

АЛГОРИТМ ПЕРЕРАСПРЕДЕЛЕНИЯ РОЛЕЙ В БЕСПРОВОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРАХ

Д. С. Неструев*, Д. Б. Борзов

Юго-Западный государственный университет, Курск, Россия

** nestruev98@mail.ru*

Аннотация. Описана алгоритмическая структура беспроводного вычислительного кластера, основанная на динамическом распределении ролей между его элементами. В результате проведенного анализа выявлено, что динамическое перераспределение ролей в беспроводном вычислительном кластере во время выполнения задач повышает отказоустойчивость. Предложен алгоритм, который может стать основой для разработки беспроводных кластерных моделей с повышенной отказоустойчивостью. Эффективность алгоритма показана на примере сравнительного анализа. Для непосредственного использования алгоритм необходимо доработать и настроить в соответствии с техническими требованиями.

Ключевые слова: беспроводной вычислительный кластер, алгоритм динамического распределения ролей, повышение отказоустойчивости, орбитальная модель, эффективность алгоритма

Ссылка для цитирования: Неструев Д. С., Борзов Д. Б. Алгоритм перераспределения ролей в беспроводных вычислительных кластерах // Изв. вузов. Приборостроение. 2024. Т. 67, № 10. С. 844–852. DOI: 10.17586/0021-3454-2024-67-10-844-852.

ALGORITHM FOR REDISTRIBUTING ROLES IN WIRELESS COMPUTING CLUSTERS

D. S. Nestruev*, D. B. Borzov

Southwest State University, Kursk, Russia

** nestruev98@mail.ru*

Abstract. The algorithmic structure of a wireless computing cluster based on the dynamic distribution of roles between its elements is described. As a result of performed analysis, it is revealed that the dynamic redistribution of roles in a wireless computing cluster during task execution increases fault tolerance. An algorithm is proposed that can become the basis for the development of wireless cluster models with increased fault tolerance. The algorithm effectiveness is demonstrated by an example of comparative analysis. For direct use, the algorithm must be modified and configured in accordance with technical requirements.

Keywords: wireless computing cluster, dynamic role allocation algorithm, increased fault tolerance, orbital model, algorithm efficiency

For citation: Nestruev D. S., Borzov D. B. Algorithm for redistributing roles in wireless computing clusters. *Journal of Instrument Engineering*. 2024. Vol. 67, N 10. P. 844–852 (in Russian). DOI: 10.17586/0021-3454-2024-67-10-844-852.

Введение. Для быстроразвивающейся области распределенных вычислений беспроводные вычислительные кластеры (БВК) стали жизненно важным компонентом, особенно в сценариях, где использование проводных соединений нецелесообразно или невозможно. Эти кластеры состоят из множества беспроводных устройств, взаимодействующих при решении вычислитель-

ных задач. Эффективное перераспределение ролей между этими устройствами имеет решающее значение для оптимизации производительности, повышения отказоустойчивости и управления использованием ресурсов [1]. В настоящей работе рассматривается алгоритм, разработанный для перераспределения ролей в беспроводных вычислительных кластерах, и выполняется сравнительный анализ их эффективности.

Перераспределение ролей в беспроводных вычислительных кластерах предполагает динамическое назначение задач различным узлам (устройствам) на основе различных критериев, таких как вычислительная мощность, время автономной работы, подключение к сети и текущая рабочая нагрузка. Эффективные алгоритмы перераспределения обеспечивают оптимальную работу кластера, адаптируясь к изменениям в среде или рабочей нагрузке без „ручного вмешательства“.

В широком смысле алгоритмы динамического перераспределения ролей принято разделять на [2]: централизованные; децентрализованные; гибридные.

В централизованных алгоритмах при перераспределении ролей полагаются на один управляющий узел, отслеживающий состояние кластера. Этот узел может самостоятельно, по мере необходимости, принимать решения о перераспределении ролей [3, 4]. Преимущества таких алгоритмов заключаются в:

- упрощении процесса принятия решения;
- простоте процесса внедрения и управления.

Отрицательные особенности:

- существование единой точки отказа;
- проблема с масштабируемостью по мере увеличения размера кластера.

В децентрализованных алгоритмах, напротив, процесс принятия решения распределяется между всеми элементами кластера. Каждый узел, в свою очередь, принимает свои собственные решения о перераспределении ролей на основе локальной информации [4]. Преимущества таких алгоритмов состоят в:

- масштабируемости;
- отсутствии отдельной точки отказа.

Также им свойственны такие недостатки, как:

- сложность принятия решений и координации;
- возможность возникновения несогласованных состояний в кластере.

Для эффективной балансировки нагрузки на элементы БВК и повышения его гибкости при адаптации к различным сценариям применяют гибридные алгоритмы, объединяющие положительные стороны централизованного и децентрализованного алгоритмов: например, наличие центрального координатора для глобальных решений и децентрализованных механизмов для локальных корректировок.

В ходе настоящей работы рассмотрены способы перераспределения и резервирования ролей в БВК, проанализированы алгоритмы перераспределения ролей в беспроводных вычислительных кластерах, которые направлены на обеспечение эффективности вычислений, масштабируемости, устойчивости к отказам и энергоэффективности. В результате исследования реализовано решение, которое совмещает в себе простоту процесса внедрения и управления, унаследованную от централизованных алгоритмов, масштабируемость и отсутствие единой точки отказа — от децентрализованных, эффективную балансировку нагрузки на элементы — от гибридных. Разработанное решение применимо в широком спектре отраслей, таких как оборонная промышленность, с целью обеспечения безопасной и гибкой связи между подразделениями, мониторинга окружающей среды, сбора данных из удаленных и труднодоступных районов. Применение решения в сфере здравоохранения позволит анализировать медицинские данные в режиме реального времени, что повысит качество обслуживания пациентов.

Материалы и методы. Основанный на ролях подход к построению сетей с возможностью динамической реконфигурации является передовой концепцией в области сетевых технологий, особенно актуальной для разработки беспроводных вычислительных кластеров и систем

интернета вещей (IoT). Такой подход позволяет адаптировать сеть к изменяющимся условиям окружающей среды и эксплуатационным требованиям, тем самым повышая ее надежность, эффективность и масштабируемость [5].

Основные принципы ролевого подхода:

1) динамическое распределение ролей — каждый узел может динамически изменять свою роль (хост, резервный хост, накопитель, резервный накопитель или датчик — рабочая машина, или „пчела“) в зависимости от текущих потребностей сети, ее состояния и состояния окружающей среды. Наглядный пример разделения на роли представлен на рис. 1;

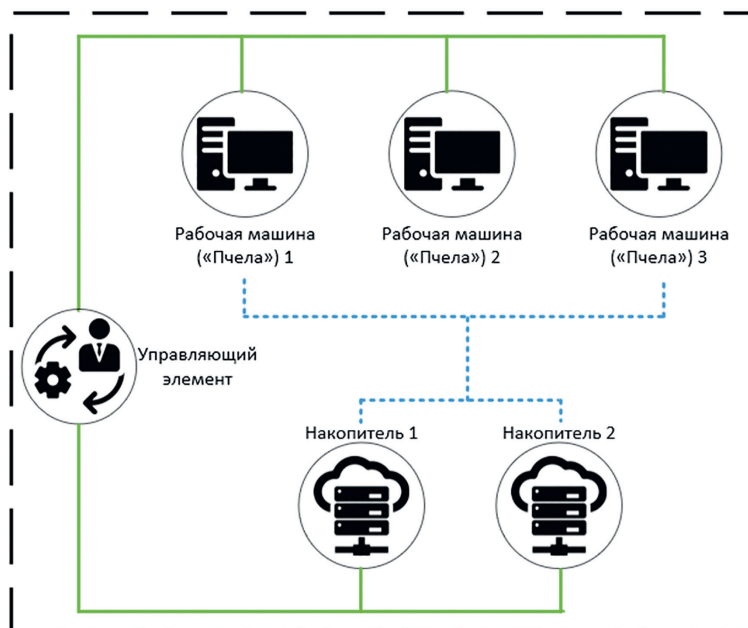


Рис. 1

2) адаптивность и масштабируемость — система способна автоматически адаптироваться к таким изменениям, как добавление или удаление узлов, изменение условий связи или требований к приложениям. Это обеспечивает гибкость при масштабировании системы [6]. Решение о переназначении ролей может быть представлено функцией

$$f(S) \rightarrow R^N,$$

где R^N — набор ролей для всех узлов в сети;

3) устойчивость к отказам — ролевой подход способствует повышению устойчивости сети к отказам отдельных узлов. Функции узла, в случае его выхода из строя, могут быть переназначены другим узлам. Оценивать вероятность сохранения работоспособности системы после возникновения некоторого числа отказов рекомендуется с учетом зависимости отказов от выполнения узлами различных функций (ролей). При этом могут быть использованы подходы, предложенные в работах [7, 8].

В настоящей статье представлены разработанные авторами алгоритмы функционирования основных элементов беспроводного вычислительного кластера.

По включению питания каждый процессор каждого из N элементов кластера выполняет:

- 1) общую программу загрузки из внутренней флеш-памяти со стартового адреса;
- 2) определение первичной роли элемента в кластере;
- 3) загрузку из флеш-памяти в ОЗУ программы, соответствующей роли элемента M кластера;
- 4) считывание контрольной суммы из конечного блока программы и сравнение с рассчитанной для контроля целостности информации;

- 5) рестарт ролевой программы из ОЗУ на выполнение;
- 6) анализ возможного изменения роли элемента путем считывания переменной *TempNum*;
- 7) в случае изменения роли элемента БВК — перезагрузка на исполнение соответствующей ролевой программы.

Стартовый алгоритм функционирования элементов кластера представлен на рис. 2.

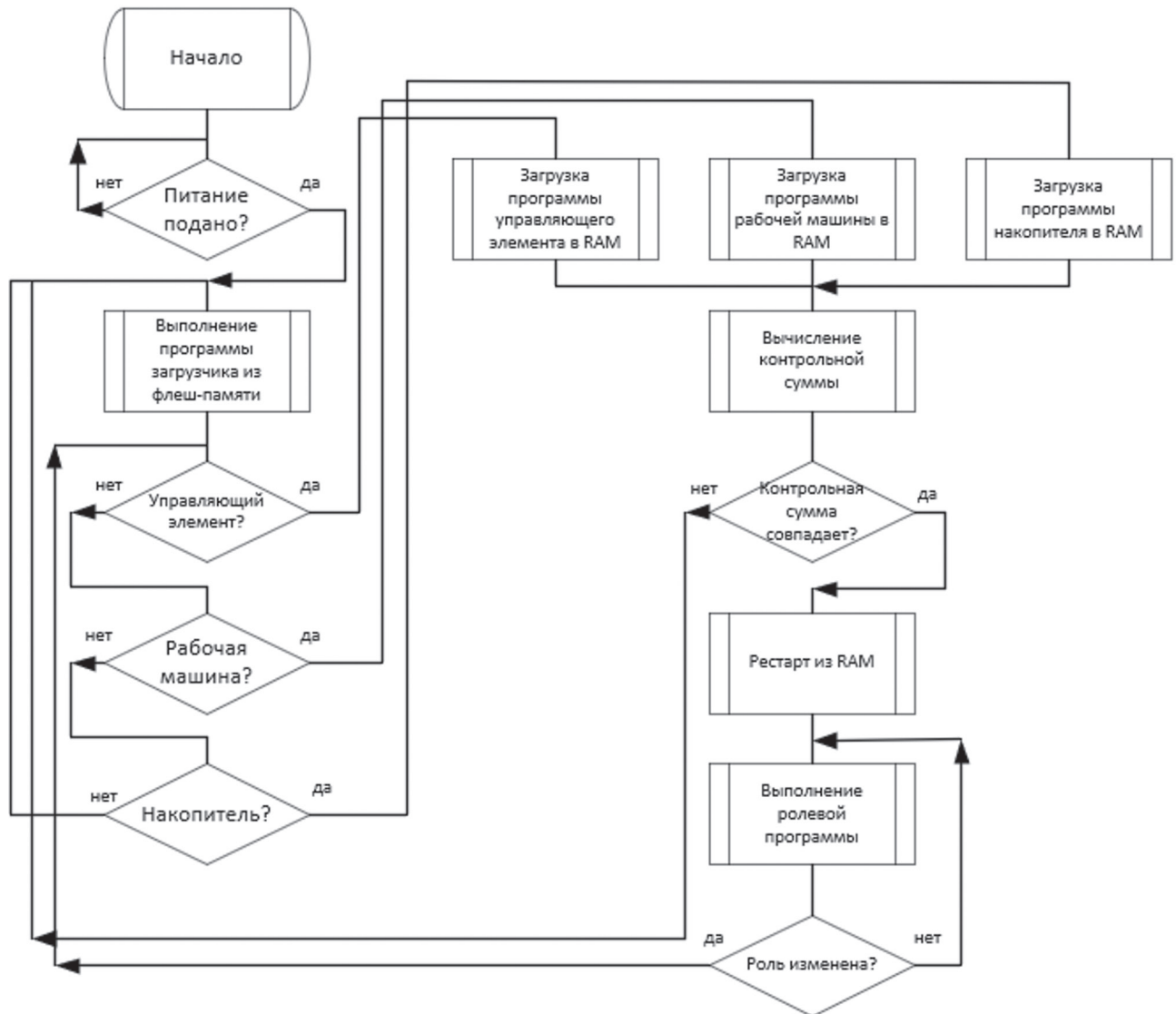


Рис. 2

Следовательно, при подаче электропитания каждый элемент системы выполняет определенные универсальные процедуры, ведущие к инициализации разнообразных функциональных программ [9–11].

Алгоритм функционирования управляющего элемента содержит такие этапы, как:

- 1) загрузка рабочей программы сервера;
- 2) определение собственных координат и опрос подмножества адресов системы для определения количества узлов, готовых к работе;
- 3) опрос готовых к работе узлов для определения технических характеристик и местоположения;
- 4) согласно полученной информации хост формирует начальный граф $C(M, K)$ с N вершинами, $M = \{1, \dots, N\}$, K — расстояние между вершинами;
- 5) по результатам анализа информации происходит распределение ролей в БВК.

На этапе выполнения ролевой программы циклически реализуется ряд функций, которые заложены в программе управляющего элемента. Алгоритм управляющего элемента представлен на рис. 3. „Самоконтроль“ предполагает прохождение элементом внутренних тестов исправности. „Арбитраж“ подразумевает сравнение собственных характеристик с характеристиками резервного элемента, что позволяет определить возможность делегирования управляющих функций [12, 13].

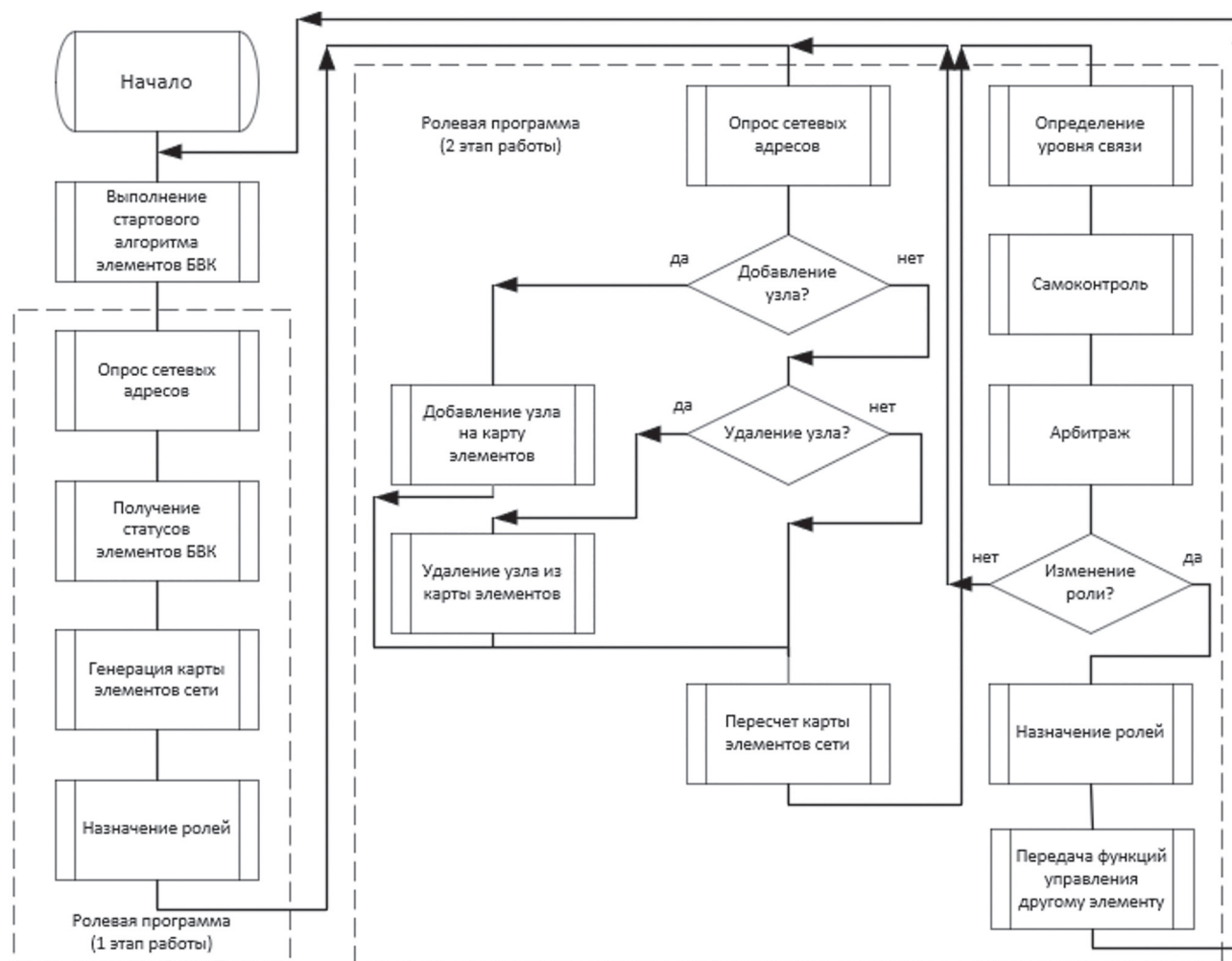


Рис. 3

Алгоритм функционирования рабочей машины (рис. 4) содержит такие этапы:

- 1) получение координаты элемента по GPS после старта рабочей программы;
 - 2) получение опросного сигнала от управляющего элемента;
 - 3) после выполнения процедуры опроса — получение командного слова от управляющего элемента;
 - 4) при изменении роли элемента — выполнение загрузки рабочей программы и рестарт.
- Для общения элементов кластера используются команды.

В команде управляющего элемента рабочей машины (рис. 5) содержится следующая информация:

- *Work* — бит, отвечающий за режим работы;
- *TempNum* — переменная, означающая текущую роль элемента в системе;
- *KvCount* — счетчик квитирования, показывающий, сколько раз „пчела“ должна передать ответ;
- *AdmNk*, *AdrRnk*, *AdrRh* — адреса основных элементов БВК;
- *INT* — поле векторного прерывания;

- *Резерв* — поле резерва;
- *КС* — контрольная сумма.

Команда ответа рабочей машины (рис. 6) состоит из следующих полей:

- *Заряд* — уровень оставшегося заряда (в процентах);
- *NMEA* — координаты элемента;
- *Бит исправности* — результат выполнения встроенного самоконтроля;
- *KvCount* — счетчик ответов, записывается управляющим элементом;
- *Num* — роль элемента в системе по умолчанию;
- *Резерв* — резервная ячейка.
- *КС* — контрольная сумма.

Алгоритм работы накопителя отличается от алгоритма функционирования „пчелы“ только выполняемой ролевой программой, а также командами и статусами. Командное слово содержит те же поля, что и для „пчелы“, но отличается наличием ячейки „Индикатор свободной памяти“, которая показывает свободное место в накопителе для записи информации. Ответная команда статуса накопителя для управляющего элемента приведена на рис. 7.

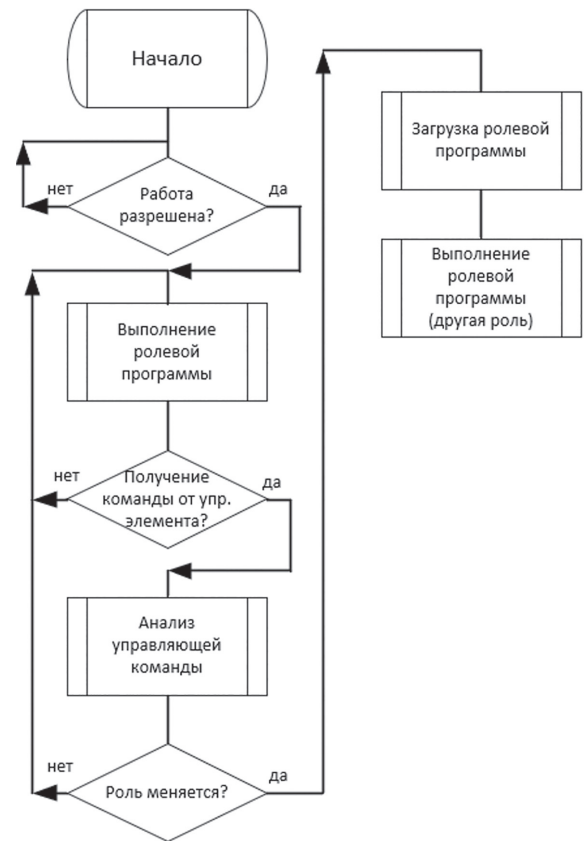


Рис. 4

Work	TempNum	KvCount	AdrNk	AdrRnk	AdrRh	INT	Резерв	КС
------	---------	---------	-------	--------	-------	-----	--------	----

Рис. 5

Заряд	NMEA	Бит исправности	KvCount	Num	Резерв	КС
-------	------	-----------------	---------	-----	--------	----

Рис. 6

Заряд	NMEA	Бит исправности	KvCount	Num	Индекс свободной памяти	Резерв	КС
-------	------	-----------------	---------	-----	-------------------------	--------	----

Рис. 7

Результаты и их обсуждение. Проведенные экспериментальные исследования позволяют лучше понять факторы, влияющие на время выполнения алгоритма реконфигурации беспроводного вычислительного кластера: размер кластера (количество элементов) и внесение дополнительных параметров (настроек) для ролей. Чтобы минимизировать ошибку загрузки центрального процессора, вызванную использованием сторонних программ, каждый сценарий был исполнен 100 раз. После этого было рассчитано среднее значение из этих 100 итераций.

Влияние размера кластера на среднее время выполнения оценивалось для 7 (~7,36 мс), 10 (~10,64), 20 (~50,47), 25 (~247,34) элементов. Дополнительные параметры для ролей не устанавливались.

Исходя из анализа результатов был сделан вывод, что рост размеров кластера ведет к увеличению времени, необходимого для выполнения алгоритма реконфигурации. Это обусловлено тем, что с увеличением числа элементов для опроса возрастает и количество вычислений для

определения суммарных расстояний между этими элементами. Следовательно, больший объем вычислений требует больше времени.

Чтобы оценить, как дополнительные параметры ролей влияют на систему, было принято решение использовать средний размер элементов в кластере, состоящем из 10 элементов. При этом все элементы находились в одинаковых начальных позициях. В качестве исследуемых параметров были выбраны объем внутреннего хранилища для роли „Накопитель“ и производительность процессора для роли „Управляющий элемент“. Среднее время выполнения при одном дополнительном параметре составило ~24,36 мс, при двух — 67,72, без дополнительных параметров — 14,43.

На основании полученных результатов сделан вывод, что добавление элементов и внесение дополнительных параметров повышает сложность алгоритма и увеличивает время выполнения алгоритма реконфигурации.

Стоит отметить, что увеличение числа элементов и добавление дополнительных параметров увеличивает нагрузку на центральный процессор элемента с текущей ролью „Управляющий элемент“. Это необходимо учитывать при построении системы беспроводного вычислительного кластера с помощью алгоритма реконфигурации ролей.

С целью визуализации работы алгоритма написана программа для моделирования работы. Интерфейс программы представлен на рис. 8.

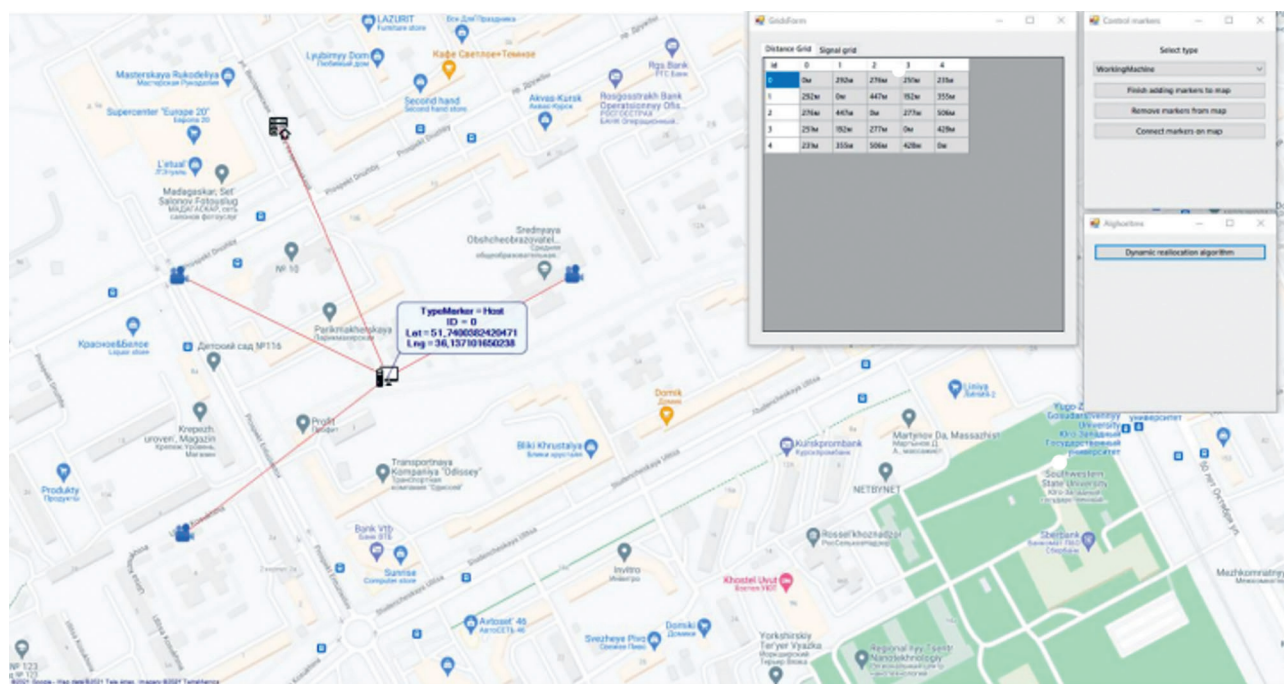


Рис. 8

В настоящей работе описывается алгоритмическая структура беспроводного вычислительного кластера, основанная на динамическом распределении ролей между его элементами. Основой для распределения элементов беспроводного кластера служит концепция, согласно которой каждый элемент кластера занимает свою „орбиту“ и выполняет определенные функции. Алгоритм, предложенный в статье, может быть использован при разработке программного обеспечения для создания собственной архитектуры беспроводного вычислительного кластера.

Использование этих алгоритмов для беспроводных вычислительных кластеров способствует повышению их надежности в различных условиях. Предложенный алгоритм может быть интегрирован с уже существующими системами для создания умной системы управления беспроводным кластером [14—18].

СПИСОК ЛИТЕРАТУРЫ

1. Муравьев К. А., Алябьев И. О., Синютина Д. С., Шушуев А. И. Алгоритмическое проектирование беспроводных сенсорных сетей // Надежность и качество сложных систем. 2020. № 4(32). С. 79–89. DOI 10.21685/2307-4205-2020-4-9.
2. Павлов А. Н., Павлов Д. А., Умаров А. Б. Метод оценивания показателей живучести бортовых систем малых космических аппаратов в условиях изменяющихся режимов функционирования и деструктивных воздействий // Труды МАИ. 2021. № 120. С. 92–101.
3. Алешин Е. Н., Зиновьев С. В., Копкин Е. В., Осипенко С. А., Павлов А. Н., Соколов Б. В. Системный анализ организационно-технических систем космического назначения. СПб: ВКА им. А.Ф. Можайского, 2018. С. 357–363.
4. Татарникова Т. М., Бимбетов Ф., Горина Е. В. Алгоритм роя пчел выбора головных узлов кластеров беспроводной сенсорной сети // Изв. СПбГЭТУ „ЛЭТИ“. 2022. Т. 15, № 3. С. 15–22.
5. Борзов Д. Б., Чернышев А. А., Сизов А. С., Соколова Ю. В. Методика и алгоритм построения вычислительной сети на основе беспроводного протокола // Труды МАИ. 2021. № 121. DOI: 10.34759/trd-2021-121-20.
6. Глушков В. Н., Миронов М. Е. Повышение производительности беспроводных кластеров за счет оптимизации ролевого распределения // Вычислительные технологии. 2024. № 6. С. 250–265.
7. Bogatyrev V. A., Bogatyrev S. V., Bogatyrev A. V. The Probability of Timely Redundant Service in a Two-Level Cluster of a Flow of Requests that is Heterogeneous in Functionality and Allowable Delays // Communications in Computer and Information Science. 2023. Vol. 1748. P. 122–134.
8. Богатырев В. А., Богатырев А. В. Оптимизация резервированного распределения запросов в кластерных системах реального времени // Информационные технологии. 2015. Т. 21, № 7. С. 495–502.
9. Татарникова Т. М., Бимбетов Ф., Горина Е. В. Алгоритм энергоэффективного взаимодействия узлов беспроводной сенсорной сети // Труды МАИ. 2022. № 131. С. 55–63.
10. Галушина Т. Ю., Николаева Е. А., Красавин Д. С., Летнер О. Н. Применение методов машинного обучения для классификации резонансного движения астероидов // Вестник Томского государственного университета. Математика и механика. 2022. № 76. С. 87–100.
11. Богатырев В. А., Богатырев С. В., Богатырев А. В. Оценка готовности компьютерной системы к своевременному обслуживанию запросов при его совмещении с информационным восстановлением памяти после отказов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 3(145). С. 608–617.
12. Красавин Д. С., Александрова А. Г., Томилова И. В. Применение искусственных нейронных сетей в исследовании динамической структуры околоземного орбитального пространства // Изв. вузов. Физика. 2021. Т. 64, № 10. С. 38–43.
13. Татарникова Т. М. Статистические методы исследования сетевого трафика // Информационно-управляющие системы. 2018. № 5(96). С. 35–43.
14. Bogatyrev V. A., Bogatyrev A. V., Bogatyrev S. V. Redundant servicing of a flow of heterogeneous requests critical to the total waiting time during the multi-path passage of a sequence of info-communication nodes // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2020. Vol. 12563. P. 100–112.
15. Basford P. J., Johnston S. J., Perkins C. S., Garnock-Jones T., Tso F. P., Pezaros D., Mullins R. D., Yoneki E., Singer J., Cox S. J. Performance analysis of single board computer clusters // Future Generation Computer Systems. 2020. Vol. 102. P. 278–291.
16. Монтлевич В. М., Попов А. Д. Математическая модель размещения виртуальных машин на физических серверах вычислительных сетей // Вестник Самарского университета. Экономика и управление. 2020. Т. 11, № 2. С. 125–131.
17. Воробьев А. А., Данг С. Б. Формализация задач оптимизации размещения виртуальных машин и распределения сетевых ресурсов в облачной вычислительной системе // Системы управления и информационные технологии. 2016. № 3. С. 28–32.
18. Пальчевский Е. В., Халиков А. Р. Равномерное распараллеливание сетевой нагрузки по физическим серверам кластера // „Актуальные вопросы современных научных исследований“. Матер. Междунар. науч.-практич. конф. М., 2017. № 1. С. 119–122.

СВЕДЕНИЯ ОБ АВТОРАХ

Денис Сергеевич Неструев

— аспирант; Юго-Западный государственный университет, кафедра вычислительной техники; E-mail: nestruiev98@mail.ru

Дмитрий Борисович Борзов

— д-р техн. наук, профессор; Юго-Западный государственный университет, кафедра вычислительной техники; доцент; E-mail: borzovdb@mail.ru

Поступила в редакцию 22.04.24; одобрена после рецензирования 16.06.24; принята к публикации 23.08.24.

REFERENCES

1. Muravyev K.A., Alyabyev I.O., Sinyutina D.S., Shushuev A.I. *Reliability and quality of complex systems*, 2020, no. 4(32), pp. 79–89, DOI 10.21685/2307-4205-2020-4-9. (in Russ.)
2. Pavlov A.N., Pavlov D.A., Umarov A.B. *Trudy MAI*, 2021, no. 120, pp. 92–101. (in Russ.)
3. Aleshin E.N., Zinoviev S.V., Kopkin E.V., Osipenko S.A., Pavlov A.N., Sokolov B.V. *Sistemnyy analiz organizatsionno-tekhnicheskikh sistem kosmicheskogo naznacheniya* (System Analysis of Organizational and Technical Systems for Space Purposes), St. Petersburg, 2018. pp. 357–363. (in Russ.)
4. Tatarnikova T.M., Bimbetov F., Gorina E.V. *Izvestiya SPbSETU "LETI"*, 2022, no. 3(15), pp. 15–22. (in Russ.)
5. Borzov D.B., Chernyshev A.A., Sizov A.S., Sokolov Yu.V. *Trudy MAI*, 2021, no. 121, DOI: 10.34759/trd-2021-121-20. (in Russ.)
6. Glushkov V.N., Mironov M.E. *Computing Technologies*, 2024, no. 6, pp. 250–265. (in Russ.)
7. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. *Communications in Computer and Information Science*, 2023, vol. 1748, pp. 122–134.
8. Bogatyrev V.A., Bogatyrev A.V. *Information Technologies*, 2015, no. 7(21), pp. 495–502. (in Russ.)
9. Tatarnikova T.M., Bimbetov F., Gorina E.V. *Trudy MAI*, 2022, no. 131, pp. 55–63. (in Russ.)
10. Galushina T.Yu., Nikolaeva E.A., Krasavin D.S., Letner O.N. *Bulletin of Tomsk State University. Mathematics and mechanics*, 2022, no. 76, pp. 87–100. (in Russ.)
11. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. *Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics*, 2023, no. 3(23), pp. 608–617. (in Russ.)
12. Krasavin D.S., Alexandrova A.G., Tomilova I.V. *News of universities. Physics*, 2021, no. 10(64), pp. 38–43. (in Russ.)
13. Tatarnikova T.M. *Information and control systems*, 2018, no. 5(96), pp. 35–43. (in Russ.)
14. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12563, pp. 100–112. (in Russ.)
15. Basford P.J., Johnston S.J., Perkins C.S., Garnock-Jones T., Tso F.P., Pezaros D., Mullins R.D., Yoneki E., Singer J., Cox S.J. *Future Generation Computer Systems*, 2020, vol. 102, pp. 278–291.
16. Montlevich V.M., Popov A.D. *Bulletin of Samara University. Economics and management*, 2020, no. 2(11), pp. 125131. (in Russ.)
17. Vorobyov A.A., Dang S.B. *Management systems and information technologies*, 2016, no. 3, pp. 28–32. (in Russ.)
18. Palchevsky E.V., Khalikov A.R. *Aktual'nyye voprosy sovremennykh nauchnykh issledovaniy* (Uniform Parallelization of Network Load across Physical Cluster servers — Topical Issues of Modern Scientific Research), Materials of the International Scientific and Practical Conference, Moscow, 2017, no. 1, pp. 119–122.

DATA ON AUTHORS

- | | | |
|--------------------------|---|--|
| Denis S. Nestruev | — | Post-Graduate Student; Southwest State University, Department of Computer Engineering; E-mail: nestruev98@mail.ru |
| Dmitriy B. Borzov | — | Dr. Sci., Professor; Southwest State University, Department of Computer Engineering; Associate Professor; E-mail: borzovdb@mail.ru |

Received 22.04.24; approved after reviewing 16.06.24; accepted for publication 23.08.24.