатов вычислений.

С целью обеспечение безопасности управляющие компьютеры разделяют все множество компьютеров на уровни по степени «доверия». Деление может осуществляться на два и более уровня. Количество уровней зависит от заданных критериев защиты в системе управления. По этим критериям управляющая система отслеживает и обеспечивает безопасность распределенных вычислений. Если компьютеры в системе были проверены на предыдущих вычислениях и, согласно принятым критериям, обеспечивали надежный вычислительный процесс и получали правильные результаты, то они имеют уровень «доверия» более высокий. Компьютеры, которые дискредитировали себя, имеют низкий уровень «доверия». Результаты вычислений, полученные этими компьютерами, пересчитываются и в последующих вычислениях эти компьютеры не участвуют.

Такая распределенная система может обеспечивать приемлемый уровень безопасности, если окружающая вычислительная среда не изменяется (либо изменяется в небольших пределах). В большой сети, имеющей нестабильную вычислительную среду, время получения результатов, в конечном счете, будет увеличиваться за счет времени необходимого системе управления для выполнения многочисленных проверок компьютеров в работающей распределенной вычислительной системе и определения для каждого компьютера соответствующего уровня «доверия».

Еще одним недостатком такой организации распределенных вычислений, с помощью системы управления, является угроза блокировки компьютера управления. Система распределенных вычислений останется без управления и не сможет обеспечить требуемые вычислительные процессы.

Другим подходом обеспечения безопасности вычислительных процессов под управлением многоагентной системы является деление исполняемого кода на открытые и закрытые блоки [7, 11–12].

Открытый блок может исполняться только после выполнения закрытого блока. Закрытые блоки включают в себя наиболее важные команды. Открытые блоки могут выполняться на любом компьютере, а закрытые блоки – только на проверенных компьютерах. Ранжирование на открытые и закрытые блоки выполняется заранее специальной программой по заданным критериям. При этом критерии задаются и корректируются программистом.

Основным недостатком этого подхода являются неизбежные временные затраты связанные с необходимостью задания, для каждой новой задачи, своих критериев ранжирования вычислительных блоков на открытые и закрытые. В случае использования неустойчивой вычислительной среды весь ход ранжирования по блокам будет сопровождаться ручной корректировкой. После выполнения процесса ранжирования, требуется время для ручного уточнения результата. В некоторых случаях, когда однозначно задать критерии невозможно, ранжирование придется выполнять полностью вручную. Поэтому такую систему можно применять для выполнения вычислений, которые не имеют жесткого ограничения по времени решения.

Постановка задачи. Основной проблемой, на решение которой направлена настоящая работа, является создание безопасных распределенных вычислений в компьютерной сети на основе многоагентной системы с целью сокращения времени решения большеобъемных задач. Для этого необходимо решить следующие задачи:

- разработать алгоритм агента многоагентной системы, позволяющий организовать децентрализованную распределенную вычислительную систему на основе компьютерной сети;
- организовать взаимодействие агентов таким образом, чтобы вычислительные процессы оставались работоспособными при динамическом изменении вычислительной среды;

- обеспечить защиту результатов распределенных вычислений от угрозы дискредитации результатов решения.

Создаваемая вычислительная система должна:

- свести к минимуму подготовительные этапы процесса решения большеобъемной задачи;
- быть работоспособной при любом наборе компьютеров, как по количеству, так и по производительности;
- обладать достаточной степенью безопасности вычислений с точки зрения работоспособности при изменении вычислительных ресурсов сети;
- обеспечивать защиту распределенных вычислений от угрозы дискредитации результатов решения задачи;
- сократить время решения задачи.

Организация распределенных вычислений на основе многоагентной системы. Многоагентная система представляет собой множество агентов *M* в виде одинаковых программных модулей агентов $\{m_1,m_2,\ldots,m_n\} \in M$. Множество *M* накладывается на множество $\{p_1, p_2, \ldots, p_j\} \in P$ сетевых компьютеров ($P \succ M$) так, что агент m_i располагается на соответствующем p_i компьютере сети. Вся многоагентная система *M*, управляя компьютерами $\{p_1, p_2, \ldots, p_n\} \in P$, организует систему распределенных вычислений для решения всего множества *W* заданий $\{w_1, w_2, \ldots, w_n\} \in W$. Множество *M* является одноранговым набором агентов работающих по одному алгоритму (рис. 1).

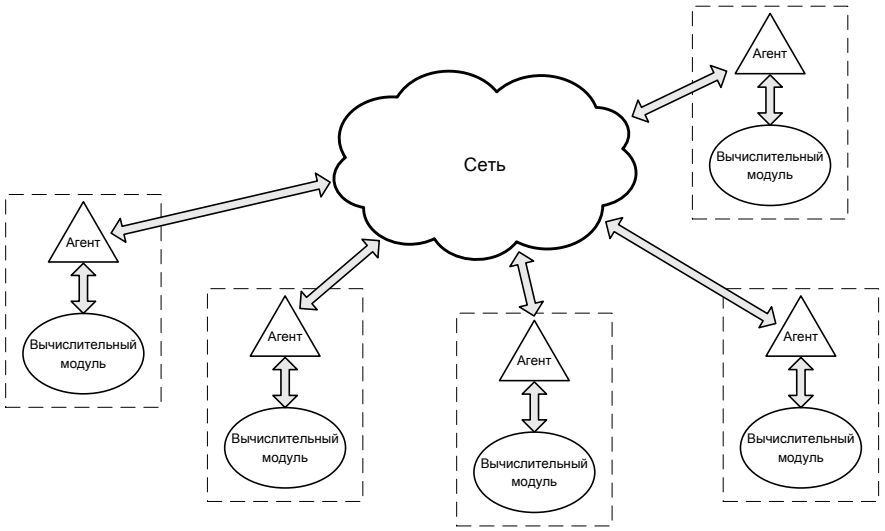

Рис. 1. Структура многоагентной системы

Для *W* системы был разработан алгоритм взаимодействия и работы *n* агентов в компьютерной сети состоящей из множества узлов *P* [13–15].

Каждый модуль агента $m_i \in M$ (агент) управляет ресурсами компьютера p_i и следит за выполняемой на нем нагрузке w_i. В свою очередь, каждый компьютер p_i, управляемый агентом m_i, работает независимо от других компьютеров. Агенты $m_i \in M$ обмениваются друг с другом информацией по компьютерной сети, используя сетевые ресурсы $p_i \in P$.

Алгоритм агента функционирует следующим образом. В начале организации распределённых вычислений в компьютерной сети P на $\{p_1, p_2, \quad , p_n\} \in P$ находятся, управляющие их работой, агенты $\{m_1, m_2, \ldots, m_n\} \in M$. На первом этапе агент $m_i \in M$, получает основную информацию для организации распределенных вычислений в множестве M. Она включает в себя вычислительную нагрузку W системы M и указание того, какую часть w_i из общего объема W агент должен выполнить. На первоначальном этапе организации распределённых вычислений в компьютерной сети $w_i \subseteq W$.

После получения агентом $m_i \in M$ нагрузки и общей информации о системе он инициирует на своем компьютере вычислительный процесс для выполнения w_{i1}. После выполнения каждого решения w_{i1} агент рассылает результат агентам $\{m_1, m_2, \ldots, m_{n-1}\} \in M$.

На фоне выполнения p_i вычислительных процессов агент m_i выполняет поиск m_s, расположенного на $p_s \in P_z$ и не имеющего нагрузки. Если m_s будет найден, и $w_s = 0$, то m_i передает m_s информацию о нагрузке W и части w_i, которая еще не выполнена, уменьшая таким образом свою собственную нагрузку.

Таким образом достаточно одному m_i получить информацию о W и нагрузка $w_i = W$, то по ходу своей работы агенты множества M распределят её между собой $\{w_1, w_2, \quad , w_n\} \in W$.

Организуемая таким образом вычислительная система на основе компьютеров сети P_z является распределенной, благодаря разработанному алгоритму работы агентов множества M. На фоне выполнения w_i агенты m_i уменьшают вычислительные нагрузки на свои $p_i \in P_z$ с целью сокращения времени T выполнения системой W [14, 15].

После того, как нагрузка $\{w_1, w_2, \quad , w_n\} \in W$ распределена между $\{m_1, m_2, \ldots, m_n\} \in M$ на $\{p_1, p_2, \quad , p_n\} \in P$, образующими вычислительную систему, каждый p_i под управлением m_i выполняет w_i. В ходе выполнения w_i агент m_i отправляет по сети остальным агентам полученные результаты вычислений и получает результаты от $\{m_1, m_2, \ldots, m_{n-1}\} \in M$. Работа системы M организована так, что каждый m_i принимает и хранит результаты, полученные в ходе работы всей распределенной вычислительной системы.

Распределение нагрузки не является окончательным. При выполнении разработанного алгоритма агентом m_i, W может быть перераспределена в процессе своего выполнения. Перераспределение выполняется в соответствии с сетевым трафиком P и вычислительными ресурсами p_i.

На фоне выполнения p_i вычислительных процессов агент m_i выполняет поиск компьютеров $p_s \in (P \backslash P_z)$. Если p_s найден и он свободен то m_i передает ему копию программного модуля агента. Переданная копия запускается на p_s и агенту m_s от m_i передается информация об общей нагрузки о множестве агентов и части своей невыполненной нагрузки.

При выполнении алгоритма агента m_i вычислительная система имеет возможность увеличивать число образующих ее компьютеров P_z путем передачи копии агента m_i в $p_s \notin P_z$. Поскольку все $m_i \in M$ работают по одному алгоритму, появляется возможность выполнения поставленных задач на системе обладающей дополнительными вычислительными ресурсами.

Каждый $m_i \in M$ не ограничивается выполнением только тех вычислительных процессов, которые входят в w_i. После выполнения своей части w_i агент просматривает – все ли результаты решений W получены. Если есть задания, результаты решения которых он еще не получил, то m_i формирует из них новую часть w_i и дает команду p_i приступить к их выполнению.

Такая ситуация с невыполненными заданиями может возникнуть по разным причинам. Либо из-за отсутствия соответствующего агента в системе, либо вычисления выполняются на p_s, который обладает небольшими вычислительными ресурсами.

Структурная схема алгоритма работы m_i на рис. 2.

Выполнение агентами разработанного алгоритма позволяет повысить живучесть системы и отказоустойчивость вычислительных процессов. При отключении p_k с агентом m_k от сети или отказе вычислительного оборудования его нагрузка w_k обязательно будет выполнена. Это будет реализовано агентами вычислительной системы, которые уже выполнили свою часть W.

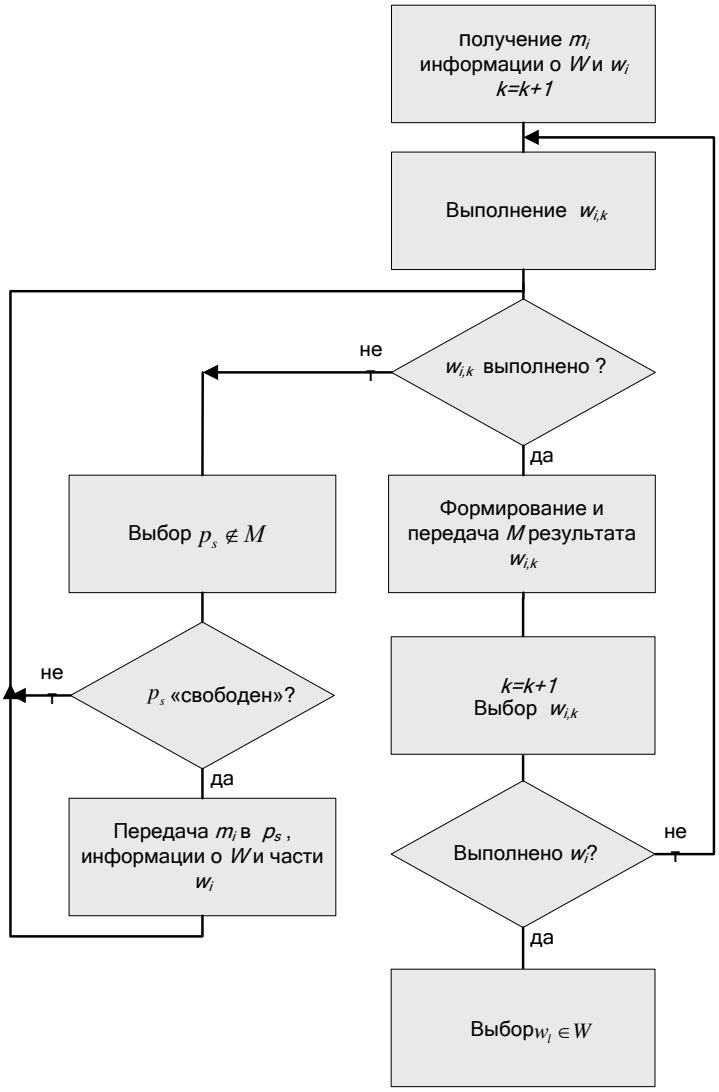

Рис. 2. Алгоритм работы агента

Поэтому в случае отключения нескольких компьютеров с агентами от вычислительной системы и уменьшении множества M, время выполнения системой распределённых вычислений T увеличится, но результаты вычислений будут получены в любом случае, даже тогда, когда в системе останется один агент $m_i \in M$ на $p_i \in P_z$.

Организация безопасных распределенных вычислений на основе многоагентной системы. Система распределённых вычислений на базе многоагентной системы может быть организована на основе вычислительных узлов любой сети, например, глобальной компьютерной сети Интернет. С одной стороны это дает возможность практически бесконечно наращивать совокупную вычислительную мощность системы. Но с другой стороны возникает угроза безопасности распределенных вычислений, поскольку повышается вероятность проникновения в многоагентную систему такого масштаба злоумышленника предпринимающего действия, ведущие к компрометации результатов вычислений [1, 6–9, 11–12, 16–19].

Результат w_i может быть фальсифицирован с целью предоставление недостоверных данных вычислений, что может привести к неправильному принятию решения. Такое происходит, когда злоумышленник имитирует работу агента $m_i \in M$. Для этого он может принимать или перехватывать сообщения, передаваемые в системе M, подменять в них результаты w_i, полученные m_i распределённой вычислительной системы, на ложные.

В случае, когда угроза безопасности вычислений является реальной, в системе выбирается $m_d \in M$ для организации рассылки результатов расчетов W. Выбор m_d производится только из множества M_s расположенного на определенном заранее множестве компьютеров $P_s \in P$, которые являются надежными и не создают угрозы безопасности вычислениям.

Агент m_d управляющий компьютером p_d в системе распределенных вычислений не отличается от агентов $\{m_1, m_2, \ldots, m_n\} \in M$. Он, как и все, выполняет $w_k \in W$, но, наряду с этим, обеспечивает процесс контроля за правильностью и безопасностью вычислений.

Каждый агент m_i, закончив выполнение очередного задания из w_i, отсылает всем $\{m_1, m_2, \ldots, m_n\} \in M$ сообщения и в том числе m_d содержащее полученные данные. В сообщении указывается идентификационная информация, однозначно определяющая w_i, p_i, агента m_i, который получил результат и значение полученного результата. Все $\{m_1, m_2, \ldots, m_n\} \in M$ распределенной вычислительной системы будут хранить у себя не только свое задание но и информацию о всей нагрузке W и результаты всех выполненных вычислений в системе.

Агент m_d, после получения информации от агентов $\{m_1, m_2, \ldots, m_n\} \in M$ записывает результаты вместе с номерами p_i, который получил эти результаты, к себе в память. После этого агент m_d проверяет, выполнялось ли задание, результат которого был передан, раньше и кто выполнял. Если задание выполнялось агентом $m_i \in M$, то проверяет – совпадает ли значение результата в сообщении с полученным им ранее результатом.

В случае, если результат не совпадает, то агент m_d принимает решение об ошибочном или фальсифицированном результате и сам выполняет перепроверку результата w_i. После этого агент m_d отправляет сообщения агентам $(m_1, m_2, \ldots, m_n) \in M$ о факте разницы результатов и правильном w_i.

В силу различных причин, может произойти отключение p_d от системы P. Когда, агент m_d перестает выполнять свои функции, то алгоритм многоагентной системы позволяет назначить агентом для организации рассылки результатов расчетов W другой $m_e \in M$. Выбор m_e также производится только из множества M_s расположенного на определенном заранее множестве компьютеров $P_s \in P$.

Это даёт возможность не прерывать вычислительные процессы в системе и продолжить работу по обеспечению процесса контроля над правильностью и безопасностью вычислений в системе M.

Другой вариант угрозы, когда компьютер злоумышленника формирует сообщения от имени $m_i \in M$. Для этого в информацию, передаваемую всему множеству M, злоумышленнику достаточно подставить в качестве номера компьютера, кото-

рый получил эти результаты, номер реального $p_i \in P$, номера $w \in W$ указать любой случайный номер из W и вместо правильного результата другой. Получив такое сообщение, m_d сравнивает указанные в сообщении номер w и номер p со своей информацией по вычислительной системе. Агент m_d проверяет, соответствует ли номер w части задания агенту m, номер которого указан в задании. Если номер в задании отсутствует, то агент m_d отправляется сообщение о фальсификации данных и уничтожает у себя полученные фальсифицированные данные.

Оценка ускорения выполнения задачи многоагентной системой. Для оценки ускорения решения задачи многоагентной системой, работающей по разработанному алгоритму, была выбрана задача построения связывающих деревьев цепей. Для решения задач большой размерности при проектировании изделий вычислительной техники наиболее эффективным является использование технологии распараллеливания алгоритмов [20, 21].

В качестве алгоритма решения этой задачи выбран алгоритм построения связывающих деревьев цепей $\{l_1, l_2, \quad , l_k\} \in L$ на многопроцессорной [22].

Конфигурация системы представляет собой граф компьютерной сети P с логической структурой информационных потоков между узлами, имеющий решетчатую структуру. Каждый узел $p_{i,j}$, где $\{p_1, p_2, \quad , p_n\} \in P$ обменивается информацией только с узлами $\{p_{i-1,j}, p_{i+1,j}, p_{i,j-1}, p_{i+1,j+1}\} \in P$, с которыми он связан по структуре. Физически это может быть всего одна линия связи, используемая всеми узлами сети.

Перед началом процесса трассировки все множество дискретов Z рабочего поля разбивается на множество частей $\{z_1, z_2, \quad , z_n\} \in Z$. Между соседними частями определяются граничные линии, состоящие из множества дискретов d_i. Множество d_i является общим $d_i \in z_i$ и $d_i \in z_{i+1}$ для двух частей рабочего поля и оказывается принадлежащим узлам p_i и p_{i+1}.

При выполнении построения связывающих деревьев цепей узел p_i работает независимо от других. Если контакты («источники» и «цели») расположены в одной части рабочего поля вычислительного узла z_i, процесс построения связи с большой вероятностью будет реализовываться автономно вычислительным узлом p_i.

Оценим ускорение выполнение процесса построения всех связей L при изменении количества компьютеров K соединенных в сеть и используемых для реализации распределенных вычислений.

В общем виде время решения задачи равно

$$T_{\text{р}} = t_{\text{д}} * d_i + t_{\text{перед}}, \quad (1)$$

где $t_{\text{д}}$ – время увеличения длины связи на один дискрет; d_i – суммарное количество дискретов во всех связях строящихся на i-ом компьютере; $t_{перед}$ – время обмена информацией между соседними компьютерами.

Для простоты будем считать, что производительность компьютеров в сети приблизительно одинакова, поэтому время $t_{\text{д}}$ будет одинаковым для всех компьютеров сети.

Время $t_{перед}$ определяется количеством передач d_i, выполняемых при построении связей i-м компьютером:

$$t_{перед} = d_i * t_{обм}. \quad (2)$$

Само число d_i равно количеству дискретов рабочего поля, которые являются общими у пары соседних компьютеров. Поскольку части дискретного рабочего поля у всех компьютеров примерно одинаковы, то можно записать:

$$d_i = 2*(a_i + b_i), \tag{3}$$

где a_i и b_i – размеры дискретного поля p_i-го компьютера.

Если размеры общего рабочего дискретного поля $A*B$, то

$$a_i = \frac{A}{\sqrt{K}};\ b_i = \frac{B}{\sqrt{K}}. \tag{4}$$

Подставляя значения в формулу (1), найдем суммарное количество дискретов во всех связях, строящихся на i-ом компьютере:

$$d_i = \frac{\left(\frac{A+B}{\sqrt{K}}+4\right)\cdot L}{2\cdot K}. \tag{5}$$

Тогда время решения задачи построения T_p будет равно:

$$T_p = \frac{\left(\frac{A+B}{\sqrt{K}}+4\right)\cdot L}{2\cdot K}\cdot\left(\frac{A+B}{\sqrt{K}}\right)\cdot 2\cdot t_{\text{д}}\cdot t_{\text{обм}}. \tag{6}$$

Используя полученную формулу (6), построим графики зависимости общего времени решения задачи построения связей от количества задействованных сетевых компьютеров (рис. 3, 4), используемых для организации распределенных вычислений.

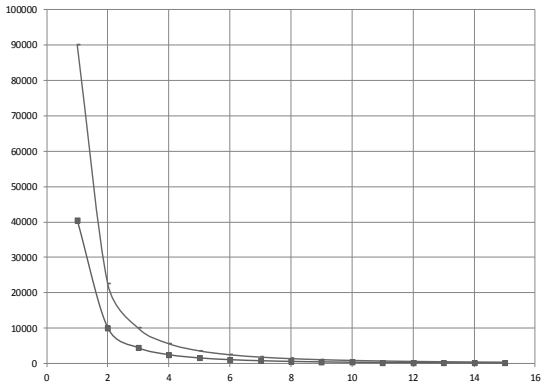

Рис. 3. Зависимость времени решения задачи от количества компьютеров для L=1000 и L=600 при A и B, равными 100 и 200 дискретов соответственно

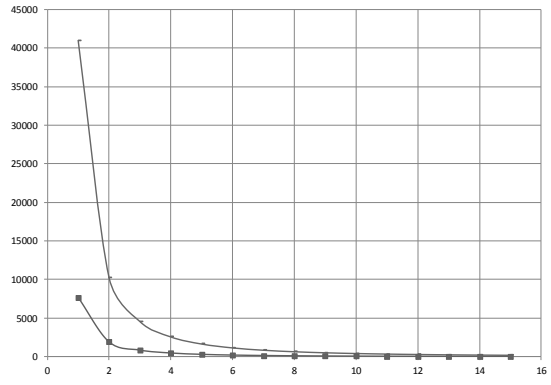

Рис. 4. Зависимость времени решения задачи от количества компьютеров для L=4000 и L=1000 при A и B, равными 200 и 3000 дискретов соответственно

Заключение. Достоинствами разработанного алгоритма агентов многоагентной системы является способность системы самостоятельно организовывать использование вычислительных мощностей компьютеров любой уже существующей сети для решения большеобъёмных задач путём создания распределенных вычислений. Агенты на основе компьютерной сети в процессе решения задачи могут: конфигурировать распределенную вычислительную систему, распределить вычислительную нагрузку между компьютерами управляемыми агентами, выполнить оптимизацию нагрузки распределенной вычислительной системы в зависимости от вычислительной мощности компьютеров сети. С целью сокращения времени решения все эти функции выполняются на фоне решения задачи без предварительных этапов подготовки к решению и прерываний вычислительного процесса. Количество компьютеров, соединенных в сеть, может увеличиваться за счет подключения новых компьютеров к многоагентной системе, что приводит к возрастанию общей вычислительной мощности. Все это позволяет уменьшить время решения задачи и повысить отказоустойчивость (живучесть) вычислительных процессов в условиях изменяемой вычислительной среды (динамическое изменение количество компьютеров в сети).

Оценка с помощью математической модели ускорения вычислительного процесса организованного по разработанному алгоритму при изменении количества компьютеров показала, что при увеличении количества компьютеров общее время решения большеобъёмной задачи уменьшается. Эффективность использования распределенных вычислений на основе многоагентной системы повышается с увеличением объема решаемой задачи. Однако для решения задачи с заданным объемом вычислений использование больше определенного количества компьютеров приводит к снижению эффективности распределенной системы. Основной причиной снижения показателя сокращения времени решения при увеличении количества задействованных в системе компьютеров является увеличение количества обменов между ними.

Достоинством такого подхода является возможность повышения степени безопасности распределённых вычислений за счет добавления в многоагентную систему специального агента для организации рассылки результатов расчетов W. Безопасность распределенных вычислений обеспечивается даже при организации системы на компьютерах глобальной сети. В этом случае многоагентная система позволяет обнаруживать факты фальсификации результатов работы распределенной системы, которые могут привести к принятию неправильных решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Kshemkalyani A.D., Singhal M.* Distributed Computing: Principles. Algorithms, and Systems. Cambridge University Press, 2008.
2. *Műller J., Fisher K.* Application Impact of Multiagent Systems and Technologies: A Survey // In Agent-Oriented Software Engineering book series. – Springer, 2013. – P. 1-26.
3. *Wooldridge M.* An introduction to multiagent systems. – New Jersey: Wiley, 2012. – 484 p.
4. *Котенко В.В., Румянцев К.Е., Котенко С.В.* Идентификационный анализ в информационно-телекоммутационных системах: монография. – Ростов-на-Дону: Изд-во ЮФУ, 2014.
5. *Kaminka Gal.* Robots Are Agents, Too! // Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007), Honolulu, Hawaii, May, 2007.
6. *Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U.* Securing Mobile-Agent-Based Systems Against Malicious Hosts // World Applied Sciences Journal. – 2014. – Vol. 29 (2). – P. 287-297.
7. *Muñoz A., Pablo A., Maña A Multiagent* Systems Protection // Advances in Software Engineering. – 2011. – Article ID 281517. – 9 p. Doi: 10.1155/2011/281517.
8. *Beydoun G. Low G. Mouratidis H. and Hendersonsellers B.* A security-Aware Metamodel for MultiAgent System (MAS) // Information and software technology. – 2009. – Vol. 51, No. 5. – P. 832-845.
9. *Poslad S, Calisti M, Charlton P.* Specifying Standard Security Mechanisms in Multi-Agent Systems // Proc. workshop on Deception, Fraud and Trust, Bologna, Italy. – http://citeseerx.ist.psu.edu/viewdoc/download?doi =10.1.1.203.878&rep=rep1.

10. *Alfalayleh M. Brankovic L.* An overview of security issues and techniques in mobile agents // 8th IFIP TC-6 TC-11, Salford, United kingdom. – 2004. – P. 59-78.
11. *Borselius N., Holloway R.* Security in Multi-Agent Systems. University of London. http://www.isrc.rhul.ac.uk/nb/publications/security-in-MAS(SAM02).pdf.
12. *Chadha Zrari, Hela Hachicha, Khaled Ghedira*. Agent's security during communication in mobile agents system // 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems Procedia Computer Science. – 2015. – No. 60. – P. 17-26.
13. *Хованcков С.А., Литвиненко В.А., Норкин О.Р.* Оптимизации распределенных вычислений на базе алгоритма реконфигурирования и продукций предметной области // Труды конгресса по интеллектуальным системам и информационным технологиям "AIS-IT'09". – М.: Физматлит, 2009. – Т. 2. − С. 153-158.
14. *Хованcков С.А., Литвиненко В.А.* Оптимизация решения задачи в распределенных системах // Известия ТРТУ. – 2005. – № 3 (47). – С. 209.
15. *Хованcков С.А., Литвиненко В.А.* Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
16. *Khovanskova V., Khovanskov S.* Мультиагентные системы: концепции защиты, Безопасность мультиагентных систем // Technical and natural sciences: Theory and practice: Proceedings of materials of international scientific e-Symposium. Russia, Moscow, 27-28 March 2015. – Kirov, 2015. – P. 167-175.
17. *Хованcков С.А., Хованcкова В.С.* Безопасность мультиагентных систем // Вопросы науки: Естественно-научные исследования и технический прогресс: Сборник статей по материалам III Международной научно-практической конференции (26 февраля 2015 г. Воронеж). – Воронеж, 2015. – Т. 2. – С. 83-87.
18. *Хованcков С.А., Хованcкова В.С.* Методы защиты распределенных вычислений // Модернизация современного общества: Проблемы, пути развития и перспективы: Сборник материалов VI Международной научно-практической конференции. – Ставрополь: Логос, 2015. – С. 104-107.
19. *Хованcков С.А., Хованcкова В.С.* Повышение степени защиты распределенных вычислений // Современное состояние естественных и технических наук: Материалы XVIII Международной научно-практической конференции (20.03.2015). – М.: Изд-во «Спутник +», 2015. – С. 96-100.
20. *Kureichik V.V., Kureichik V.M., Sorokoletov P.V.* Analysis and a survey of evolutionary models // Journal of Computer and Systems Sciences International. – 2007. – Vol. 46, No. 5. – P. 779-791.
21. *Курейчик В.В., Курейчик В.В.* Архитектура гибридного поиска при проектировании // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 22-27.
22. *Хованcков С.А., Литвиненко В.А., Норкин О.Р.* Организация распределенных вычислений для решения задач трассировки // Известия ЮФУ. Технические науки. –2010. – № 12 (113). – С. 48-55.

REFERENCES

1. *Kshemkalyani A.D., Singhal M.* Distributed Computing: Principles. Algorithms, and Systems. Cambridge University Press, 2008.
2. *Műller J., Fisher K.* Application Impact of Multiagent Systems and Technologies: A Survey, *In Agent-Oriented Software Engineering book series*. Springer, 2013, pp. 1-26.
3. *Wooldridge M.* An introduction to multiagent systems. New Jersey: Wiley, 2012, 484 p.
4. *Kotenko V.V., Rumyantsev K.E., Kotenko S.V.* Identifikatsionnyy analiz v informatsionno-telekommutatsionnykh sistemakh: monografiya [Identification analysis in information-telecommunications systems: monograph]. Rostov-on-Don: Izd-vo YuFU, 2014.
5. *Kaminka Gal.* Robots Are Agents, Too!, *Keynote Lecture. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007), Honolulu, Hawaii, May, 2007.*
6. *Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U.* Securing Mobile-Agent-Based Systems Against Malicious Hosts, *World Applied Sciences Journal,* 2014, Vol. 29 (2), pp. 287-297.
7. *Muñoz A., Pablo A., Maña A Multiagent* Systems Protection, *Advances in Software Engineering,* 2011. – Article ID 281517. – 9 p. Doi: 10.1155/2011/281517.

8. *Beydoun G. Low G. Mouratidis H. and Hendersonsellers B.* A security-Aware Metamodel for MultiAgent System (MAS), *Information and software technology,* 2009, Vol. 51, No. 5, pp. 832-845.
9. *Poslad S, Calisti M, Charlton P.* Specifying Standard Security Mechanisms in Multi-Agent Systems, *Proc. workshop on Deception, Fraud and Trust, Bologna, Italy*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi =10.1.1.203.878&rep=rep1.
10. *Alfalayleh M. Brankovic L.* An overview of security issues and techniques in mobile agents, *8th IFIP TC-6 TC-11, Salford, United kingdom,* 2004, pp. 59-78.
11. *Borselius N., Holloway R.* Security in Multi-Agent Systems. University of London. Available at: http://www.isrc.rhul.ac.uk/nb/publications/security-in-MAS(SAM02).pdf.
12. *Chadha Zrari, Hela Hachicha, Khaled Ghedira*. Agent's security during communication in mobile agents system, *19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems Procedia Computer Science,* 2015, No. 60, pp. 17-26.
13. *Khovanskov S.A., Litvinenko V.A., Norkin O.R.* Optimizatsii raspredelennykh vychisleniy na baze algoritma rekonfigurirovaniya i produktsiy predmetnoy oblasti [Optimization of distributed computing based on the reconfiguration algorithm and products subject area], *Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam* [Proceedings of Congress on intelligent systems and information technologies "AIS-IT'09"]. Moscow: Fizmatlit, 2009, Vol. 2, pp. 153-158.
14. *Khovanskov S.A., Litvinenko V.A.* Optimizatsiya resheniya zadachi v raspredelennykh sistemakh [The optimization problem solving in distributed systems], *Izvestiya TRTU* [Izvestiya TSURE], 2005, No. 3 (47), pp. 209.
15. *Khovanskov S.A., Litvinenko V.A.* Resheniya zadach putem organizatsii raspredelennykh vychisleniy v seti [The organization of fulfilling tasks by the method of a cooperative decisions making], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 3 (80), pp. 16-21.
16. *Khovanskova V., Khovanskov S.* Мультиагентные системы: концепции защиты, Безопасность мультиагентных систем, *Technical and natural sciences: Theory and practice: Proceedings of materials of international scientific e-Symposium. Russia, Moscow, 27-28 March 2015*. Kirov, 2015, pp. 167-175.
17. *Khovanskov S.A., Khovanskova V.S.* Bezopasnost' mul'tiagentnykh sistem [Security of multiagent systems], *Voprosy nauki: Estestvenno-nauchnye issledovaniya i tekhnicheskiy progress: Sbornik statey po materialam III Mezhdunarodnoy nauchno-prakticheskoy konferentsii (26 fevralya 2015 g. Voronezh)* [Problems of science: Natural science research and technical progress: proceedings of the III International scientific-practical conference (26 Feb 2015 Voronezh)]. Voronezh, 2015, Vol. 2, pp. 83-87.
18. *Khovanskov S.A., Khovanskova V.S.* Metody zashchity raspredelennykh vychisleniy [Protection methods in distributed computing], *Modernizatsiya sovremennogo obshchestva: Problemy, puti razvitiya i perspektivy: Sbornik materialov VI Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Modernization of the modern society: Problems, ways of development and prospects: proceedings of the VI International scientific-practical conference]. Stavropol': Logos, 2015, pp. 104-107.
19. *Khovanskov S.A., Khovanskova V.S.* Povyshenie stepeni zashchity raspredelennykh vychisleniy [Increasing security degree of distributed computing], *Sovremennoe sostoyanie estestvennykh i tekhnicheskikh nauk: Materialy XVIII Mezhdunarodnoy nauchno-prakticheskoy konferentsii (20.03.2015)* [The modern state of natural and technical Sciences: Materials of XVIII International scientific and practical conference (20.03.2015)]. Moscow: Izd-vo «Sputnik +», 2015, pp. 96-100.
20. *Kureichik V.V., Kureichik V.M., Sorokoletov P.V.* Analysis and a survey of evolutionary models, *Journal of Computer and Systems Sciences International,* 2007, Vol. 46, No. 5, pp. 779-791.
21. *Kureychik V.V., Kureychik V.V.* Arkhitektura gibridnogo poiska pri proektirovanii [The architecture of hybrid search for design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 22-27.
22. *Khovanskov S.A., Litvinenko V.A., Norkin O.R.* Organizatsiya raspredelennykh vychisleniy dlya resheniya zadach trassirovki [The organization of the distributed calculations for the decision of problems of trace], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 48-55.

Статью рекомендовала к опубликованию к.т.н., доцент Е.А. Ищукова.

Хованcков Сергей Андреевич – Южный федеральный университет; e-mail: sah59@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 8634371902; кафедра информационной безопасности телекоммуникационных систем; к.т.н.; доцент.

Хованскова Вера Сергеевна – e-mail: v.s.khovanskova@gmail.com; кафедра информационной безопасности телекоммуникационных систем; аспирантка.

Литвиненко Василий Афанасьевич – e-mail: litv_va@mail.ru; 347928, г. Таганрог, ул. Энгельса, 1; тел.: 88634377651; кафедра систем автоматизированного проектирования; к.т.н.; доцент.

Khovanskov Sergey Andreevich – Southern Federal University; e-mail: sah59@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634371902; the department of information security of telecommunication systems; cand. of eng. sc.; associate professor.

Khovanskova Vera Sergeevna – e-mail: v.s.khovanskova@gmail.com; the department of information security of telecommunication systems; postgraduate student.

Litvinenko Vasily Afanas'evich – e-mail: litv_va@mail.ru; 1, Engels street, Taganrog, 347928, Russia; phone: +78634377651; the department of computer aided design; cand. of eng. sc.; associate professor.