

Реализация агентно-базированных метакомпьютерных систем и приложений

Н.С. Карамышева¹, Д.С. Свищев¹, К.В. Попов¹, С.А. Зинкин¹ 

¹ Пензенский государственный университет
д. 40, ул. Красная, г. Пенза 440026, Российская Федерация

 e-mail: zsa49@yandex.ru

Резюме

Цель исследования. Создание методики проектирования прототипа метакомпьютерной распределенной вычислительной системы с учетом современного этапа эволюции аппаратного и облачно-сетевого программного обеспечения для предоставления пользователям средства создания приложений, обладающих межпрограммным параллелизмом и возможностью совместной работы компонент.

Методы. Логические модели искусственного интеллекта - концептуальные графы, агентно-базированная технология программирования, виртуализация сетевых ресурсов. Метод проведения натурного эксперимента заключался в том, что при запуске приложения на выполнение в виртуальном агентно-базированном метакомпьютере была использована сетевая инфраструктура с удаленным доступом к лабораторной сети Fast Ethernet через Интернет и затем проводилось измерение временных характеристик.

Результаты. Предложена методика проектирования облачно-сетевых метакомпьютерных систем и приложений, создано прототипное программное обеспечение промежуточного уровня middleware на базе мультиагентной технологии. Поставленная цель достигнута, так как разработанная агентно-базированная среда позволяет реализовывать универсальные управляющие конструкции программирования – переход по одному или нескольким условиям, цикл, последовательность, распараллеливание, для которых введены исполнимые концептуальные спецификации.

Заключение. Предложен подход к реализации распределенного метакомпьютерного приложения в среде компьютерной сети на основе использования предметно-ориентированной мультиагентной системы и концепции агентно-ориентированного программирования на основе концептуальных графов, описывающих обмен сообщениями и обработку данных программными агентами. В агентно-базированной среде эффективно реализуются межузловые взаимодействия по типу пиринговых; возможна организация параллельной обработки данных как несколькими, так и одним агентом. Предлагаемые методы решения задач позволяют оперативно создавать распределенные приложения, работающие в облачно-сетевой среде; время разработки и отладки при этом сокращается примерно на порядок.

Ключевые слова: метакомпьютерные системы; концептуальные графы; мультиагентные системы; технологии виртуализации; сетевые технологии.

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Реализация агентно-базированных метакомпьютерных систем и приложений / Н.С. Карамышева, Д.С. Свищев, К.В. Попов, С.А. Зинкин // Известия Юго-Западного государственного университета. 2022; 26(1): 148-171. <https://doi.org/10.21869/2223-1560-2022-26-1-148-171>.

Поступила в редакцию 22.12.2021

Подписана в печать 27.01.2022

Опубликована 31.03.2022

Implementation of Agent-Based Metacomputersystems and Applications

Nadezhda S. Karamysheva¹, Danil S. Svishchev¹, Konstantin V. Popov¹,
Sergey A. Zinkin¹ ✉

¹ Penza State University

40 Krasnaya str., Penza 440026, Russian Federation

✉ e-mail: zsa49@yandex.ru

Abstract

Purpose of research. Creation of a methodology for designing a prototype of a metacomputer distributed computing system, taking into account the current stage of the evolution of hardware and cloud-network software to provide users with the means to create applications with inter-program parallelism and the ability of components to work together.

Methods. Logical models of artificial intelligence, semantic networks and conceptual graphs, agent-based technology, virtualization of network resources. The method of conducting a full-scale experiment was that when the application was launched for execution in a virtual agent-based metacomputer, a network infrastructure was used with remote access to the Fast Ethernet laboratory network via the Internet, and then time characteristics were measured.

Results. A technique for designing cloud-network metacomputer systems and applications is proposed, and prototype middleware software based on multi-agent technology is created. The goal of the study has been achieved, since the developed agent-based environment allows the implementation of universal programming control structures - transition by one or more conditions, cycle, sequence, parallelization, for which executable conceptual specifications have been introduced.

Conclusion. An approach to the implementation of a distributed metacomputer application in a computer network environment based on conceptual graphs describing the exchange of messages and data processing by software agents is proposed. The performance of the application under conditions of extraneous load on the network was demonstrated.

Keywords: metacomputer systems; logical models; conceptual graphs; multiagent systems; virtualization technologies; network technologies; technology integration.

Conflict of interest. The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Karamysheva N. S., Svishchev D. S., Popov K. V., Zinkin S. A. Implementation of Agent-Based Metacomputersystems and Applications. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2022; 26(1): 148-171 (In Russ.). <https://doi.org/10.21869/2223-1560-2022-26-1-148-171>.

Received 22.12.2021

Accepted 27.01.2022

Published 31.03.2022

Введение

Достижения в области сетевых и вычислительных технологий позволяют создавать крупномасштабные высокопроизводительные распределенные вычислительные среды. В то же время появляются новые требования к приложениям по использованию разнообразных, географически распределенных ресурсов. Появляются новые классы приложений – грид, облачных, коммунальных, кластерных, требующих обеспечения функциональных возможностей, недоступных на одном компьютере. Большинство крупных компьютерных компаний занимается разработкой распределенных сетевых архитектур компонентно базированных приложений, обеспечивающих параллельную и распределенную обработку информации. В этой связи в классе компонентно-базированных приложений большой интерес представляют агентно-базированные приложения, обеспечивающие при реализации в распределенных вычислительных системах высокий уровень масштабируемости, гибкости, мобильности и надежности.

Проблемы развития Интернета и будущее цифровой экономики рассмотрены в работах [1, 2], где отмечается важное значение развитие международной экономики, управляемой потоками трансграничных данных.

На сайте [3] уточняется понятие “метакомпьютинг”, сделан краткий обзор технологий организации распре-

ленных вычислений в Интернете и отмечается, что современный метакомпьютинг основан на однородном доступе к вычислительным ресурсам большого числа (вплоть до нескольких тысяч) компьютеров в локальных или глобальных сетях. Подобное же мнение поддерживается организаторами и пользователями крупного международного проекта “World Community Grid” [4], где большое внимание уделяется развитию “волонтерского” распределения вычислительных мощностей, принадлежащих различным собственникам. В названии проекта присутствует термин “грид”, хотя термин “метакомпьютинг” более соответствует решаемым задачам. В числе этих задач, как отмечается в источниках [3, 4], задачи ядерной физики, защиты окружающей среды, предсказания погоды, численного моделирования, биологического моделирования, фармацевтики, астрономии, астрофизики, медицины и др.

Агентно-базированная массивно-параллельная система виртуальных машин разработана авторами работ [5, 6]. Система, созданная по технологии мобильных агентов, предназначена для высокопроизводительных вычислений в различных предметных областях.

Агентно-ориентированный язык программирования JADEL для платформы JADE предложен в работе [7]. Этот язык предназначен для упрощения программирования сетевых приложений и

создан на основе мультиагентной технологии.

Технология аглетов – мобильных JAVA-агентов [8] уже долгое время привлекает внимание разработчиков сетевого программного обеспечения. Однако эта технология в настоящее время поддерживается практически только программистами-любителями, так как аглеты (агенты-апплеты) оказались слишком громоздкими для использования в промышленных проектах. Более успешными оказались такие инструменты, как Voyager, Grasshopper, Tryllian, JADE, Trasy, SPRINGS. В этой же работе отмечается, что использование мобильных агентов устраняет недостатки существующей парадигмы “клиент-сервер” за счет снижения сетевой нагрузки и задержек, а также за счет асинхронного выполнения действий. Они также делают систему адаптивной к изменениям, надежной и отказоустойчивой [8].

Подробные обзоры агентно-базированных платформ и фреймворков приведены, например, в работах [9, 10, 11, 12].

На базе программных агентов эффективно реализуются и так называемые сервис-ориентированные архитектуры SOA (англ. SOA – Service-Oriented Architecture), в том числе архитектуры типа “функции как сервис” FaaS (англ. FaaS – Functions as a Service) [13, 14, 15]. В работе [16] предложена новая архитектура агентно-базированной облачной системы AaaS “агент как сервис”

(англ. AaaS – Agent as a Service). В данной архитектуре сервисные функции реализуются мобильными или стационарными агентами. Практический опыт создания объектно-ориентированных, агентно-ориентированных и AI-ориентированных (англ. AI – Artificial Intelligence – искусственный интеллект) параллельных и распределенных приложений, описанный в работах [16, 17, 18, 19], показывает, что при разработке современного программного обеспечения целесообразно использовать формальные методы и средства, в том числе логические методы искусственного интеллекта.

Материалы и методы

Интеграция технологий. В настоящей работе при реализации проекта прототипа метакомпьютера используются подход к описанию системной и функциональной архитектуры распределенных приложений на основе концепции фирмы Sun Microsystems “Сеть – это компьютер” [20], а также концептуальные модели функциональной архитектуры, определенные в работах [21, 22]. Агентно-базированный метакомпьютер создается в результате интеграции по меньшей мере трех технологий, что иллюстрирует рис. 1. К этим технологиям можно добавить и технологию создания грид-систем на основе мобильных агентов [4].

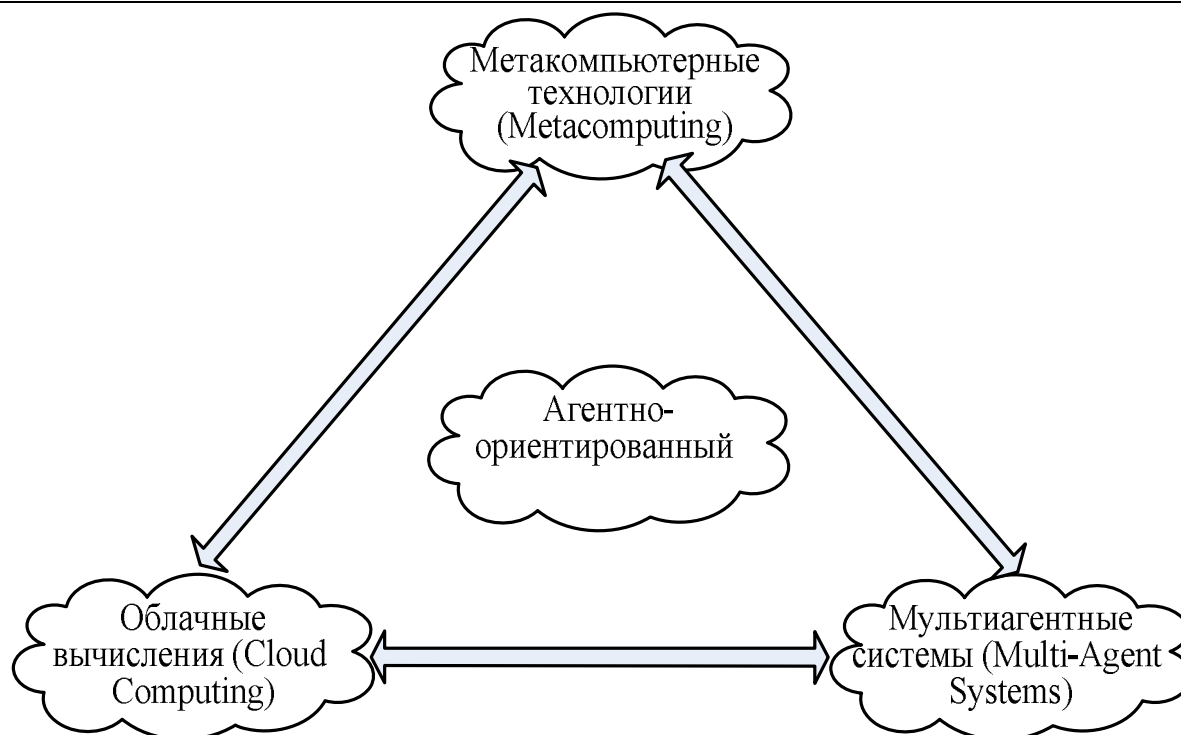


Рис. 1. Интеграция технологий при проектировании агентно-базируемого метакomпьютера

Fig. 1. Integration of technologies in the design of an agent-based metacomputer

Программные агенты начали создаваться на современных этапах эволюции программного обеспечения, предоставляя пользователям средства для создания приложений, обладающих межпрограммным параллелизмом и возможностью совместной работы компонент. Программный агент отличается от робота тем, что он действует в киберпространстве, образуемом компьютерной сетью. Для работы в динамичной, изменяющейся среде обычно используется модульная или компонентная, архитектура приложений, нередко использующая модели представления знаний в интеллектуальных системах.

В настоящей работе не учитываются антропоморфные свойства интеллектуальных агентов, хотя в плане развития архитектуры смешанного типа

AaaS-FaaS возможно придать агентам искусственные нейронные сети как дополнительные функции, а также функции машинного обучения MLaaS – Machine Learning as a Service.

Инфраструктура. Инфраструктура для развертывания рассматриваемого в настоящей работе метакomпьютерного приложения – это обычная глобальная или локальная TCP/IP сеть, в среде которой развернута апробированная в международной практике платформа JADE (Java Agent Development Framework) и проект которой разработан и сопровождается компанией Telecom Italia Lab [23, 24]. Размещение виртуальных узлов с агентами и виртуальные каналы связи в метакomпьютерной системе, развернутой в среде TCP/IP сети, показаны на рис. 2.

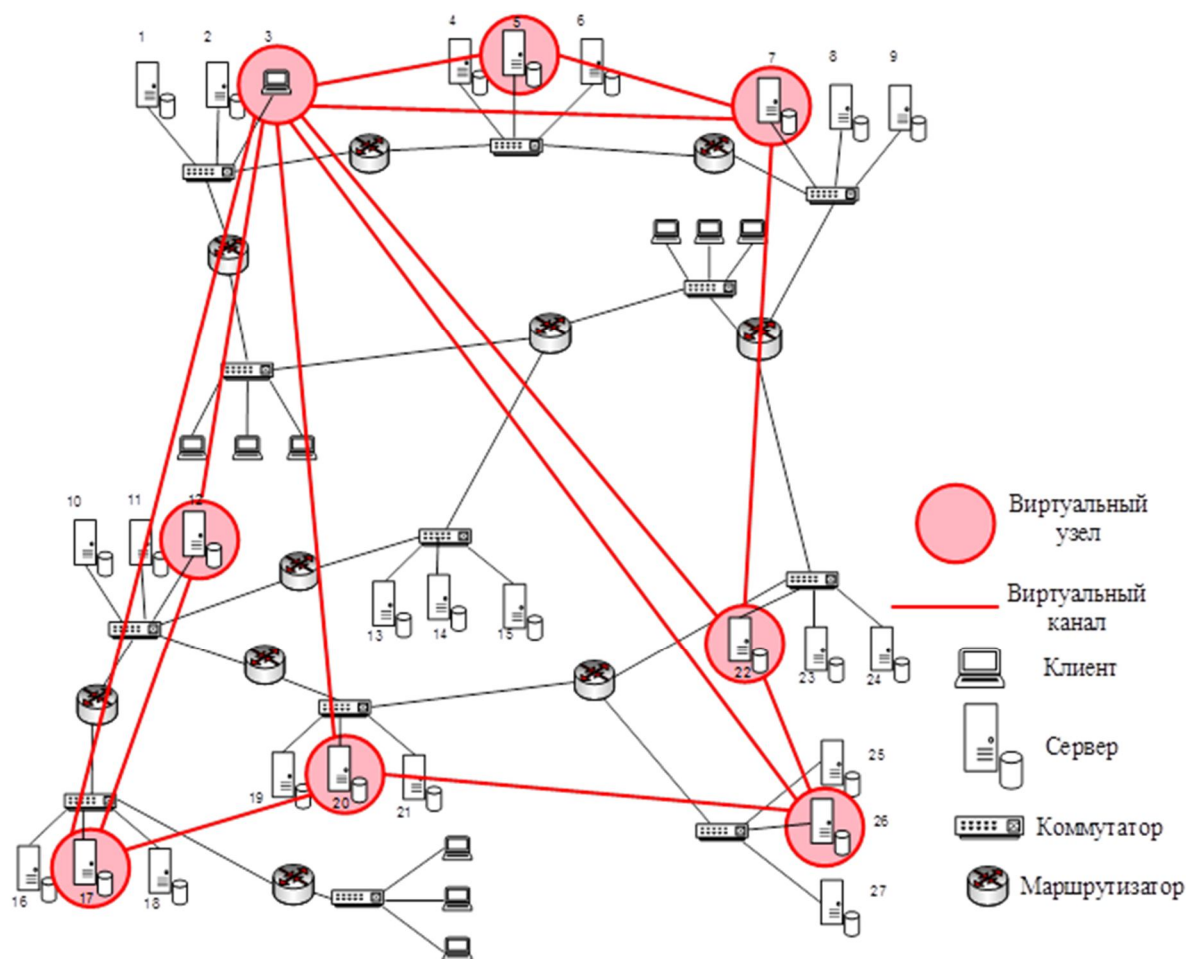


Рис. 2. Виртуальные узлы с агентами и виртуальные каналы связи метакомпьютерной системы, развернутой в среде TCP/IP сети

Fig. 2. Virtual nodes with agents and virtual communication channels of a metacomputer system deployed in a TCP/IP network environment

Концептуальная модель приложения. В качестве примера полнофункционального приложения рассмотрена реализация сценария *A* обработки данных, заданного концептуальным графом на рис. 2 и следующим операторным выражением:

$A = A0/Y0, A1/Y0, A2/Y0 \parallel (A3/Y1;$
 $A4/Y2; A5/Y3), J(A6/Y3) \parallel, A7/Y4, A8/Y4,$
 $S[A9[k/Y4]((k = 1): A10/Y5 \oplus (k = 2):$
 $A11/Y6 \oplus (k = 3): A12/Y7) Or(A13/Y8),$
 $A14/Y8,$

где A_i/Y_j – обозначение агента-оператора A_i , размещенного на узле Y_j ;

$\parallel F \dots J \parallel$ – обозначение операции “Fork-Join” распараллеливания сообщений;

$S \dots \oplus \dots \oplus \dots Or$ – обозначение операции “Switch-Or” выбора одной из нескольких альтернатив по значению некоторого параметра;

$A_i[k/Y_j]$ – эта запись имеет следующий смысл: “выполнение агента-оператора A_i на узле Y_j сопровождается вычислением параметра k ”.

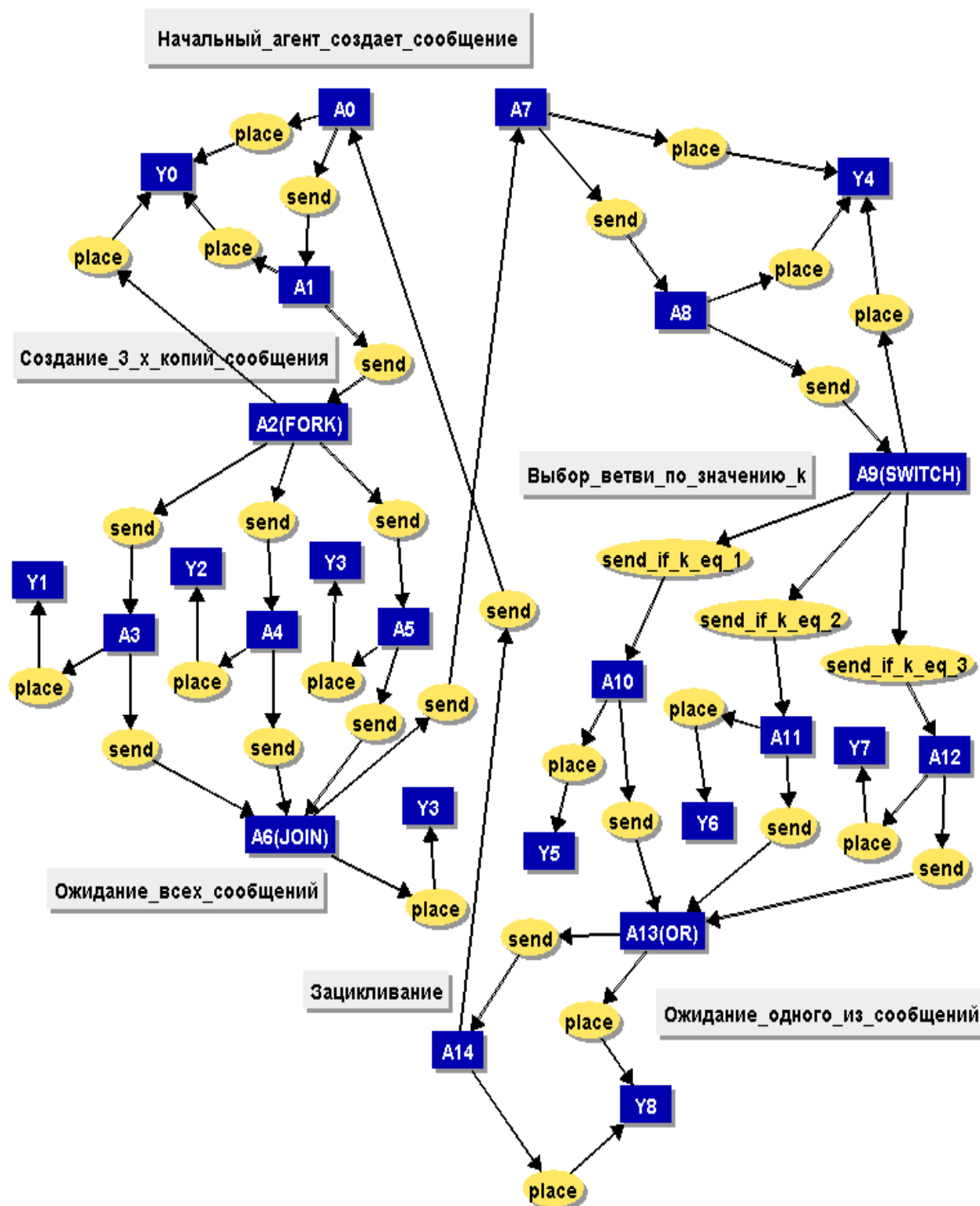


Рис. 3. Концептуальный граф сценария распределенной обработки данных

Fig. 3. Conceptual graph of a distributed data processing scenario

На рис. 4 представлена UML диаграмма, отображающая развертывание агентов, используемых в сценарии рас-

пределенной обработки данных, в мета-компьютере.

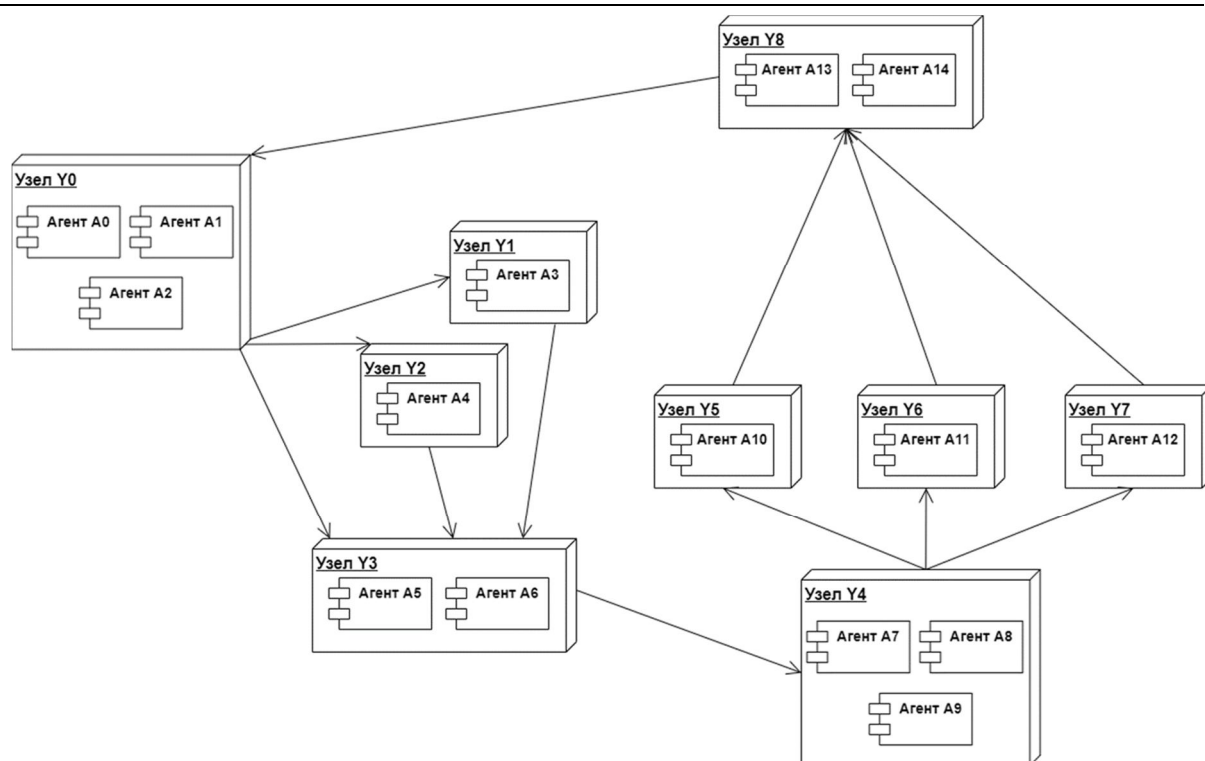


Рис. 4. Диаграмма развертывания метакомпьютерной системы

Fig. 4. Diagram of the deployment of a metacomputer system

Сценарий А “настроен” на выполнение решаемой задачи после того, как будут конкретизированы отношения send передачи сообщений и place размещения агентов на логических узлах сети, о чем на формальном уровне свидетельствует истинность следующего составного высказывания Р, которое основано на фактах, составляющие экстенциональную базу знаний:

$P = \text{place}(a0, y0) \ \& \ \text{place}(a1, y0) \ \& \ \text{send}(a0, a1) \ \& \ \text{place}(a2, y0) \ \& \ \text{send}(a1, a2) \ \& \ \text{place}(a3, y1) \ \& \ \text{place}(a4, y2) \ \& \ \text{place}(a5, y3) \ \& \ \text{send}(a2, a3) \ \& \ \text{send}(a2, a4) \ \& \ \text{send}(a2, a5) \ \& \ \text{place}(a6, y3) \ \& \ \text{send}(a3, a6) \ \& \ \text{send}(a4, a6) \ \& \ \text{send}(a5, a6) \ \& \ \text{place}(a7, y4) \ \& \ \text{send}(a6, a7) \ \& \ \text{place}(a8, y4) \ \& \ \text{send}(a7, a8) \ \& \ \text{place}(a9, y4) \ \&$

$\ \& \ \text{send}(a8, a9) \ \& \ \text{place}(a10, y5) \ \& \ \text{place}(a11, y6) \ \& \ \text{place}(a12, y7) \ \& \ ((k=1) \ \& \ \text{send}(a9, a10) \ \vee \ (k=2) \ \& \ \text{send}(a9, a11) \ \vee \ (k=3) \ \& \ \text{send}(a9, a12)) \ \& \ \text{place}(a13, y8) \ \& \ ((k=1) \ \& \ \text{send}(a10, a13) \ \vee \ (k=2) \ \& \ \text{send}(a11, a13) \ \vee \ (k=3) \ \& \ \text{send}(a12, a13)) \ \& \ \text{place}(a14, y8) \ \& \ \text{send}(a13, a14) \ \& \ \text{send}(a14, a0).$

Мультиагентная технология.

Фрейм-ворк JADE реализует мультиагентную технологию и представляет собой средство для разработки программного обеспечения промежуточного уровня *middleware*, предназначенное для создания многоагентных систем. JADE включает в свой состав среду выполнения, в которой агенты JADE могут реализовывать свой функционал [23, 24].

Поставлена задача разработки кроссплатформенного Java-приложения, обес-

печивающего реализацию некоторого сценария обработки данных и знаний, выполняемого в сети ЭВМ мобильными или стационарными агентами на основе парадигмы передачи сообщений. Программа должна реализовывать распределенный алгоритм обмена сообщениями между агентами и иметь графический пользовательский интерфейс для взаимодействия пользователя с системой и некоторыми агентами.

Общие примерные требования к составу программных средств: язык программирования Java, среда разработки IntelliJ IDEA, фреймворк JADE. Рекомендуется использование технологий

Java: Collection Framework, механизм обработки исключительных ситуаций, Java Stream API, Java Multithreading, сетевое взаимодействие. В рамках данной работы без ограничения общности рассматривается разработка распределенного алгоритма взаимодействия агентов в рамках одного контейнера.

Работа распределенного приложения начинается с инициализации агентов и последующей их регистрации в главном контейнере платформы. Для задания последовательности создания экземпляров агентов должен быть сформирован конфигурационный файл со следующим содержимым:

```
-gui – graphical user interface (графический интерфейс пользователя);
ta:agent.TestAgent;dev1:agent.DeveloperAgent;dev2:agent.DeveloperAgent;
dev3:agent.DeveloperAgent;tsr1:agent.TesterAgent;tsr2:agent.TesterAgent;
tsr3:agent.TesterAgent;do:agent.DevOpsAgent;tsl:agent.TestLeadAgent;
tsm:agent.TestManagerAgent;tsd:agent.TestDirectorAgent;tl:agent.TeamLeadAgent;
pm:agent.ProjectManagerAgent;dm:agent.DeliveryManagerAgent;
td:agent.TechnicalDirectorAgent;cl:agent.ClientAgent.
```

Приведенные далее необходимые элементы описания, которое касается программирования в среде JADE, соответствуют работам [23, 24].

Регистрация агентов. В целях обеспечения динамического поиска агентов каждый созданный экземпляр агента должен быть зарегистрирован в сервисе желтых страниц. При регистрации необходимо указать идентификатор агента AID (Agent Identifier) и его тип. После регистрации агент может выполнять поиск агента-адресата сообщения.

В качестве критериев поиска используются тип опубликованного (зарегистрированного) агента и, возможно, его имя. Каждый агент может обладать совокупностью “поведений”, которые описывают выполняемые агентом действия. Тип каждого поведения определяется функциональностью агента.

После регистрации агент готов к работе. При наличии подготовленного управляющего сообщения текущий агент выполняет поиск агента, которому следует передать это сообщение, то есть

поиск агента-адресата по его типу и, возможно, по имени.

Агент, работающий на платформе JADE, регистрирует свои так называемые “поведения”, то есть сценарии возможного функционирования. Поведения задают последовательность действий агента и реализуются в одном потоке.

Для распределенного алгоритма, представленного на рис. 3, сообщение задано в виде сложного унифицированного объекта, предоставляющего гибкую структуру данных, включающую контент и вспомогательные поля.

Пользовательские интерфейсы агентов. В соответствии со сценарием действий *A* взаимодействия агентов для выполнения распределенного алгоритма начинаются с создания стартового сообщения с помощью пользовательского интерфейса агента A0 (рис. 5).

Интерфейс агента A0 распределенного алгоритма является формой отображения входящих сообщений от агента A14 (при зацикливании) и обеспечивает управление добавлением, удалением, редактированием и отправкой вводимых сообщений. Необходимым условием успешных действий агента A0 является регистрация агента A1 в сервисе желтых страниц, только тогда он будет найден как адресат агентом A0. Use-case диаграмма (UML диаграмма вариантов использования) агента A0 представлена на рис. 6. Используемый в диаграмме термин “логирование” (англ. *logging*) действий агента обозначает его регистрацию с возможностью последующего отслеживания событий, которые происходят при работе распределенного приложения. Use-Case диаграммы для остальных агентов строятся по такому же образцу.

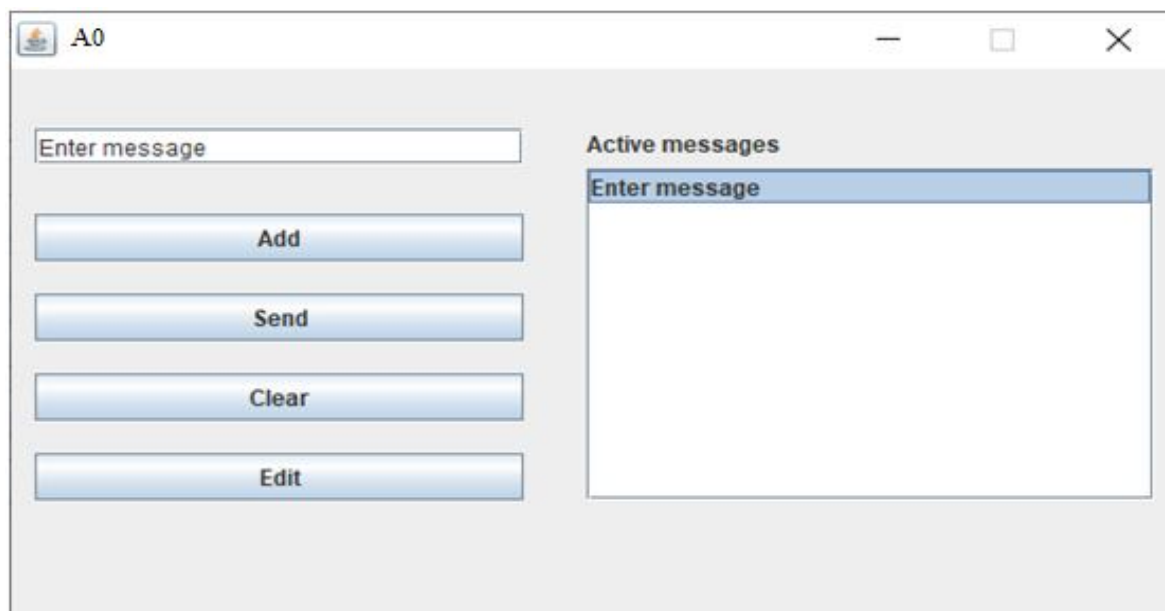


Рис. 5. Интерфейс агента A0

Fig. 5. Interface of the agent A0

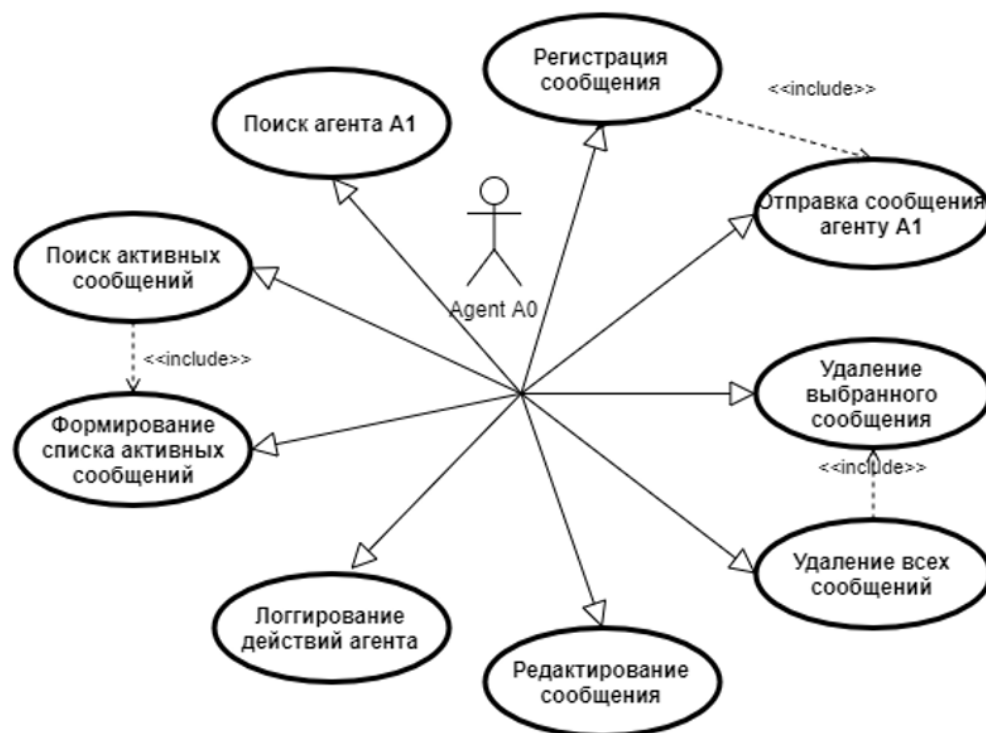


Рис. 6. Use-case диаграмма агента A0

Fig. 6. Use-case diagram of the agent A0

При нажатии кнопки “Send” выбранное сообщение отправляется агенту A1, далее проходит через цепочку агентов A2-A3-A4-A5-A6-A7-A8 и реги-

стрируется в интерфейсе агента A9 (рис. 7), информируя об этом пользователя всплывающим окном (рис. 8).

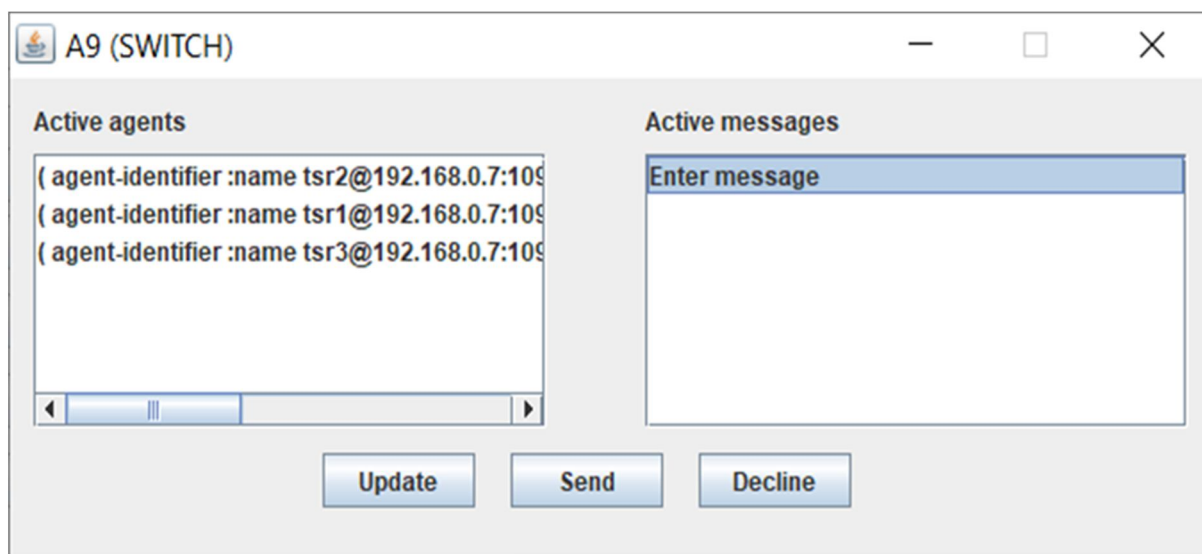


Рис. 7. Интерфейс агента A9

Fig. 7. Interface of the agent A9

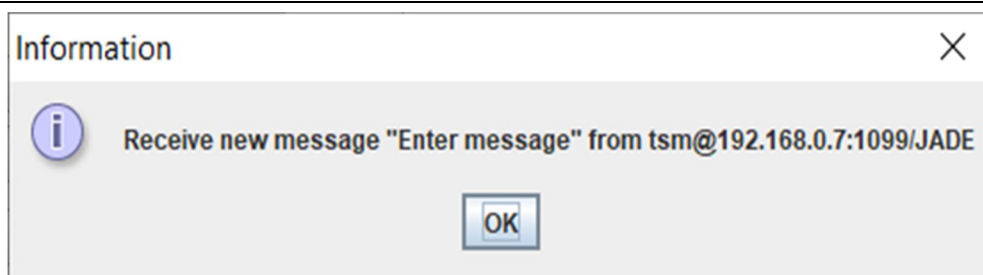


Рис. 8. Информационное окно с сообщением, выдаваемое агентом A9

Fig. 8. Information window with a message issued by the agent A9

Далее регистрируется передача сообщения через следующую цепочку агентов A10-A11-A12-A13, отображаемая в интерфейсе агента A14 (рис. 9), с последующим информированием пользователя при помощи информационного окна (рис. 10).

Взаимодействия агентов. Работа метакомпьютерного приложения начинается с регистрации и инициализации начального агента A0. Последующие

инициализации могут происходить при заиклировании распределенного алгоритма. В этом случае агент A14 должен отправить сообщение агенту A0, запуская новую итерацию распределенного алгоритма и уведомляя агента A0 о появлении передаваемого сообщения. Если же распределенное приложение должно выполняться однократно, то агент A14 завершает работу приложения.

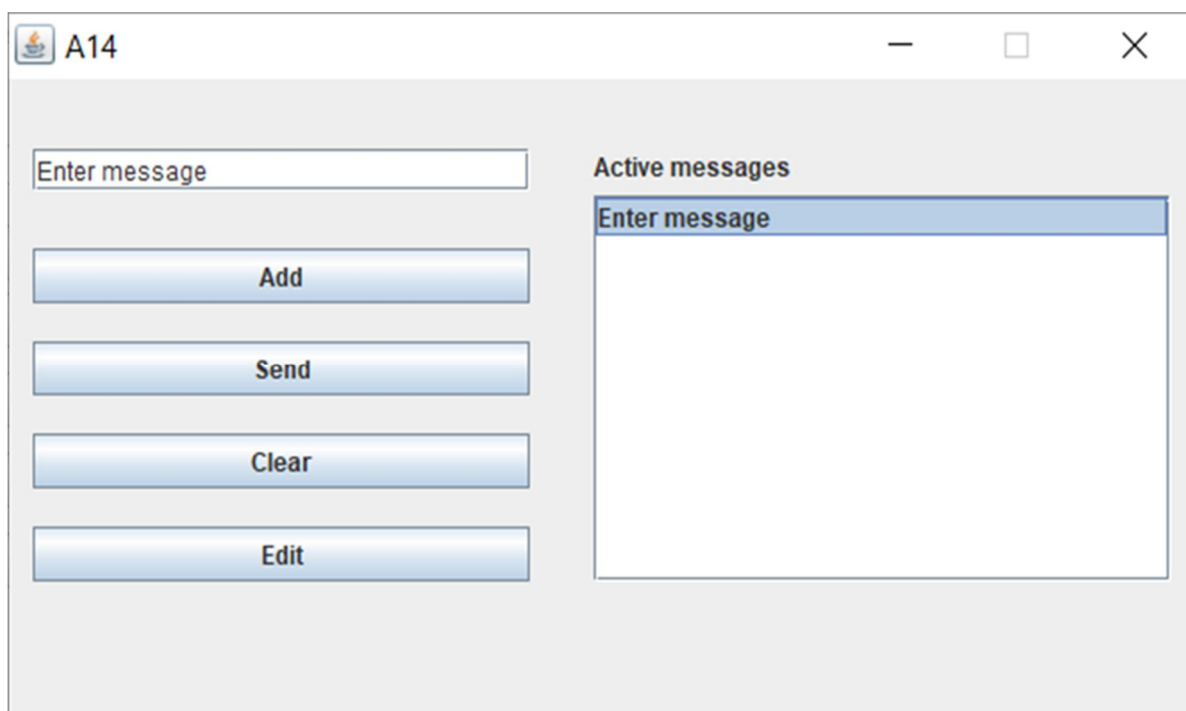


Рис. 9. Интерфейс агента A14

Fig. 9. Interface of the agent A14

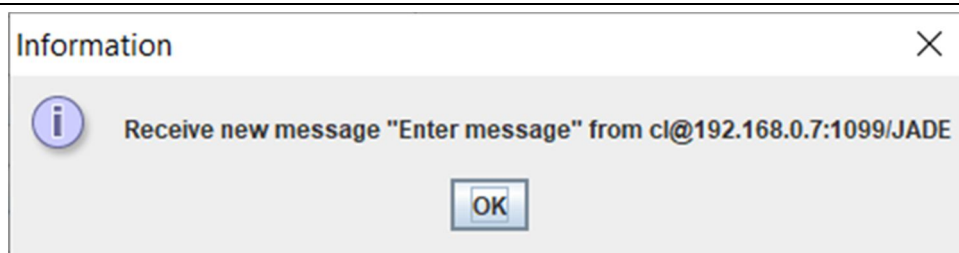


Рис. 10. Информационное окно, выдаваемое агентом A14

Fig. 10. Information window with a message issued by the agent A14

Агент A1, зарегистрировав свое поведение для поиска и обработки сообщений, ожидает входящее сообщение с данными от агента A0. Поиск сообщений осуществляется с помощью встроенного в платформу механизма обмена сообщениями между агентами. При успешном получении сообщения и наличии адресата агент A1 отправляет сообщение с данными агенту A2.

Агент A2 распараллеливает обработку получаемых данных, создавая копии полученных сообщений и рассылая их найденным агентам звена A3-A4-A5 в соответствии с заданным концептуальным графом на рис. 3.

Агенты звена A3-A4-A5 получают сообщения и далее осуществляют их обработку и отправку агенту-сборщику данных A6, который осуществляет реализацию сложных переговоров, ожидая поставки сообщений от всех агентов звена. Только в том случае, когда все сообщения получены, выполняется операция объединения данных и отправка сообщения с данными агенту A7.

Агент A7 ожидает сообщение от агента A6 и при его получении осуществляет его обработку, логирование данных и их передачу следующему

агенту A8. Агент A8 обладает поведением, подобным поведению агента A7, и при получении данных транслирует их агенту A9, который, как и агент A0, обладает пользовательским интерфейсом, позволяющим выбрать активное сообщение и следующего агента из списка.

Агент A9 является переключателем сообщений между агентами, отдавая определенную задачу одному выбранному агенту из группы агентов A10-A11-A12.

В свою очередь, каждый из агентов группы A10-A11-A12 ожидает задание на обработку сообщения, но получить задание может только один из них. Этот агент после получения сообщения запускает процесс логирования полученных данных и их отправку вместе с сообщением агенту A13.

Агент A13, получив сообщение от одного из агентов группы A10-A11-A12, обрабатывает его в сервисе сообщений, описанном в поведении агента, и отправляет конечному агенту A14 распределенного приложения.

Агент A14 отображает в графическом интерфейсе полученные сообщения и позволяет пользователю (в про-

цессе отладки) выбрать сообщение для передачи агенту A0 и запуска распределенного алгоритма на новую итерацию. Агент A0 прослушивает свой “почтовый ящик” и при наличии входящих со-

общений от агента A14 осуществляет запуск распределенного приложения с новыми данными. Use-case диаграмма агента A14 приведена на рис. 11.

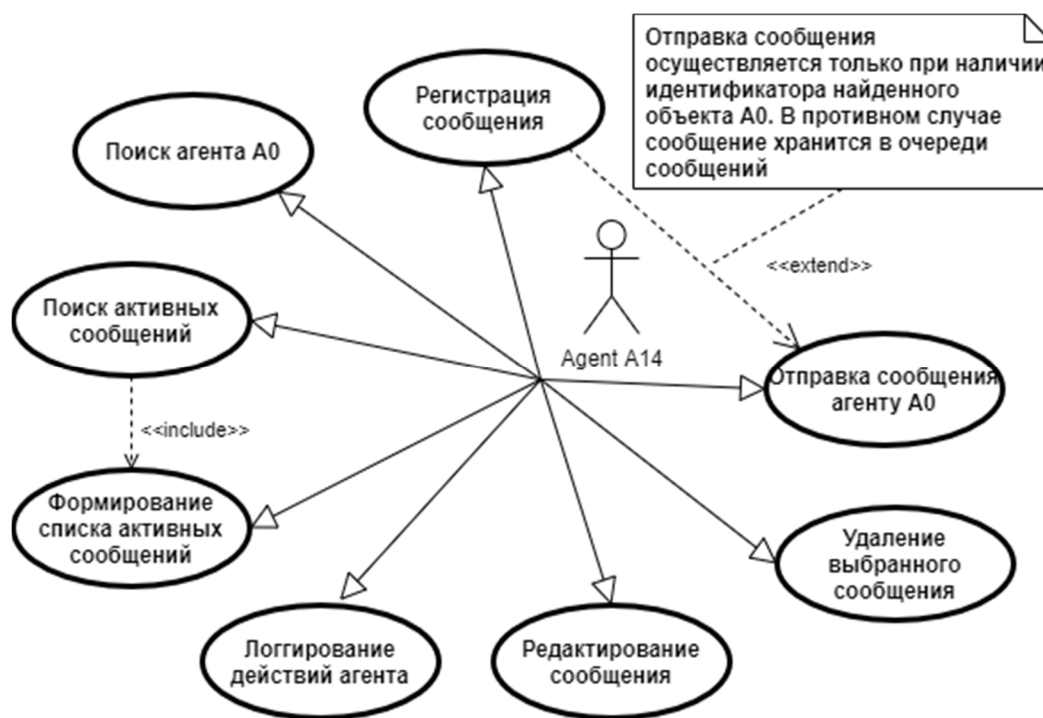


Рис. 11. Use-case диаграмма агента A14

Fig. 11. Use-case diagram of the agent A14

Описанное метакомпьютерное приложение способно обрабатывать данные, передаваемые совместно с управляющими сообщениями, в режиме “конвейерной параллельности”, то есть приложение способно принимать на обработку следующее сообщение с данными, не дожидаясь, пока будет обработано предыдущее сообщение. Для предотвращения тупиковых ситуаций предусмотрена проверка занятости каждого агента обработкой текущего сообщения.

При запуске приложения помимо создания и регистрации агентов произ-

водится инициализация пользовательского интерфейса (рис. 12), а именно административного GUI и выбранных для проверки работы приложения окон агентов A0, A9, A14.

Административный пользовательский интерфейс работает под управлением агента RMA (англ. Remote Management Agent); необходимы также утилиты Dummy agent и Sniffer agent, предназначенные для организации очередей и просмотра потока сообщений между агентами, как и предписано описаниями платформы JADE [23, 24].

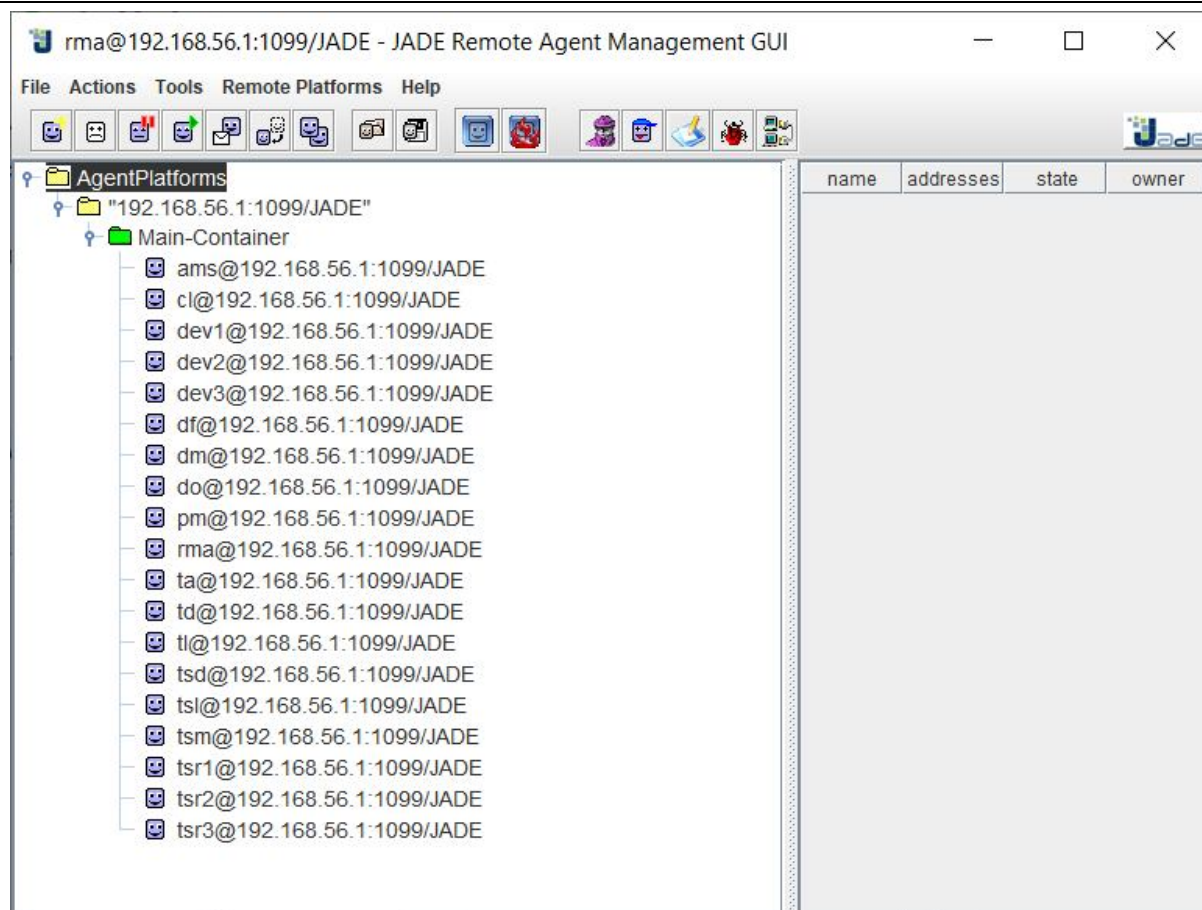


Рис. 12. Административный пользовательский интерфейс

Fig. 12. Administrative user interface

Результаты и их обсуждение

Главной целью создания агентно-базированной метакомпьютерной системы АБ МКС (*англ.* AB MCS – agent-based metacomputer system) являлась реализация концепции “Сеть – это компьютер”. Указанная цель достигнута, так как разработанная агентно-базированная среда позволяет реализовывать универсальные управляющие конструкции программирования – переход по одному или нескольким условиям, цикл, последовательность, распараллеливание, для которых введены исполнимые концептуальные спецификации.

Отличительной особенностью разработки является ее кардинальное отличие от различного рода пиринговых и клиент-серверных приложений и систем. Как и многие распределенные приложения, разработанное программное обеспечение АБ МКС работает эффективно при обработке больших объемах данных и при малом числе межузловых передач сообщений с данными между узлами. С другой стороны, необходимость создания АБ МКС может быть обусловлена территориальным размещением источников информации, местонахождение которых нецелесообразно изменять.

В процессе проведения экспериментов с агентно-базированным метакомпьютером АБ МКС была использована сетевая инфраструктура с удаленным доступом к лабораторной сети Fast Ethernet через Интернет.

Во-первых, были произведены эксперименты с АБ МКС на локальной сети Fast Ethernet, результаты которых приведены на рис. 13. Эксперименты состояли в следующем. Первый агент – управляющий, размещенный на управляющем узле, передавал свою копию с именем А0 по локальной сети и далее выполнялось приложение, концептуальный граф которого был ранее представлен на рис. 3, а сценарий действий описан выражением А. Агенты сценария на компьютеры локальной сети загружались последовательно, при помощи агента А0. Время загрузки одного агента 4 мс. “Теоретическое” время $T_{обр.}$ обработки сценария А принимало одно из двух значений – 3200 мс и 1600 мс (это время задавалось путем подбора времени выполнения операторов сценария). Все приложение в целом размещалось на N узлах локальной сети ($N = 1, 2, 4, 8, 16$).

На рис. 13 словом “Эксп. 1” обозначено экспериментальное среднее время с учетом служебного трафика (во внеучебном времени, передавались только управляющие пакеты локальной сети и, следовательно, посторонний для приложения уровень трафика был небольшим); словом “Эксп. 2” – экспериментальное среднее время с учетом

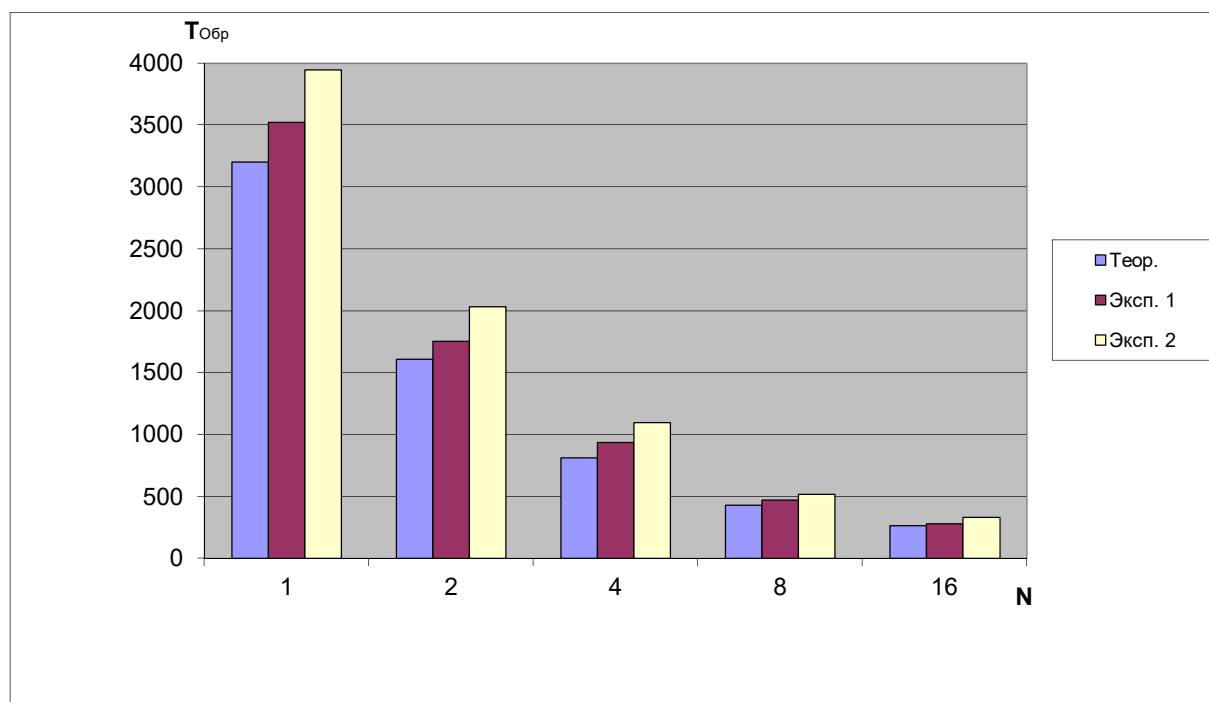
постороннего трафика повышенного уровня в локальной сети (в учебное время).

На рис. 14 представлены результаты экспериментов по передаче управляющих сообщений и копии агента с управляющего узла в Пензе на управляющий узел локальной сети в Москве через Интернет. Каждый агент загружал свою копию с одного компьютера в Пензе на другой волонтерский компьютер сети в Москве и копия затем возвращала обратно подтверждающее сообщение. На рис. 14 представлены зависимости среднего времени t одного «оборота» мобильного агента от номера эксперимента, при котором агент многократно (100 раз) запускался на удаленные узлы сети. Рисунки соответствуют четырем экспериментам, проведенным в различное время “будних” суток: рис. 14, а – утро, с 8 часов; рис. 14, б – обед, с 13 часов; рис. 14, в – вечер, с 19 часов; рис. 14, г – ночь, с 2 часов. Среднее время доставки копии управляющего агента составило 50-60 мс и в отдельные моменты времени изменялось скачками в связи с неравномерной загрузкой сети Интернет; тем не менее, все запланированные передачи агента состоялись и метакомпьютерное приложение получало задания на обработку условных данных по сценарию А.

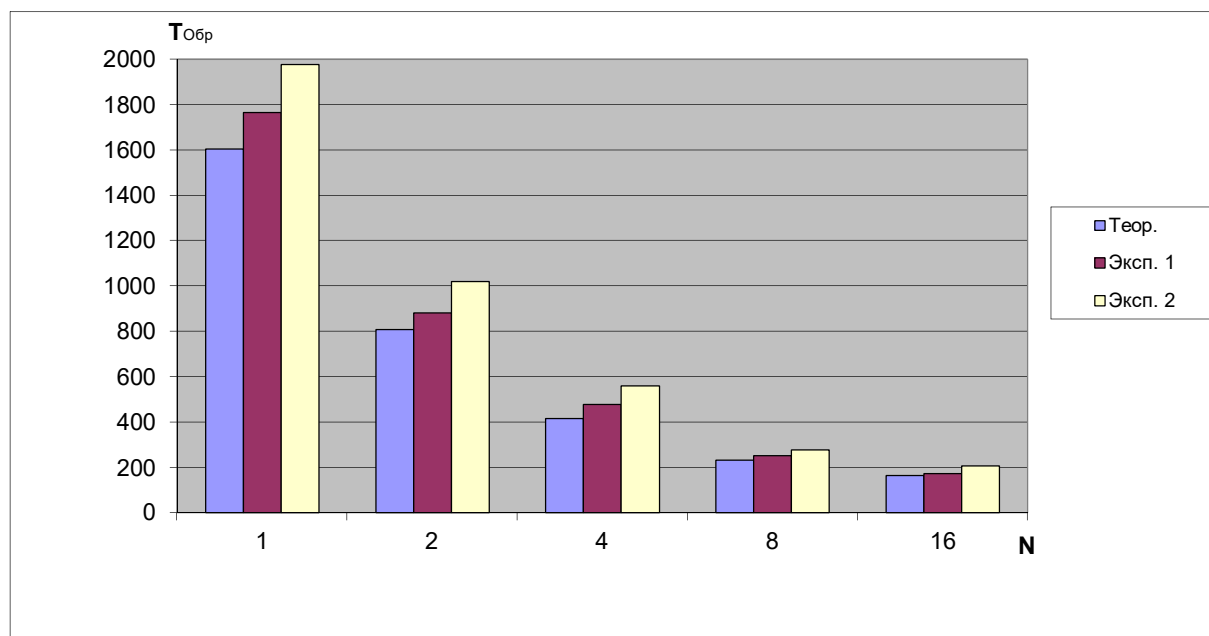
Главной целью проведенного эксперимента было доказательство работоспособности метакомпьютерного приложения в реальной сетевой среде. Эксперименты показали устойчивую рабо-

ту АБ МКС как в условиях постороннего трафика локальной сети, так и при использовании сети Интернет на неко-

тором ее участке с высоким уровнем постороннего для решаемой задачи трафика.



а) $T_{\text{Теор.}} = 3200$ мс, $T' = 4$ мс



б) $T_{\text{Теор.}} = 1600$ мс, $T' = 4$ мс

Рис. 13. Время решения задачи по сценарию А в локальной сети Fast Ethernet, 100 Мб ит/с, АБ МКС на платформе “JADE”

Fig. 13. Job execution time according to scenario A in the local network Fast Ethernet, 100 Mbps, AB MCS on the platform JADE

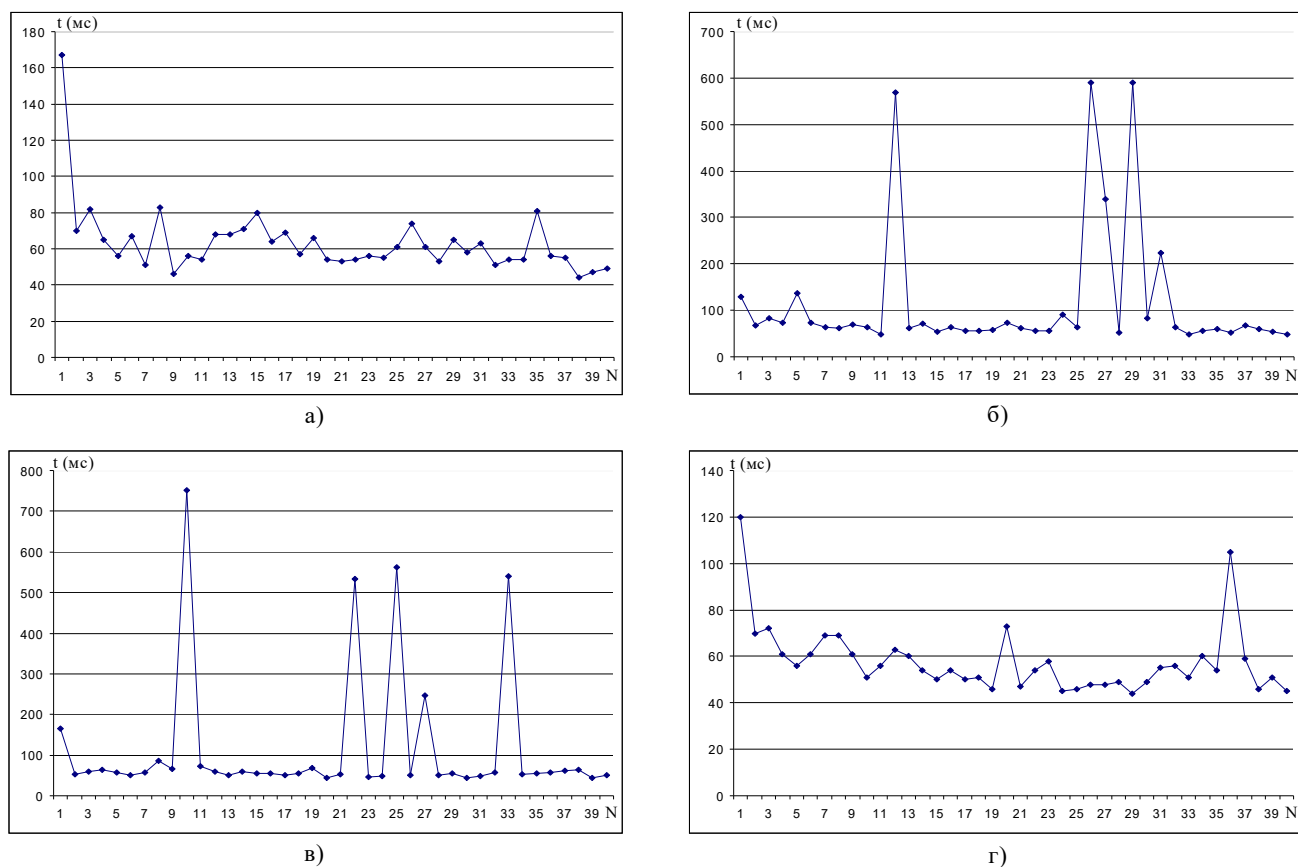


Рис. 14. Сеть Интернет, время передачи копии агента с квитированием на участке Пенза-Москва

Fig. 14. Average transfer time of an agent A0 with confirmation of its receipt on the Internet section Penza-Moscow

Сравнение показателей работы приложения по сценарию А с каким-либо прототипом не проводилось, поскольку почти все реальные метакомпьютерные и подобные им приложения в настоящее время ориентированы на обработку массовых данных в режиме “клиенты-сервер” или “мастер-слуги” и анализ производительности таких систем тривиален.

Выводы

В работе предложен подход к реализации распределенного метакомпьютерного приложения в среде компьютерной

сети на основе комплексного использования предметно-ориентированной мультиагентной системы и концепции агентно-ориентированного программирования в соответствии с заданным концептуальным графом, описывающим обмен сообщениями и обработку данных программными агентами. Использована динамично развивающаяся открытая мультиагентная платформа JADE, обеспечивающая широкий спектр функциональных возможностей при работе с агентами. Эта платформа является эффективной средой разработки и управления агентными приложениями в раз-

личных предметных областях, в том числе в области метакомпьютинга. В агентно-базированной среде эффективно реализуются межузловые взаимодействия по типу пиринговых; возможна организация параллельной обработки данных одним или многими агентами. Предлагаемые методы решения задач позволяют оперативно создавать

распределенные приложения, работающие в облачно-сетевой среде; время разработки и отладки при этом сокращается примерно на порядок. Проведенные эксперименты с базовой конфигурацией удаленного метакомпьютера показали работоспособность предложенной технологии создания распределенных агентно-базированных приложений.

Список литературы

1. Cohen-almagor R. The Future of the Internet. Academia Letters, 2021. P. 1-5. <https://doi.org/10.20935/AL1962>.
2. Digital economy report 2021. Cross-border data flows and development: for whom the data flow. Geneva, United Nations, 2021. 238 p.
3. Что такое метакомпьютинг (краткий обзор технологий организации распределенных вычислений в Интернете). URL: <https://parallel.ru/computers/reviews/meta-computing.html>. [Электронный ресурс]. Вид с экрана. Метод доступа свободный.
4. World Community Grid: Request for Proposal (RFP). 2020. P. 1–4. URL: <https://www.worldcommunitygrid.org/bg/rfp.pdf>. [Electronic resource]. Blank from the screen.
5. A New Massively Parallel And Distributed Virtual Machine Model Using Mobile Agents / M. Youssfi, O. Bouattane, J. Bakkoury, M. Bensalah // Conference: IEEE Conference: 4th International Conference on Multimedia Computing and Systems April 14-16, 2014 Marrakech - MOROCCO MSTI Mediterranean Space of Technology and Innovation. Morocco Chapter. April 14-16, 2014. P. 407-414. Marrakech, Morocco. <https://doi.org/10.1109/ICMCS.2014.6911306>.
6. Youssfi M., Bouattane O., Bensalah M. A Parallel Computational Model Based on Mobile Agents for High Performance Computing // Contemporary Engineering Sciences, 2015. Vol. 8, No. 15. P. 677-698. <http://dx.doi.org/10.12988/ces.2015.55153>.
7. Agent-oriented model-driven development for JADE with the JADEL programming language / F. Bergenti, E. Iotti, S. Monica, A. Poggi // Comput. Lang. Syst. Struct. 2017. Vol. 50. P. 142–158. https://doi.org/10.1007/978-3-319-93581-2_9.
8. Development of Mobile Agents With Aglets (A Java Based Tool) / M. Yadav, P. Sethi, D. Juneja, N. Chauhan // International Journal of Innovations & Advancement in Computer Science (IJIACS), Vol. 4, Special Issue, May 2015. P. 245-251.
9. Kravari K., Bassiliades N. A Survey of Agent Platforms // Journal of Artificial Societies and Social Simulation. 2015. Vol. 18 (1), No. 11. P. 1-18.

10. Cynthia N., Gregory M. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms // *Journal of Artificial Societies and Social Simulation*. 2009. Vol. 12, № 2. URL: <http://jasss.soc.surrey.ac.uk/12/2/2.html>. [Electronic resource]. Blank from the screen.

11. Survey: Agent-based Software Technology Under the Eyes of Cyber Security, Security Controls, Attacks and Challenges / B. Alluhaybi1, M. S. Alrahal, A. Alzhrani, V. A. Thayanathan // *International Journal of Advanced Computer Science and Applications (IJACSA)*, King Abdulaziz University (KAU), Jeddah, Saudi Arabia. 2019. Vol. 10, No. 8. P. 211-230.

12. Silva L., Meneguzzi F., Logan B. BDI Agent Architectures: A Survey // *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. Survey track*. – 2020. P. 4914-4921. DOI: <https://doi.org/10.24963/ijcai.2020/684>.

13. Faastlane: Accelerating Function-as-a-Service Workflows / S. Kotni, A. Nayak, V. Ganapathy, A. Basu // *Proceedings of the 2021 USENIX Annual Technical Conference*. July 14–16. 2021. P. 957-971.

14. Faas orchestration of parallel workloads / D. Barcelona-Pons, P. Garcia-Lopez, A. Ruiz, A. Gomez-Gomez, G. Paris, M. Sanchez-Artigas // *Proceedings of the 5th International Workshop on Serverless Computing*. 2019. P. 25-30. <https://doi.org/10.1145/3366623.3368137>.

15. On the FaaS Track: Building stateful distributed applications with serverless architectures / D. Barcelona-Pons, M. Sanchez-Artigas, G. Paris, P. Sutra, P. Garcia-Lopez // *ACM Middleware Conference*. 2019. P. 41-54. <https://doi.org/10.1145/3361525.3361535>.

16. Волчихин В. И., Зинкин С. А., Карамышева Н. С. Организация функционирования облачно-сетевых распределенных вычислительных систем с архитектурой «агенты как сервисы» // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2019. № 4. С. 27–50. <https://doi.org/10.21685/2072-3059-2019-4-3>.

17. Разработка сетевых агентно-базированных приложений на основе метакомпьютерной технологии / В. И. Волчихин, Н. С. Карамышева, А. В. Горынина, С. А. Зинкин // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2021. № 4. С. 3–25. <https://doi.org/10.21685/2072-3059-2021-4-1>.

18. Synthesis of the Functional Architecture of Agent-based Computing Systems when Using Conceptual Models of Artificial Intelligence / S. A. Zinkin, V. I. Volchihin, N. S. Karamysheva, M. S. Jaafar // *2021 International Conference on Engineering Management of Communication and Technology, EMCTECH 2021. Proceedings, Austria, Vienna, 2021*. P. 1-6. <https://doi.org/10.1109/EMCTECH53459.2021.9619174>.

19. Dynamic Topology Transformation of Cloud-Network Computer Systems: Conceptual Level / S. A. Zinkin, V. I. Volchihin, N. S. Karamysheva, M. S. Jaafar // *2020 International Conference on Engineering Management of Communication and Technology*,

ЕМСТЕCH 2020. Proceedings, Austria, Vienna, 2020. P. 1-10. <https://doi.org/10.1109/ЕМСТЕCH49634.2020.9261554>.

20. Зинкин С. А., Джафар М. С. Развитие информационно-коммуникационных инфраструктур распределенных вычислительных систем на основе концепции «Сеть – это компьютер» // Известия Юго-Западного государственного университета. 2018. Т. 22. №4 (79). С. 75-93.

21. Зинкин С. А., Джафар М. С., Карамышева Н. С. Организация функционирования распределенных вычислительных систем с переменной архитектурой в виде облачного сервиса, формируемого по запросу клиента (реализация изменяемой системной архитектуры) // XXI век: итоги прошлого и проблемы настоящего *плюс*. 2018. Т. 7. № 4 (44). С. 54-60.

22. Зинкин С. А., Джафар М. С., Карамышева Н. С. Организация функционирования распределенных вычислительных систем с переменной архитектурой в виде облачного сервиса, формируемого по запросу клиента (концептуальные графы распределенных алгоритмов) // XXI век: итоги прошлого и проблемы настоящего *плюс*. 2018. Т. 7. № 4 (44). С. 136-146.

23. Bellifemine F. L., Caire G., Greenwood D. Developing multi-agent systems with JADE. Wiley, 2007. 300 p.

24. Java Agent Development Environment (JADE). URL: <http://jade.tilab.com/>. [Electronic resource]. Blank from the screen.

References

1. Cohen-almagor R. The Future of the Internet. Academia Letters, 2021. P. 1-5. <https://doi.org/10.20935/AL1962>.

2. Digital economy report 2021. Cross-border data flows and development: for whom the data flow. Geneva, United Nations, 2021. 238 p.

3. Chto takoe metakomp'yuting (kratkii obzor tekhnologii organizatsii graspredelennykh vychislenii v Internetе) [What is metacomputing (a brief overview of distributed computing technologies on the Internet)]. URL: <https://parallel.ru/computers/reviews/metacomputing.html>. [Elektronnyi resurs]. Vid s ekrana. Metod dostupa svobodnyi.

4. World Community Grid: Request for Proposal (RFP). 2020, pp. 1–4. URL: <https://www.worldcommunitygrid.org/bg/rfp.pdf>. [Electronic resource]. Blank from the screen.

5. Youssfi M., Bouattane O., Bakkoury J., Bensalah M. A New Massively Parallel And Distributed Virtual Machine Model Using Mobile Agents. *Conference: IEEE Conference: 4th International Conference on Multimedia Computing and Systems April 14-16, 2014 Marrakech - MOROCCO MSTI Mediterranean Space of Technology and Innovation*. Morocco

Chapter. April 14-16, 2014. pp. 407-414. Marrakech, Morocco. <https://doi.org/10.1109/ICMCS.2014.6911306>.

6. Youssfi M., Bouattane O., Bensalah M. A Parallel Computational Model Based on Mobile Agents for High Performance Computing. *Contemporary Engineering Sciences*, 2015, vol. 8, no. 15, pp. 677-698. <http://dx.doi.org/10.12988/ces.2015.55153>.

7. Bergenti F., Iotti E., Monica S., Poggi A. Agent-oriented model-driven development for JADE with the JADEL programming language. *Comput. Lang. Syst. Struct.*, 2017, vol. 50, pp. 142–158. https://doi.org/10.1007/978-3-319-93581-2_9.

8. Yadav M., Sethi P., Juneja D., Chauhan N. Development of Mobile Agents With Agents (A Java Based Tool). *International Journal of Innovations & Advancement in Computer Science (IJACS)*, may 2015, vol. 4, Special Issue, pp. 245-251.

9. Kravari K., Bassiliades N. A Survey of Agent Platforms. *Journal of Artificial Societies and Social Simulation*, 2015, vol. 18 (1), no. 11, pp. 1-18.

10. Cynthia N., Gregory M. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 2009, vol. 12, no. 2. URL: <http://jasss.soc.surrey.ac.uk/12/2/2.html>. [Electronic resource]. Blank from the screen.

11. Alluhaybil B., Alrahhal M. S., Alzhrani A., Thayananthan V. A Survey: Agent-based Software Technology Under the Eyes of Cyber Security, Security Controls, Attacks and Challenges. *International Journal of Advanced Computer Science and Applications (IJACSA)*, King Abdulaziz University (KAU), Jeddah, Saudi Arabia, 2019, vol. 10, no. 8. P. 211-230.

12. Silva L., Meneguzzi F., Logan B. BDI Agent Architectures: A Survey. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. Survey track*, 2020, pp. 4914-4921. <https://doi.org/10.24963/ijcai.2020/684>.

13. Kotni S., Nayak A., Ganapathy V., Basu A. Faastlane: Accelerating Function-as-a-Service Workflows. *Proceedings of the 2021 USENIX Annual Technical Conference*. July 14–16, 2021, pp. 957-971.

14. Barcelona-Pons D., Garcia-Lopez P., Ruiz A., Gomez-Gomez A., Paris G., Sanchez-Artigas M. Faas orchestration of parallel workloads. *Proceedings of the 5th International Workshop on Serverless Computing*, 2019, pp. 25-30. <https://doi.org/10.1145/3366623.3368137>.

15. Barcelona-Pons D., Sanchez-Artigas M., Paris G., Sutra P., Garcia-Lopez P. On the FaaS Track: Building stateful distributed applications with serverless architectures. *ACM Middleware Conference*, 2019, pp. 41-54. <https://doi.org/10.1145/3361525.3361535>.

16. Volchikhin V. I., Zinkin S. A., Karamysheva N. S. Organizatsiya funktsionirovaniya oblachno-setevykh raspredelennykh vychislitel'nykh sistem s arkhitekturoi «agenty kak servisy» [Organization of functioning of cloud-network distributed computing systems

with the architecture "agents as services"]. *Izvestiya vysshikh uchebnykh zavedenii. Povolzhskii region. Tekhnicheskie nauki = Izvestiya vyshev uchebnykh zavedeniy. Volga region. Technical sciences*, 2019. no. 4, pp. 27–50. <https://doi.org/10.21685/2072-3059-2019-4-3>

17. Volchikhin V. I., Karamysheva N. S., Gorynina A. V., Zinkin S. A. Razrabotka setevykh agentno-bazirovannykh prilozhenii na osnove metakomp'yuternoi tekhnologii [Development of network agent-based applications based on metacomputer technology]. *Izvestiya vysshikh uchebnykh zavedenii. Povolzhskii region. Tekhnicheskie nauki = Izvestia of higher educational institutions. Volga region. Technical Sciences*. 2021, no. 4, pp. 3–25. <https://doi.org/10.21685/2072-3059-2021-4-1>.

18. Zinkin S. A., Volchihin V. I., Karamysheva N. S., Jaafar M. S. Synthesis of the Functional Architecture of Agent-based Computing Systems when Using Conceptual Models of Artificial Intelligence. *2021 International Conference on Engineering Management of Communication and Technology, EMCTECH 2021. Proceedings*, Austria, Vienna, 2021, pp. 1-6. <https://doi.org/10.1109/EMCTECH53459.2021.9619174>.

19. Zinkin S. A., Volchihin V. I., Karamysheva N. S., Jaafar M. S. Dynamic Topology Transformation of Cloud-Network Computer Systems: Conceptual Level. *2020 International Conference on Engineering Management of Communication and Technology, EMCTECH 2020. Proceedings*, Austria, Vienna, 2020, pp. 1-10. <https://doi.org/10.1109/EMCTECH49634.2020.9261554>.

20. Zinkin S. A., Dzhafer M. S. Razvitiye informatsionno-kommunikatsionnykh infrastruktur raspredelennykh vychislitel'nykh sistem na osnove kontseptsii «Set' – eto komp'yuter» [Development of information and communication infrastructure of distributed computing systems based on the concept "Network is a computer"]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*, 2018, vol. 22, no. (79), pp. 75-93.

21. Zinkin S. A., Dzhafer M. S., Karamysheva N. S. Organizatsiya funktsionirovaniya raspredelennykh vychislitel'nykh sistem s peremennoi arkhitekturoi v vide oblachnogo servisa, formiruемого po zaprosu klienta (realizatsiya izmenyaemoi sistemnoi arkhitektury) [Organization of the functioning of distributed computing systems with variable architecture in the form of a cloud service formed at the request of the client (implementation of a changeable system architecture)]. *XXI vek: itogi proshlogo i problemy nastoyashchego plyus = XXI century: the results of the past and the problems of the present plus*, 2018, vol. 7, no. 4 (44), pp. 54-60.

22. Zinkin S. A., Dzhafer M. S., Karamysheva N. S. Organizatsiya funktsionirovaniya raspredelennykh vychislitel'nykh sistem s peremennoi arkhitekturoi v vide oblachnogo servisa, formiruемого po zaprosu klienta (kontseptual'nye grafy raspredelennykh algoritmov) [Organization of functioning of distributed computing systems with variable architecture in the form of a cloud service formed at the request of the client (conceptual graphs of distrib-

uted algorithms]. *XXI vek: itogi proshlogo i problemy nastoyashchego plyus = XXI century: results of the past and problems of the present plus*. 2018, vol. 7, no. 4 (44), pp. 136-146.

23. Bellifemine F. L., Caire G., Greenwood D. Developing multi-agent systems with JADE. Wiley, 2007, 300 p.

24. Java Agent Development Environment (JADE). URL: <http://jade.tilab.com/>. Blank from the screen.

Информация об авторах / Information about the Authors

Карамышева Надежда Сергеевна, кандидат технических наук, доцент, Пензенский государственный университет, г. Пенза, Российская Федерация, e-mail: karamyshevans@yandex.ru

Nadezhda S. Karamysheva, Cand. of Sci. (Engineering), Associate Professor, Penza State University, Penza, Russian Federation, e-mail: karamyshevans@yandex.ru

Свищев Данил Сергеевич, магистрант, Пензенский государственный университет, г. Пенза, Российская Федерация, e-mail: vt@pnzgu.ru

Danil S. Svishchev, Master Student, Penza State University, Penza, Russian Federation, e-mail: vt@pnzgu.ru

Попов Константин Владимирович, кандидат технических наук, доцент, Пензенский государственный университет, г. Пенза, Российская Федерация, e-mail: vt@pnzgu.ru

Konstantin V. Popov, Cand. of Sci. (Engineering), Associate Professor, Penza State University, Penza, Russian Federation, e-mail: vt@pnzgu.ru

Зинкин Сергей Александрович, доктор технических наук, профессор, Пензенский государственный университет, г. Пенза, Российская Федерация, e-mail: zsa49@yandex.ru

Sergey A. Zinkin, Dr. of Sci. (Engineering), Professor, Penza State University, Penza, Russian Federation, e-mail: zsa49@yandex.ru