

УДК 004.9

DOI: [10.26102/2310-6018/2023.43.4.002](https://doi.org/10.26102/2310-6018/2023.43.4.002)

Обобщенная экологическая модель динамической распределенной вычислительной системы

Е.Р. Брюханова^{1,2}, О.А. Антамошкин^{1,2}

¹*Сибирский государственный университет науки и технологий им. М.Ф. Решетнева,
Красноярск, Российская Федерация*

²*Сибирский федеральный университет, Красноярск, Российская Федерация*

Резюме. В работе представлена обобщенная модель, позволяющая проводить структурный анализ распределенной вычислительной динамической системы, а также исследовать применимость различных методов управления с учетом параметров экологичности ее работы. С приходом эпохи информационного общества интенсивность использования распределенных вычислительных систем для обработки данных и выполнения разнообразных задач безостановочно растет. Однако с ростом их количества и масштабов остро встают вопросы энергопотребления и негативного воздействия на окружающую среду. Предложенная модель предоставляет инструментарий для оценки воздействия таких систем на окружающую среду, а также для принятия мер по минимизации их экологического следа. Она включает в себя комплекс параметров, позволяющих анализировать и учитывать факторы, такие как энергопотребление, выбросы углерода и эффективность использования ресурсов. Данная модель призвана способствовать развитию более экологически позитивных подходов к управлению распределенными вычислительными системами. Это имеет особую важность в свете нарастающего внимания к экологической проблематике и стремления общества к более ответственному использованию ресурсов. Результаты данного исследования открывают путь к созданию более эффективных и экологически дружелюбных вычислительных решений, способствуя снижению негативного воздействия на окружающую среду и более устойчивому будущему, обеспечивая баланс между производительностью и экологичностью распределенных систем вычислений.

Ключевые слова: распределенные вычислительные системы, динамические системы, экологическая устойчивость, энергопотребление, оптимизация.

Для цитирования: Брюханова Е.Р., Антамошкин О.А., Обобщенная экологическая модель динамической распределенной вычислительной системы. *Моделирование, оптимизация и информационные технологии.* 2023;11(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=1439> DOI: 10.26102/2310-6018/2023.43.4.002

Generalized ecological model of a dynamic distributed computing system

E.R. Bryukhanova^{1,2}, O.A. Antamoshkin^{1,2}

¹*Reshetnev Siberian State University of Science and Technology, Krasnoyarsk,
the Russian Federation*

²*Siberian Federal University, Krasnoyarsk, the Russian Federation*

Abstract. The paper presents a generalized model that enables a structural analysis of a distributed computational dynamic system and makes it possible to investigate the applicability of various control methods taking into account the environmental parameters of its operation. With the advent of the information society era, distributed computing systems for data processing and performing various tasks are being increasingly used. However, with the growth of their number and scale, the issues of energy consumption and negative impact on the environment are becoming more acute. The proposed model

provides tools for assessing the impact of such systems on the environment as well as for taking measures to minimize their ecological footprint. It includes a set of parameters that help to analyze and take into account such factors as energy consumption, carbon emissions and resource efficiency. This model is designed to promote the development of more environmentally positive approaches to the management of distributed computing systems. This is of particular importance in the light of the growing attention to environmental issues and the desire of society for a more responsible use of resources. The results of this study open the way to creating more efficient and environmentally friendly computing solutions reducing the negative impact on the environment and a more sustainable future ensuring a balance between performance and environmental friendliness of distributed computing systems.

Keywords: distributed computing systems, dynamic systems, environmental sustainability, energy consumption, optimization.

For citation: Bryukhanova E.R, Antamoshkin O.A. Generalized ecological model of a dynamic distributed computing system. *Modeling, Optimization and Information Technology*. 2023;11(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=1439> DOI: 10.26102/2310-6018/2023.43.4.002 (In Russ.).

Введение

В современных условиях, когда вопросы экологии и устойчивого развития становятся все более актуальными, необходимость интеграции экологических аспектов в управление динамическими распределенными системами становится неотъемлемой частью обеспечения эффективности и устойчивости процессов. В контексте этой проблемы, разработка обобщенной модели, способной учитывать экологические параметры работы таких систем, приобретает особую важность.

Исследование и учет экологических аспектов в динамических распределенных системах сталкивается с рядом вызовов. Эти системы включают в себя сети, процессы и ресурсы, динамично взаимодействующие между собой для достижения определенных целей. При этом, помимо обычных критериев эффективности, необходимо учитывать воздействие на окружающую среду, энергопотребление, выбросы и другие экологически значимые параметры.

Цель данной работы заключается в разработке обобщенной модели, способной внедрить экологические параметры в управление динамическими распределенными системами. В ходе исследования планируется разработать алгоритмы и методики, позволяющие эффективно управлять ресурсами и процессами, учитывая как традиционные, так и экологические критерии.

Управление распределенными динамическими системами требует больших вычислительных ресурсов. Таким образом, эффективное управление ресурсами имеет важное значение из-за неоднородности и динамизма окружающей среды. Алгоритмы планирования – это набор политик, процедур и правил, реализованных для назначения наилучшего ресурса для выполнения задач с целью достижения целей поставщика услуг и пользователя вычислительных мощностей. Каждый из существующих методов планирования [1-4] учитывает несколько показателей производительности. Наиболее распространенные показатели упомянуты ниже:

- 1) пропускная способность [5, 6];
- 2) время выполнения [7, 8];
- 3) время отклика [9, 10];
- 4) стоимость выполнения [11];
- 5) крайние сроки и бюджетные ограничения [12-15];
- 6) балансировка нагрузки [16, 17];
- 7) отказоустойчивость [18, 19];
- 8) нарушение SLA [20];

9) энергопотребление [21-22];

10) передача данных [23].

Очевидна необходимость разработки модели, которая описывает поведение ресурсов в системе и позволяет оценить их эффективное использование. Разработанная модель в будущем позволит использовать для оптимизации управления ресурсами обнуляющие нейронные сети.

Исследование имеет большое практическое значение для применения в различных областях, включая энергетику, логистику, управление ресурсами и другие. Разработка обобщенной модели, учитывающей экологические параметры, может способствовать более устойчивому и эффективному функционированию динамических распределенных систем в условиях современных экологических вызовов.

Материалы и методы

Одной из основных задач данного исследования является эффективное распределение технологических процессов и соответствующих ресурсов. Эта задача имеет существенное значение для обеспечения оптимальной производительности и использования ресурсов. Планирование в данном контексте – сложная задача, особенно при большом числе ресурсов и процессов. Для решения этой задачи используются специализированные инструменты и алгоритмы. Планирование включает в себя временные границы выполнения процессов и оптимальное распределение ресурсов для эффективной работы системы. Задачи оптимального распределения ресурсов являются NP-трудными, что означает отсутствие алгоритмов полиномиальной сложности для их решения. В практике применяются разнообразные алгоритмические подходы, включая эвристические, жадные, графовые, генетические алгоритмы и нейросетевые методы.

Предлагаемый подход схож с аналогичными исследованиями, описанными в работах [24-29], однако отличие заключается в учете критериев экологичности работы системы. Это подтверждает практическую значимость и потенциальную ценность разработанного подхода.

Постановка задачи построения модели, учитывающей экологические параметры работы динамических распределенных систем

В современных сферах жизни, оперирующих динамическими распределенными системами, возникает потребность в разработке обобщенной модели, способной учесть экологические параметры в процессе функционирования. Такие системы могут включать в себя сети облачных вычислений, распределенные вычислительные платформы, интернет вещей и другие, где эффективность, производительность и воздействие на окружающую среду становятся ключевыми аспектами.

Современные предприятия располагают следующими ресурсами:

Вычислительные ресурсы: в состав динамических распределенных систем входят серверы, компьютеры, кластеры и облачные инфраструктуры. Эти ресурсы используются для выполнения вычислительных задач, обработки данных и предоставления услуг. Оптимальное распределение и использование этих ресурсов с учетом экологических параметров может значительно влиять на эффективность и стойкость системы.

Энергетические ресурсы: для обеспечения работы вычислительных и технических устройств требуется энергия. Потребление электроэнергии оказывает влияние на эффективность и экологическую стойкость системы. Рассмотрение и оптимизация энергопотребления становятся ключевыми аспектами для снижения негативного воздействия на окружающую среду.

Технические устройства: оборудование, такое как серверы, маршрутизаторы, коммутаторы и датчики, играют важную роль в функционировании динамических распределенных систем. Оптимальное управление и использование технических устройств может существенно повысить производительность системы и уменьшить потребление ресурсов.

Временные ресурсы: эффективное управление временными ресурсами, такими как время выполнения задач, временные интервалы, дедлайны и периоды активности, является важным аспектом для обеспечения непрерывности работы системы и снижения негативного воздействия на окружающую среду.

Информационные ресурсы: информация и данные являются ключевыми активами в динамических распределенных системах. Эффективное управление информационными потоками, их передачей и обработкой влияет на производительность и эффективность системы.

Требуемые ресурсы, необходимые для выполнения процесса, проявляются на этапе его проектирования. На данном этапе осуществляется определение последовательности операций и анализ ресурсов, требуемых для успешного их осуществления, включая человеческие, временные и технические аспекты. Кроме того, для каждой задачи, поступающей в систему, определяется степень ее приоритетности, учитывая установленные сроки выполнения. Эти факторы составляют фундаментальную основу для оценки эффективности функционирования системы с учетом экологических параметров.

В общем виде, модель распределенной динамической вычислительной системы, учитывающей экологические факторы, можно представить в виде функциональной модели системы, описанной в нотации IDEF0, которая представлена на Рисунке 1.

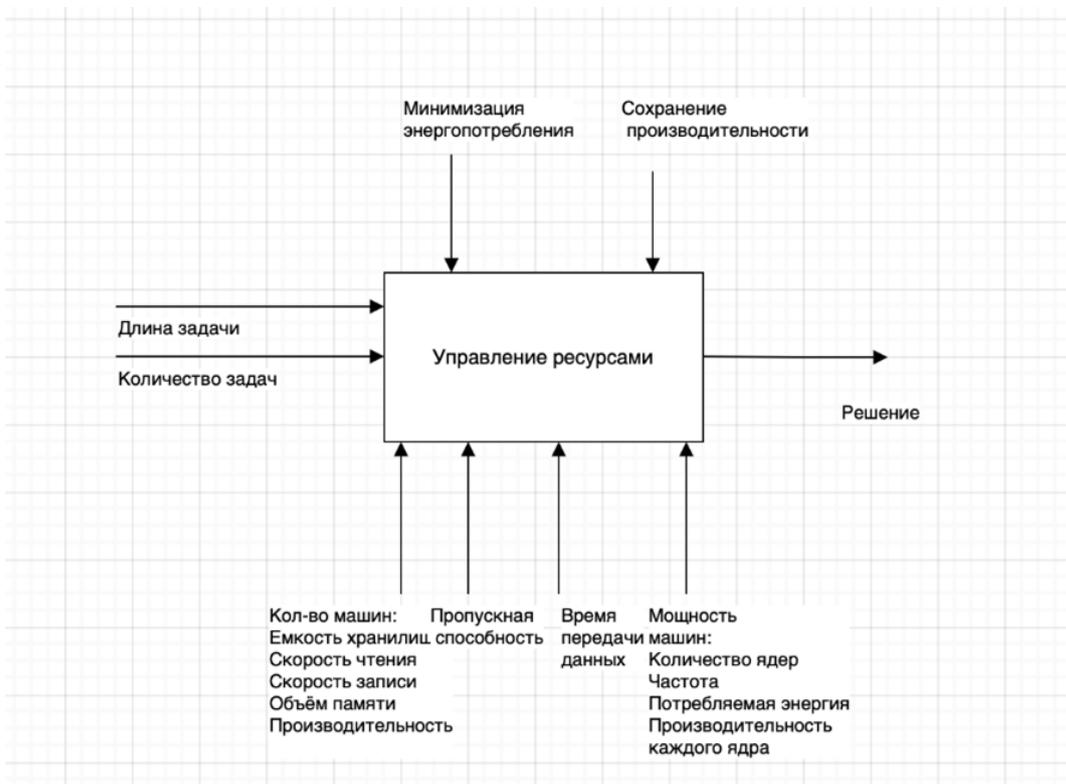


Рисунок 1 – Функциональная модель системы
Figure 1 – Functional model of the system

Общее математическое описание распределенной вычислительной системы

Система представлена гетерогенными вычислительными узлами и каналами передачи их связывающими, состоящими из двух типов обработки: обработки задач и передачи данных.

Характеристики узлов и их обозначения перечислены следующим образом:

$U = \{u_i\}$ множество узлов, где u_i обозначает i узел;

sc_i – это целочисленное значение, представляющее емкость хранилища машины u_i (МБ);

r_i – это значение с плавающей запятой, представляющее скорость чтения (получения данных) машины u_i (МБ/с);

w_i – это значение с плавающей запятой, представляющее скорость записи (передачи данных) машины u_i (МБ/с);

RAM [u_i] – это целочисленное значение, представляющее доступный объем памяти машины u_i (Байт);

N_CPU [u_i] – количество ядер машины u_i ;

P_CPU [u_i] – это целочисленное значение, представляющее производительность процессора каждого ядра машины u_i (миллион инструкций в секунду – MIPS);

b_{ij} – значение с плавающей запятой, представляющее пропускную способность соединения между машинами u_i и u_j (МБ/с);

β_{ij} – элементарное время передачи данных между машинами u_i и u_j , определяемое формулой:

$$\beta_{ij} = \begin{cases} 0 & \text{if } i = j \\ \frac{1}{b_{ij}} & \text{otherwise} \end{cases} \quad (1)$$

PP [u_i] – это целочисленное значение, представляющее вычислительную мощность машины u_i (в миллионах инструкций в секунду – MIPS), которое вычисляется следующим образом:

$$PP[u_i] = N_CPU[u_i] \times P_CPU[u_i], \quad (2)$$

где N_CPU [u_i] – количество ядер u_i , а P_CPU [u_i] – производительность процессора каждого ядра в u_i .

TP_i определяет список выполняемых задач в u_i .

Независимые пользователи отправляют задания на выполнение. Мы рассматриваем, что задачи поступают в режиме реального времени. Все онлайн-задачи совместно используют ресурсы и данные на серверах. Поскольку выполняемые задачи требуют больших объемов данных, с каждой задачей связаны два важных фактора: необходимые данные и ресурсы. Задачи выполняются неупреждающим способом. Однако каждая задача определяется следующим образом:

$T = \{t_n\}$ набор задач, где t_n – n -я задача;

l_n обозначает длину n -й задачи (в миллионе инструкций – MI);

RAM [t_n] – целое число, представляющее объем памяти, необходимый для выполнения задачи t_n (Байт);

CPU [t_n] – целое число, представляющее количество MIPS, необходимое для выполнения задачи t_n ;

V [t_n] – это целочисленное значение, представляющее общий размер всех необходимых наборов данных по задаче t_n ;

α_n – индекс конечного назначения машины (mci) задачи;

ω_n – десятичное значение, представляющее время прибытия t_n ;
 UR_{in} – коэффициент использования CPU, определяющий, имеет ли машина u_i достаточное количество ресурсов для поддержки задачи t_n или нет.

Как упоминалось ранее, балансировка нагрузки является критически важным аспектом, который следует учитывать при разработке алгоритма планирования любых задач таким образом, чтобы оптимизировать использование ресурсов, максимизировать пропускную способность и минимизировать время отклика. Для этого определяем рабочую нагрузку каждого сервера следующим образом:

$$load[u_i] = \frac{\sum_{t_n \in TP_i} l[t_n]}{PP[u_i]}, \quad (3)$$

где $load[u_i]$ – процентная ставка, указывающая на то, что u_i перегружен или недогружен. $load[u_i]$ вычисляется путем деления суммы всех выполняемых задач в u_i на вычислительную мощность $PP[u_i]$.

Мы предполагаем, что на серверах изначально хранится фиксированное количество наборов данных. Каждый набор данных определяется следующим образом:

$D = \{d_k\}$ множество наборов данных, где d_k – k-й набор данных;

v_k целочисленное значение, обозначающее объем k-го набора данных (в Байтах);

$\psi = \{\psi_{ki}\}$ – это наборы данных для матрицы назначения машин. Уравнение (4) описывает вычисление матрицы ψ .

$$\psi = \begin{cases} 1 & \text{если } d_k \text{ входит в } u_i \\ 0 & \text{другое} \end{cases} \quad (4)$$

$F = \{f_{ki}\}$ – это присвоение наборов данных матрице задач. Мы устанавливаем матрицу F , потому что для выполнения задачи может потребоваться один или несколько наборов данных, и многие задачи могут использовать один и тот же набор данных. Матрица F генерируется в соответствии с уравнением (5).

$$f_{kn} = \begin{cases} 1 & \text{если } d_k \text{ требует } t_n \\ 0 & \text{в других случаях} \end{cases} \quad (5)$$

Для набора данных d_k может быть два варианта использования (1). Локальное использование – когда набор данных и его задача потребителя находятся на одном узле, в этом случае доступ к набору данных осуществляется локально. (2) Удаленное использование – когда требуемый набор данных хранится в узле, отличном от того, на котором размещена задача; в этом случае требуется перенос данных из удаленного источника. Очевидно что из-за процесса миграции на время выполнения потребительской задачи влияет добавление времени переноса данных DMT , где DMT_{ni} – время миграции всех наборов данных, требуемых t_n , из их местоположений в u_i (u_i также является местом, где назначается t_n) [30].

Описание целевой функции

Для возможности применения подхода и алгоритма оптимизации опишем модели целевой функции, используемой в задаче планирования рабочего процесса в распределенных вычислительных системах. Мы предположили, что распределенная вычислительная система включают в себя множество вычислительных узлов (CU), представленных $CU = \{cu_1, cu_2, cu_3, \dots\}$. Процессоры вычислительных узлов, в момент времени t_n , могут работать на различной частоте, что, в свою очередь, потребует различное потребление энергии. Множество $DCSCP(t_n) = \{CU_i(P_j(F_j, V_j, E_j))\}$, где CU_i – i – ый вычислительный узел, P_j – j – ый процессор этого узла, работающий на

чистоте F_j , вольтаже V_j и потребляющий для работы E_j энергии. Для оценки экологичности процесса вычисления введем переменную E_{eco} , показывающую количество выброса углерода для получения этой энергии.

Для расчета экологических последствий (environmental impacts) работы распределенной вычислительной системы в любой момент времени будем использовать формулу.

$$EI = \sum_{i,j} CU_i(P_j(F, V, E(t))) \times E_{eco}(t)_i. \quad (6)$$

Важно отметить, что величина E_{eco} на каждом из вычислительных узлов i будет иметь различное значение, т. к. система является распределенной в пространстве, и экологичность потребляемой таким узлом энергии будет отличаться в зависимости от механизма ее получения.

Для вычисления среднего времени выполнения (AETIME) задачи T_k на j -ом процессоре P_j i -ого вычислительного узла CN_i будем использовать формулу

$$AETIME(T_k, CU_i(P_j)) = \frac{1}{J} \sum_{c=1}^N \frac{Task_len(T_k)}{CU_i(P_j)_c}, \quad (7)$$

где $Task_len(T_k)$ является сложностью k -ой задачи, измеряемой в миллионах инструкций процессора в секунду, необходимых для вычисления этой задачи. $CU_i(P_j)_c$ подразумевает характеристики работы j -ого вычислительного узла CU, включающего j процессоров P , из J множества процессоров.

Самое раннее время начала (EST) каждой задачи в этой модели можно рассчитать следующим образом:

$$EST(T_k, CU_i(P_j)) = \begin{cases} 0, & \text{если } T_k \text{ является входной задачей для узла} \\ \max(avail(CU_i(P_j), \max(FT(T_k) + CT(T_i, T_k))), & \\ & \text{в противном случае, для каждого} \\ & T_i \in \text{предшествующего}(T_k) \end{cases}, \quad (8)$$

где $avail(CU_i(P_j))$ является моментом, когда i -я виртуальная машина готова к выполнению указанной задачи. Время выполнения каждой задачи, FT, можно определить с помощью следующего уравнения:

$$FT(T_k, CU_i(P_j)) = \begin{cases} deadline(w_k), & \text{если } T_k \text{ является выходной задачей} \\ EStime(CU_i(P_j)) + Ave Exection Time(CU_i(P_j)), & \\ & \text{в противном случае} \end{cases}, \quad (9)$$

Ave Exection Time – среднее время выполнения.

Здесь $deadline(w_k)$ – крайний срок выполнения k -го рабочего процесса, CT – время передачи данных между T_k и T_i , которое может быть рассчитано с использованием следующего уравнения:

$$CT(T_k, T_i) = \begin{cases} 0, & \text{если } CU(P(T_k)) = CU(P(T_i)) \\ \frac{Date(T_k, T_i)}{Bandwidth(CU(P(T_k)), CU(P(T_i)))}, & \\ & \text{в противном случае} \end{cases}. \quad (10)$$

$Date(T_k, T_i)$ указывают объем данных, которыми необходимо обмениваться между задачами, а $Bandwidth(CU(P(T_k)), CU(P(T_i)))$ обозначает пропускную способность между машинами, выполняющими задачи T_k и T_i . Кроме того, продолжительность рабочего процесса может быть рассчитана на основе:

$$makespan(w_i) = \{\max(FT(T_k)) | T_k \in w_i\}. \quad (11)$$

Рассмотрено моделирование процессов оптимизации управления ресурсами в распределенных динамических системах: создание математической модели, которая описывает поведение ресурсов в системе и позволяет оценить их эффективное использование. Модель будет использоваться для оптимизации управления ресурсами в системе с помощью обнуляющих нейронных сетей.

Результаты

Основываясь на наблюдениях работы обобщенной модели, мы обнаружили снижение энергопотребления без потери производительности с помощью управления энергопотреблением процессоров путем понижения частоты работы процессора. В моменты низкой нагрузки можно существенно снизить энергопотребление устройства с помощью метода Dynamic Voltage and Frequency Scaling (DVFS), что продемонстрировано в Таблице 1. Существуют возможность динамически увеличивать частоту работы процессора, когда требуется высокая производительность, и уменьшать ее в моменты меньшей активности. Это помогает балансировать производительность и энергопотребление, что является важным аспектом для достижения поставленной цели. Мы предлагаем подход, который использует преимущества методов DVFS для потребления меньшего количества энергии и удовлетворения требований сохранения производительности и минимизации энергопотребления. Результаты применения которого представлены в Таблице 1.

Таблица 1 – Энергопотребление задач

Table 1 – Task power consumption

Частота (ГГц)	Мощность (Вт)	Время (мс)	Энергия (мДж)
1.5	1.43	48	68.64
1.0	0.91	65	59.15

Таким образом, предлагаемая нами стратегия планирования задач, основанная на DVFS, миграции и репликации данных, направлена на выбор задач из очереди и их перенос на соответствующий сервер, чтобы обеспечить лучшее время отклика, систему балансировки нагрузки и минимизацию углеродного следа. Время отклика задачи включает в себя два основных фактора: производительность ресурсов и, что более важно, управление данными об их местоположении, состоянии и репликации в системе.

Разработанная математическая модель распределенной динамической системы может быть использована в том числе для включения в алгоритм оптимизации обнуляющих нейронных сетей, представлена на Рисунке 2.

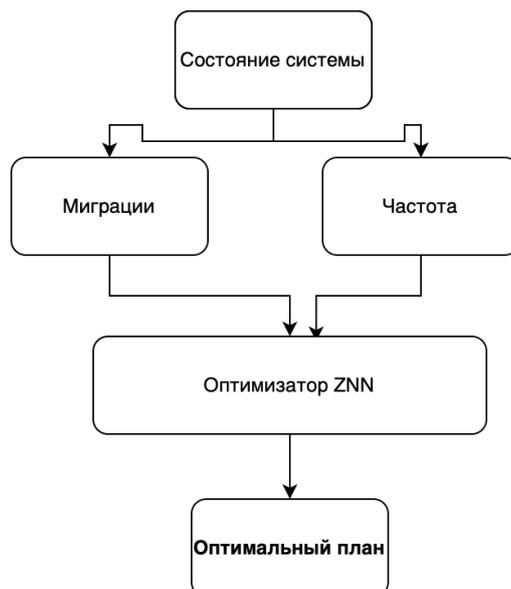


Рисунок 2 – Алгоритм предлагаемого подхода среды выполнения
 Figure 2 – Algorithm of the proposed runtime approach

Рабочий процесс алгоритма показан на Рисунке 2. На этапе модели задач мы сначала собираем время работы с различными рабочими частотами для разных длин входа, затем мы проводим обучение данных для построения регрессионной модели и прогнозирования времени работы в различных условиях эксплуатации.

Обсуждение

Таким образом, данное исследование является важным шагом в направлении разработки инновационных решений для управления динамическими системами с учетом экологической устойчивости, что может привести к более эффективному использованию ресурсов и снижению негативного воздействия на окружающую среду.

Применение алгоритма обнуляющей нейронной сети (ZNN) для адаптивного управления ресурсами представляет собой инновационный подход, который может привести к более эффективному использованию ресурсов, уменьшению времени простоя и снижению негативного воздействия на окружающую среду. Этот подход может найти применение в различных областях, включая управление производством, информационными системами и облачными вычислениями.

Заключение

Основным достижением данной работы стало создание обобщенной модели, способной интегрировать экологические параметры в процесс управления динамическими распределенными вычислительными системами. Модель обеспечивает возможность учета разнообразных экологических аспектов, таких как энергопотребление, углеродный след и эффективное использование ресурсов.

В будущем предполагается разработка прототипа программного продукта с целью его полной интеграции в процессы планирования задач в распределенных динамических вычислительных системах. Планируется расширение функциональности с использованием методов интеллектуального планирования и прогнозирования, что позволит создать более точную и эффективную модель управления экологически устойчивыми динамическими распределенными системами.

СПИСОК ИСТОЧНИКОВ

1. Dubey K., Kumar M. Sharma S. Modified HEFT algorithm for task scheduling in cloud environment. *Procedia Computer Science*. 2018;125:725–732. DOI: 10.1016/j.procs.2017.12.093.
2. Mondal R., Nandi E., Sarddar D. Load balancing scheduling with shortest load first. *International Journal of Grid and Distributed Computing*. 2015;8:171–178. DOI: 10.14257/ijgdc.2015.8.4.17.
3. Lakra A.V., Yadav D.K. Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. *Procedia Computer Science*. 2015;48:107–113. DOI: 10.1016/j.procs.2015.04.158.
4. Wang H., Wang F., Liu J., Wang D., Groen J. Enabling customer-provided resources for cloud computing: Potentials, challenges, and implementation. *IEEE Transactions on Parallel and Distributed Systems*. 2015;26:1874–1886.
5. Gill S.S., Chana I., Singh M., Buyya R. CHOPPER: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing*. 2018;21:1203–1241. DOI: 10.1007/s10586-017-1040-z.
6. Thomas A., Krishnalal G., Raj P.V. Credit based scheduling algorithm in cloud computing environment. *Procedia Computer Science*. 2015;46:913–920. DOI: 10.1016/j.procs.2015.02.162.
7. Sajid M., Raza, Z. Turnaround time minimization-based static scheduling model using task duplication for fine-grained parallel applications onto hybrid cloud environment. *IETE Journal of Research*. 2015;62(3):1–13. DOI: 10.1080/03772063.2015.1075911.
8. Hadji M., Zeghlache D. Minimum cost maximum flow algorithm for dynamic resource allocation in clouds. *2012 IEEE 5th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA*. 2012. p. 876–882. DOI: 10.1109/CLOUD.2012.36.
9. Elzeki O., Reshad M., Abu Elsoud, M. Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*. 2012;50(12):22–27. DOI: 10.5120/7823-1009.
10. Fernández Cerero D., Fernández-Montes A., Jakóbič A., Kołodziej J., Toro M. SCORE: Simulator for cloud optimization of resources and energy consumption. *Simulation Modelling Practice and Theory*. 2018;82:160–173. DOI: 10.1016/j.simpat.2018.01.004.
11. Ma T., Chu Y., Zhao L., Otgonbayar A. Resource allocation and scheduling in cloud computing: policy and algorithm. *IETE Technical Review*. 2014;31(1):4–16. DOI: 10.1080/02564602.2014.890837.
12. Carrasco R., Iyengar G., Stein C. Resource cost aware scheduling. *European Journal of Operational Research*. 2018;269(2):621–632. DOI: 10.1016/j.ejor.2018.02.059.
13. Coninck E., Verbelen T., Vankeirsbilck B., Bohez S., Simoens P., Dhoedt, B. Dynamic auto-scaling and scheduling of deadline constrained service workloads on IaaS clouds. *Journal of Systems and Software*. 2016;118:101–114. DOI: 10.1016/j.jss.2016.05.011.
14. Yi P., Ding H., Ramamurthy B. Budget-minimized resource allocation and task scheduling in distributed grid/clouds. *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Nassau, Bahamas*. 2013. p. 1–8. DOI: 10.1109/ANTS.2013.6802891.
15. Reddy G. A deadline and budget constrained cost and time optimization algorithm for cloud computing. *Commun. Comput. Inf. Sci*. 2011;193:455–462.
16. Xin Y., Xie Z.Q., Yang J. A load balance oriented cost efficient scheduling method for parallel tasks. *Journal of Network and Computer Applications*. 2018;81:37–46. DOI: 10.1016/j.jnea.2016.12.032.

17. Yang S.J., Chen Y.R. Design adaptive task allocation scheduler to improve MapReduce performance in heterogeneous Clouds. *Journal of Network and Computer Applications*. 2015;57:61–70. DOI: 10.1016/j.jnca.2015.07.012.
18. Li Z., Chang V., Hu Haiyang, Hu Hua. Real-time and dynamic fault-tolerant scheduling for scientific workflows in Clouds. *Information Science*. 2021;568(12). DOI: 10.1016/j.ins.2021.03.003.
19. Zhou Z., Abawajy J., Chowdhury M., Hu Z., Li K., Cheng H., Alelaiwi A., Li F. Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*. 2017;86:836–850. DOI: 10.1016/j.future.2017.07.048.
20. Pradhan R., Satapathy S. Energy-aware cloud task scheduling algorithm in heterogeneous multi-cloud environment. *Intelligent Decision Technologies*. 2022;16(8):1–6. DOI: 10.3233/IDT-210048.
21. Bryukhanova E.R., Antamoshkin O.A. Minimizing the carbon footprint with the use of zeroing neural networks. *The European Proceedings of Computers and Technology*. 2023. DOI: 10.15405/epct.23021.20.
22. Duan H., Chen C., Min G., Wu Y. Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*. 2017;74:142–150. DOI: 10.1016/j.future.2016.02.016.
23. Shaikh M.B., Waghmare Shinde K., Borde S. Challenges of big data processing and scheduling of processes using various Hadoop schedulers: a survey. *Int. J. Multifaceted Multiling. Stud.* 2019;III:1–6.
24. Reddy G., Kumar S. MACO-MOTS: Modified ant colony optimization for multi objective task scheduling in cloud environment. *International Journal of Intelligent Systems and Applications*. 2019;11(1):73–79. DOI: 10.5815/ijisa.2019.01.08.
25. Biswas D., Samsuddoha M., Asif M.R.A., Ahmed M.M. Optimized round robin scheduling algorithm using dynamic time quantum approach in cloud computing environment. *International Journal of Intelligent Systems and Applications*. 2023;15(1):22–34. DOI: 10.5815/ijisa.2023.01.03.
26. Soltani N., Barekatin B., Soleimani Neysiani B. MTC: Minimizing time and cost of cloud task scheduling based on customers and providers needs using genetic algorithm. *I.J. Intelligent Systems and Applications*. 2021;2:38–51. DOI: 10.5815/ijisa.2021.02.03.
27. Mohseni Z., Kiani V., Rahmani A. A Task scheduling model for multi-CPU and multi-hard disk drive in soft real-time systems. *International Journal of Information Technology and Computer Science*. 2019;11(1):1–13. DOI: 10.5815/ijitcs.2019.01.01.
28. Zaharia M., Borthakur D., Sen Sarma J., Elmeleegy K., Shenker S., Stoica I. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. *European Conference on Computer Systems, Proceedings of the 5th European conference on Computer systems, EuroSys 2010, April 13–16 2010, Paris, France*. p. 265–278. DOI: 10.1145/1755913.1755940.
29. Bouhouch L., Zbakh M., Tadonki C. Dynamic data replication and placement strategy in geographically distributed data centers. *Concurrency and Computation Practice and Experience*. 2022;35(11). DOI: 10.1002/cpe.6858.
30. Samadi Y., Zbakh M., Tadonki C. DT-MG: Many-to-one matching game for tasks scheduling towards resources optimization in cloud computing. *International Journal of Computers and Applications*. 2020;43(6):1–13. DOI: 10.1080/1206212X.2018.1519630.

REFERENCES

1. Dubey K., Kumar M. Sharma S. Modified HEFT algorithm for task scheduling in cloud environment. *Procedia Computer Science*. 2018;125:725–732. DOI: 10.1016/j.procs.2017.12.093.
2. Mondal R., Nandi E., Sarddar D. Load balancing scheduling with shortest load first. *International Journal of Grid and Distributed Computing*. 2015;8:171–178. DOI: 10.14257/ijgdc.2015.8.4.17.
3. Lakra A.V., Yadav D.K. Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. *Procedia Computer Science*. 2015;48:107–113. DOI: 10.1016/j.procs.2015.04.158.
4. Wang H., Wang F., Liu J., Wang D., Groen J. Enabling customer-provided resources for cloud computing: Potentials, challenges, and implementation. *IEEE Transactions on Parallel and Distributed Systems*. 2015;26:1874–1886.
5. Gill S.S., Chana I., Singh M., Buyya R. CHOPPER: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing*. 2018;21:1203–1241. DOI: 10.1007/s10586-017-1040-z.
6. Thomas A., Krishnalal G., Raj P.V. Credit based scheduling algorithm in cloud computing environment. *Procedia Computer Science*. 2015;46:913–920. DOI: 10.1016/j.procs.2015.02.162.
7. Sajid M., Raza, Z. Turnaround time minimization-based static scheduling model using task duplication for fine-grained parallel applications onto hybrid cloud environment. *IETE Journal of Research*. 2015;62(3):1–13. DOI: 10.1080/03772063.2015.1075911.
8. Hadji M., Zeghlache D. Minimum cost maximum flow algorithm for dynamic resource allocation in clouds. *2012 IEEE 5th International Conference on Cloud Computing (CLOUD), Honolulu, HI, USA*. 2012. p. 876–882. DOI: 10.1109/CLOUD.2012.36.
9. Elzeki O., Reshad M., Abu Elsoud, M. Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*. 2012;50(12):22–27. DOI: 10.5120/7823-1009.
10. Fernández Cerero D., Fernández-Montes A., Jakóbič A., Kołodziej J., Toro M. SCORE: Simulator for cloud optimization of resources and energy consumption. *Simulation Modelling Practice and Theory*. 2018;82:160–173. DOI: 10.1016/j.simpat.2018.01.004.
11. Ma T., Chu Y., Zhao L., Otgonbayar A. Resource allocation and scheduling in cloud computing: policy and algorithm. *IETE Technical Review*. 2014;31(1):4–16. DOI: 10.1080/02564602.2014.890837.
12. Carrasco R., Iyengar G., Stein C. Resource cost aware scheduling. *European Journal of Operational Research*. 2018;269(2):621–632. DOI: 10.1016/j.ejor.2018.02.059.
13. Coninck E., Verbelen T., Vankeirsbilck B., Bohez S., Simoens P., Dhoedt, B. Dynamic auto-scaling and scheduling of deadline constrained service workloads on IaaS clouds. *Journal of Systems and Software*. 2016;118:101–114. DOI: 10.1016/j.jss.2016.05.011.
14. Yi P., Ding H., Ramamurthy B. Budget-minimized resource allocation and task scheduling in distributed grid/clouds. *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Nassau, Bahamas*. 2013. p. 1–8. DOI: 10.1109/ANTS.2013.6802891.
15. Reddy G. A deadline and budget constrained cost and time optimization algorithm for cloud computing. *Commun. Comput. Inf. Sci*. 2011;193:455–462.
16. Xin Y., Xie Z.Q., Yang J. A load balance oriented cost efficient scheduling method for parallel tasks. *Journal of Network and Computer Applications*. 2018;81:37–46. DOI: 10.1016/j.jnea.2016.12.032.

17. Yang S.J., Chen Y.R. Design adaptive task allocation scheduler to improve MapReduce performance in heterogeneous Clouds. *Journal of Network and Computer Applications*. 2015;57:61–70. DOI: 10.1016/j.jnca.2015.07.012.
18. Li Z., Chang V., Hu Haiyang, Hu Hua. Real-time and dynamic fault-tolerant scheduling for scientific workflows in Clouds. *Information Science*. 2021;568(12). DOI: 10.1016/j.ins.2021.03.003.
19. Zhou Z., Abawajy J., Chowdhury M., Hu Z., Li K., Cheng H., Alelaiwi A., Li F. Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*. 2017;86:836–850. DOI: 10.1016/j.future.2017.07.048.
20. Pradhan R., Satapathy S. Energy-aware cloud task scheduling algorithm in heterogeneous multi-cloud environment. *Intelligent Decision Technologies*. 2022;16(8):1–6. DOI: 10.3233/IDT-210048.
21. Bryukhanova E.R., Antamoshkin O.A. Minimizing the carbon footprint with the use of zeroing neural networks. *The European Proceedings of Computers and Technology*. 2023. DOI: 10.15405/epct.23021.20.
22. Duan H., Chen C., Min G., Wu Y. Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*. 2017;74:142–150. DOI: 10.1016/j.future.2016.02.016.
23. Shaikh M.B., Waghmare Shinde K., Borde S. Challenges of big data processing and scheduling of processes using various Hadoop schedulers: a survey. *Int. J. Multifaceted Multiling. Stud.* 2019;III:1–6.
24. Reddy G., Kumar S. MACO-MOTS: Modified ant colony optimization for multi objective task scheduling in cloud environment. *International Journal of Intelligent Systems and Applications*. 2019;11(1):73–79. DOI: 10.5815/ijisa.2019.01.08.
25. Biswas D., Samsuddoha M., Asif M.R.A., Ahmed M.M. Optimized round robin scheduling algorithm using dynamic time quantum approach in cloud computing environment. *International Journal of Intelligent Systems and Applications*. 2023;15(1):22–34. DOI: 10.5815/ijisa.2023.01.03.
26. Soltani N., Barekatin B., Soleimani Neysiani B. MTC: Minimizing time and cost of cloud task scheduling based on customers and providers needs using genetic algorithm. *I.J. Intelligent Systems and Applications*. 2021;2:38–51. DOI: 10.5815/ijisa.2021.02.03.
27. Mohseni Z., Kiani V., Rahmani A. A Task scheduling model for multi-CPU and multi-hard disk drive in soft real-time systems. *International Journal of Information Technology and Computer Science*. 2019;11(1):1–13. DOI: 10.5815/ijitcs.2019.01.01.
28. Zaharia M., Borthakur D., Sen Sarma J., Elmeleegy K., Shenker S., Stoica I. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. *European Conference on Computer Systems, Proceedings of the 5th European conference on Computer systems, EuroSys 2010, April 13–16 2010, Paris, France*. p. 265–278. DOI: 10.1145/1755913.1755940.
29. Bouhouch L., Zbakh M., Tadonki C. Dynamic data replication and placement strategy in geographically distributed data centers. *Concurrency and Computation Practice and Experience*. 2022;35(11). DOI: 10.1002/cpe.6858.
30. Samadi Y., Zbakh M., Tadonki C. DT-MG: Many-to-one matching game for tasks scheduling towards resources optimization in cloud computing. *International Journal of Computers and Applications*. 2020;43(6):1–13. DOI: 10.1080/1206212X.2018.1519630.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Брюханова Евгения Романовна, аспирант
Сибирского государственного университета
науки и технологий им. М.Ф. Решетнева,
инженер-исследователь
Сибирского
федерального университета,
Красноярск,
Российская Федерация.

e-mail: evgbryuhanova@gmail.com

ORCID: [0000-0001-6176-837X](https://orcid.org/0000-0001-6176-837X)

Evgeniia R. Bryukhanova, Postgraduate
Student, Reshetnev Siberian State University of
Science and Technology, Research Engineer,
Siberian Federal University, Krasnoyarsk,
the Russian Federation.

Антамошкин Олеслав Александрович,
доктор технических наук, профессор,
Сибирский федеральный университет,
Сибирский государственный университет
науки и технологий им. М.Ф. Решетнева
Красноярск, Российская Федерация.

e-mail: oleslav24@gmail.com

ORCID: [0000-0002-5976-5847](https://orcid.org/0000-0002-5976-5847)

Oleslav A. Antamoshkin, Doctor of Technical
Sciences, Professor, Reshetnev Siberian State
University of Science and Technology, Siberian
Federal University, Krasnoyarsk, the Russian
Federation.

*Статья поступила в редакцию 12.09.2023; одобрена после рецензирования 20.09.2023;
принята к публикации 04.10.2023.*

*The article was submitted 12.09.2023; approved after reviewing 20.09.2023;
accepted for publication 04.10.2023.*