

## Раздел V. Вычислительные системы и программирование

УДК 004.056.5

DOI 10.23683/2311-3103-2018-4-198-210

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

### ОРГАНИЗАЦИЯ И ЗАЩИТА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ МНОГОАГЕНТНОЙ СИСТЕМЫ В КОМПЬЮТЕРНОЙ СЕТИ С ЦЕЛЬЮ СОКРАЩЕНИЯ ВРЕМЕНИ РЕШЕНИЯ МАСШТАБНЫХ ЗАДАЧ\*

*Для решения многовариантных задач, имеющих временные ограничения, часто используются спецвычислители. Однако это значительно повышает стоимость решения задачи и требует временных затрат для организации доступа к такой вычислительной среде. Из наиболее доступных и распространённых технологий сокращения времени решения масштабных задач в настоящее время является использование распределённых вычислений организованных в компьютерной сети. В настоящее время существует множество различных подходов для организации распределённых вычислений в компьютерной сети – технология grid, metacomputing (BOINC, PVM и другие). Основным недостатком большинства существующих подходов является то, что они предназначены для создания централизованных систем распределённых вычислений. Предлагается организовать решения такой задачи как многовариантное моделирование, путем создания распределённых вычислений в компьютерной сети на основе децентрализованной многоагентной (мультиагентной) системы. В качестве вычислительной среды выбрана обычная компьютерная сеть, являющаяся в большинстве случаев неустойчивой вычислительной средой. В качестве вычислительной системы предлагается самоорганизующаяся распределённая вычислительная система на основе децентрализованной многоагентной системы. Система представляет собой множество агентов выполняющих один и тот же алгоритм. Предлагается алгоритм агента децентрализованной многоагентной системы. Агенты работающие по этому алгоритму создают самоорганизующуюся распределённую вычислительную систему и обеспечивают защиту результатов вычислений от такой грозы как "отказ в обслуживании".*

*Распределённые вычисления; защита информации; вычислительный процесс; организация вычислений.*

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

### ORGANIZATION AND PROTECTION IN DISTRIBUTED COMPUTING BASED ON MULTI-AGENT SYSTEMS IN COMPUTER NETWORKS TO REDUCE THE LARGE-SCALE PROBLEMS SOLUTION TIME

*Special calculators are often used to solve multi-variant problems with time constraints. However, this significantly increases the cost of solving the problem and requires time to organize access to such a computing environment. Among the most accessible and common technologies to reduce the time of solving large-scale problems is the use of distributed computing organized in a computer network. Currently, there are many different approaches to the organization of distributed computing in a computer network-grid technology, metacomputing (BOINC, PVM and others). The main disadvantage of most existing approaches is that they are designed to create centralized systems of distributed computing. In this article it is proposed to organize solutions of*

\* Работа выполнена при поддержке РФФИ (проект № 18-01-00041).

*such problem as multivariate modeling by creating distributed computing in a computer network based on a decentralized multi-agent system. As a computing environment, a conventional computer network is chosen, which in most cases is an unstable computing environment. As a computing system, a self-organizing distributed computing system based on a decentralized multi-agent system is proposed. The system is a set of agents that perform the same algorithm. The article offers the agent algorithm of a decentralized multi-agent system. Agents working on this algorithm create a self-organizing distributed computing system and protect the results of calculations from such a thunderstorm as "denial of service".*

*Distributed computing; information security; computing process; organization of calculations.*

**Введение.** Для решения многовариантных задач, имеющих временные ограничения, часто используются спецвычислители. Однако это значительно повышает стоимость решения задачи и требует временных затрат для организации доступа к такой вычислительной среде.

Из наиболее доступных и распространённых технологий сокращения времени решения сложных многовариантных задач [1–4] в настоящее время является использование распределенных вычислений.

Для реализации таких вычислений используются различные вычислительные среды. В качестве вычислительной среды могут использоваться многопроцессорная вычислительная машина, кластерная вычислительная система, многомашинная вычислительная система, либо обычная компьютерная сеть. Наиболее доступной вычислительной средой, на которой возможно выполнять распределенные вычисления, является компьютерная сеть, обладающая достаточным или избыточным количеством центров обработки данных (локальная, глобальная сети).

В настоящее время существует множество различных подходов для организации распределенных вычислений в компьютерной сети – технология grid, метакомпьютинг (боинг, PVM и другие). Основным недостатком большинства существующих подходов является то, что они предназначены для создания централизованных систем распределенных вычислений. В глобальной компьютерной сети существует реальная угроза нарушение работоспособности распределенных вычислительных процессов из-за крайней нестабильности вычислительной среды и действий злоумышленников. Для снижения угроз существованию распределенных вычислений и безопасности полученных результатов предлагается использовать, для решения большеобъемных задач, самоорганизующуюся систему распределенных вычислений на основе децентрализованной многоагентной системы [5, 6] (рис. 1).

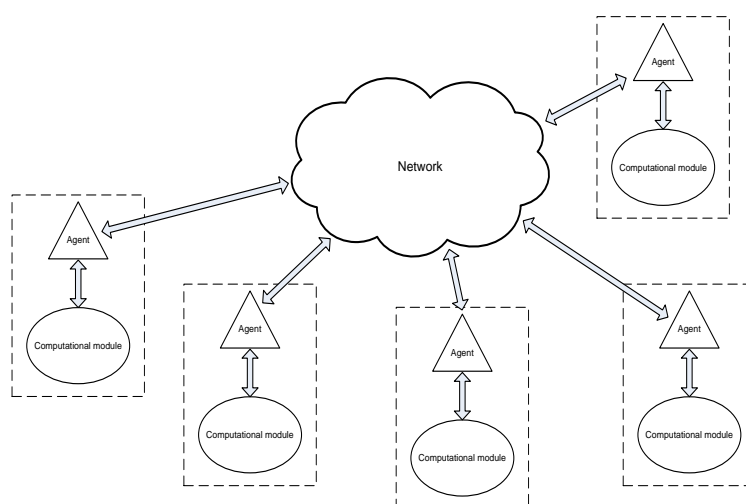


Рис. 1. Структура многоагентной системы

Под многоагентной системой понимается набор агентов, каждый из которых представляет собой программный модуль и помещается на отдельном компьютере. Агент выполняет управление только своим компьютером и поэтому его работа является независимой. Он организует решение задач на своем компьютере, инициирует обмен данными с компьютерами других агентов, выполняет обработку полученной от них информации и на ее основе принимает решения.

Децентрализованная многоагентная система представляет собой множество сетевых компьютеров. Каждый компьютер находится под управлением своего агента. Все агенты выполняют один и тот же алгоритм работы. В результате получается одноранговая вычислительная система. Каждый агент работает независимо от других агентов, обмен между агентами осуществляется с помощью широкополосных сообщений. Это позволяет в процессе выполнения распределенных вычислений масштабировать многоагентную систему, не нарушая работоспособности процессов вычислений.

Для реализации и защиты распределенных вычислений в компьютерных сетях разработан алгоритм для агентов, многоагентной системы. Алгоритм позволяет организовать распределенную вычислительную систему на основе узлов компьютерной сети [7–11].

**Реализация многоагентной системы.** Система должна быть децентрализованной – каждый агент должен обладать равными правами и иметь возможность обмениваться сообщениями с другими агентами

Сформулируем требования к алгоритму работы агента:

- ◆ агент должен следить за вычислительными процессами, выполняемыми на управляемом им компьютере;
- ◆ агенты должны самостоятельно распределять между собой вычислительную нагрузку для организации распределенных вычислений;
- ◆ агенты должны выполнять перераспределение своей вычислительной нагрузки в зависимости от производительности своих компьютеров;
- ◆ каждый агент должен хранить все результаты выполнения большеобъемной задачи;
- ◆ многоагентная система должна обеспечивать защиту распределенных вычислений от угроз со стороны злоумышленников.

Для организации распределенных вычислений в компьютерной сети и реализации требований был разработан алгоритм содержащий *ряд правил*, которые должен выполнять каждый агент.

Многоагентная система представляет собой множество агентов  $M$  в виде одинаковых программных модулей агентов  $\{m_1, m_2, \dots, m_n\} \in M$ . Множество  $M$  накладывается на множество  $\{p_1, p_2, \dots, p_j\} \in P$  сетевых компьютеров ( $P > M$ ) так, что агент  $m_i$  располагается на соответствующем  $p_i$  компьютере сети. Каждый модуль агента  $m_i \in M$  (агент) управляет ресурсами компьютера  $p_i$  и следит за выполняемой на нем нагрузке  $W_i$ . Вся многоагентная система  $M$ , управляемая компьютерами  $\{p_1, p_2, \dots, p_n\} \in P$ , организует систему распределенных вычислений для решения всего множества заданий  $\{w_1, w_2, \dots, w_n\} \in W$ . Множество  $M$  является одноранговым набором агентов работающих по одной программе.

В начале организации распределенных вычислений в компьютерной сети  $P$  на  $\{p_1, p_2, \dots, p_n\} \in P$  находятся управляющие их работой агенты  $\{m_1, m_2, \dots, m_n\} \in M$ . 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. Для отслеживания процесса выполнения вычислительной на-

грузки каждый агент для работы имеет две таблицы первая таблица  $W_{nrez}$  включает в себя все невыполненные задания вторая таблица  $W_{rez}$  включает выполненные задания с результатами выполнения  $\{W_{nrez}, W_{rez}\} \in W$ .

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  $p_i$  вычислительный процесс для выполнения  $w_i$ , выполняя действия в соответствии с правилом выполнения вычислительной нагрузки.

**Алгоритм агента для организации распределённых вычислений и обеспечения безопасности результатов вычислительных процессов.** Каждый агент многоагентной системы располагается на своём сетевом компьютере. Каждый компьютер представляет собой автономную вычислительную систему, работа которой не зависит от других компьютеров сети.

Агент  $m_i$  следит за состоянием компьютера  $p_i$  и управляет его работой в соответствии с правилом «выполнение вычислительной нагрузки». Если  $p_i$  не выполняет вычислений, то агент  $m_i$  выбирает из списка своей вычислительной нагрузки  $W_i$  следующее по порядку задание и передаёт его для выполнения вычислительному блоку компьютера  $p_i$ . Выбор ведется путем последовательного просмотра агентом списка вычислительной нагрузки  $W_i \in W_{nrez}$ .

Алгоритм действий агента  $m_i$  по правилу «выполнение вычислительной нагрузки».

1°. Агент  $m_i \in M$  проверяет выполнена ли очередная задача  $w_j \in W_i$ ? Если нет то к пункту 7°.

2°. Агент  $m_i$  получает результат выполнения задачи  $w_j \in W_i$ .

3°. Агент  $m_i$  просматривает список невыполненных заданий  $W_{nrez}$ .

4°. Нагрузка  $W_i$  выполнена полностью  $W_{nrez} = 0$ ? Если да переход к пункту 7°.

5°. Выбор агентом  $m_i$  следующего по порядку задания  $w_{j+1}$  из списка невыполненной вычислительной нагрузки  $W_{nrez}$ .

6°. Передача выбранного задания  $w_{j+1}$  на выполнение компьютеру  $p_i \in P_z$ .

7°. Переход к выполнению другого правила.

Благодаря правилу «выполнение вычислительной нагрузки» каждый вычислительный узел  $p_i$  непрерывно выполняет вычислительную нагрузку  $W_i$ . Процесс выполняется полностью под контролем агента  $m_i$ . Это позволяет оптимально использовать вычислительные ресурсы каждого компьютера.

Для реализации взаимодействия между агентами  $\{m_1, m_2, \dots, m_n\} \in M$  в процессе выполнения распределённых вычислений агенты обмениваются между собой сообщениями с результатами. Обмен между агентами происходит на фоне выполнения вычислительной нагрузки компьютерами, которыми управляют агенты  $\{m_1, m_2, \dots, m_n\} \in M$ . Агент  $m_i$ , получив результат от другого агента записывает  $w_j \in W$  в свою таблицу результатов выполнения общей вычислительной нагрузки. В конце работы у каждого агента хранятся все результаты решений заданий  $W$ .

Алгоритм действий агента  $m_i$  по правилу «передача полученного результата другим агентам».

1°. Проверка есть ли переданный результат выполнения своей вычислительной нагрузки  $w_j \in W_i$ ? Если да то переход к пункту 2°, если нет то к пункту 5°.

2°. Сформировать пакет для передачи информации агентам  $\{m_1, m_2, \dots, m_n\} \in M$ .

3°. Свободна среда передачи данных? Если да то переход к пункту 4°, если нет к пункту 5°.

4°. Передать пакет с информацией о результате вычислений  $w_j \in W_i$  сразу всем агентам  $\{m_1, m_2, \dots, m_n\} \in M$ .

5°. Перейти к выбору правила поведения.

Время выполнения всей вычислительной нагрузки  $W$  многоагентной системой  $M$  равно времени выполнения средней нагрузки  $W_i$  агентом  $m_i \in M$ .

Неполное или медленное выполнение вычислительной нагрузки агентами системы распределенных вычислений может быть вызвано не только низким быстродействием отдельных компьютеров, но и следствием реализация угрозы типа "отказ в обслуживании". Атака типа "отказ в обслуживании" может привести к нарушению работоспособности некоторых вычислительных узлов и как следствие этого – прекращение функционирования самой системы распределенных вычислений. Для обеспечения защиты процессов распределённых вычислений от этой угрозы каждый агент выполняет действия по правилу слежения за полнотой выполнения общей вычислительной нагрузки  $W$ .

Агент  $m_i$  выполняет слежения за полнотой выполнения  $W$  при каждом получении результатов как от агентов  $m_j \in M$  так и от компьютера  $p_i$  и записью получаемых результатов в список.  $W_{rez}$ . Если вся нагрузка  $W_i$  выполнена в полном объеме то агент просматривает и выбирает из  $W_{nrez}$ . Задание и и передает его на выполнение  $p_i$ .

Алгоритм действий агента  $m_i$  по правилу слежения за полнотой выполнения общей вычислительной нагрузки.

1°. Агент  $m_i \in M$  проверяет все ли задания входящие в его вычислительную нагрузку выполнены  $W_{nrez} = 0$ ? Если да то переход к пункту 5°, если нет к пункту 2°.

2°. Агент  $m_i$  выбирает из таблицы общей вычислительной нагрузки  $W_{nrez}$  задание  $w_j$  по которому не получен результат.

3°. Агент  $m_i$  проверяет выполнял он  $w_j$  раньше? Если да то переход к пункту 5°, если нет то переход к пункту 4°.

4°. Агент  $m_i$  передает выбранное задание на выполнение своему компьютеру  $p_i \in P_z$ .

5°. Переход к выполнению следующего правила.

Графическое представление алгоритма на рис. 2.

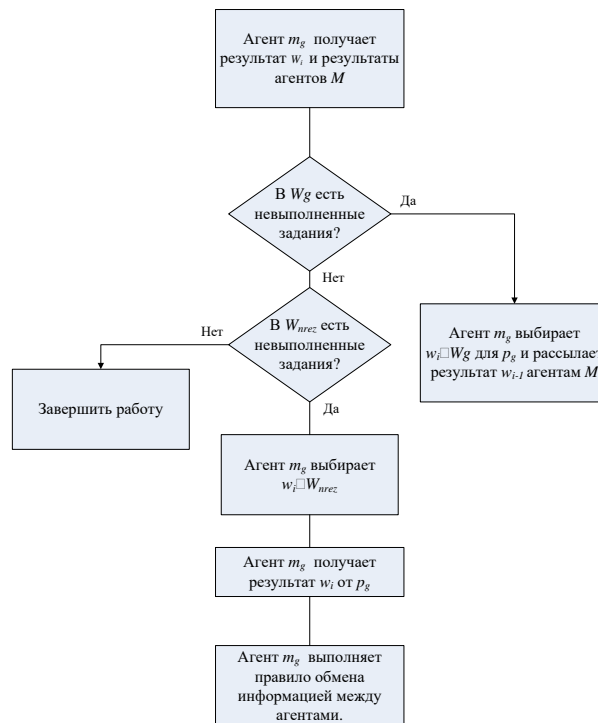


Рис. 2. Структура программы агента многоагентной системы

Благодаря выполнению правила слежения за полнотой выполнения общей вычислительной нагрузки, происходит перераспределение нагрузки между агентами многоагентной системы. При атаке "отказ в обслуживании" и выходе из строя одного или нескольких агентов нагрузка перераспределяется между оставшимися работоспособными вычислительными узлами многоагентной системы. Это обеспечивает высокую отказоустойчивость системы распределенных вычислений созданной на основе многоагентной системы в компьютерной сети.

На основе разработанного алгоритма была написана на языке Python и отлажена программа работы агента многоагентной системы. Структура программы приведена на рис. 3.

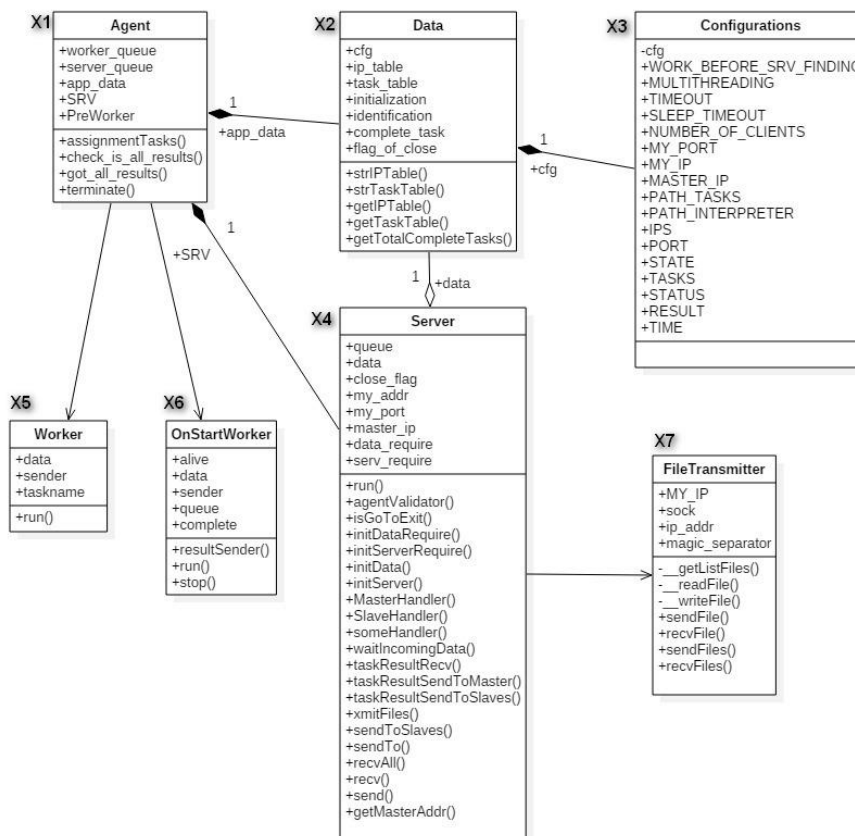


Рис. 3. Структура программы агента многоагентной системы

**Оценка обнаружения ложного результата в централизованной многоагентной системе.** Каждый агент многоагентной системы, выполняя разработанный алгоритм организации распределенных вычислений, позволяет расширять систему за счёт включения в неё свободных компьютеров. Для этого на свободный компьютер передается модуль агента и его вычислительная нагрузка. Масштабирование многоагентной системы уменьшает вычислительную нагрузку на каждого агента и приводит к сокращению времени выполнения большеобъемной задачи. Модуль агента может располагаться на любом сетевом компьютере, в том числе, и на компьютерах глобальной сети Интернет. Это увеличивает степень угрозы безопасности процессов и результатов распределенных вычислений. Для защиты рас-

пределенных вычислений от угрозы получения ложного результата управляющие агенты проверяют правильность результатов получаемых от агентов многоагентной системы.

Рассчитаем на конкретном примере вероятность обнаружения ложного результата для централизованной многоагентной системы. Расчёт выполним для многоагентной системы, состоящей из тысячи агентов, выполняющих вычисления, и одного управляющего агента. Допустим, что среди множества агентов, составляющих многоагентную систему, есть злоумышленники, каждый из которых передаёт ложные результаты вычислений. Управляющий агент из результатов, полученных от агентов, выбирает некоторые и проверяет их правильность путем повторения вычислений. Проверяемые результаты выбираются случайным образом, поскольку управляющий агент, обладая ограниченными вычислительными ресурсами, точно не знает, какой из агентов является злоумышленником.

Вероятность  $P_{olr}$  обнаружения одного ложного результата в централизованной многоагентной системе, который постоянно формирует злоумышленник, определяется по формуле Бернулли. Формула Бернулли позволяет определять вероятность появления определенного события при независимых условиях. Это говорит о том, что появление события в эксперименте не зависит от появления или не появления того же события в ранее проведенных или последующих испытаниях.

$$P_n(m) = \frac{n!}{m!(n-m)!} * p^m * (1-p)^{n-m}, \quad (1)$$

где  $m$  – количество раз наступление события;  $p$  – вероятность, что событие произойдет;  $n$  – количество повторений эксперимента.

Для нашего случая количество повторений эксперимента  $n$  – это средняя вычислительная нагрузка  $kr$  на каждого агента. Она зависит от общего объёма вычислительной нагрузки  $W$  и количества агентов  $N$  в многоагентной системе  $M$ . Нагрузка  $kr$  рассчитывается по формуле

$$n = kr = \frac{W}{N}, \quad (2)$$

где  $N$  – количество агентов в многоагентной системе  $M$ ;  $W$  – общий объём вычислительной нагрузки.

Вероятность наступления события в одном эксперименте определяется количеством агентов многоагентной системы:

$$p = \frac{1}{N}; \quad (3)$$

Для расчёта вероятности обнаружения ложного результата подставим в формулу (1) наши данные из формул (2) и (3).

$$P_{olr} = \frac{\left(\frac{W}{N}\right)!}{m! * \left(\frac{W}{N} - m\right)!} * \left(\frac{1}{N}\right)^m * \left(1 - \frac{1}{N}\right)^{\frac{W}{N} - m}, \quad (4)$$

где  $m$  – количество обнаруженных ложных результатов.

Поскольку злоумышленник может формировать ложные результаты для всей своей вычислительной нагрузки, то вероятность формирования им ложного результата  $ko = 1$ . Если вероятность формирования ложного результата  $ko = 0,5$ , это означает, что он формирует ложные результаты только для половины своей вычислительной нагрузки.

В соответствии со структурой многоагентной системы, для расчёта вероятности обнаружения ложного результата используется формула (5), отражающая зависимость вероятности от количества управляющих агентов и вероятности формирования ложных результатов злоумышленником:

$$p = \frac{b}{N} * ko; \quad (5)$$

где  $ko$  – вероятность формирования ложных результатов злоумышленником;  $b$  – количество управляющих агентов.

Подставляя в формулу (4) вместо вероятности  $p$ , рассчитываемое по формуле (3), значение  $p$ , рассчитываемое по формуле (5) сформируем графики вероятности обнаружения управляющим агентом ( $b = 1$ ), хотя бы одного ложного результата ( $m=1$ ) из передаваемых злоумышленником. Вероятности обнаружения хотя бы одного ложного результата в централизованной многоагентной системе которые формируются злоумышленником с вероятностями  $ko = 1$ ,  $ko = 0.8$  и  $ko = 0.6$  (рис. 4).

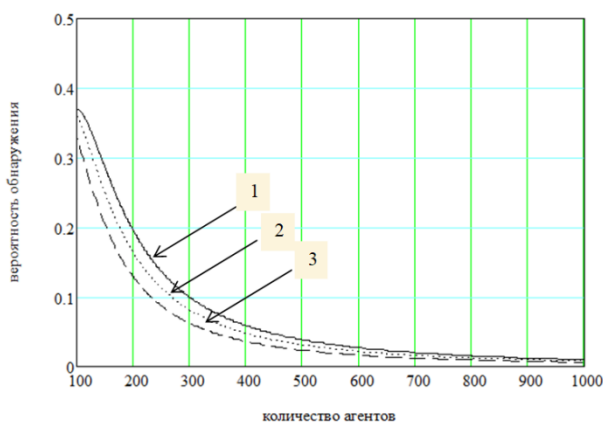


Рис. 4. Вероятности обнаружения хотя бы одного ложного результата при вероятности формирования «ложного результата»: 1)  $ko = 1$ ; 2)  $ko = 0.8$ ; 3)  $ko = 0.6$

Расчёт выполнен для централизованной многоагентной системы с одним управляющим агентом. Количество заданий большеобъемной задачи  $W=10000$  при различном количестве агентов  $N$  многоагентной системы от 100 до 1000. При анализе полученных графиков видно, что вероятность обнаружения хотя бы одного ложного результата управляющим агентом снижается с увеличением количества агентов системы. При количестве агентов  $N=1000$  вероятность обнаружения хотя бы одного ложного результата составляет  $P_{olr} = 0,005$ . А при  $N \rightarrow \infty$  вероятность  $P_{olr} \rightarrow 0$ . Это связано с тем, что при масштабировании многоагентной системы количество результатов, получаемых управляющим агентом, возрастает. Управляющий агент не в состоянии проверить все результаты, передаваемые ему агентами многоагентной системы  $M$ .

На снижение вероятности обнаружения ложных результатов также влияет снижение вероятности формирования ложных результатов злоумышленником, поскольку в этом случае общее количество ложных результатов в многоагентной системе уменьшается, что приводит к снижению вероятности обнаружения ложных результатов управляющим агентом.



Аналогично рассчитывается вероятность обнаружения управляющим агентом хотя бы 2 ложных результатов при одном злоумышленнике, аналогично 3 и 4 ( $P_{olr2}$ ,  $P_{olr3}$ ,  $P_{olr4}$ ). В формулу (4) подставляем  $m=2,3,4$ .

Рассчитаны вероятности  $P_{olr2}$ ,  $P_{olr3}$ ,  $P_{olr4}$  обнаружения управляющим агентом ложных результатов при одном злоумышленнике. Вероятности обнаружения 2, 3 и 4 ложных результатов в централизованной многоагентной системе из тех которые формируются злоумышленником с вероятностями  $ko = 1$ , показаны на рис. 5.

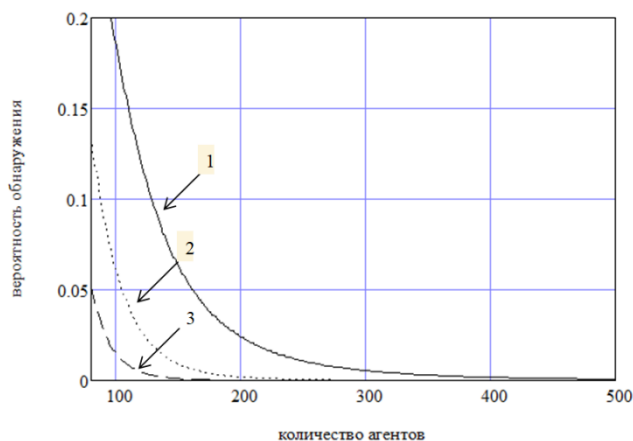


Рис. 5. Вероятности обнаружения с  $ko = 1$ : 1) 2х; 2) 3х; 3) 4х ложных результатов

Сравнив графики на рис. 4 и 5, можно сделать вывод, что в централизованной многоагентной системе вероятность обнаружения ложных результатов снижается с увеличением количества ложных результатов, которые должен обнаружить управляющий агент.

**Выводы.** Был разработан алгоритм для организации и защиты распределенных вычислений в компьютерной сети на основе многоагентной системы с целью сокращения времени решения масштабных задач.

Децентрализованная вычислительная система обеспечивает более высокую защиту работоспособности процессов распределенных вычислений, чем централизованное в нестабильной вычислительной среде компьютерной сети. Агенты, работая по приведённому в статье алгоритму, самостоятельно выполняют распределение между собой заданной вычислительной нагрузки для организации распределенных вычислений. В процессе выполнения распределенных вычислений агенты обмениваются между собой широковещательными сообщениями передают друг другу результаты вычислений и перераспределяют между собой заданную вычислительную нагрузку в зависимости от производительности компьютеров. Это позволяет обеспечить кроме сокращения времени решения масштабных задач, защиту работоспособности вычислительных процессов и результатов вычислений от подмены. Это повышает степень защиты распределенных вычислений от угрозы «отказ в обслуживании» и безопасности результатов решения от угрозы «ложного результата» по сравнению с централизованной вычислительной системой.

Для реализации алгоритма децентрализованной многоагентной системы была написана программа на языке Python. Программа агента была установлена на 3 компьютера. Результаты работы многоагентной системы в сети показали, что организованная система распределенных вычислений работает. Система выполняет

контроль за целостностью результатов решений большеобъемной задачи. При организации системы распределенных вычислений снижается вычислительная нагрузка на компьютеры, имеющие низкую производительность, путем перераспределения вычислительной нагрузки между агентами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Литвиненко В.А., Ховансков С.А., Хованскова В.С., Литвиненко Е.В.* Применение параметрической адаптации в алгоритмах построения ортогонального дерева Штейнера // Информатика, вычислительная техника и инженерное образование. – 2016. – № 4 (28). – С. 9-16.
2. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Разбиение на основе моделирования адаптивного поведения биологических систем // Нейрокомпьютеры: разработка, применение. – 2010. – № 2. – С. 28-34.
3. *Литвиненко В.А., Ховансков С.А., Рябов О.В., Платонов В.А.* Разработка модуля размещения учебно-исследовательской САПР на базе САПР Ki-CAD // Известия ЮФУ. Технические науки. – 2008. – № 4 (81). – С. 68-73.
4. *Ховансков С.А., Литвиненко В.А., Литвиненко Е.В.* Модифицированный адаптивный алгоритм определения максимальных полных подграфов // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 227-231.
5. *Ховансков С.А., Литвиненко В.А., Литвиненко Е.В.* Гибридный метод управления точностью решения экстремальных задач на графах // Известия ЮФУ. Технические науки. – 2013. – № 7 (144). – С. 112-116.
6. *Максюта Д.Ю., Ховансков С.А., Литвиненко В.А.* Адаптивный алгоритм построения дерева Штейнера // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 152-160.
7. *Ховансков С.А., Литвиненко В.А., Максюта Д.Ю.* Построение дерева Штейнера с использованием параметрической адаптации // Информатика, вычислительная техника и инженерное образование. – 2014. – № 4 (19). – С. 34-43.
8. *Ховансков С.А., Литвиненко В.А., Калашников В.А.* Алгоритм трассировки на многопроцессорной вычислительной системе. – Депонировано в ВИНТИ 19.04.91 г., № 1684-В91.
9. *Ховансков С.А., Норкин О.Р., Хованскова В.С.* Адаптация алгоритма трассировки к реализации в распределенной информационной системе // Материалы III Международной научно-технической конференции «Технологии разработки информационных систем ТРИС-2012». – 2012. – С. 81-88.
10. *Ховансков С.А.* Распараллеливание алгоритмов построения связывающего дерева для решения на многопроцессорной вычислительной системе // Известия ТРТУ. – 1997. – № 3 (6). – С. 226.
11. *Ховансков С.А.* Исследование и разработка параллельных алгоритмов трассировки БИС: автореф. дисс. ... канд. техн. наук. – Таганрог, 1998.
12. *Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U.* Mobile-Agent-Based Systems Against Malicious Hosts // World Applied Sciences Journal. – 2014. – Vol. 29 (2). – P. 287-297.
13. *Muñoz A., Pablo A., Maña A.* Multiagent Systems Protection // Advances in Software Engineering. – 2011. – Article ID 281517. – 9 p. – Doi: 10.1155/2011/281517.
14. *Xudong G., Yiling Ya., Yinyuan Y.* POM-a mobile model against malicious hosts // Proceedings of High Performance Computing in the Asia-Pacific Region. – 2000. – Vol. 2. – P. 1165-1166.
15. *Ховансков С.А., Литвиненко В.А.* Решения задач путем организации распределенных вычислений в сети // Известия ЮФУ. Технические науки. – 2008. – № 3 (80). – С. 16-21.
16. *Litvinenko V.A., Khovanskov S.A., Norkin O.R., Khovanskova V.S.* The algorithm for determining the direction of building relations in a distributed computing system // IS&IT'12: proceedings of the Congress on intelligent systems and information technologies. Scientific publication in 4 volumes. – Moscow: Physmathlit, 2012. – Vol. 4. – P. 49.
17. *Ховансков 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.

18. Лобашев Д.Г., Ховансков С.А. Организация распределенных вычислений в иерархической вычислительной структуре // Теория и практика современной науки. – 2016. – № 6-1 (12). – С. 748-750.
19. Ховансков С.А., Норкин О.Р., Парфенова С.С., Хованскова В.С. Алгоритмическое обеспечение распределённых вычислений с использованием иерархической вычислительной структуры // Информатизация и связь. – 2014. – № 2. – С. 71-74.
20. Ховансков С.А., Румянцев К.Е., Хованскова В.С. Алгоритм программного модуля децентрализованной системы управления для создания безопасных распределенных вычислений в неустойчивой вычислительной среде // Известия ЮФУ. Технические науки. – 2013. – № 12 (149). – С. 247-253.
21. Ховансков С.А., Литвиненко В.А., Хованскова В.С. Алгоритм организации безопасных распределенных вычислений на основе многоагентной системы // Известия ЮФУ. Технические науки. – 2016. – № 10 (183). – С. 146-158.
22. Ховансков С.А., Хованскова В.С., Литвиненко В.А. Оценка сокращения времени построения связывающих деревьев цепей с помощью распределённой вычислительной системы // Информатика, вычислительная техника и инженерное образование. – 2016. – № 4 (28). – С. 34-42.
23. Шумигин В.К., Лобашев Д.Г., Бембеев Д.А., Ховансков С.А. Комплексный подход для обеспечения информационной безопасности в корпоративной организации // Теория и практика современной науки. – 2016. – № 6-2 (12). – С. 427-430.
24. Хованскова В.С., Ховансков С.А. Безопасность мультиагентных систем // Вопросы науки. – 2015. – Т. 2. – С. 83-87.
25. Хованскова В.С., Ховансков С.А. Повышение степени защиты распределенных вычислений // Современное состояние естественных и технических наук. – 2015. – № XVIII. – С. 96-99.
26. Хованскова В.С., Ховансков С.А. Методы защиты распределенных вычислений // Модернизация современного общества: проблемы, пути развития и перспективы. – 2015. – № 6. – С. 104-107.

## REFERENCES

1. Litvinenko V.A., Khovanskov S.A., Khovanskova V.S., Litvinenko E.V. Primenenie parametricheskoy adaptatsii v algoritmakh postroeniya ortogonal'nogo dereva Shteynera [Application of parametric adaptation in algorithms of construction of an orthogonal Steiner tree], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Computer science, computer engineering and engineering education], 2016, No. 4 (28), pp. 9-16.
2. Kureychik V.M., Lebedev B.K., Lebedev O.B. Razbienie na osnove modelirovaniya adaptivnogo povedeniya biologicheskikh sistem [Partition on the basis of modeling of adaptive behavior of biological systems], *Neyrokompyutery: razrabotka, primeneniye* [Neurocomputers: development, application], 2010, No. 2, pp. 28-34.
3. Litvinenko V.A., Khovanskov S.A., Ryabov O.V., Platonov V.A. Razrabotka modulya razmeshcheniya uchebno-issledovatel'skoy SAPR na baze SAPR Ki-CAD [Module development of accommodation of educational research CAD of CAD on the basis of Ki-CAD], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 4 (81), pp. 68-73.
4. Khovanskov S.A., Litvinenko V.A., Litvinenko E.V. Modifitsirovanny adaptivnyy algoritm opredeleniya maksimal'nykh polnykh podgrafov [A modified adaptive algorithm for determining the maximum full sub-graphs], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 227-231.
5. Khovanskov S.A., Litvinenko V.A., Litvinenko E.V. Gibridnyy metod upravleniya tochnost'yu resheniya ekstremal'nykh zadach na grafakh [The hybrid method of controlling the precision of solutions of extremal problems on graphs], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 112-116.
6. Maksyuta D.Yu. Khovanskov S.A., Litvinenko V.A. Adaptivnyy algoritm postroeniya dereva Shteynera [Adaptive algorithm for construction of Steiner tree], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 152-160.

7. Khovanskov S.A., Litvinenko V.A., Maksyuta D.Yu. Postroenie dereva SHteynera s ispol'zovaniem parametricheskoy adaptatsii [Construction of the Steiner tree using parametric adaptation], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Computer science, computer engineering and engineering education], 2014, No. 4 (19), pp. 34-43.
8. Khovanskov S.A., Litvinenko V.A., Kalashnikov V.A. Algoritm trassirovki na mnogoprotsessornoy vychislitel'noy sisteme [Trace algorithm on a multiprocessor computer system]. Deposited in 19.04.91, No. 1684-V91.
9. Khovanskov S.A., Norkin O.R., Khovanskova V.S. Adaptatsiya algoritma trassirovki k realizatsii v raspredelennoy informatsionnoy sisteme [Adaptation of the trace algorithm to implementation in a distributed information system], *Materialy III Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii «Tekhnologii razrabotki informatsionnykh sistem TRIS-2012»* [Proceedings of the III International scientific and technical conference "technologies for development of information systems TRIS-2012"], 2012, pp. 81-88.
10. Khovanskov C.A. Rasparallelvanie algoritmov postroeniya svyazyvayushchego dereva dlya resheniya na mnogoprotsessornoy vychislitel'noy sisteme [Parallelization of algorithms for constructing a linking tree for solving on a multiprocessor computer system], *Izvestiya TRTU* [Izvestiya TSURE], 1997, No. 3 (6), pp. 226.
11. Khovanskov S.A. Issledovanie i razrabotka parallel'nykh algoritmov trassirovki BIS: avtoref. diss. ... kand. tekhn. nauk [Research and development of parallel algorithms for BIS tracing: autoref. Cand. of eng. sc. diss.]. Taganrog, 1998.
12. Madkour A.M., Eassa F.E., Ali A.M., Qayyum N.U. Mobile-Agent-Based Systems Against Malicious Hosts, *World Applied Sciences Journal*, 2014, Vol. 29 (2), pp. 287-297.
13. Muñoz A., Pablo A., Maña A. Multiagent Systems Protection, *Advances in Software Engineering*, 2011. Article ID 281517, 9 p. Doi: 10.1155/2011/281517.
14. Xudong G., Yiling Ya., Yinyuan Y. POM-a mobile model against malicious hosts, *Proceedings of High Performance Computing in the Asia-Pacific Region*, 2000, Vol. 2, pp. 1165-1166.
15. Khovanskov S.A., Litvinenko V.A. Resheniya zadach putem organizatsii raspredelennykh vychisleniy v seti [Solving problems by organizing distributed computing in the network], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 3 (80), pp. 16-21.
16. Litvinenko V.A., Khovanskov S.A., Norkin O.R., Khovanskova V.S. The algorithm for determining the direction of building relations in a distributed computing system, *IS&IT'12: proceedings of the Congress on intelligent systems and information technologies. Scientific publication in 4 volumes*. Moscow: Phymathlit, 2012, Vol. 4, pp. 49.
17. 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 domain products], *Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam "AIS-IT'09"* [Proceedings of the Congress on intelligent systems and information technologies "AIS-IT'09"]. Moscow: Fizmatlit, 2009, Vol. 2, pp. 153-158.
18. Lobashev D.G., Khovanskov S.A. Organizatsiya raspredelennykh vychisleniy v ierarkhicheskoy vychislitel'noy strukture [Organization of distributed computing in a hierarchical computational structure], *Teoriya i praktika sovremennoy nauki* [Theory and practice of modern science], 2016, No. 6-1 (12), pp. 748-750.
19. Lobashev D.G., Khovanskov S.A. Organizatsiya raspredelennykh vychisleniy v ierarkhicheskoy vychislitel'noy strukture [Algorithmic support of distributed computing using a hierarchical computational structure], *Teoriya i praktika sovremennoy nauki* [Informatization and communication], 2016, No. 6-1 (12), pp. 748-750.
20. Khovanskov S.A., Rumyantsev K.E., Khovanskova V.S. Algoritm programmogo modulya detsentralizovannoy sistemy upravleniya dlya sozdaniya bezopasnykh raspredelennykh vychisleniy v neustoychivoy vychislitel'noy srede [Algorithm of the software module of the decentralized control system for creation of safe distributed calculations in the unstable computing environment], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 12 (149), pp. 247-253.
21. Khovanskov S.A., Litvinenko V.A., Khovanskova V.S. Algoritm organizatsii bezopasnykh raspredelennykh vychisleniy na osnove mnogoagentnoy sistemy [Algorithm for secure distributed computing based on multi-agent systems], *Izvestiya YUFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 10 (183), pp. 146-158.

22. *Khovanskov S.A., Khovanskova V.S., Litvinenko V.A.* Otsenka sokrashcheniya vremeni postroeniya svyazyvayushchikh derev'ev tsepey s pomoshch'yu raspredelennoy vychislitel'noy sistemy [Estimation of reduction of time of construction of connecting trees of chains by means of the distributed computing system,] *Informatika, vychislitel'naya tekhnika i inzhenerное образование* [Computer science, computer engineering and engineering education], 2016, No. 4 (28), pp. 34-42.
23. *Shumigin V.K., Lobashev D.G., Bembeev D.A., Khovanskov S.A.* Kompleksnyy podkhod dlya obespecheniya informatsionnoy bezopasnosti v korporativnoy organizatsii [An integrated approach to information security in a corporate organization], *Teoriya i praktika sovremennoy nauki* [Theory and practice of modern science], 2016, No. 6-2 (12), pp. 427-430.
24. *Khovanskova V.S., Khovanskov S.A.* Bezopasnost' mul'tiagentnykh sistem [Safety of multi-agent systems], *Voprosy nauki* [Questions of science], 2015, Vol. 2, pp. 83-87.
25. *Khovanskova V.S., Khovanskov S.A.* Povyshenie stepeni zashchity raspredelennykh vychisleniy [increasing the degree of protection of distributed computing], *Sovremennoe sostoyanie estestvennykh i tekhnicheskikh nauk* [Current state of natural and technical Sciences], 2015, No. XVIII, pp. 96-99.
26. *Khovanskova V.S., Khovanskov S.A.* Metody zashchity raspredelennykh vychisleniy [Methods of protection of distributed computing], *Modernizatsiya sovremennoogo obshchestva: problemy, puti razvitiya i perspektivy* [Modernization of modern society: problems, ways of development and prospects], 2015, No. 6, pp. 104-107.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.О. Чернышев.

**Ховансков Сергей Андреевич** – Южный федеральный университет; e-mail: sah59@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634676616; кафедра информационной безопасности телекоммуникационных систем; доцент.

**Хованскова Вера Сергеевна** – e-mail: bepok2010@gmail.com; кафедра информационной безопасности телекоммуникационных систем; аспирантка.

**Литвиненко Василий Афанасьевич** – e-mail: litv\_va@mail.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

**Khovanskov Sergey Andreevich** – Southern Federal University; e-mail: sah59@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634676616; the department of information security of telecommunication systems; associate professor.

**Khovanskova Vera Sergeevna** – e-mail: bepok2010@gmail.com; the department of information security of telecommunication systems; postgraduate student.

**Litvinenko Vasily Avanasjevich** – e-mail: litv\_va@mail.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

УДК 004.023

DOI 10.23683/2311-3103-2018-4-210-223

**А.С. Аргюхова**

### **ИССЛЕДОВАНИЕ ЗАВИСИМОСТИ ОПТИМАЛЬНОСТИ (ЭФФЕКТИВНОСТИ) ТЕСТ-КЕЙСА ОТ РАЗЛИЧНЫХ ПОКАЗАТЕЛЕЙ\***

*В настоящее время процесс испытания программного продукта является трудоемким, и затратным процессом в материальном плане и по времени, отводимом на этот этап разработки. Сложность программ неуклонно растет, что усложняет процесс их верификации. Разработка методов позволяющих оптимизировать этот этап является актуальной задачей. Возможным решением этой задачи является создание метода оптимизации процесса верифика-*

\* Работа выполнена при финансовой поддержке грантов РФФИ (проекты: № 16-07-00336, № 16-07-00335) в Южном федеральном университете.