

Научная статья
УДК 004.725
DOI: [10.34759/trd-2023-128-19](https://doi.org/10.34759/trd-2023-128-19)

МОДЕЛЬ РЕОРГАНИЗАЦИИ ЭЛЕМЕНТОВ БЕСПРОВОДНОГО ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА С ОРБИТАЛЬНЫМ РАСПОЛОЖЕНИЕМ ЭЛЕМЕНТОВ

Денис Сергеевич Неструев^{1✉}, Дмитрий Борисович Борзов²

^{1,2}Юго-Западный государственный университет, ЮЗГУ,

Курск, Россия

¹nestrujev98@mail.ru✉

²borzovdb@mail.ru

Аннотация. Кластер - это модульная многопроцессорная система, созданная на базе стандартных вычислительных узлов, соединенных высокоскоростной коммуникационной средой. Типичный кластер представляет из себя множество ЭВМ или процессорных ядер под централизованным управлением, которые пользователь воспринимает как единое целое. Основной характеристикой вычислительного кластера является отказоустойчивость. Для обеспечения большей живучести элементы кластера должны иметь возможность перемещения в пространстве и быть зарезервированы. В первую очередь это касается управляющего элемента (Хоста) и узлов, в которых хранится накопленная информация (Хранилище). Научной новизной является метод орбитальной динамической реконфигурации ролей. Этот метод

позволяет распределить элементы кластера по орбитам относительно управляющего элемента, что в свою очередь обеспечивает лучшее соединение с остальными элементами кластера. Сравнение производится по результатам моделирования. Моделирование выполняется в разработанной программе для ЭВМ. В качестве параметров сравнения используется параметр времени работы беспроводного вычислительного кластера в нештатной ситуации (исчезновение сигнала между элементами кластера, изменение положение элемента кластера и т.п.). В статье рассматривается алгоритмическая модель инициализации беспроводного вычислительного кластера с динамической реконфигурацией ролей орбитальным методом, что существенно повышает отказоустойчивость кластера. В данной работе был проведен анализ и сравнение результатов работы описанного алгоритма с существующими. В результате проведенного анализа выявлено, что динамическая реконфигурация ролей позволяет повысить отказоустойчивость беспроводного вычислительного кластера благодаря тому, что управляющим элементом БВК может выступать любой из элементов.

Ключевые слова: беспроводной вычислительный кластер, алгоритмическая модель, отказоустойчивость, граф, алгоритм инициализации

Для цитирования: Неструев Д.С., Борзов Д.Б. Модель реорганизации элементов беспроводного вычислительного кластера с орбитальным расположением элементов

// Труды МАИ. 2023. № 128. DOI: [10.34759/trd-2023-128-19](https://doi.org/10.34759/trd-2023-128-19)

Original article

MODEL OF REORGANIZATION OF ELEMENTS OF A WIRELESS COMPUTING CLUSTER WITH AN ORBITAL ARRANGEMENT OF ELEMENTS

Denis S. Nestruev¹✉, Dmitry B. Borzov²,

^{1,2}Southwestern State University,

Kursk, Russia

¹nestruev98@mail.ru✉

²borzovdb@mail.ru

Abstract. A cluster is a modular multiprocessor system created on the ground of standard computing nodes connected by a high-speed communication medium. A typical cluster is a set of computers or processor cores under centralized control, which the user perceives as a single entity. The main characteristic of a computing cluster is fault tolerance. To ensure greater survivability, cluster elements must be able to move in space and be reserved. This concerns, first of all, the control element (Host) and the nodes in which the accumulated information is stored (Storage). Scientific novelty consists in the method of orbital dynamic reconfiguration of roles. This method allows distributing the cluster elements in orbits relative to the control element, which, in turn, ensures a better connection with the rest of the cluster elements. Comparison is performed by the simulation results. Modeling is carried out with the developed computer program. Parameter of the wireless computing cluster running time in an extraordinary situation (disappearance of a signal between cluster elements, change in the cluster element position, etc.) is used as the comparison parameter.

The article considers an algorithmic model for initializing a wireless computing cluster with dynamic reconfiguration of roles by the orbital method, which significantly increases the fault tolerance of the cluster. The authors performed the analysis and comparison of the results of the described algorithm operation with existing ones. The results of the analysis revealed that dynamic reconfiguration of roles allows increasing the fault tolerance of a wireless computing cluster due to the fact that any of the elements are able to act as the ICD control element.

Keywords: wireless computing cluster, algorithmic model, fault tolerance, graph, initialization algorithm

For citation: Nestruev D.S., Borzov D.B. Model of reorganization of elements of a wireless computing cluster with an orbital arrangement of elements. *Trudy MAI*, 2023, no. 128. DOI: [10.34759/trd-2023-128-19](https://doi.org/10.34759/trd-2023-128-19)

Состояние рассматриваемого вопроса и постановка задачи

Кластер – это модульная многопроцессорная система, созданная на базе стандартных вычислительных узлов, соединенных коммуникационной средой. Типичный кластер представляет из себя множество ЭВМ или процессорных ядер под централизованным управлением, которые пользователь воспринимает как единое целое. Кластеры используются в вычислительных целях, в частности в научных исследованиях. Основной характеристикой вычислительного кластера является отказоустойчивость.

Требования, предъявляемые к кластеру [1]:

– надежность;

- доступность (готовность);
- масштабируемость;
- вычислительная мощность.

На сегодня расположение узлов кластера (координаты в пространстве), технология построения и топология каналов связи как правило стационарны, т.е. «вероятный противник» может, зная «узкие места», парализовать всю работу системы [2]. Для обеспечения большей живучести объекты беспроводного вычислительного кластера должны иметь возможность перемещения в пространстве и быть зарезервированы. В первую очередь это касается управляющего элемента (хоста) и узлов, в которых хранится важная накопленная информация.

Собранную информацию может хранить хост, но при выходе из строя этого элемента прекращает работу не только система БВК, но и вся накопленная информация. Именно поэтому отдается предпочтение отдельному элементу накопления данных (хранилище).

Таким образом, для выполнения требований, предъявляемых к кластеру, необходима разработка динамических ролевых моделей кластерных систем.

Методы решения задачи и принятые допущения

Модель нацелена на выполнение таких действий как: инициализация первого управляющего элемента, сбор исходных данных для алгоритма, выполнение алгоритма реорганизации элементов беспроводного вычислительного кластера, распределение ролей, выполнение задач, поставленных кластеру.

Данная модель реорганизации беспроводных вычислительных кластеров может использоваться в группе беспилотных летательных аппаратов, устройстве автоматического прицеливания, разведывательных дронах и др.

Основная часть

Для обеспечения живучести объектов кластерной системы, каждому объекту необходимо иметь возможность резервирования и перемещения. Системы беспроводных вычислительных кластеров подразумевают использование беспроводной сети из-за особенности их динамического перемещения. Для этого будет использоваться радиоканал Wi-Fi, который не зависит от поставщиков услуг [3]. В связи с тем, что объекты кластера чаще всего неоднородны, уместно ввести ролевое разделение, как показано на рис. 1.

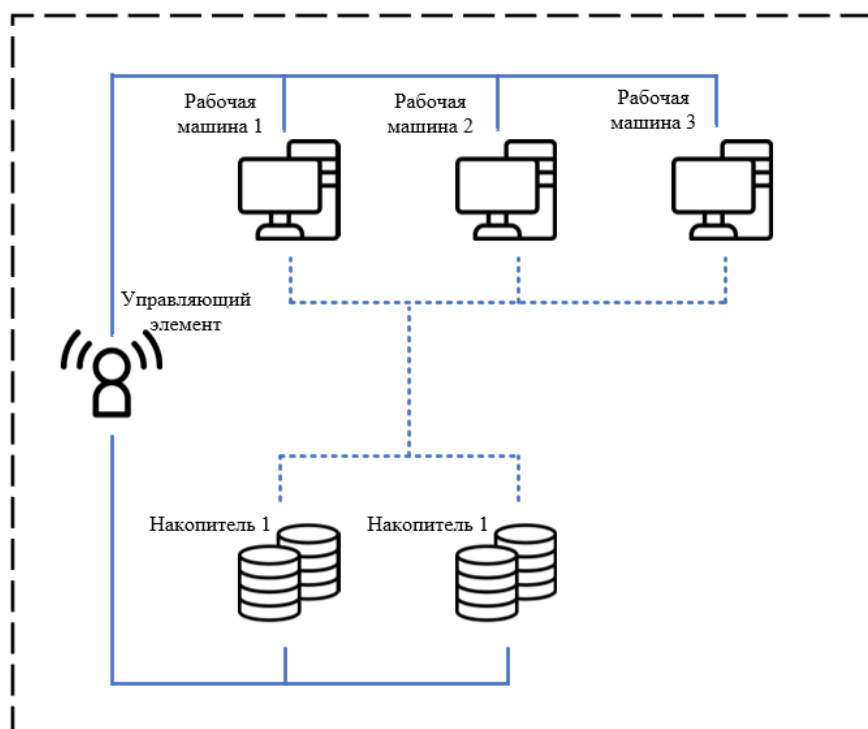


Рис.1. Кластерная система с ролевым разделением.

Отличительной особенностью таких сетей является невозможность резервирования каналов связи между объектами. При наличии угрозы исчезновения канала связи, сеть должна быть тут же реконфигурирована таким образом, чтобы между задействованными в процессе объектами имелись стабильные каналы связи. Реконфигурация подразумевает динамическое изменение топологии, вызванное постоянным перемещением N объектов, критическим падением заряда подмножества Z объектов, критическим падением уровня сигнала множества S каналов связи, изменением характера выполняемых задач [7].

Основные характеристики предлагаемого беспроводного вычислительного кластера:

- динамическое изменение топологии вследствие естественных причин;
- резервирование важнейших элементов;
- ролевое разделение;
- общая память;
- возможность динамической реконфигурации.

Ввиду частой реконфигурации сети общие данные невыгодно держать в распределенной памяти. Это приводит к скачкообразному росту трафика, увеличению сложности алгоритмов реконфигурации и управления. Поэтому в беспроводных вычислительных сетях целесообразно использовать общую память с поэтапным заполнением. Для предотвращения потерь информации память накопителя необходимо зарезервировать на другом элементе беспроводного кластера [4]. Степень резервирования такой сети определяется по числу объектов, которые могут выполнять функции критичных элементов, без которых сеть не сможет

функционировать. При определенной степени резервирования надежность беспроводного кластера будет близка к надежности критических систем [5].

При инициализации предлагаемого БК должны выполняться следующие действия:

1. Элемент M_0 из множества N , изначально инициализированный как «Управляющий элемент» (Хост) запрашивает со спутника координаты и получает их. То же самое делают каждый MN элемент. Образуется начальный граф G с N вершинами (рисунок 2а), каждая из которых не знает о местоположении остальных.
2. «Управляющий элемент» по очереди опрашивает каждый из элементов кластера, получая координаты и вычисляя расстояние до элемента по формуле
$$\Delta\sigma = 2\arcsin\left\{\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right)} + \cos\varphi_1 * \cos\varphi_2 * \sin^2\left(\frac{\Delta\gamma}{2}\right)\right\} \quad (1)$$
где $\varphi_1, \varphi_2, \gamma_1, \gamma_2$ – широта и долгота двух точек в радианах, $\Delta\gamma$ – разница координат по долготе.
3. «Управляющий элемент» высчитывает параметр Sum (сумма расстояний от точки до точки) для каждого элемента и находит идентификатор элемента с минимальным значением. По результатам формируется новый псевдограф PG (рисунок 2б).
4. Если есть необходимость «Управляющий элемент» посылает передатчик команду найденному элементу на принятие управления. Найденный элемент получает роль «Управляющий элемент», а Хост приобретает роль найденного элемента.
5. Новый «Управляющий элемент» выполняет пункты 2 и 3 переписывая граф PG (рисунок 2в).

6. Новый хост распределяет роли элементов системы и выдает команду на запуск этими элементами ролевых функций.

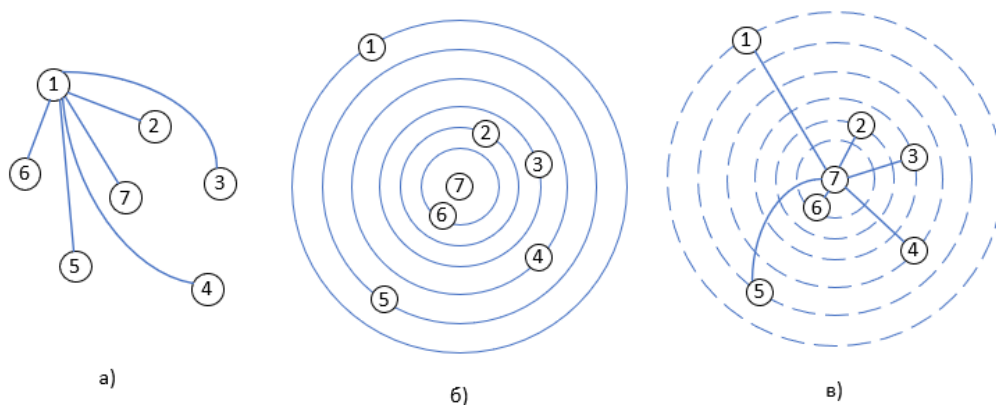


Рис.2. Преобразование графа G в графы PG1 и PG2.

На рис 2. мы видим, что при инициализации беспроводного вычислительного кластера роль «Управляющий элемент» была у элемента с идентификатором 1. После вычислений элемент 1 передал свои функции элементу 7. Таким образом вокруг нового Хоста теперь расположены элементы, лежащие каждый на своей орбите.

Основные элементы вычислительной сети — это сервер и память, они могут быть реализованы в составе одного узла, но для повышения надежности функциями, что соответствует простейшему потоку отказов, реализация будет в отдельных узлах, что приведёт к ролевому разделению на несколько субъектов:

- хост-элемент с серверными функциями;
- резервный хост-элемент;
- накопитель информации;
- резервный накопитель;
- множество рабочих элементов (пчёлы).

Предположим, то все субъекты однородны по возможностям оборудования. Тогда беспроводную систему представим графом $S(M,K)$, где M – множество вершин (элементов беспроводного кластера), K – множество топологических связей, причём $M=\{1..N\}$, а N – число элементов системы. Тогда в беспроводной системе, аналогичной представленной на рисунке 4, будет существовать один хост-элемент H (сервер), один резервный хост-элемент RH , один накопитель N_k , один резервный накопитель RN_k , и множество P рабочих элементов (“пчёл”) [9].

Из сказанного следует, что $M=\{H,RH,N_k, [RN]_k,p1,p2,\dots,pN-4\}$.

Основные функции хост элемента:

- отслеживание работоспособности действующих элементов кластера и подключения новых элементов;
- проверка свойств каналов связи при изменяющемся местоположением элементов;
- реализация алгоритмов управления;
- поиск оптимальных вариантов реконфигурации и их хранение в своей внутренней памяти.

Основные функции накопителя:

- приём информации от рабочих элементов;
- сохранение информации во внутренней памяти.

Основные функции рабочего элемента (пчелы):

- поддержка алгоритмов управления;
- реализация множества задач пользовательского уровня (предположительно сбор и обработка информации);

– передача информации в общую память.

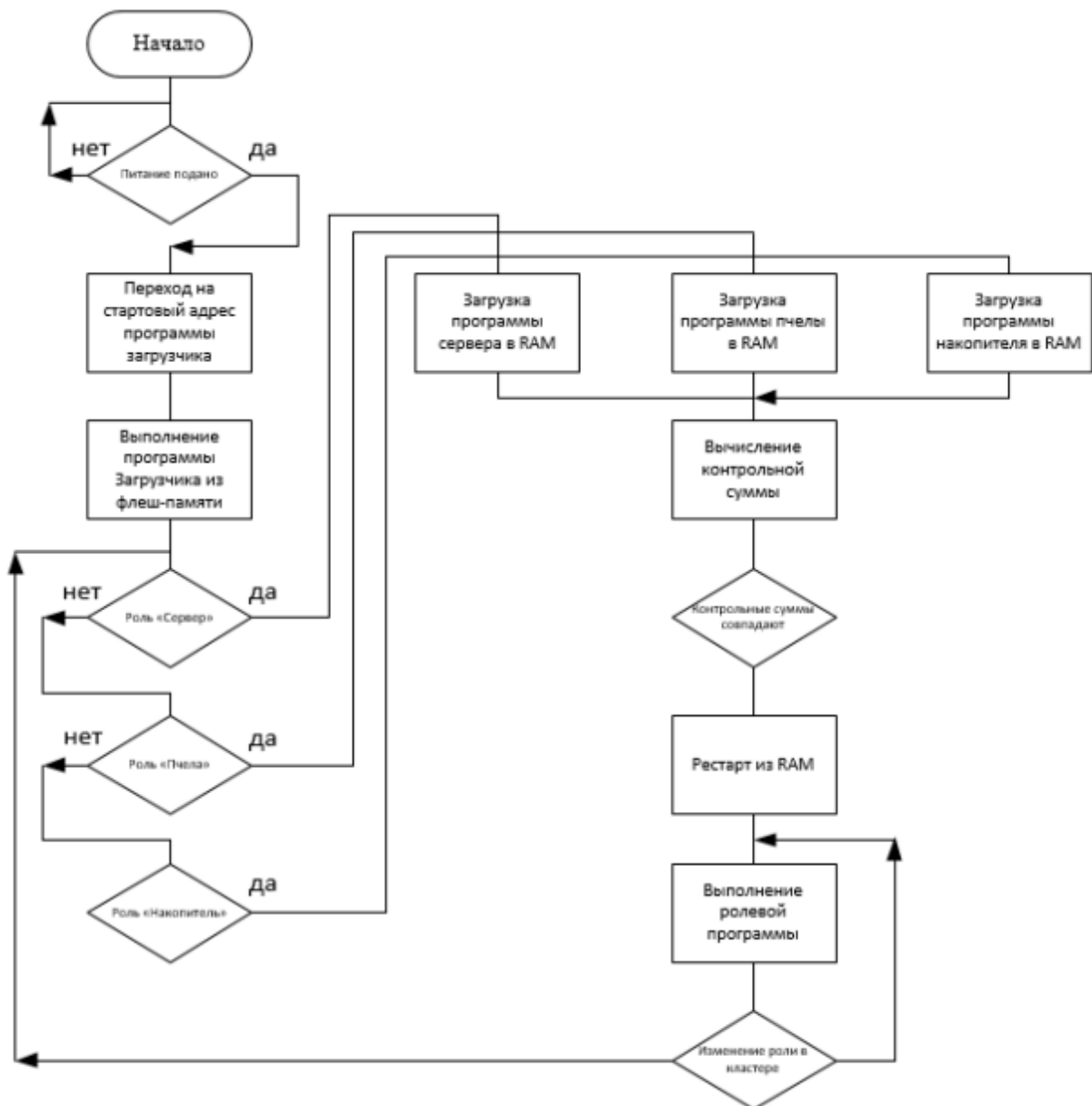


Рис.3. Обобщенный алгоритм инициализации элементов кластера.

На рисунке 3 представлен обобщенный алгоритм инициализации элементов кластера.

По включению питания каждый из N элементов кластера выполняет следующие процедуры:

1. Выполнение общей программы загрузчика из внутренней флеш-памяти со стартового адреса;

2. Определение первичной роли элемента в кластере;
3. Загрузка из флэш-памяти в ОЗУ программы, соответствующей роли элемента М кластера;
4. Считывание контрольной суммы из хвоста программы и сравнение с рассчитанной в процессе загрузки в ОЗУ, что позволяет контролировать целостность информации. Если контрольные суммы не совпадают, то на сервер выдаётся сообщение о неисправности элемента;
5. Рестарт ролевой программы из оперативной памяти на выполнение;
6. Анализ возможного изменения роли элемента в кластере путём периодического считывания переменной TempNum, реализуемый на основе командного протокола, за реализацию управления в котором отвечает сервер.
7. В случае изменения роли в системе субъект перезагружает на исполнение соответствующую ролевую программу и переходит на неё через soft-reset.

В данной работе рассматривается алгоритм работы «Сервера» («Управляющий элемент, Хост»). Алгоритм, представленный на рис. 4 – процедурный алгоритм работы хоста, который содержит следующие основные стадии:

1. Загрузка программой-загрузчиком рабочей программы;
2. Определение собственных координат элемента;
3. Отправка команды на получение координат и технических характеристик другим элементам кластера.
4. Вычисление подходящего на роль «Управляющий элемент» элемента кластера.

5. При необходимости, передача управления выбранному элементу и переход на роль найденного элемента.

Пункты 3 и 4 алгоритма подразумевает извлечения из вложенного в слово статуса кадра NMEA географических координат (Lng, Lat), вычисления расстояния до точки и, в случае накопителя, анализ оставшегося объёма.

Определение уровня сигнала (связи) происходит косвенным путём, а именно вычислением с накоплением времени TimeSv прихода нескольких (число поле управляющего слова KvCount) квитанций. Далее полученное суммарное время ранжируется для определения орбит элементов. Суммарное TimeSv далее участвует в процедуре арбитража.

Второй этап представляет из себя циклическое выполнения ряда функций. Пингование сетевых адресов предназначено для обнаружения в составе сети новых подключенных узлов. Каждый новый узел, ответивший на пинг, добавляется в систему как «Рабочий элемент». Далее в следующем цикле происходит отправка командных слов всем узлам для оценки их технического состояния.

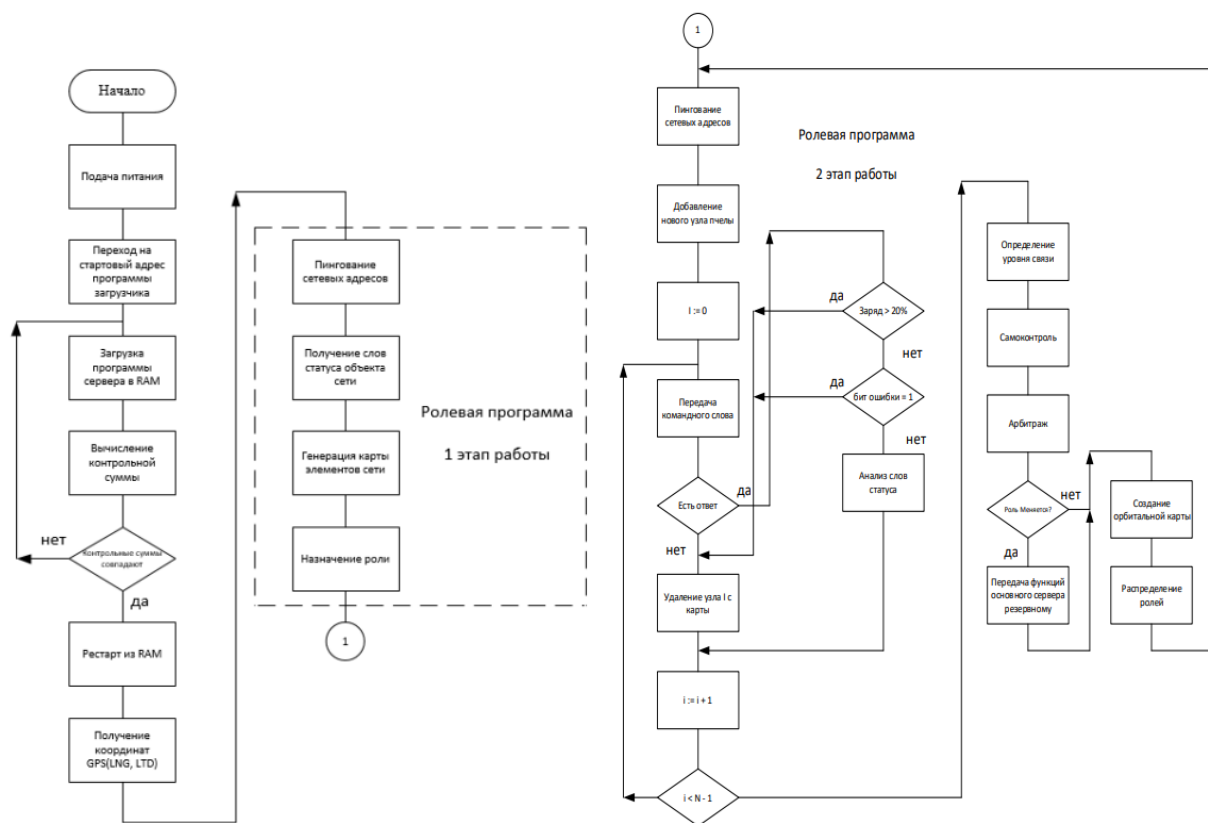


Рис.4. Алгоритм работы управляющего элемента.

Результаты и их обсуждение

Были проведены эксперименты, чтобы понять какие факторы и насколько сильно влияют на время выполнения алгоритма реконфигурации беспроводного вычислительного кластера. Были рассмотрены влияние размеров кластера (количество элементов) и внесение дополнительных параметров (размер внутреннего хранилища, возможности ЦП и т.д.) для ролей. Для того, чтобы убрать погрешность загрузки WG сторонними программами каждый вариант был просчитан 100 раз. Затем было выведено среднее значение среди 100 запусков.

Для оценки влияния размера кластера были выбраны размеры кластера равные 7, 10, 20, 25 элементам. Дополнительные параметры для ролей не устанавливалось. В таблице 1 приведены полученные результаты.

Таблица 1. Результаты оценки влияния размеров кластера

| Количество элементов | 7 | 10 | 20 | 25 |
|-------------------------------|--------|---------|---------|----------|
| Среднее время выполнения (мс) | ~ 7.36 | ~ 10.64 | ~ 50.47 | ~ 247.34 |

На основании полученных результатов сделан вывод, что увеличение размерности кластера приводит к увеличению времени выполнения алгоритма реконфигурации за счет того, что увеличивается количество опрашиваемых элементов и, соответственно, количество расчетов сумм расстояний каждого из элементов. Большее количество расчетов требует большего количества времени.

Для оценки влияния дополнительных параметров ролей было выбрано среднее значение элементов кластера. Во всех случаях 10 элементов. Начальные позиции всех элементов одинаковые. В качестве параметров были выбраны: размер внутреннего хранилища для роли «Накопитель» и скорость работы ЦП для роли «Управляющий элемент».

Таблица 2. Результаты оценки влияния дополнительных параметров

| Количество параметров | Без параметров | 1 | 2 |
|-------------------------------|----------------|---------|---------|
| Среднее время выполнения (мс) | ~ 14.43 | ~ 24.36 | ~ 67.72 |

На основании полученных результатов сделан вывод, что добавление и увеличение дополнительных параметров увеличивают сложность алгоритма и, соответственно, увеличивают время выполнения алгоритма реконфигурации.

Стоит отметить, что увеличение количества элементов и добавление дополнительных параметров увеличивает нагрузку на ЦП элемента с текущей ролью «Управляющий элемент». Это необходимо учитывать при построении системы беспроводного вычислительного кластера с помощью алгоритма реконфигурации ролей.

Также было проведено сравнение обозреваемого алгоритма с алгоритмом выбора головного элемента LEACH. Количество элементов было одинаковое, а именно 7, 10, 20, 25. Дополнительные параметры не добавлялись. Время рассчитано в миллисекундах. Время выполнения является средним из 100 запусков. Результаты сравнения приведены на рис. 5.

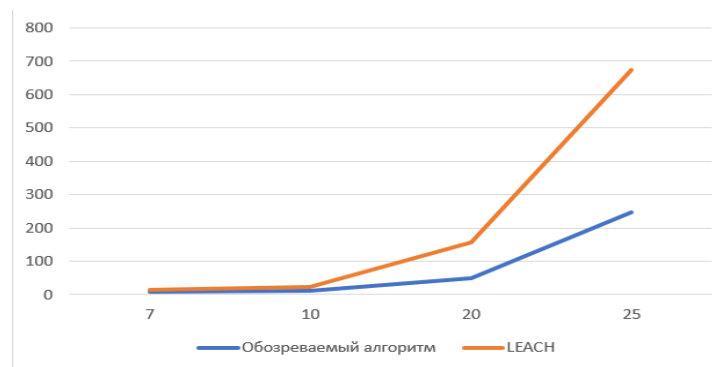


Рис. 5. Результат сравнения обозреваемого алгоритма и алгоритма выбора головного элемента LEACH.

При сравнении алгоритмов заметно приблизительно одинаковое время выполнения при маленьком количестве элементов кластера. С увеличением элементов кластера у алгоритма LEACH видно явное увеличение времени

реорганизации кластера. Стоит отметить, что алгоритм LEACH не подразумевает разделение элементов вычислительного кластера по ролям, а находит только головной элемент, т.е. «Управляющий элемент». Обозреваемый алгоритм за выделенное время распределил не только роль Хоста, но и «Накопитель» и «Пчела».

Выводы

В работе представлена алгоритмическая модель беспроводного вычислительного кластера с динамическим распределением ролей. В рамках данного алгоритма приводится концептуальная модель распределения элементов беспроводного вычислительного кластера по типу орбитальной системы, где каждый элемент БВК находится на своей орбите и выполняет возложенные на него функции. Представленные в работе алгоритмы могут найти применение в разработке программного обеспечения для построения собственной модели беспроводного вычислительного кластера.

Беспроводные вычислительные кластеры, работающие по представленному алгоритму, позволят повысить их отказоустойчивость при разного рода условиях. Предлагаемый алгоритм может быть совмещен с существующими для построения интеллектуальной системы управления беспроводным вычислительным кластером. В будущем исследовании планируется изучение влияния размеров вычислительного кластера и введение дополнительных параметров на отказоустойчивость и время реорганизации беспроводного вычислительного кластера.

Список источников

1. Вишнеvский В.М. Теоретические основы проектирования компьютерных систем. - М.: Техносфера. 2003. – 512 с.
2. Волков А.А., Петрова С.Н., Гинзбург А.В., Иванова Н.А. и др. Информационные системы и технологии в строительстве. – М.: Московский государственный строительный университет, 2015. - 424 с.
3. Дмитриев В.Т. Технологии передачи данных // Компоненты и технологии. 2006. № 12. С. 132 – 135.
4. Коржук В.М. Идентификация атак на беспроводную сенсорную сеть на основе анализа поведения сети // Региональная информатика и информационная безопасность. 2017. № 4. С. 468-469.
5. Финогеев А.А., Финогеев А.Г., Нефедова И.С. Распределенная обработка данных в беспроводных сенсорных сетях на основе мультиагентного подхода и туманных вычислений // Труды международного симпозиума "Надежность и качество". 2016. № 1. С. 258-260.
6. Аминова Р.Р. Разработка алгоритма первоначального размещения базовых станций сетей широкополосного радиодоступа на этапе частотно-территориального планирования // Всероссийская научно-практическая конференция «Новые технологии, материалы и оборудование российской авиакосмической отрасли - АКТО-2016» (Казань, 10–12 августа 2016): сборник докладов. – Казань: Академия наук Республики Татарстан, 2016. Т. 2. С. 338-342.
7. Огородникова О.В. Анализ программного обеспечения, средств обработки информации используемого в УИС // Всероссийская научно-практическая

конференция «Актуальные проблемы деятельности подразделений УИС» (Воронеж, 25 октября 2018): сборник трудов. – Воронеж: Научная книга, 2018. С. 127-130.

8. Борзов Д.Б., Титов В.С. Параллельные вычислительные системы (архитектура, принципы размещения задач. – М.: Изд-во LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. - 152 с.

9. Пальгубев Д.А. Сочетание алгоритмов обработки информации и структуры информационной системы как инструмент построения информационной системы сетевой структуры // Радиопромышленность. 2021. Т. 31. № 2. С. 49-60.

DOI: [10.21778/2413-9599-2021-31-2-49-60](https://doi.org/10.21778/2413-9599-2021-31-2-49-60)

10. Осипова В.А., Дубинина К.С. Применение алгоритмов теории графов к упрощенному методу анализа иерархий // Моделирование и анализ данных. 2019. № 3. С. 24-31.

11. Ананьев А.В, Иванников К.С., Филатов С.В. Основные принципы построения систем связи на базе беспилотных летательных аппаратов // Труды МАИ. 2022. № 125. URL: <https://trudymai.ru/published.php?ID=168188>. DOI: [10.34759/trd-2022-125-16](https://doi.org/10.34759/trd-2022-125-16)

12. 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. URL: <https://trudymai.ru/published.php?ID=161425>. DOI: [10.34759/trd-2021-120-11](https://doi.org/10.34759/trd-2021-120-11)

13. Борзов Д.Б., Кошелев М.А., Соколова Ю.В. Моделирование реконфигурируемой системы реального времени с учетом уменьшения временных затрат обработки информации // Труды МАИ. 2021. № 117. URL: <https://trudymai.ru/published.php?ID=156284>. DOI: [10.34759/trd-2021-117-13](https://doi.org/10.34759/trd-2021-117-13)

14. Кондрашин М.А., Арсенов О.Ю., Козлов И.В. Применение технологии виртуализации и облачных вычислений при построении сложных распределенных моделирующих систем // Труды МАИ. 2016. № 89. URL: <http://trudymai.ru/published.php?ID=73411>
15. Осипов Н.А., Шавин А.С., Тарасов А.Г. Методика идентификации помех, действующих в канале передачи информации робототехнических систем // Труды МАИ. 2017. № 94. URL: <http://trudymai.ru/published.php?ID=81085>
16. Davidovic T., Teodorovic D., Selmic M. Bee colony optimization, Part I: The algorithm overview // Yugoslav Journal Of Operations Research, 2015, vol. 25 (1), pp. 33-56. DOI:[10.2298/YJOR131011017D](https://doi.org/10.2298/YJOR131011017D)
17. Dawood N., Dawood H., Rodriguez-Trejo S. et al. Visualising urban energy use: the use of LiDAR and remote sensing data in urban energy planning // Visualization in Engineering, 2017, vol. 5 (1). DOI: 10.1186/s40327-017-0060-3
18. Fabian Bock, Monika Sester. Improving Parking Availability Maps using Information from Nearby Roads // Transportation Research Procedia, 2016, vol. 19, pp. 207-214. URL: <https://doi.org/10.1016/j.trpro.2016.12.081>
19. Parkhurst J., Darringer J., Grundmann B. From single core to multi-core: preparing for a new exponential // Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design, 2019, no. 1, pp. 67–72. URL: <https://doi.org/10.1016/j.ifacol.2016.08.029>

20. Perković T., Šolić P., Zargariasl H., Čoko D., Rodrigues Joel J.P.C. Smart Parking Sensors: State of the Art and Performance Evaluation // Journal of Cleaner Production, 2020, vol. 1262, pp. 121181. [URL: https://doi.org/10.1016/j.jclepro.2020.121181](https://doi.org/10.1016/j.jclepro.2020.121181)

References

1. Vishnevskii V.M. *Teoreticheskie osnovy proektirovaniya komp'yuternykh sistem.* (Theoretical foundations of designing computer systems), Moscow, Tekhnosfera, 2003, 512 p.
2. Volkov A.A., Petrova S.N., Ginzburg A.V., Ivanova N.A. et al. *Informatsionnye sistemy i tekhnologii v stroitel'stve* (Information systems and technologies in construction), Moscow, Moskovskii gosudarstvennyi stroitel'nyi universitet|, 2015, 424 p.
3. Dmitriev V.T. *Komponenty i tekhnologii*, 2006, no. 12, pp. 132 – 135.
4. Korzhuk V.M. *Regional'naya informatika i informatsionnaya bezopasnost'*, 2017, no. 4, pp. 468-469.
5. Finogeev A.A., Finogeev A.G., Nefedova I.S. *Trudy mezhdunarodnogo simpoziuma "Nadezhnost' i kachestvo"*, 2016, no. 1, pp. 258-260.
6. Aminova R.R. *Vserossiiskaya nauchno-prakticheskaya konferentsiya «Novye tekhnologii, materialy i oborudovanie rossiiskoi aviakosmicheskoi otrasli - AKTO-2016»:* sbornik dokladov. Kazan', Akademiya nauk Respubliki Tatarstan, 2016, vol. 2, pp. 338-342.
7. Ogorodnikova O.V. *Vserossiiskaya nauchno-prakticheskaya konferentsiya «Aktual'nye problemy deyatel'nosti podrazdelenii UIS»:* sbornik trudov. Voronezh, Nauchnaya kniga, 2018, pp. 127-130.

8. Borzov D.B., Titov V.S. *Parallel'nye vychislitel'nye sistemy (arkhitektura, printsipy razmeshcheniya zadach* (Parallel computing systems (architecture, principles of task placement), Izd-vo LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012, 152 p.
9. Pal'guev D.A. *Radiopromyshlennost'*, 2021, vol. 31, no. 2, pp. 49-60. DOI: 10.21778/2413-9599-2021-31-2-49-60
10. Osipova V.A., Dubinina K.S. *Modelirovanie i analiz dannykh*, 2019, no. 3, pp. 24-31.
11. Anan'ev A.V, Ivannikov K.S., Filatov S.V. *Trudy MAI*, 2022, no. 125. URL: <https://trudymai.ru/eng/published.php?ID=168188>. DOI: 10.34759/trd-2022-125-16
12. Pavlov A.N., Pavlov D.A., Umarov A.B. *Trudy MAI*, 2021, no. 120. URL: <https://trudymai.ru/eng/published.php?ID=161425>. DOI: 10.34759/trd-2021-120-11
13. Borzov D.B., Koshelev M.A., Sokolova Yu.V. *Trudy MAI*, 2021, no. 117. URL: <https://trudymai.ru/eng/published.php?ID=156284>. DOI: 10.34759/trd-2021-117-13
14. Kondrashin M.A., Arsenov O.Yu., Kozlov I.V. *Trudy MAI*, 2016, no. 89. URL: <http://trudymai.ru/eng/published.php?ID=73411>
15. Osipov N.A., Shavin A.S., Tarasov A.G. *Trudy MAI*, 2017, no. 94. URL: <http://trudymai.ru/eng/published.php?ID=81085>
16. Davidovic T., Teodorovic D., Selmic M. Bee colony optimization, Part I: The algorithm overview, *Yugoslav Journal Of Operations Research*, 2015, vol. 25 (1), pp. 33-56. DOI:10.2298/YJOR131011017D
17. Dawood N., Dawood H., Rodriguez-Trejo S. et al. Visualising urban energy use: the use of LiDAR and remote sensing data in urban energy planning, *Visualization in Engineering*, 2017, vol. 5 (1). DOI: 10.1186/s40327-017-0060-3

18. Fabian Bock, Monika Sester. Improving Parking Availability Maps using Information from Nearby Roads, *Transportation Research Procedia*, 2016, vol. 19, pp. 207-214. URL: <https://doi.org/10.1016/j.trpro.2016.12.081>
19. Parkhurst J., Darringer J., Grundmann B. From single core to multi-core: preparing for a new exponential, *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, 2019, no. 1, pp. 67–72. URL: <https://doi.org/10.1016/j.ifacol.2016.08.029>
20. Perković T., Šolić P., Zargariasl H., Čoko D. et al. Smart Parking Sensors: State of the Art and Performance Evaluation, *Journal of Cleaner Production*, 2020, vol. 1262, pp. 121181. URL: <https://doi.org/10.1016/j.jclepro.2020.121181>

Статья поступила в редакцию 11.01.2023

Одобрена после рецензирования 18.01.2023

Принята к публикации 27.02.2023

The article was submitted on 11.01.2023; approved after reviewing on 18.01.2023; accepted for publication on 27.02.2023