Оригинальная статья / Original article

УДК 681.3

https://doi.org/10.21869/2223-1560-2025-29-2-201-220

Оценка производительности вычислительной системы

Г. В. Петушков [1] ✉

[1] Российский технологический университет «РТУ МИРЭА»,
пр. Вернадского, д. 78, Москва 119454, Российская Федерация

✉ e-mail: petushkov@mirea.ru

Резюме

Цель работы. *Провести анализ и моделирование производительности вычислительных систем, включая расчет и сравнение различных показателей, таких как загрузка системы, пиковая и асимптотическая производительность, ускорение системы и ее реальная производительность, с использованием математических моделей для оценки эффективности работы систем в условиях динамичных задач и многозадачности. Особое внимание уделяется влиянию различных параметров системы на ее способность эффективно выполнять вычислительные операции и управлять ресурсами.*

Методы. *В данной работе были использованы методы математического моделирования для анализа характеристик вычислительных систем, расчет загрузки системы как среднего арифметического загрузок всех устройств, определение пиковой производительности системы через количество устройств и производительность каждого, вычисление ускорения системы как суммы загрузок устройств и соотношения выполненных операций и времени, оценку реальной и асимптотической производительности через минимальные пиковые значения, сравнение различных систем по показателям производительности и ускорения, симуляция и анализ для оценки влияния параметров на общую эффективность системы.*

Результаты. *В ходе исследования проведен анализ производительности гетерогенных вычислительных систем, включающих процессоры Intel Xeon и сопроцессоры Intel Xeon Phi. Было выявлено, что классическая модель оценки производительности, основанная на простой сумме возможностей узлов, существенно переоценивает реальные показатели из-за игнорирования архитектурных и системных особенностей, таких как задержки передачи данных и пропускная способность межсоединений. Современная модель, учитывающая векторизацию AVX-512, многоуровневую память и ограничение шины PCIe 4.0, позволила получить более точную оценку – около 1.99 ТФлопс для однородной конфигурации CPU+GPU. При этом пропускная способность PCIe выступает ограничением в совместной работе CPU и GPU. Анализ гетерогенных конфигураций с Xeon Phi 7120P и Xeon E5-2683 v4 показал значительный прирост производительности – до 2.67 ТФлопс, что превышает возможности однородных систем. Ключевым параметром, влияющим на эффективность, стал коэффициент размера очереди выгрузки mmm, определяющий максимальный размер обрабатываемого блока данных. Эксперименты показали, что при малых значениях mmm издержки передачи данных увеличивают общее время расчета, тогда как при оптимальном диапазоне m=25–35m = 25\text{–}35m=25–35 достигается минимальное время выполнения за счет баланса между размером очереди и накладными расходами на коммуникацию. Дальнейшее увеличение mmm приводит к стабилизации или незначительному росту времени работы из-за усложнения балансировки нагрузки и задержек. Полученные данные подтверждают, что правильный подбор параметров очереди является важным фактором оптимизации гетерогенных систем.*

Заключение. *Проведенное исследование подтвердило необходимость использования современных моделей оценки производительности, учитывающих архитектурные особенности, пропускную способность, межсоединений и системные ограничения, для точного прогнозирования вычислительных возможностей гетерогенных платформ. Классические методы оценки оказываются недостаточными, так как не учитывают задержки передачи данных, особенности памяти и параллелизм, что приводит к завышенным и нереалистичным прогнозам. Современные модели с учетом AVX-векторизации, многоуровневой памяти и пропускной способности PCIe позволяют получить адекватную оценку и выявить ограничения, важные для оптимизации.*

Ключевые слова*: вычислительные системы; производительность; реальная производительность; пиковая производительность; загрузка..*

Конфликт интересов: *Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.*

Для цитирования: Петушков Г. В. Оценка производительности вычислительной системы // Известия Юго-Западного государственного университета. 2025; 29(2): 201-220. https://doi.org/ 10.21869/2223-1560-2025-29-2-201-220.

Поступила в редакцию 04.04.2025 *Подписана в печать 30.05.2025* *Опубликована 23.07.2025*

Computational system performance evaluation

Grigory V. Petushkov [1] ✉

[1] MIREA – Russian Technological University
78, Vernadskogo str., Moscow 119454, Russian Federation

✉ e-mail: petushkov@mirea.ru

Abstract

Purpose of research. *To analyze and model the performance of computing systems, including the calculation and comparison of various metrics such as system utilization, peak and asymptotic performance, system acceleration and real performance, using mathematical models to evaluate the performance of systems under dynamic tasks and multitasking. Special attention is given to the effect of various system parameters on the system's ability to perform computational operations and resource management efficiently.*

Methods. *In this paper, mathematical modeling techniques were used to analyze the performance of computing systems, calculating the system load as the arithmetic average of the loads of all devices, determining the peak performance of the system through the number of devices and the performance of each, calculating the system acceleration as the sum of device loads and the ratio of operations performed to time, estimating the real and asymptotic performance through minimum peak values, comparing different systems in terms.*

Results*. The study analyzes the performance of heterogeneous computing systems including Intel Xeon processors and Intel Xeon Phi coprocessors. It was revealed that the classical performance evaluation model based on a simple sum of nodes' capabilities significantly overestimates real performance due to ignoring architectural and system peculiarities such as data transfer latency and interconnect bandwidth. A modern model that takes into account AVX-512 vectorization, multi-level memory, and PCIe 4.0 bus limitation resulted in a more accurate estimate of about 1.99 TFLops for a homogeneous CPU+GPU configuration. In this case, the PCIe bandwidth acts as a bottleneck in the joint operation of CPU and GPU. Analysis of heterogeneous configurations with Xeon Phi 7120P and Xeon E5-2683 v4 showed a significant performance gain of up to 2.67 TFLops, which exceeds the capabilities of homogeneous systems. The key parameter affecting the performance was the unload queue size factor mmm, which determines the maximum size of the processed data block. Experiments have shown that for small values of mmm, the*

communication overhead increases the total computation time, whereas the optimal range of m=25-35m = 25\text{-}35m=25-35 achieves the minimum execution time due to the balance between queue size and communication overhead. Further increase in mmm leads to stabilization or slight increase in runtime due to increased complexity of load balancing and delays. The obtained data confirm that proper selection of queueing parameters is an important factor in the optimization of heterogeneous systems.

***Conclusion**. This research has confirmed the necessity of using modern performance evaluation models that take into account architectural features, bandwidth, interconnects and system limitations to accurately predict the computational capabilities of heterogeneous platforms. Classical evaluation methods prove to be insufficient as they do not take into account data transfer latency, memory features and parallelism, resulting in overestimated and unrealistic predictions. Modern models taking into account AVX vectorization, multi-level memory and PCIe bandwidth allow us to obtain an adequate evaluation and identify real bottlenecks important for optimization.*

***Keywords**: computational systems; performance; real performance; peak performance; utilization.*

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

For citation: Petushkov G. V. Computational system performance evaluation. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2025; 29(2): 201-220 (In Russ.). https://doi.org/10.21869/2223-1560-2025-29-2-201-220.

Received 04.04.2025 *Accepted 30.05.2025* *Published 23.07.2025*

***

Введение

В современных условиях задача оценки производительности вычислительных систем приобретает особую значимость, так как их применение находит место в практически во всех областях знаний. Применение стандартных методик позволяет проводить сопоставление различных архитектур и аппаратных платформ, что, в свою очередь, предоставляет разработчикам и пользователям объективную основу для выбора оптимальных решений.

Под производительностью понимается количественная характеристика вычислительной мощности системы, отражающая объем выполняемых вычислений за единицу времени. Однако в силу отсутствия универсальной меры вычислительной работы, не существует единого, общепринятого подхода к ее измерению. В результате разработка методик оценки производительности требует учета множества факторов и контекстных особенностей и обоснований.

Типовая методология построения оценки включает последовательность этапов и выбор метрик, по которым будет проводиться количественная оценка производительности, установление зависимостей этих метрик от архитектуры системы и характера вычислительной нагрузки, что требует построения моделей как рабочей нагрузки, так и самой вычислительной системы. Важно, чтобы степень детализации обеих моделей была сопоставимой, обеспечивая корректность взаимных соотношений.

На основе указанных моделей формируется обобщенная модель производительности, включающая выбранные параметры как со стороны нагрузки, так и системы. Подстановка конкретных зна-

чений параметров позволяет получить численные оценки производительности в заданных условиях эксплуатации.

В условиях отсутствия государственного стандарта, проверка соответствия производительности вычислительных систем требованиям, осуществляется с опорой на нормативно-технические документы, разработанные самими организациями-разработчиками, либо на экспертные заключения специалистов этих организаций. При этом оценка производительности, как правило, проводится не для всего комплекса в целом, а для отдельных его компонентов [1, 2].

Проектирование и разработка нового программного обеспечения для вычислительных систем напрямую зависит от их производительности, что требует точного анализа их технических и архитектурных характеристик и обусловлено необходимостью обеспечения эффективности и устойчивости работы в условиях высокой нагрузки, что делает такие исследования особенно актуальными в контексте развития современных информационных технологий [3, 4].

Целью оценки производительности может быть, как сравнение уже существующих вычислительных систем, так и прогнозирование эффективности перспективных архитектур. В обоих случаях ключевым показателем остается быстродействие – количество операций, выполняемых системой в единицу времени, так же очень важно учитывать, как надежность программного обеспечения, так и аппаратных средств [5,6], при такой оценке важно использовать математическое моделирование, численные методы и комплексы программы [7].

В Российской Федерации принят курс, направленный на реализацию политики импортозамещения и переход на российские аппаратно-программные средства, в соответствии с этим были разработаны требования к вычислительным средствам, которые постоянно совершенствуются исходя из современных условий [8].

Современные вычислительные задачи, такие как обучение нейросетей, численное моделирование, параллельная обработка больших массивов данных, требуют высокопроизводительных вычислительных систем, способных эффективно использовать как центральные процессоры (CPU) и сопроцессоры, графические ускорители (GPU), так и специализированные многоядерной архитектуры Intel Xeon Phi, сочетающие в себе принципы CPU и GPU. При этом одной из ключевых проблем остается адекватная оценка производительности таких гетерогенных систем, учитывающая особенности архитектуры, межсоединений (PCI), памяти, загрузки компонентов.

Классические модели [9, 10] оценки, как правило, предполагают линейное суммирование пиковых или реальных мощностей отдельных вычислителей и не учитывают такие критически важные факторы, как пропускная способность, межсоединения (PCIe), наличие иерархии памяти, асимметрию загрузки между CPU и GPU, а также влияние параллелизма и архитектурных

ограничений. Это приводит к систематическим завышениям производительности при прогнозировании и не позволяет корректно масштабировать решения под реальные нагрузки.

В данном исследовании рассматривается применение современной, расширенной модели оценки производительности, учитывающей векторизацию (AVX-512), многоуровневую структуру памяти (L1–L3, HBM), ограничения шины PCIe и реальные коэффициенты загрузки устройств. Для этого проведено моделирование гетерогенной вычислительной системы, включающей процессоры Intel Xeon Gold и E5-2683 v4, ускорители NVIDIA A100 и Intel Xeon Phi 7120P, с целью сопоставления результатов, полученных по классической и современной моделям. Особое внимание уделено влиянию коэффициента размера очереди задач m, как ключевого параметра управления распределением нагрузки между CPU и сопроцессорами.

Результаты показали, что применение современной модели позволяет получить более точную и реалистичную оценку производительности вычислительной системы, выявить ограничения по пропускной способности PCIe и обосновать необходимость архитектурных и программных оптимизаций. Кроме того, проведен анализ влияния параметра m на производительность, что позволяет формировать практические рекомендации по настройке гетерогенных комплексов под конкретные вычислительные задачи. Полученные данные могут иметь прикладное значение для разработки и эксплуатации высокопроизводительных систем в научных и инженерных приложениях.

Материалы и методы

При использовании классического метода оценки производительности [11] для каждого устройства в системе определяются пиковая производительность π_i и загрузка p_i, характеризующие его способность выполнять операции под реальной нагрузкой. Система состоит из s устройств, как простых, так и конвейерных. Если устройства обладают пиковыми производительностями $\pi_1,\ldots,\pi_s$ и работают с загрузками $p_1,\ldots,p_s$, то реальная производительность всей системы вычисляется по формуле

$$r = \sum_{i=1}^{s} p_i \cdot \pi_i, \quad (1)$$

где r – совокупная производительность системы; s – число устройств; p_i – загрузка i-го устройства; π_i – пиковая производительность.

Поскольку суммарная производительность определяется через индивидуальные показатели всех функциональных устройств (ФУ), для анализа достаточно рассмотреть одно устройство.

Пусть выполнение одной операции на устройстве занимает время τ, за интервал T выполняется N операций. Тогда общая стоимость выполненной работы составит $N\tau$.

Для простого устройства максимальная стоимость работы равна T, и загрузка будет:

$$p=\frac{N\tau}{T}. \quad (2)$$

Реальная производительность устройства:

$$r=\frac{N}{T},\ \pi=\frac{1}{\tau}\Rightarrow r=p\cdot\pi. \quad (3)$$

Для конвейерного устройства с длиной конвейера n, максимальная стоимость работы составляет nT, загрузка:

$$p=\frac{N\tau}{nT}. \quad (4)$$

Реальная производительность остается:

$$r=\frac{N}{T},\ \pi=\frac{n}{\tau}\Rightarrow r=p\cdot\pi. \quad (5)$$

Таким образом, формула $r=p\cdot\pi$ справедлива как для простых, так и для конвейерных устройств.

Из этого следует важный практический вывод: для достижения наибольшей производительности устройства необходимо обеспечить его максимальную загрузку. При этом загрузка служит вспомогательной характеристикой, позволяющей определить, насколько эффективно устройство выполняет полезную работу и где возможно улучшение.

Загрузка всей системы в этом случае определяется как взвешенная сумма загрузок отдельных компонентов:

$$\mathrm{p}=\sum_{\mathrm{i}=1}^{\mathrm{s}}\alpha_{\mathrm{i}}p_{\mathrm{i}},\ \alpha_{\mathrm{i}}=\frac{\pi_{\mathrm{i}}}{\sum_{\mathrm{j}=1}^{\mathrm{s}}\pi_{\mathrm{j}}}, \quad (6)$$

где α_i – вес, соответствующий доле пиковой производительности i-го устройства в общей пиковой производительности системы. Эти веса удовлетворяют условиям нормировки:

$$\sum_{\mathrm{i}=1}^{\mathrm{s}}\alpha_{\mathrm{i}}=1,\ \alpha_{\mathrm{i}}\geq 0. \quad (7)$$

Таким образом, загрузка системы соответствует средневзвешенной загрузке ее компонентов, с учетом их относительной вычислительной мощности. Это определение согласуется с интуитивным представлением: в случае единственного устройства (s=1) системная загрузка совпадает с загрузкой самого устройства.

Реальная производительность системы в рамках данной модели выражается через произведение ее общей загрузки и суммарной пиковой производительности:

$$\mathrm{r}=p\cdot\pi_{\text{система}},\ \pi_{\text{система}}=\sum_{\mathrm{i}=1}^{\mathrm{s}}\pi_{\mathrm{i}}. \quad (8)$$

В случае однородной архитектуры (все устройства имеют одинаковую пиковую производительность) загрузка системы определяется как среднее арифметическое загрузок всех устройств, а реальная производительность равна сумме реальных производительностей каждого компонента.

Кроме того, в модели учитывается понятие ускорения, которое может быть определено как сумма загрузок всех устройств или, при использовании простых устройств, как отношение времени выполнения задачи на одном устройстве ко времени выполнения задачи на системе из s устройств:

$$A=\frac{\mathrm{T}_{1\text{ устройство}}}{\mathrm{T}_{\text{система}}}. \quad (9)$$

Подобный подход позволяет на-глядно оценить, как степень загрузки и однородность устройств влияют на эффективность распределения вычислительной нагрузки и общее ускорение системы.

Несмотря на строгость и логическую завершенность, данная модель обладает рядом существенных ограничений, которые делают ее недостаточной для анализа современных вычислительных систем. В частности, она предполагает независимость устройств и игнорирует сложные взаимодействия между ними, возникающие в условиях конвейерной, параллельной или распределенной обработки данных.

Быстрые и медленные устройства рассматриваются одинаково, что может привести к неправильным решениям при оптимизации. Например, медленное устройство может быть полностью загружено, но сдерживать производительность всей системы, тогда как быстрое устройство простаивает из-за нехватки задач.

Для более точной оценки необходимо учитывать влияние каждого устройства на общую производительность, а также применять дополнительные метрики:

- Время отклика – сколько времени уходит на выполнение одной задачи;
- Пропускная способность – число операций в единицу времени;
- Влияние на другие компоненты – насколько загрузка одного устройства ограничивает другие.

Таким образом, базовое определение загрузки как отношения выполненной работы к максимально возможной работе полезный ориентир, но недостаточный. Для эффективной оптимизации требуется расширенный анализ, включающий характеристики взаимодействий между устройствами и их вклад в общую производительность. Особенно важно понимать, что для всей системы равенство $r=p\cdot\pi$ может не соблюдаться, и это требует более глубокой интерпретации показателей.

Если суммировать существенные ограничения в современных условиях в оценке производительности ВС с помощью классической модели можно выявить такие недостатки - изолированный анализ компонентов без учета их взаимодействия в сложных архитектурах, не учитывает ключевые аспекты современных систем: параллельную обработку (SIMD, SMT) [15], иерархию памяти и связанные с ней задержки, энергоэффективность вычислений, особенности гетерогенных архитектур (CPU/GPU/TPU).

Особенно проблематично применение модели для облачных и распределенных систем, где критически важны сетевые задержки и синхронизация. Модель также не отражает временные характеристики (latency) и не учитывает динамические изменения производительности из-за thermal throttling или DVFS. Эти ограничения делают традиционный подход недостаточным для комплексной оценки современных вычислительных систем, требуя дополнения более совершенными метриками и методами анализа.

Современные расширения модели производительности учитывают широкий спектр факторов [12,13,14], отражающих усложнение вычислительных архитектур и появление новых техноло-

гических подходов. В отличие от традиционной модели, современные методы анализа опираются на учет архитектурных особенностей, глубокой параллельности и более точных метрик, способных охарактеризовать поведение системы в реальных условиях.

Современные процессоры обладают развитой иерархией памяти, включающей несколько уровней кэша (L1, L2, L3) и оперативную память (DRAM), с различной задержкой и пропускной способностью. Это требует учета времени доступа к данным и частоты промахов кэша. Кроме того, широкое распространение векторных инструкций, таких как AVX в архитектуре x86 или NEON в ARM, позволяет выполнять несколько операций над данными за один такт, что существенно увеличивает пиковую производительность. При этом важно учитывать архитектурные ограничения, связанные с шириной векторных регистров и типами операций.

Также все чаще используются гетерогенные вычислительные системы, объединяющие центральные процессоры, графические ускорители и специализированные блоки, такие как TPU. Их совместная работа требует учета различий в архитектуре, производительности, пропускной способности межсоединений и времени передачи данных между компонентами. Дополнительным фактором является влияние технологий энергосбережения, таких как динамическое изменение напряжения и частоты (DVFS), которые могут снижать производительность ради оптимизации энергопотребления, особенно в мобильных и облачных средах.

Параллелизм в современных системах представлен на разных уровнях. Многопоточность, hyper-threading, позволяет одному физическому ядру исполнять несколько логических потоков, повышая эффективность использования ресурсов. Многоядерные архитектуры, особенно с неоднородным доступом к памяти (NUMA), требуют учета расположения данных и привязки потоков. Кроме того, важную роль играют задержки, возникающие при работе конвейеров, например, из-за так называемых «пузырей» (pipeline bubbles), и механизмы внеочередного выполнения команд (out-of-order execution), повышающие среднюю производительность, но затрудняющие точное прогнозирование поведения системы.

Поэтому для эффективной оценки производительности применяются усовершенствованные метрики[1] [15], временные характеристики, латентность выполнения операций на различных квантилях (P50, P90, P99), которые позволяют выявить системные задержки, пропускная способность в операциях в секунду, а также общее время отклика системы дают представление об общей эффективности обработки задач.

Ключевые ресурсные показатели включают среднее число инструкций на

[1] URL:https://factorycode.wordpress.com/2024/04/13/latency-metrics. Retrieved: April, 2024.

такт (IPC), соотношение попаданий и промахов кэша, использование пропускной способности памяти и энергозатраты на выполнение операций. Эти метрики позволяют оценить, насколько эффективно используется оборудование в конкретных условиях.

Системные параметры, такие как закон Амдала и закон Густавсона, дают представление о теоретических и практических пределах масштабируемости системы при параллельной обработке. Универсальный закон масштабируемости учитывает влияние различных факторов параллельных, последовательных и конкурентных на общую эффективность увеличения числа ресурсов[1].

Таким образом, современные расширения модели производительности представляют собой комплексный подход, объединяющий архитектурный анализ, параметры параллелизма и точные количественные метрики, что позволяет адекватно оценивать вычислительные системы с учетом их реального поведения и потенциальных ограничений, которая не ограничивается лишь теоретическим анализом.

Для повышения эффективности вычислительных систем важно использовать практические методы, позволяющие выявлять ограничения, балансировать нагрузку и оперативно реагировать на изменения в поведении программ и инфраструктуры.

Одним из ключевых направлений является применение инструментов анализа - профилировщиков, таких как perf, Intel VTune и NVIDIA Nsight [16], позволяют детально исследовать поведение приложений на уровне инструкций, кэша и пропускной способности, помогают определить, какие участки кода потребляют наибольшее количество ресурсов или вызывают задержки. В дополнение к этому, системы трассировки, eBPF и LTTng, предоставляют подробную информацию о событиях в ядре и пользовательском пространстве, позволяя отслеживать системные вызовы, блокировки и другие аспекты взаимодействия компонентов. Метрики в реальном времени, собираемые с помощью систем мониторинга вроде Prometheus, дают возможность наблюдать за изменениями производительности при реальной нагрузке и выявлять отклонения от нормы.

Для эффективного использования вычислительных ресурсов важно грамотно организовать распределение задач. Современные подходы к балансировке включают алгоритмы work-stealing, при которых потоки или процессы, завершившие свои задачи, могут "перехватывать" работу у более загруженных. Это позволяет избежать неравномерной загрузки и простаивания ресурсов. Динамическое распределение нагрузки адаптирует выполнение задач в зависимости от текущего состояния системы, а методы предиктивного автомасштабирования используют прогно-

[1] URL: https://www3.nd.edu/~zxu2/acms60212-40212/ Lec-06.pdf. Retrieved: May, 2022

зирование на основе исторических данных для управления количеством вычислительных узлов или ресурсов в облачных средах.

Инструменты визуализации, такие как flame graphs [17, 18], позволяют наглядно представить, где тратится процессорное время и какие вызовы занимают ключевое место в вычислениях.

Тестирование реконфигурируемой вычислительной системы (РВС) (то есть системы, способной динамически менять свою архитектуру или конфигурацию, как, FPGA или гибридные системы CPU+FPGA) нужно проводить в автоматическом режиме. Это может значительно упростить ее разработку и отладку, чем в случае, если бы тестирование производилось вручную [19].

В табл. 2 показан сравнительный анализ традиционного метода с современным (учет архитектурных особенностей, параллелизм и конвейеризация, расширенные метрики производительности, динамическое профилирование и мониторинг, моделирование и симуляция, облачные и распределенные системы и др.).

Таблица 1. Сравнительный анализ моделей оценки производительности вычислительных систем

Table 1. Comparative analysis of models for evaluating the performance of computing systems

Подход / Approach	Погрешность для CPU-системы / Error for the CPU system	Погрешность для CPU+GPU / Error for the CPU+GPU	Определение ограничивающих факторов системы / Determining the limiting factors of the system	Учет латентности / Accounting for latency
Классическая модель	~8%	~30%	Нет	Нет
Современная модель	<3%	<7%	Да	Да

Результаты и их обсуждение

Для оценки точности классической и современной моделей, а также оценки производительности было проведено моделирование гетерогенной вычислительной системы, ориентированной на выполнение ресурсоемких задач, таких как машинное обучение, численные методы и параллельная обработка данных. Цель моделирования заключалась в сравнении прогнозируемой производительности на основе традиционного подхода с результатами, полученными при использовании современных методов анализа, учитывающих архитектурные, ресурсные и временные характеристики системы.

В табл. 2 приведена структура и основные характеристики всех узлов моделируемой вы-числительной системы, на которых были основаны расчеты и наблюдения.

Таблица 2. Состав и характеристики узлов моделируемой вычислительной системы

Table 2. Composition and characteristics of nodes of the simulated computer system

№	Компонент / Component	Тип / Модель / Type / Model	Назначение / Purpose	Характеристики / Characteristics
1	CPU1	Intel Xeon Gold 6338	Центральный процессор	32 потока, 2.0 ГГц, AVX-512, 40 МБ L3, TDP 205 Вт
2	CPU2	Intel Xeon Gold 6338	Центральный процессор	32 потока, 2.0 ГГц, AVX-512, 40 МБ L3, TDP 205 Вт
3	GPU	NVIDIA A100 40GB	Графический ускоритель	6912 CUDA-ядер, HBM2e 40 ГБ, 1.6 ТФлопс FP64, NVLink, PCIe 4.0
4	RAM	DDR4 ECC Registered	Основная оперативная память	256 ГБ, 3200 МГц, двухканальная, NUMA-разметка
5	Кэш-память CPU	L1 / L2 / L3	Многоуровневая память CPU	L1: 32КБ, L2: 1МБ на ядро, L3: 40 МБ общая
6	Память GPU	HBM2e	Локальная память GPU	40 ГБ, 1555 ГБ/с, широкая шина

Расчет по традиционной модели основывается на предположении, что производительность системы определяется как сумма реальных производительностей всех ее устройств. Реальная производительность отдельного устройства r_i определяется как произведение его загрузки p_i на пиковую производительность π_i(1), тогда общая реальная производительность системы $r_{\text{система}}$ рассчитывается как сумма по всем устройствам (2).

В табл. 3 показан расчет для гетерогенной вычислительной системы, использование только трех узлов из табл. 2 (CPU1, CPU2, GPU) в расчетах по традиционной модели – это умышленное упрощение, соответствующее классическому подходу, в котором в фокусе только вычислительные узлы, непосредственно выполняющие полезную работу (в операциях в секунду).

Получаем конечную производительность:

$r_{\text{система}}$=100·0.80+100·0.75+500·0.40=

=80+75+200=355 оп./с

В рамках традиционной модели, общая реальная производительность системы оценивается как 355 условных операций в секунду. Как уже указывалось, данная модель не учитывает влияние архитектурных задержек, асинхронных вычислений, особенностей передачи данных и других факторов, присущих гетерогенным вычислительным платформам. Такие ограничения станут очевидны при сравнении с расчетами по современной модели.

Таблица 3. Традиционный оценки производительности вычислительной системы

Table 3: Traditional performance evaluation of a computing system

Устройство / Device	Пиковая производительность π_i, оп/с / Peak performance π_i, op/c	Загрузка p_i / Load p_i	$r_i = p_i \cdot \pi_i$, оп/с
CPU1	100	0.80	$100 \cdot 0.80 = 80$
CPU2	100	0.75	$100 \cdot 0.75 = 75$
GPU	500	0.40	$500 \cdot 0.40 = 200$

Сделаем более точный анализ и прогнозирование производительности для этого необходимо применять усовершенствованные модели, учитывающие ключевые архитектурные характеристики и реальные режимы работы компонентов вычислительной системы:

1. Расчет производительности CPU.

Рассмотрим процессор Intel Xeon Gold 6338 с 32 физическими ядрами и поддержкой AVX-512. AVX-512 позволяет выполнять 16 операций с плавающей точкой двойной точности (FP64) за такт на ядро.

Тактовая частота процессора: 2.0 ГГц.

Пиковое количество операций FP64 в секунду (теоретически):

$\pi_{CPU}=32\times2\times10^9\times16=$

$=1.024\times10^{12}$операций/с=1.024ТФлопс.

В реальных условиях вычислительные блоки не всегда загружены полностью, часть времени уходит на ожидание данных из памяти, переключение потоков и другие накладные расходы. Принимается коэффициент загрузки $p_{CPU} = 0.85$, отражающий уровень оптимизации кода и эффективность использования векторных инструкций.

Тогда эффективная производительность CPU:

$r_{CPU}=\pi_{CPU}\times p_{CPU}=1.024\times0.85=$

$=0.870$ ТФлопс.

2. Расчет производительности GPU.

Для вычислительных нагрузок, требующих массовой параллельности, используется графический ускоритель NVIDIA A100.

Теоретическая пиковая производительность для операций FP64 (двойной точности) составляет около 1.6 ТФлопс.

Однако реальная производительность снижается из-за ограничений пропускной способности памяти, задержек синхронизации и PCIe, а также неидеальной загрузки CUDA-ядер. Принимается коэффициент загрузки $p_{GPU} = 0.70$.

Таким образом, эффективная производительность GPU:

$r_{GPU}=1.6\times0.70=1.12$ ТФлопс.

3. Учет межсоединений (PCIe 4.0).

Межпроцессорное соединение PCIe 4.0 играет ключевую роль в передаче данных между CPU и GPU.

Дуплексная пропускная способность PCIe 4.0 – около 32 ГБ/с.

Для операций FP64, где каждый элемент занимает 8 байт, это соответствует:

$$\pi_{PCIe}=\frac{32\times10^{9}\text{байт/с}}{8\text{байт/операция}}=$$

$$=4\times10^{9}\text{операций/с}=4\text{Гоп/с}.$$

На практике загрузка составляет порядка $p_{PCIe}=0.60$ из-за накладных расходов протокола, латентности и др.

Итоговая пропускная способность шины:

$r_{PCIe}=4\times0.60=2.4$Гоп/с.

4. Итоговая производительность системы.

Рассмотрим совместную работу CPU и GPU, ограниченную пропускной способностью шины PCIe.

Суммарная вычислительная мощность CPU и GPU:

$r_{CPU}+r_{GPU}=0.870+1.12=1.99$ТФлопс.

Поскольку PCIe пропускная способность в операциях в секунду значительно ниже, она становится ограничением для общего потока данных и, следовательно, ограничивает максимальную производительность системы.

Итоговая производительность вычислительной системы определяется минимумом между суммарной вычислительной мощностью и пропускной способностью межсоединения:

$r_{система}=\min(1.99, 2.4)=1.99$ ТФлопс.

Таблица 4. Сравнение с традиционной моделью

Table 4. Comparison with the traditional model

Метрика / Metrica	Традиционная модель / Traditional model	Современная модель / Modern model
Производительность CPU	80 оп/с	0.870 ТФлопс
Производительность GPU	200 оп/с	1.12 ТФлопс
Учет межсоединений	Отсутствует	2.4 Гоп/с
Итоговая производительность	355 оп/с (упрощенно)	1.99 ТФлопс

Традиционные оценки оперируют единицами операций без учета параллелизма и особенностей архитектуры, что ведет к завышению результатов и искажению представления о реальной вычислительной мощности.

Представленная современная модель существенно расширяет традиционный подход, вводя параметры, учитывающие специфику современных процессоров и систем:

• Векторизация позволяет существенно повысить плотность вычислений за так, параллелизм на уровне потоков и GPU обеспечивают масштабируемость, ог-раничения памяти и межсоединений, становятся ограничивающим фактором и влияют на производительность.

• Энергоэффективность и тепловые параметры косвенно влияют на стабильность частот и длительность работы под нагрузкой.

Фактическая производительность систем часто существенно ниже пиковых теоретических значений из-за множества факторов: задержек, неоптимального кода, архитектурных ограничений.

Для повышения производительности вычислительных систем необходимо:

• Повышать коэффициент загрузки GPU, приближая его к $p_{GPU} \rightarrow 0.9$ за счет оптимизации алгоритмов и кода.

• Использовать более современные и быстрые межсоединения (PCIe 5.0, NVLink) для устранения пропускных ограничений.

• Внедрять NUMA-aware и memory-bound оптимизации для эффективного распределения данных и вычислений по ресурсам.

Таким образом, расширенная модель обеспечивает более реалистичный и адекватный анализ вычислительной системы, выявляя критические места и позволяя целенаправленно улучшать архитектуру и программное обеспечение.

Так же при исследовании произведен расчет эффективного использования сопроцессоров Intel Xeon Phi 7120P в различных конфигурациях, включая гетерогенные системы с процессорами Intel Xeon E5-2683 v4. Исходные параметры сопроцессора Intel Xeon Phi 7120P включают 61 ядро с поддержкой 4 потоков на ядро, что в сумме дает 244 потока. Согласно стандарту, 4 потока резервируются системой, поэтому для вычислений используется 240 потоков. Коэффициент размера очереди задан по умолчанию равным 40, что позволяет вычислить максимальный размер обрабатываемого блока данных как произведение данного коэффициента на количество потоков.

Пиковая производительность Intel Xeon Phi 7120P по операциям с плавающей точкой двойной точности (FP64) составляет порядка 1.2 терафлопс. Принимая во внимание коэффициент загрузки 0.75, достигается эффективная производительность около 0.9 терафлопс. Для процессоров Intel Xeon E5-2683 v4, имеющих 16 ядер и 32 потока с базовой частотой 2.1 ГГц, пиковая производительность составляет примерно 512 гигафлопс на один процессор. Для двух таких процессоров суммарная пиковая мощность равна 1.024 терафлопс, а с учетом коэффициента загрузки 0.85 эффективная производительность достигает примерно 0.87 терафлопс.

Рассмотрены пять конфигураций: один сопроцессор Xeon Phi 7120P; один Xeon Phi 7120P в сочетании с двумя процессорами Xeon E5-2683 v4; два сопроцессора Xeon Phi 7120P; два Xeon Phi 7120P с двумя Xeon E5-2683 v4 и, наконец, две Xeon E5-2683 v4 без сопроцессоров. Для каждой из конфигураций рассчитано количество вычислительных потоков, максимальный размер блока данных, пиковая и эффективная производительность. Результаты расчетов представлены в табл. 5.

Таблица 5. Расчет эффективного использования процессоров

Table 5. Calculation of effective processors

Конфигурация / Configuration	Потоки (N) / Flows (N)	Максимальный размер блока $S=m\times N$ / Maximum block size $S=m\times N$	Пиковая производительность (ТФлопс) / Peak Performance (TFlops	Эффективная производительность (ТФлопс) / Effective Performance (TFlops)
1x Xeon Phi 7120P	240	9600	1.2	0.9
1x Xeon Phi 7120P + 2x Xeon E5-2683 v4	304	12160	2.224	1.77
2x Xeon Phi 7120P	480	19200	2.4	1.8
2x Xeon Phi 7120P + 2x Xeon E5-2683 v4	544	21760	3.424	2.67
2x Xeon E5-2683 v4	64	2560	1.024	0.87

Полученные данные демонстрируют, что увеличение количества вычислительных потоков (рис. 1) и использование нескольких сопроцессоров и процессоров позволяет существенно повысить максимальный размер обрабатываемого блока и общую вычислительную производительность системы.

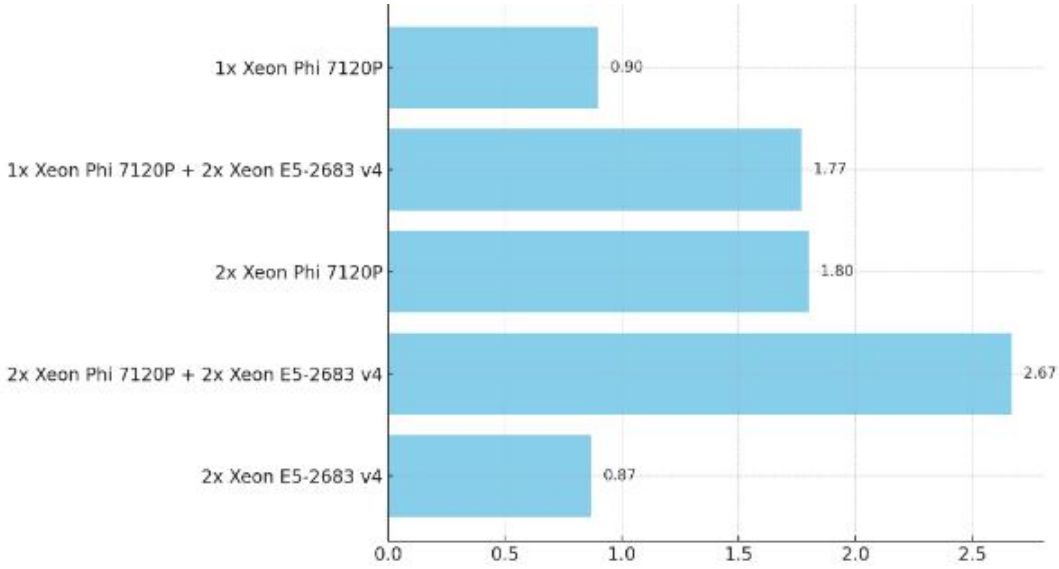

Рис. 1. Вычислительная производительность (ТФлопс)

Fig. 1. Computing Performance (TFlops)

В то же время коэффициенты загрузки, обусловленные архитектурными и системными особенностями, снижают теоретический пиковый уровень, подчеркивая важность учета реальных условий эксплуатации. Использование гетерогенных конфигураций с объединением мощностей Xeon Phi и Xeon CPU обеспечивает заметный прирост по сравнению с однородными системами.

Так же проведен анализ влияния коэффициента размера очереди выгрузки m на производительность гетерогенных вычислительных систем с использованием сопроцессоров Intel Xeon Phi 7120P и процессоров Intel Xeon E5-2683 v4. Коэффициент m определяет множитель, используемый для вычисления максимального размера обрабатываемого блока данных, или очереди заданий, посредством формулы $S=m\times N$, где N – количество вычислительных потоков, задействованных на сопроцессоре. Значение m напрямую влияет на эффективность передачи данных и распределение нагрузки между процессорными компонентами системы.

Экспериментальные данные показывают, что при малых значениях m наблюдается значительное увеличение общего времени расчета во всех исследованных конфигурациях. Это объясняется тем, что небольшие размеры очереди заданий приводят к частым операциям передачи данных между основным процессором и сопроцессорами, что увеличивает накладные расходы и снижает эффективность вычислительного процесса. По мере увеличения коэффициента m общее время расчета заметно сокращается, достигая минимальных значений в диапазоне примерно от 20 до 40. В этом интервале достигается оптимальный баланс между размером обрабатываемого блока и системными затратами на передачу данных, обеспечивающий наиболее полную загрузку вычислительных ресурсов и эффективную работу всей системы. При дальнейшем увеличении m за пределами оптимального диапазона наблюдается либо стабилизация, либо незначительный рост времени расчета. Это связано с тем, что чрезмерно большие очереди заданий увеличивают задержки и усложняют балансировку нагрузки, что в итоге снижает масштабируемость и общую производительность вычислительной системы. Для конфигураций, включающих гетерогенные сочетания Xeon Phi и Xeon, минимальное время работы достигается при значениях коэффициента m, близких к 25–35, что указывает на необходимость тщательного выбора параметров работы для конкретных аппаратных комплексов и задач.

Таким образом, коэффициент m является ключевым параметром настройки гетерогенных вычислительных систем, существенно влияющим на время выполнения и эффективность использования аппаратных ресурсов. Его правильный подбор позволяет оптимизировать распределение нагрузки, снизить издержки передачи данных и достичь высокой производительности при решении сложных вычислительных задач.

Выводы

Проведенное исследование позволило провести анализ классической и современной моделей оценки производительности гетерогенных вычислительных систем, а также определить влияние ключевых параметров на эффективность работы вычислительных систем с реальными аппаратными конфигурациями.

Классический подход, основанный на суммировании производительности отдельных узлов без учета архитектурных особенностей и системных ограничений, показал свою ограниченность, выдавая завышенные оценки (примерно 355 условных операций в секунду). Игнорирование факторов, таких как задержки передачи данных, пропускная способность межсоединений, особенности иерархии памяти и параллелизма, приводит к значительным расхождениям с реальной производительностью современных вычислительных систем.

Современная модель, учитывающая возможности векторизации (AVX-512), многоуровневой памяти (кэш CPU, HBM GPU), а также пропускную способность шины PCIe 4.0, позволила получить более точную и обоснованную оценку – около 1.99 ТФлопс суммарной вычислительной мощности системы. При этом выявлено, что пропускная способность PCIe является ограничивающим фактором, совместной работы CPU (0.87 ТФлопс) и GPU (1.12 ТФлопс). Анализ конфигураций с использованием сопроцессоров Intel Xeon Phi 7120P в сочетании с процессорами Intel Xeon E5-2683 v4 показал, что гетерогенные системы значительно превосходят однородные по вычислительной мощности, достигая эффективной производитель-ности до 2.67 ТФлопс. Важным фактором оптимизации является коэффициент размера очереди выгрузки m, который влияет на максимальный размер обрабатываемого блока данных и эффективность передачи заданий между CPU и сопроцессорами. Оптимальные значения m в диапазоне 25–35 позволяют минимизировать накладные расходы и повысить эффективность использования вычислительных ресурсов. Эти наблюдения согласуются с результатами исследований, где параметр m рассматривался как ключевой элемент балансировки нагрузки и управления памятью.

Практические рекомендации, сделанные при исследовании, включают необходимость применения современных межсоединений (PCIe 5.0, NVLink), способных снизить задержки передачи данных и увеличить пропускную способность. Оптимизация загрузки GPU, повышение коэффициента использования до 0.9 и более, достигается за счет совершенствования алгоритмов, эффективного распределения задач и балансировки потоков. Не менее важен учет NUMA-архитектуры и иерархии памяти, что позволяет минимизировать задержки доступа и повысить пропускную способность памяти.

Данные выводы согласуются с современными трендами в области высокопроизводительных вычислений, отраженными в литературе, где подчеркивается интеграция аппаратных и про-

граммных оптимизаций для достижения максимальной эффективности.

В целом, использование расширенных моделей, учитывающих архитектурные и системные особенности, позволяет более точно оценивать и прогнозировать производительность, выявлять ограничения в сложных гетерогенных вычислительных системах. Это особенно важно при решении ресурсоемких задач машинного обучения, численного моделирования и обработки больших баз данных.

Дальнейшие исследования могут быть направлены на автоматизацию подбора оптимальных параметров конфигурации, включая коэффициент размера очереди m, с применением методов машинного обучения и адаптивных алгоритмов, что позволит повысить адаптивность и эффективность вычислительных систем в динамических условиях эксплуатации.

Список литературы

1. Леонтьева О.Ю., Климанова Е.Ю., Зеленко Б.В. Оценка производительности вычислительных систем // Вестник технологического университета. 2015. Т.18, № 24. С. 102-105.

2. Сорокин А.П., Бенсон М.З., Методики оценки производительности гетерогенных вычислительных систем // Russian Technological Journal. 2017. №5(6). С. 11-19.

3. Bahnam B.S., Dawwod S.A. Younis M.C. Optimizing software reliability growth models through simulated annealing algorithm: parameters estimation and performance analysis // The Journal of Supercomputing. April 2024. DOI: 10.1007/ s11227-024-06046-4

4. Вепаев Ш.В. Исследование Марковских моделей обслуживания // Молодой ученый. 2022. № 49 (444). С. 26–28.

5. Гачаев А.М., Датаев А.А., Вазкаева С.С.-А. Исследование надежности программного обеспечения компьютерных информационных технологий // Прикладные экономические исследования. 2023. №2. С. 80-84. https://doi.org/10.47576/2949-1908_2023_2_80.

6. Вадейко В.С., Манько А.В. Марковская модель надежности. Минск: БНТУ, 2022. С. 222–22.

7. Терсков В.А., Сакаш И.Ю. Математическая модель оценки надежности функционирования многопроцессорных вычислительных комплексов // Computational Nanotechnology. 2024. Т. 11, № 2. С. 22–28. DOI: 10.33693/2313-223X-2024-11-2-22-28. EDN: MHZWBU

8. Михалок В.В. Технические требования к программно-аппаратному комплексу (ПАК) исполнителя. URL: https://intellectexport.ru/site/assets/ files/1035/prilozhenie_2.doc

9. Оценка производительности вычислительных систем / Е.Ю. Климанова, А.Р. Субханкулова, Б.В. Зеленко, О.Ю. Леонтьева // Вестник технологического университета. 2015. Т.18, № 24. С. 102-105

10. Гоголевский А.С., Романов А.В., Трепкова С.А. Методика оценки производительности аппаратно-программного комплекса информационно-управляющей системы // Известия ТулГУ. Технические науки. 2022. Т 10. С. 87-91.

11. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети. Л.: Энергоатомиздат. Ленингр. отд-ние, 1987. 288 с.

12. Сравнительный анализ методов оценки производительности узлов в распределенных системах / Мин Тху Кхаинг, С.А. Лупин, Ай Мин Тайк, Д.А. Федяшин // Международный журнал открытых информационных технологий. 2023. Т 11, №6.

13. Lorenzo Luciano , Imre Kiss, Peter William Beardshear, Esther Kadosh, A. Ben Hamza WISE: a computer system performance index scoring framework // Journal of Cloud Computing: Advances, Systems and Applications. 2021. 10:8.

14. Альбертьян А.М., Курочкин И.И., Ватутин Э.И. Использование гетерогенных вычислительных узлов в грид-системах при решении комбинарных задач // Известия ЮФУ. 2022. С. 142-153.

15. Jim Holtmana, Neil J. Gunther Getting in the Zone for Successful Scalability // Performance Dynamcis Company, Castro Valley, California, USA, 2018.

16. Xin Li Scalability: strong and weak scaling // Royal Institute of Technology. 2018.

17. Rupak Roy, JaeHyuk Kwack Intel Analyzers // Argonne Leadership Computing Facility. 2025.

18. Brendan Gregg Visualizing Performance: The Developer's Guide to Flame Graphs // Communications of the ACM. 2022.

19. Мартышкин А. И., Кирюткин И. А., Мереняшева Е. А. Автотестирование встраиваемой реконфигурируемой вычислительной системы // Известия Юго-Западного государственного университета. 2023; 27(1): 140-152. https://doi.org/10.21869/2223-1560-2023-27-1-140-152.

References

1. Leontieva O.Yu., Klimanova E.Yu., Zelenko B.V. Performance evaluation of computing systems. *Vestnik tekhnologicheskogo universiteta = Bulletin of the Technological University*. 2015; 18(24): 102-105. (In Russ.).

2. Sorokin A.P., Benenson M.Z. Methodologies for performance evaluation of heterogeneous computing systems. *Russian Technological Journal*. 2017; (5): 11-19. (In Russ.).

3. Bahnam B.S., Dawwod S.A., Younis M.C. Optimizing software reliability growth models through simulated annealing algorithm: parameter estimation and performance analysis. *The Journal of Supercomputing*. April 2024. DOI: 10.1007/s11227-024-06046-4

4. Vepaev Sh.V. Study of Markov service models. *Molodoi uchenyi = Young Scientist*. 2022; (49): 26–28. (In Russ.).

5. Gachaev A.M., Dataev A.A., Vazkaeva S.S.-A. Study of software reliability in computer information technologies. *Prikladnye ekonomicheskie issledovaniya = Applied economic research.* 2023; (2):80-84. (In Russ.). https://doi.org/10.47576/2949-1908_2023_2_80.

6. Vadeyko V.S., Manko A.V. Markov reliability model. Minsk; 2022. P. 222–22. (In Russ.).

7. Terskov V.A., Sakash I.Yu. Mathematical model for reliability evaluation of multiprocessor computing complexes. *Computational Nanotechnology*. 2024; 11(2): 22–28. (In Russ.). https://doi.org/10.33693/2313-223X-2024-11-2-22-28. EDN: MHZWBU

8. Mikhalok V.V. Technical requirements for the software-hardware complex (SHC) of the executor. (In Russ.). Available at: https://intellectexport.ru/site/assets/files/1035/ prilozhenie_2.doc

9. Klimanova E.Yu., Subkhankulova A.R., Zelenko B.V., Leontieva O.Yu. Performance evaluation of computing systems. *Vestnik tekhnologicheskogo universiteta = Bulletin of the Technological University*. 2015; 18(24):102-105. (In Russ.).

10. Gogolevsky A.S., Romanov A.V., Trepkova S.A. Methodology for performance evaluation of hardware-software complex of information control system. *Izvestiya TulGU. Tekhnicheskie nauki = Izvestiya Tula State University. Technical Sciences*. 2022; 10: 87-91. (In Russ.).

11. Larionov A.M., Mayorov S.A., Novikov G.I. Computing complexes, systems and networks. Leningrad: Energoatomizdat; 1987. 288 p. (In Russ.).

12. Min Thu Khaing, Lupin S.A., Ai Min Taik, Fedyashin D.A. Comparative analysis of node performance evaluation methods in distributed systems. *Mezhdunarodnyi zhurnal otkrytykh informatsionnykh tekhnologii = International Journal of Open Information Technologies*. 2023; 11(6). (In Russ.).

13. Lorenzo Luciano, Imre Kiss, Peter William Beardshear, Esther Kadosh, A. Ben Hamza WISE: a computer system performance index scoring framework. *Journal of Cloud Computing: Advances, Systems and Applications*. 2021; 10:8.

14. Albertyan A.M., Kurochkin I.I., Vatutin E.I. Use of heterogeneous computing nodes in grid systems for solving combinatorial problems. *Izvestiya YuFU = Izvestiya of Southern Federal University*. 2022: 142-153. (In Russ.).

15. Jim Holtman, Neil J. Gunther Getting in the Zone for Successful Scalability. Performance Dynamics Company, Castro Valley, California, USA, 2018.

16. Xin Li Scalability: strong and weak scaling. *Royal Institute of Technology*, 2018.

17. Rupak Roy, JaeHyuk Kwack Intel Analyzers. *Argonne Leadership Computing Facility*. 2025.

18. Brendan Gregg Visualizing Performance: The Developer's Guide to Flame Graphs. Communications of the ACM. 2022.

19. Martyshkin A. I., Kiryutkin I. A., Merenyasheva E. A. Autotesting an Embedded Reconfigurable Computing System. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2023; 27(1): 140-152 (In Russ.). https://doi.org/10.21869/2223-1560-2023-27-1-140-152.

Информация об авторе / Information about the Author

Петушков Григорий Валерьевич,
младший научный сотрудник Центра популяризации науки и высшего образования, Институт молодежной политики и международных отношений РТУ МИРЭА, г. Москва, Российская Федерация,
e-mail: petushkov@mirea.ru

Grigory V. Petushkov, Junior Researcher, Centre for Popularisation of Science and Higher Education, Institute of Youth Policy and International Relations, MIREA – Russian Technological University, Moscow, Russian Federation,
e-mail: petushkov@mirea.ru