

УДК 004.75

DOI: [10.26102/2310-6018/2026.55.4.014](https://doi.org/10.26102/2310-6018/2026.55.4.014)

## Распределенная вычислительная система на базе мобильных устройств

Ф.И. Исаев✉, Г.Н. Исаева

*Национальный исследовательский ядерный университет, Москва,  
Российская Федерация*

**Резюме.** В работе рассматривается архитектура распределенной вычислительной системы, построенной на базе гетерогенных мобильных устройств и использующей комбинированный метод динамической балансировки нагрузки. Данный подход ориентирован на беспроводные среды, где состав узлов и их производительность изменяются во времени. Проведен анализ производительности смартфонов как вычислительных узлов, исследованы факторы, ограничивающие эффективность их применения: гетерогенность аппаратных платформ, тепловой троттлинг, неоднородность вычислительных ядер и загруженность фоновой активностью. Предложен алгоритм, объединяющий статическую оценку мощности узлов и динамическую корректировку коэффициентов производительности с учетом частоты, температуры и текущей загрузки процессора. Алгоритм имеет механизм отказоустойчивого перераспределения подзадач: при отключении или зависании узла незавершенные подзадачи автоматически возвращаются в очередь и назначаются другим исполнителям. Предложенный подход обеспечивает адаптацию распределения нагрузки к текущему состоянию вычислительных узлов, поддерживая стабильность общей производительности при колебаниях их ресурсов. Экспериментальная проверка выполнена на наборе смартфонов разных классов, а в качестве тестовой нагрузки использовалась задача без межузлового обмена данными. Экспериментальная оценка подтверждает, что разработанный метод обеспечивает значительное снижение времени выполнения задач и минимизацию дисперсии нагрузки по сравнению со статическими подходами.

**Ключевые слова:** распределенные вычисления, динамическая балансировка нагрузки, отказоустойчивость, грид-подход, тепловой троттлинг.

**Для цитирования:** Исаев Ф.И., Исаева Г.Н. Распределенная вычислительная система на базе мобильных устройств. *Моделирование, оптимизация и информационные технологии*. 2026;14(4). URL: <https://moitvivr.ru/ru/journal/article?id=2251> DOI: 10.26102/2310-6018/2026.55.4.014

## Distributed computing system based on mobile devices

F.I. Isaev✉, G.N. Isaeva

*National Research Nuclear University, Moscow, the Russian Federation*

**Abstract.** This paper examines the architecture of a distributed computing system built on heterogeneous mobile devices and employing a combined dynamic load balancing method. This approach is focused on wireless environments where the composition of nodes and their performance vary over time. The performance of smartphones as computing nodes is analyzed, and the factors limiting their effectiveness are investigated: heterogeneity of hardware platforms, thermal throttling, heterogeneity of computing cores, and background activity load. An algorithm is proposed that combines a static assessment of node capacity and dynamic adjustment of performance factors taking into account the frequency, temperature, and current processor load. The algorithm incorporates a fault-tolerant subtask redistribution mechanism: if a node is disconnected or freezes, unfinished subtasks are automatically returned to the queue and assigned to other workers. The proposed approach ensures adaptation of load distribution to the current state of computing nodes, maintaining stability of overall performance during fluctuations

in their resources. Experimental testing was performed on a set of smartphones of different classes, using a task without inter-node data exchange as the test load. The experimental evaluation confirms that the developed method significantly reduces task execution time and minimizes load variance compared to static approaches.

**Keywords:** distributed computing, dynamic load balancing, fault tolerance, grid approach, thermal throttling.

**For citation:** Isaev F.I., Isaeva G.N. Distributed computing system based on mobile devices. *Modeling, Optimization and Information Technology*. 2026;14(4). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2251> DOI: 10.26102/2310-6018/2026.55.4.014

## Введение

Современные вычислительные технологии все чаще сталкиваются с необходимостью обработки больших объемов данных в условиях ограниченного доступа к традиционной серверной инфраструктуре. Аренда облачных ресурсов или приобретение даже небольшого стационарного кластера электронных хранилищ остаются экономически недоступными для многих научных групп, образовательных учреждений и малых предприятий. В то же время, по состоянию на 2025 год в мире насчитывается более 6,8 миллиарда активных смартфонов<sup>1</sup>, вычислительная мощность которых уже сравнима с настольными компьютерами пяти-семилетней давности [1]. Это делает мобильные устройства перспективной платформой для построения распределенных вычислительных систем с минимальными капитальными затратами.

Исследования по использованию смартфонов в качестве вычислительных узлов проводятся более пятнадцати лет. В ранних работах [1, 2], была доказана принципиальная возможность построения полностью мобильных MPI-кластеров (проект Droid Cluster) и выполнения на данных объектах задач научных вычислений с приемлемой эффективностью. Позднее, в исследованиях 2012–2014 гг. [3, 4], предложена концепция Computing While Charging, ориентированная на беспроводные среды, а также разработаны методы балансировки с учетом пропускной способности сети. В публикациях последних лет [5, 6] созданы программно-аппаратные платформы для объединения гетерогенных смартфонов и микрокомпьютеров, а также предложены подходы к мониторингу реального теплового состояния процессоров.

Несмотря на достигнутый прогресс, существующие решения имеют ряд критических недостатков при применении в реальных беспроводных гетерогенных средах:

- большинство методов балансировки нагрузки ориентировано на оценку пропускной способности канала связи [3, 4], что делает их неприменимыми при использовании в полностью децентрализованных ad-hoc- и mesh-сетях без стационарного канала;
- недостаточно проработаны механизмы реакции на динамическое снижение производительности из-за достаточно высокого значения таких характеристик вычислительной системы, как тепловыделение и загруженность процессора, что сказывается на общей продуктивности кластера;
- отсутствуют эффективные алгоритмы миграции задач в условиях сильной гетерогенности (разница в производительности узлов 10–12 раз).

<sup>1</sup> GSMA Intelligence. *The Mobile Economy 2025*. GSMA. URL: <https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2026/01/The-Mobile-Economy-2025.pdf> (дата обращения: 14.09.2025).

## Материалы и методы

В настоящей работе приводятся результаты исследований распределенной вычислительной системы на базе смартфонов с комбинированным методом балансировки нагрузки, который обеспечивает отказоустойчивость, перераспределение подзадач, динамически реагирует на троттлинг и текущую частоту ядер. Цель исследования – проанализировать и экспериментально подтвердить эффективность распределенной вычислительной системы на базе гетерогенных мобильных устройств, обеспечивающей высокую производительность и устойчивость к снижению вычислительного потенциала узлов в условиях ограниченной инфраструктуры.

*Вычисления на мобильных узлах.* Современные смартфоны обладают значительным вычислительным потенциалом; однако их использование в качестве узлов распределенных систем требует учета ряда принципиальных ограничений. Ключевой проблемой является существенная гетерогенность аппаратных платформ (Таблица 1) – даже устройства одного поколения могут значительно различаться по количеству ядер, тактовой частоте и объему оперативной памяти. Как видно из таблицы, указанные характеристики различаются в два и более раз. Особенностью мобильных процессоров стала популярная архитектура big.LITTLE или ее аналоги (performance + efficiency ядра), где мощные и энергоэффективные ядра комбинируются в одном чипе, что создает дополнительные сложности при распределении нагрузки [7]. Серьезным ограничением подобных архитектур выступает тепловыделение – отсутствие активного охлаждения приводит к троттлингу и снижению производительности на 50–60 % при длительных вычислениях.

Таблица 1 – Характеристики узлов кластера  
Table 1 – Cluster node characteristics

Модель	SoC (чипсет)	Процессор	Оперативная память (Гб)	Тактовая частота (ГГц)
Samsung S24	Snapdragon 8 Gen 3	ARM 1×Cortex-X4, ARM 2×A720 ARM 3×A720, ARM 4×A520	10	3,2×2,9× ×2,6×2,0
Samsung A52	Snapdragon 720G	ARM 4×Cortex-A73, ARM 4×Cortex-A53	4	2,4×1,8
Huawei P20	Kirin 970	2×Kryo 465 Gold, 6×Kryo 465 Silver	6	2,3×1,8
Xiaomi Mi 9SE	Snapdragon 712	2×Cortex-A75, 6×Cortex-A55	6	2,3×1,7
ПК	Intel Core i7-13700H	6×P-core (Raptor Cove), 8×E-core (Gracemont)	16	4,5×3,2

Чтобы добиться эффективного использования мобильных устройств в распределенных вычислениях необходимо использовать грид-подход, объединяющий разнородные вычислительные ресурсы в единую инфраструктуру [8, 9]. Такая система строится по иерархическому принципу, где управляющий узел координирует работу подчиненных узлов, которые выполняют вычислительные задачи (Рисунок 1). Коммуникация между узлами осуществляется через стандартные сетевые протоколы с оптимизацией для беспроводных соединений (Nearby Connections API). В отличие от традиционных грид-систем, ориентированных на стационарные ресурсы, предложенная

локальная архитектура учитывает динамическое подключение и отключение узлов, вызванное мобильностью или изменением состояния устройств.

Для оценки гетерогенности такой системы вычисляемая задача должна обладать возможностью ее параллельной реализации в распределенной системе, что предполагает разделение задачи на произвольное количество независимых подзадач без обмена данными в процессе вычисления. Численное интегрирование по методу трапеций является подходящим математическим методом для количественной оценки гетерогенности системы. Предлагаемый подход хорошо удовлетворяет исходным данным: каждая подзадача характеризуется только начальной и конечной точками отрезка и количеством шагов.

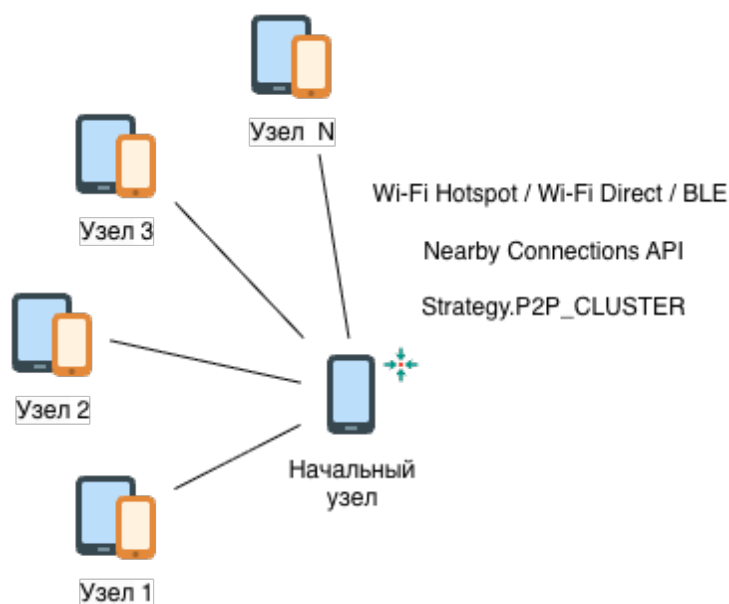


Рисунок 1 – Топология распределенной системы на базе мобильных устройств  
 Figure 1 – Topology of a distributed system based on mobile devices

Ниже представлена выбранная для исследования функция – прототип задачи, решаемой в системе, содержащая основные математические операции:

$$y = \int_{100}^0 x^3 \sin(x) dx. \quad (1)$$

Вычисления на узлах оценивались по формуле:

$$\int_b^a(x) dx \approx h \sum_{i=0}^{N-1} f(x_i), \quad (2)$$

где  $h = \frac{b-a}{N}$ ,  $N$  – число разбиений задачи в процессе вычислений.

Относительные показатели гетерогенности для рассматриваемых систем и используемого математического подхода приведены в Таблице 2. Основными показателями являются время выполнения задачи на каждом из узлов кластера и относительная производительность узла в системе. Исследование проводилось при числе шагов интегрирования  $N = 10^9$  и на максимально возможном числе физических ядер узла.

Полученные данные констатируют тот факт, что в рассматриваемой системе имеется существенный дисбаланс в производительности узлов кластера. Это напрямую влияет на критерий эффективности работы распределенной системы – минимизацию времени выполнения задачи. Только гетерогенность среды вычисления не позволяет

добиться повышения производительности системы в целом простейшим способом – декомпозицией задачи на равные части, требуется еще учитывать характеристики каждого узла, то есть выполнять балансировку нагрузки на каждом узле системы.

Таблица 2 – Показатели качества выполнения задачи интегрирования на узлах кластера  
Table 2 – Performance indicators of the integration task on cluster nodes

Узел	Время,с	Относительная производительность
Samsung S24	4,29	4,47
Samsung A52	7,62	2,52
Huawei P20	11,83	1,63
Xiaomi Mi 9SE	19,21	1

*Балансировка нагрузки в гетерогенной системе.* Для балансировки нагрузки узлов вычислительной системы используются статические и динамические методы. Статическая балансировка происходит до запуска задачи, при этом структура системы остается неизменной на протяжении всего времени ее работы. Такой метод балансировки прост в реализации и используется в большинстве подобных исследований, отдельно выделяются два подтипа [10].

Первый способ базируется на распределении, учитывающем пиковую производительность устройств. Он включает в себя формирование коэффициента относительной мощности узла и использует следующие соотношения:

$$K(i) = \frac{P_{abs}(i)}{\sum_{j=1}^n P_{abs}(j)}, \quad P_{abs} = N_{cores} \cdot F_{cores} \cdot OpC, \quad (3)$$

где  $N_{cores}$  – количество ядер,  $F_{cores}$  – частота каждого ядра (ГГц),  $OpC$  – количество операций за такт.

Второй метод основан на реальной производительности узлов. Для оценки учитываются временные параметры  $T(i)$ , полученные при равномерном распределении. Разнородность узлов отражают следующие формулы:

$$S(i) = \frac{minT}{T(i)}, \quad Load(i) = \frac{S(i)}{\sum_{j=1}^n S(j)}. \quad (4)$$

Однако данные методы являются малоэффективными. Любое изменение мощности узла в процессе работы делает полученные статические коэффициенты недостоверными. Не учитывается и тротлинг, возникающий при длительной высокой нагрузке. Из-за разнородности ядер даже смартфоны с высоким значением пиковой производительности могут иметь показатели ниже устройств с малым коэффициентом, но большим количеством энергоэффективных ядер. Важным показателем в рассматриваемых системах является и загруженность процессора множеством фоновых задач. Выходом из приведенных положений, ограничивающим достоверность получаемых характеристик распределенной системы, является создание и анализ динамической балансировки нагрузки узлов.

В данном исследовании для реализации распределенных вычислений на гетерогенных мобильных устройствах был разработан алгоритм комбинированной балансировки нагрузки, сочетающий статическую и динамическую оценку производительности узлов, а также встроенные механизмы отказоустойчивости (Рисунок 2).

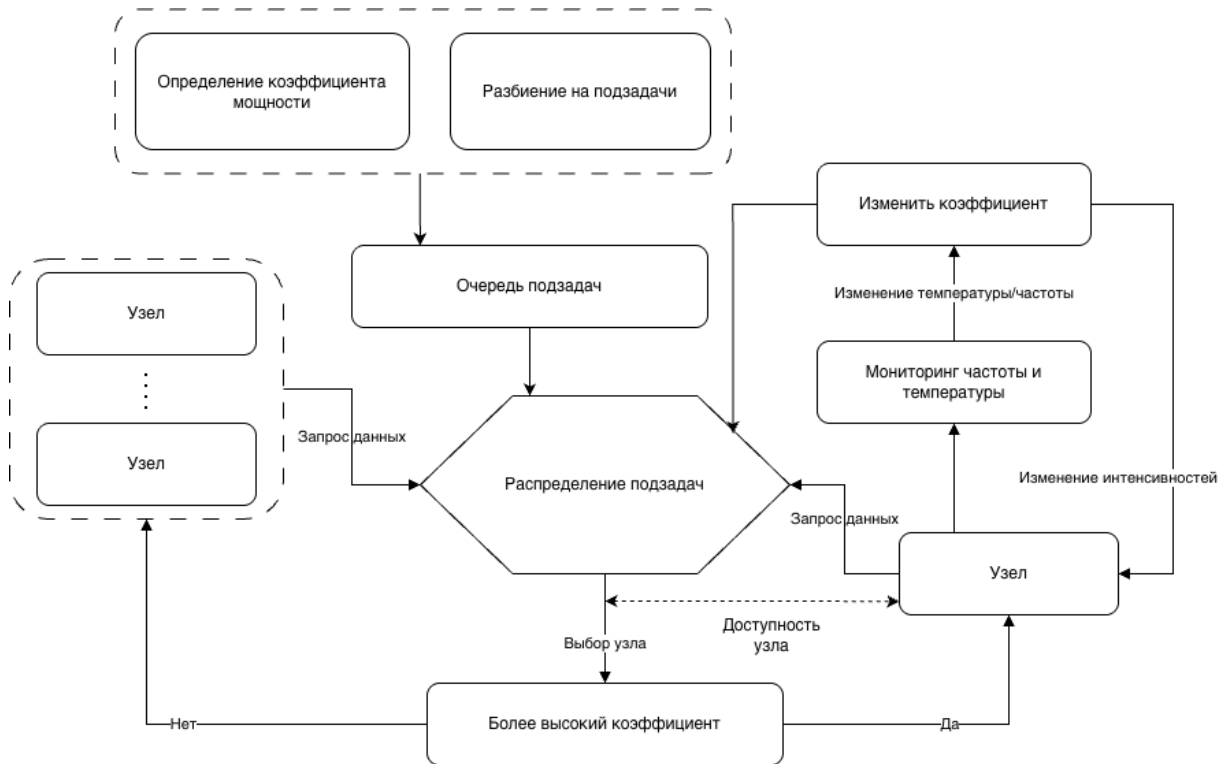


Рисунок 2 – Схема работы комбинированного метода динамической балансировки  
Figure 2 – The operating diagram of the combined dynamic balancing method

Проводится измерение пиковой производительности на стадии инициализации каждого локального узла. Она может быть определена как через количество ядер узла, их тактовые частоты и эффективную производительность архитектуры (3). На основе этих значений формируется нормированная система статических коэффициентов, определяющих относительную долю вычислений, первоначально передаваемых узлу. Глобальная задача разбивается на множество подзадач, формируя очередь.

Во время выполнения вычислений используется динамический компонент алгоритма, отражающий поведение системы при балансировке (Рисунок 3). Каждый вычислительный узел в процессе эксплуатации системы передает управляющему узлу информацию о своей частоте, температуре, загрузке центрального процессора и наличии признаков деградации. На основе этих данных вычисляется динамический коэффициент производительности, который отражает текущее состояние узла и корректирует первоначальную статическую оценку. Таким образом, вводятся обратные связи для узла и коэффициента, характеризующего долю текущих вычислений на узле системы.

В случае обнаружения недоступности узла – как вследствие отключения, так и из-за зависания – алгоритм немедленно переводит незавершенные подзадачи в очередь и перераспределяет их между оставшимися активными узлами в соответствии с актуальными динамическими коэффициентами.

Algorithm 1: Combined Dynamic Load Balancing

Input: Global task T, number of subtasks N, set of worker nodes U

Output: Distributed execution of T with dynamic coefficients and failover

```

1: // ----- ЭТАП ИНИЦИАЛИЗАЦИИ -----
2: for each node u in U do
3:   P_static(u) ← measurePeakPerformance(u)
4: end for
5: normalize(P_static)           // формирование статических коэффициентов
6: Q ← partitionTask(T, N, P_static) // очередь подзадач с учётом статической
    мощности

7: // ----- ОСНОВНОЙ ЦИКЛ -----
8: while Q is not empty do
9:   for each node u in U do
10:    metrics ← monitorNode(u)
11:    // metrics: frequency, temperature, CPU load, availability
12:
13:    if metrics.available = false then
14:      failedSubtasks ← collectUnfinished(u)
15:      Q ← Q ∪ failedSubtasks // возврат подзадач в очередь
16:      continue
17:    end if
18:
19:    K_dynamic(u) ← updateCoefficient(metrics)
20:  end for
21:
22:  u* ← selectBestNode(U, K_dynamic)
23:  s ← dequeue(Q)
24:  assign(s, u*) // отправка подзадачи узлу
25: end while
26:
27: // ----- ОБРАБОТКА ПАДЕНИЯ УЗЛОВ -----
28: onNodeFailure(u):
29:   failedSubtasks ← collectUnfinished(u)
30:   reenqueue(Q, failedSubtasks)
31:   redistribute(Q, U)

32: return SUCCESS
    
```

Рисунок 3 – Алгоритм комбинированной динамической балансировки  
 Figure 3 – Combined dynamic balancing algorithm

Такой метод эффективен и при длительных вычислениях: за счет замыкания по температуре, узел сам снимет нагрузку с hard-потоков, не допуская троттлинга, а при восстановлении оптимального режима, продолжает вычисления при полной мощности.

Результаты оценки эффективности по выбранным показателям для рассматриваемых методов балансировки в распределенной системе приведены в Таблице 3.

Комбинированный метод балансировки GRID (грид-вычисления) обеспечивает максимальную эффективность вычислений и близок по времени решения к современным стационарным станциям. Другие виды балансировок уступают ей, особенно при длительной нагрузке. Объективной оценкой уровня сбалансированности кластера служит дисперсия по выборке значений времени работы узлов при разных видах балансировки. Для комбинированного метода она имеет минимальное значение  $D = 0,31$ , что также подтверждает его преимущество.

Таблица 3 – Сравнение подходов балансировки  
 Table 3 – Comparison of balancing approaches

Тип балансировки	Время вычисления (сек)	Время вычисления длительной задачи (мин)
По пиковой частоте	13,53	26,9
Статическая	7,04	24,7
Комбинированная динамическая	2,94	7,3

### Заключение

Проведенное исследование демонстрирует значительный потенциал смартфонов как вычислительных узлов, несмотря на такие ограничения, как гетерогенность устройств и тепловыделение.

Предложенная архитектура, основанная на грид-подходе, и комбинированный метод динамической балансировки нагрузки обеспечивают высокую производительность и масштабируемость системы. В дальнейших исследованиях целесообразно углубить анализ энергоэффективности и адаптации системы к реальным сценариям эксплуатации.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

- Rodríguez J.M., Mateos C., Zunino A. Are smartphones really useful for scientific computing? In: *Advances in New Technologies, Interactive Interfaces and Communicability: Second International Conference, ADNTIIC 2011, 05–07 December 2011, Huerta Grande, Argentina*. Berlin, Heidelberg: Springer; 2012. P. 38–47. [https://doi.org/10.1007/978-3-642-34010-9\\_4](https://doi.org/10.1007/978-3-642-34010-9_4)
- Büsching F., Schildt S., Wolf L. DroidCluster: Towards smartphone cluster computing – The streets are paved with potential computer clusters. In: *2012 32<sup>nd</sup> International Conference on Distributed Computing Systems Workshops, 18–21 June 2012, Macau, China*. IEEE; 2012. P. 114–117. <https://doi.org/10.1109/ICDCSW.2012.59>
- Arslan M.Y., Singh I., Singh Sh., et al. Computing while charging: building a distributed computing infrastructure using smartphones. In: *CoNEXT '12: Proceedings of the 8<sup>th</sup> International Conference on Emerging Networking Experiments and Technologies, 10–13 December 2012, Nice, France*. New York: ACM; 2012. P. 193–204. <https://doi.org/10.1145/2413176.2413199>
- Shiraz M., Gani A., Khokhar R.H., Buyya R. A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Communications Surveys & Tutorials*. 2013;15(3):1294–1313. <https://doi.org/10.1109/SURV.2012.111412.00045>
- Балабаев С.А., Лупин С.А., Телегин П.Н., Шабанов Б.М. Повышение вычислительной мощности персонального компьютера за счёт интеграции с распределённой системой из смартфонов. *Программные продукты и системы*. 2024;(4):504–513. <https://doi.org/10.15827/0236-235X.148.504-513>  
 Balabaev S.A., Lupin S.A., Telegin P.N., Shabanov B.M. Increasing the PC computing power: integration with a distributed smartphone system. *Software & Systems*. 2024;(4):504–513. (In Russ.). <https://doi.org/10.15827/0236-235X.148.504-513>
- Исаев Ф.И., Исаева Г.Н. Анализ ограничений мобильных сетей и потенциала распределенных вычислений на смартфонах. *Известия Кабардино-Балкарского*

- научного центра РАН. 2025;27(4):24–34. <https://doi.org/10.35330/1991-6639-2025-27-4-24-34>
- Isaev F.I., Isaeva G.N. Analysis of limited mobile networks and the potential of distributed mobile computing. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2025;27(4):24–34. (In Russ.). <https://doi.org/10.35330/1991-6639-2025-27-4-24-34>
7. Qin Y., Zeng G., Kurachi R., Matsubara Y., Takada H. Execution-variance-aware task allocation for energy minimization on the big.LITTLE architecture. *Sustainable Computing: Informatics and Systems*. 2019;22:155–166. <https://doi.org/10.1016/j.suscom.2018.10.001>
  8. Долгов А.А. Разворачивание грид-системы из мобильных устройств на платформе VOINC. В сборнике: *Облачные и распределенные вычислительные системы в электронном управлении: Сборник трудов 3-й международной научно-технической конференции, 29 ноября – 02 декабря 2022 года, Переславль-Залесский, Россия*. Переславль-Залесский: Институт программных систем им. А.К. Айламазяна РАН; 2023. С. 24–29.
  9. Krioukov A., Mohan P., Alspaugh S., et al. NapSAC: Design and implementation of a power-proportional web cluster. In: *Green Networking '10: Proceedings of the first ACM SIGCOMM workshop on Green networking, 30 August 2010, New Delhi, India*. New York: ACM; 2010. P. 15–22. <https://doi.org/10.1145/1851290.1851294>
  10. Кхайнг М.Т., Лупин С.А., Тху А. Оценка эффективности методов балансировки нагрузки в распределенных вычислительных системах. *International Journal of Open Information Technologies*. 2021;9(11):30–36.  
Khaing M.T., Lupin S., Thu A. Evaluating the effectiveness of load balancing methods in distributed computing systems. *International Journal of Open Information Technologies*. 2021;9(11):30–36. (In Russ.).

#### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Исаев Федор Игоревич**, аспирант, **Fedor I. Isaev**, Postgraduate, National Research  
Национальный исследовательский ядерный университет, Москва, Российская Федерация. Nuclear University, Moscow, the Russian  
Federation.  
*e-mail:* [isaev.fed2014@yandex.ru](mailto:isaev.fed2014@yandex.ru)  
ORCID: [0009-0007-7308-1381](https://orcid.org/0009-0007-7308-1381)

**Исаева Галина Николаевна**, кандидат технических наук, доцент, Национальный исследовательский ядерный университет, Москва, Российская Федерация. **Galina N. Isaeva**, Candidate of Engineering Sciences, Associate Professor, National Research Nuclear University, Moscow, the Russian Federation.  
*e-mail:* [gnisaeva@mephi.ru](mailto:gnisaeva@mephi.ru)  
ORCID: [0009-0008-0371-5972](https://orcid.org/0009-0008-0371-5972)

*Статья поступила в редакцию 25.02.2026; одобрена после рецензирования 16.04.2026; принята к публикации 21.04.2026.*

*The article was submitted 25.02.2026; approved after reviewing 16.04.2026; accepted for publication 21.04.2026.*