Інформаційні технології

УДК 004.9 **doi: 10.26906/SUNZ.2026.1.035**

Ю. О. Андрусенко, В. О. Радченко

Харківський національний університет радіоелектроніки, Харків, Україна

ОПТИМІЗАЦІЯ ВИКОРИСТАННЯ РЕСУРСІВ У ГЕТЕРОГЕННИХ ГРІД -СИСТЕМАХ З НЕСТАЦІОНАРНИМ ВХІДНИМ ПОТОКОМ

Анотація. Актуальність дослідження зумовлена зростанням масштабів використання гетерогенних грід-систем для виконання обчислювально складних задач в умовах динамічного та нестаціонарного вхідного потоку. Різнорідність обчислювальних ресурсів і нерівномірність навантаження призводять до зниження ефективності використання обчислювальної інфраструктури та потребують застосування адаптивних методів балансування навантаження. **Метою роботи** є підвищення ефективності використання ресурсів гетерогенних грід-систем шляхом розробки та дослідження алгоритмів балансування навантаження на основі імітаційного моделювання в умовах нестаціонарного вхідного потоку задач. **Об'єктом дослідження** є процеси управління та розподілу обчислювальних ресурсів у гетерогенних грід-системах. **Предметом дослідження** є методи та алгоритми балансування навантаження й оптимізації використання ресурсів гетерогенних грід-систем з нестаціонарним вхідним потоком задач на основі імітаційного моделювання. **Результати.** У роботі розроблено імітаційну модель гетерогенної грід-системи, що враховує різнорідність обчислювальних вузлів та змінну інтенсивність надходження задач. Проведено експериментальні дослідження ефективності алгоритмів балансування навантаження у нормальному та піковому режимах роботи системи. **Висновок.** Отримані результати показали зменшення середнього часу очікування виконання задач, підвищення рівномірності завантаження ресурсів та зростання загальної пропускної здатності системи. **Напрямами подальших досліджень** є розширення імітаційної моделі з урахуванням пріоритетності задач, обмежень якості обслуговування (QoS), енергоспоживання обчислювальних вузлів, а також застосування інтелектуальних і прогнозних методів балансування навантаження у гетерогенних розподілених обчислювальних середовищах.

Ключові слова: грід-система, гетерогенність, нестаціонарність, балансування навантаження, розподіл ресурсів, обчислювальний вузол..

Вступ

Постановка проблеми. Гетерогенні грід-системи є одним із ключових інструментів реалізації високопродуктивних розподілених обчислень, що забезпечують спільне використання різнорідних обчислювальних ресурсів, об'єднаних у єдине інформаційно-обчислювальне середовище. Відмінності у продуктивності вузлів, архітектурі апаратного забезпечення, характеристиках мережевої взаємодії та режимах доступу до ресурсів зумовлюють підвищену складність задач управління такими системами [1].

В умовах нестаціонарного вхідного потоку задач, коли інтенсивність надходження запитів змінюється у часі та має виражені пікові навантаження, особливої актуальності набуває проблема балансування навантаження між обчислювальними вузлами грід-системи [2]. Неадекватний розподіл задач призводить до перевантаження окремих ресурсів, простою інших вузлів та зниження загальної ефективності функціонування системи [3].

Ефективним інструментом дослідження процесів функціонування гетерогенних грід-систем є імітаційне моделювання, яке дозволяє відтворювати динаміку вхідних потоків, поведінку обчислювальних ресурсів і алгоритмів балансування навантаження без втручання у реальну інфраструктуру. Застосування імітаційних моделей надає можливість аналізувати вплив різних стратегій планування та розподілу ресурсів, оцінювати показники продуктивності, такі як час очікування задач, коефіцієнт завантаження вузлів і пропускна здатність системи.

Використання імітаційного підходу є особливо доцільним у задачах оптимізації, пов'язаних із нестаціонарними потоками, оскільки дозволяє враховувати випадковий характер надходження задач, зміну їхніх вимог до ресурсів та адаптивну поведінку алгоритмів балансування [4]. Це створює передумови для розробки та дослідження адаптивних і динамічних методів балансування навантаження, орієнтованих на підвищення ефективності використання ресурсів гетерогенних грід-систем.

Метою роботи є розробка та дослідження імітаційної моделі гетерогенної грід-системи з нестаціонарним вхідним потоком задач, а також аналіз ефективності алгоритмів балансування навантаження за різних режимів функціонування системи.

Основний матеріал

На додаток до аналізу b n-pack ng та оцінки навантаження розглянемо досягнення в алгоритмах балансування навантаження, які можуть дати ідеї для того, щоб запропонувати адекватну техніку розподілу викликів.

VectorDot (VD) [5] - це алгоритм балансування навантаження для роботи з ресурсами в ієрархічних системах розподілених обчислень. Алгоритм натхненний методом Тойоди для багатовимірних ранців. Він дозволяє переміщати VMs, віртуальні сховища та трафік комутаторів з одного вузла на інший без простою працюючого в ньому додатки. Функція «

© Андрусенко Ю. О., Радченко В. О., 2026

точкового продукту » описує перевантажені сервери через надмірне використання ЦП, пам'яті, мережі або дискового введення-виводу. Вона використовується для пошуку вузлів серверів або комутаторів для міграції VMs або трафіку. Алгоритм перевірено в ході експериментів як у реальному центрі обробки даних, так і в змодельованій середовищі великомасштабного центру обробки даних.

Virtual Web Resource (VWR) [6] - це алгоритм балансування навантаження для розподілу клієнтських запитів між реплікованими серверами. Він використовує протокол для обмеження швидкості перенаправлення та запобігання навантаження віддалених серверів. Коли сервер перевантажений, він отримує дозвіл на зниження навантаження та перенаправляє перевищуючу навантаження на найближчий веб-сервер із меншим навантаженням. Сервери можуть прийняти або відхилити запит, якщо запит відхилений, то для його виконання вибирається інший сервер.

Стратегія планування при балансуванні навантаження на VM-ресурсах – SVB [7] – це алгоритм балансування навантаження для міграції VMs у середовищі хмарних обчислень. Алгоритм, заснований на генетичних алгоритмах GA, враховує варіації системи, історичні дані та вартість міграції. Він намагається зменшити або уникнути динамічної міграції. Хромосома представлена у вигляді деревоподібної структури. Кореневий вузол – це планування, проміжні вузли – реальні машини, а листові вузли - VMs. Пропонуються відповідні стратегії вибору, гібридизації та варіації. Цей метод здатний реалізувати балансування навантаження та розумне використання ресурсів, коли навантаження на систему стабільна та варіативна.

Honeybee Foraging Behavior (HF) [8] - це децентралізований алгоритм балансування навантаження для великомасштабних хмарних систем. Він динамічно розподіляє вебсервіси на серверах для регулювання системи залежно від попиту. Цей алгоритм використовується для самоорганізації та досягнення глобальною балансування навантаження за допомогою локальних дій сервера. Кожен фізичний сервер групує віртуальні сервери. Поняття « дошка оголошень » використовується для повідомлення про глобальну прибутку колонії. Незайняті сервери читають дошку оголошень, вибирають оголошення та обслуговують запит або випадковим чином обслуговують запит черги серверів.

Particle Swarm Optimization (TBSLB-PSO) [9] – це механізм балансування навантаження для хмарних обчислень. Цей алгоритм використовує центральний планувальник завдань CTS для передачі додаткових завдань з перевантаженою VM на нову аналогічну VM шляхом застосування інформації на дошці. Дошка містить всю інформацію планувальників про характеристики VMs, виконувані завдання та якість обслуговування QoS. Процес міграції враховує обсяг даних, пам'яті, пропускну здатність і число VMs. Непрацюючі фізичні машини ФМ не будуть обрані як нових хостів для VMs; це дозволяє знизити енергоспоживання.

Теорія мурашиних колоній та складних мереж ACCLB [10] – це механізм балансування навантаження для відкритого федерації хмарних обчислень. Мураха періодично вирушає недовантаженим вузлом для пошуку навантаження. Під час подорожі він запам'ятовує вузол, який має максимальну / мінімальну навантаження, і зупиняється у вузлі, що перевищує поріг навантаження різниця між ах і n або загальна кількість кроків переміщення. Потім мураха повідомляє двом вузлам N ах і N n щоб збалансувати навантаження між ними. Ця система спрямована на рішення складною та динамічною завдання балансування, при цьому враховуються характеристики складної мережі малий світ і свобода від масштабованості.

Event Driven (ED) [11] - це алгоритм балансування навантаження для розрахованих на багато користувачів онлайн- ігор MMOG в реальному часу на хмарних ресурсах. Він представляє собою економічно ефективний хостинг для сесій MMOG Алгоритм використовує подієво-орієнтоване рішення, яке отримує інформацію про пропускну здібності, аналізує її в контексті та приймає рішення про надання послуг. Події пропускний здібності складаються з сесії та ресурсів, статусу події, моніторингу та прогнозування.

Fuzzy Based Round Robin (FBRR) [12] - це алгоритм балансування навантаження на основі нечіткою логіки та RR у центрі обробки даних у хмарі. Він зберігає інформацію про кожну віртуальною машині та їх запити, які в даний час час розподілені. Найменш завантажена машина визначається при вступі запиту на виділення. Фаззифікатор враховує швидкість процесора та навантаження VM для балансування системи. Fuzzy Infe ence Syste FIS імітує прийняття рішень людиною на основі нечітких правил управління та відповідних вхідних лінгвістичних параметрів.

У табл. 1 наведено огляд середовищ алгоритмів балансування навантаження. У табл. 2 представлені основні характеристики алгоритмів, а у табл. 3 узагальнено метрики, використані для вивчення якості алгоритмів, тобто метрики балансування навантаження, що дозволяють оцінити стан ресурсів, якість обслуговування та ефективність розподілу запитів між вузлами системи. Вибір метрик визначається типом середовища та вимогами до продуктивності та надійності.

Наведено основні метрики, що застосовуються в алгоритмах балансування навантаження для оцінювання стану обчислювальних та мережевих ресурсів, вивчають окремий аспект функціонування системи, зокрема рівень завантаження апаратних ресурсів, якість обслуговування користувачів, енергоефективність та надійність вузлів. Метрики, пов'язані з використанням процесора, оперативної пам'яті та дискової підсистеми, дозволяють визначити обчислювальне навантаження та запобігти перевантаженню окремих вузлів. Мережні показники, такі як затримка, пропускна здатність та втрати пакетів, застосовуються для забезпечення якості сервісу та мінімізації часу відкликання системи.

Таблиця 1 – **Середовище алгоритмів балансування навантаження**

Середовище	Характеристика середовища	Типові алгоритми балансування	Переваги	Недоліки / обмеження
Локальні обчислювальні мережі (LAN)	Невелика кількість вузлів, низькі затримки, стабільні канали зв'язку	Round Robin , Weighted Round Robin	Простота реалізації, мінімальні накладні витрати	Не враховується поточне навантаження вузлів
Кластерні системи	Однорідні або слабо неоднорідні вузли, спільна мета обчислень	Least Connections , Least Load	Вища ефективність використання ресурсів	Потребує постійного моніторингу стану вузлів
Хмарні обчислення (Cloud Computing)	Динамічне масштабування, віртуалізація, змінна кількість ресурсів	Dynamic Load Balancing , Auto-scaling алгоритми	Висока адаптивність, еластичність	Складність реалізації, залежність від провайдера
Розподілені системи	Географічно рознесені вузли, асинхронна взаємодія	Hash-based , Consistent Hashing	Масштабованість, відмовостійкість	Можлива нерівномірність розподілу
Мережі мобільного зв'язку	Висока мобільність користувачів, змінна якість каналів	Load-aware , QoS-based	Підтримка якості сервісу (QoS)	Складність обчислень у реальному часі
Центри обробки даних (ЦОД)	Висока щільність серверів, великі обсяги трафіку	ECMP, Flow-based	Висока пропускна здатність	Не враховує навантаження серверів
Інтернет-сервіси	Велика кількість користувачів, нерівномірний трафік	Least Response Time	Оптимізація часу відгуку	Потребує збору статистики
IoT- системи	Обмежені ресурси вузлів, велика кількість пристроїв	Lightweight , Edge-based	Енергоефективність	Обмежена точність балансування

Таблиця 2 – **Основні характеристики алгоритмів балансування навантаження**

Характеристика	Статичні алгоритми	Динамічні алгоритми	Адаптивні (інтелектуальні) алгоритми
Принцип роботи	Розподіл за наперед заданими правилами	Розподіл з урахуванням поточного стану системи	Розподіл з урахуванням історії та прогнозу навантаження
Урахування стану ресурсів	Не враховуються	Враховуватись частично	Повне урахування
Динамічність	Низька	Середня-висока	Дуже висока
Масштабованість	обмежена	Висока	Дуже висока
Складність реалізації	Низька	Середня	Висока
Накладні витрати	Мінімальні	Помірні	Значні
Затримка прийняття рішень	Мінімальна	Помірна	Вища через аналіз даних
Відмовостійкість	Низька	Середня	Висока
Справедливість розподілу	обмежена	Висока	Оптимізована
Підтримка QoS	Відсутня	Часткова	Повна
Адаптація до змін середовища	Відсутня	обмежена	Повна
Енергетична ефективність	Низька	Середня	Висока
Типові алгоритми	Round Robin, Hashing	Least Connections	Reinforcement Learning
Сфера застосування	Малі системи, LAN	ЦОД, веб-сервіси	Хмарні системи, 5G, IoT

Таблиця 3 – **Метрики балансування навантаження**

Метрика	Опис	Що характеризує	Де застосовується
Завантаження CPU	Використання процесору	Навантаження вузла	Кластери, ЦОД, Cloud
Використання ОП	Обсяг зайнятої RAM	Ресурсні обмеження вузла	Віртуалізовані середовища
Пропускна здатність мережі	Обсяг даних годину	Мережеве навантаження	Розподілені системи, 5G
Затримка (Latency)	Час доставки	Швидкодію системи	Реальної години, URLLC
Час відгуку	Час: відповіддю – запит	Якість обслуговування	Веб-сервіси, API
Число активних з'єднань	Число відкритих сесій	Навантаження сервера	Load balancer рівня L4/L7
Втрати пакетів	Частина втрачених пакетів	Якість мережі	Мобільні мережі
Рівень використання диска	Швидкість та обсяг I/O	Стан підсистеми зберігання	Бази даних, сховища
Енергоспоживання	Споживання вузлом	Енергоефективність	IoT , “зелені” ЦОД
Надійність вузла	Імовірність відмови	Стійкість системи	Критичні системи
Пріоритет трафіку (QoS)	Важливість потоку	Гарантія сервісу	4G/5G, MC мережі
Відхилення навантаження	Різниця між вузлами	Баланс системи	Аналітична оцінка
Пропускна здатність	Число оброблених запитів	Загальну продуктивність	Високонавантажені системи
Година простою вузла	Період неактивності	Неоптимальне використання	ЦОД, серверні ферми

Окрему групу складатимуть метрики, орієнтовані на продуктивність та ефективність роботи системи в цілому, зокрема пропускну здатність, кількість активних з'єднань та середнє відхилення навантаження. Також у таблиці враховано енергетичні та надійнісні показники, що є критично важливими для хмарних середовищ, центрів обробки даних та мобільних мереж. Таким чином, наведені метрики є основою для прийняття рішень у процесі балансування навантаження та дозволяють адаптувати алгоритми до вимог конкретної середовища функціонування системи.

Приклад: оптимізація використання ресурсів у гетерогенних грід-системах з нестаціонарним вхідним потоком

Розглянемо гетерогенну грід- систему, що складається з обчислювальних вузлів із різною продуктивністю процесорів, обсягом оперативної пам'яті та пропускною здатністю мережі. У систему надходить нестаціонарний вхідний потік завдань , інтенсивність якого змінюється в години, наприклад, відповідно до добових піків навантаження або випадкових сплесків запитів. Для оптимізації використання ресурсів застосовується динамічний алгоритм балансування навантаження , який у реальному часі аналізує такі метрики: завантаження CPU, обсяг вільної оперативної пам'яті, кількість активних завдань на вузлі та середня година виконання завдань. На основі зібраної інформації формується важливий показник доступності шкірного вузла, який визначає його пріоритет для обробки нових завдань.

У періоди низької інтенсивності вхідного потоку задачі рівномірно розподіляються між вузлами з урахуванням їх продуктивності, що забезпечує максимальне використання обчислювальних ресурсів. У моменти різкого зростання навантаження алгоритм адаптивно перенаправляє нові завдання на менш завантажені та більш продуктивні вузли, а також може тимчасово обмежувати призначення завдань на вузли з високим рівнем завантаження.

Результатом такого підходу є зменшення середньої години очікування завдань , підвищення пропускної здатності грід-системи та зниження ймовірності перевантаження окремих вузлів. Таким чином, адаптивне балансування навантаження дозволяє ефективно оптимізувати використання ресурсів у гетерогенних грід- системах в умовах нестаціонарного вхідного потоку.

Введемо інтегральний показник вибору вузла

$$S_i = \alpha \cdot \frac{1}{CPU_i} + \beta \cdot \frac{1}{MEM_i} + \gamma \cdot \frac{1}{Q_i},$$

де CPU_i – завантаження процесора, MEM_i – використання пам'яті, Q_i – довжина черги завдань, α, β, γ – вагові коефіцієнти, α + β + γ = 1.

Нехай гетерогенна грід-система складається з трьох обчислювальних вузлів N_1, N_2 та N_3. Вагові коефіцієнти вибрано такими: α = 0.5, β = 0.3,γ = 0.2.

Поточний стан вузлів та розрахунок інтегрального показника наведені у табл. 1.

Таблиця 1 – **Аналіз поточного стану вузлів**

Вузол	CPU_i	MEM_i	Q_i	S_i
N_1	0.70	0.60	5	**1.254**
N_2	0.50	0.80	8	**1.400**
N_3	0.30	0.40	3	**2.484**

Найбільше значення інтегрального показника отримано для вузла N_3 : $S_3 > S_2 > S_1$. Отже, нове завдання необхідно призначити вузлу N_3 , оскільки він має найменше завантаження процесора, нижче використання пам'яті та коротку чергу завдань.

Графік рис. 1 демонструє зростання значення показника $S(t)$ у часі, що свідчить про зменшення навантаження на вузол внаслідок адаптивного перерозподілу завдань. Після початкового періоду спостерігається стабілізація показника, що вказує на досягнення квазістаціонарного режиму роботи системи.

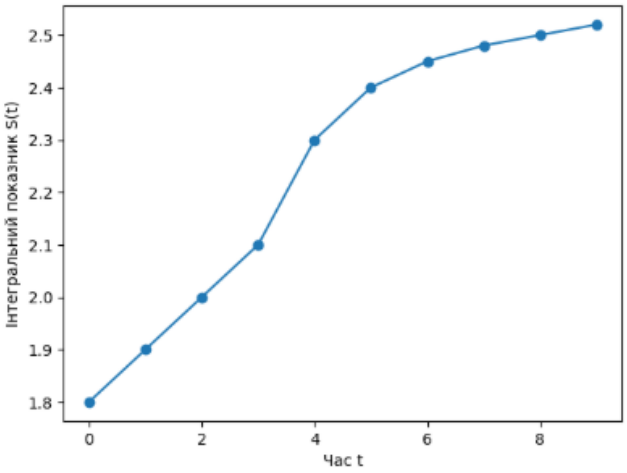

Рис. 1. Зміна інтегрального показника $S_i(t)$ у часі

Зміна інтегрального показника $S_i(t)$ у часі підтверджує ефективність адаптивного алгоритму балансування, що забезпечує стабілізацію навантаження в умовах нестаціонарного вхідного потоку.

Висновки

У роботі досліджено проблему оптимізації використання обчислювальних ресурсів у гетерогенних грід-системах в умовах нестаціонарного вхідного потоку задач. Показано, що різнорідність обчислювальних вузлів і динамічний характер навантаження істотно ускладнюють процеси планування та розподілу задач, що потребує застосування адаптивних методів балансування навантаження. Проаналізовано існуючі підходи до балансування навантаження у розподілених обчислювальних системах та обґрунтовано доцільність використання імітаційного моделювання як ефективного інструмента дослідження поведінки грід-систем у різних режимах функціонування. Отримані результати підтвердили, що застосування адаптивних алгоритмів балансування навантаження забезпечує зменшення середнього часу очікування виконання задач та покращення загальної пропускної здатності грід-системи порівняно з базовими методами розподілу задач.

Конфлікт інтересів

Автори декларують, що не мають конфлікту інтересів стосовно даного дослідження, в тому числі фінансового, особистісного характеру, авторства чи іншого характеру, що міг би вплинути на дослідження та його результати, представлені в даній статті.

Використання засобів штучного інтелекту

Автори підтверджують, що не використовували технології штучного інтелекту при створенні представленої роботи.

СПИСОК ЛІТЕРАТУРИ

1. Radchenko, V. and Andrusenko Yu. (2025), “Intelligent approach to planning taking into accountthe concept of acceptable work balance”, *Control, Navigation and Communication Systems*, Vol. 4, No. 82 (2025), pp. 121–125, doi: https://doi.org/10.26906/SUNZ.2025.4.121
2. Hung, P.T., Truong, M.D.D., Hung, P.D. (2022). Tuning Proximal Policy Optimization Algorithm in Maze Solving with ML-Agents. In: Singh, M., Tyagi, V., (eds) Advances in Computing and Data Sciences. ICACDS 2022. Communications in Computer and Information Science, vol 1614. Springer, doi: https://doi.org/10.1007/978-3-031-12641-3_21
3. Jothi, G., and Saravanan, P. (2017), “A New Algorithm to Find the Optimal Feasible Assignment for an Assignment Problem”, Int. journal of engineering research & technology, vol. 5, issue 04, doi: https://doi.org/10.17577/IJERTCONV5IS04013
4. Tan, M. & Le, Q. V. Efficientnetv2: Smaller models and faster training. ArXiv (Cornell University), Preprint arXiv:2104.00298, 2021. DOI: https://doi.org/10.48550/arXiv.2104.00298
5. Kuchuk, H. and Malokhvii, E. (2024), “Integration of IOT with Cloud, Fog, and Edge Computing: A Review”, *Advanced Information Systems*, vol. *8*(2), pp. 65–78, doi: https://doi.org/10.20998/2522-9052.2024.2.08
6. Kristensen, J.T., Valdivia, A. and Burelli, P. (2020), “Estimating player completion rate in mobile puzzle games using reinforcement learning”, Proc. of the IEEE Conference Computational Intelligence and Games, pp. 636–639, doi: https://doi.org/10.1109/CoG47356.2020.9231581
7. Zhu, W. and Rosendo, A. (2021), “A functional clipping approach for policy optimization algorithms”. IEEE Access, vol. 9, pp. 96056–96063, doi: https://doi.org/10.1109/ACCESS.2021.3094566
8. Barabash, O., Bandurka, O., Svynchuk, O. & Tverdenko, H. Method of identification of tree species composition of forests on the basis of geographic information database. *Advanced Information Systems*, 2022, vol. 6, no. 4, pp 5-10. DOI: https://doi.org/10.20998/2522-9052.2022.4.01
9. Yaloveha, V., Podorozhniak, A. & Kuchuk, H. CNN hyperparameter optimization applied to land cover classification. Radioelectronic and computer systems, 2022, no. 1 (101), pp. 115-128. DOI: https://doi.org/10.32620/reks.2022.1.09
10. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778. DOI: https://doi.org/10.1109/CVPR.2016.90
11. Singh, U., Dey, M., and Patel, P. (2025), “Enabling Grid Stability: Harnessing µPMU Data for Data-Driven Analysis of Grid Frequency Event”, *Smart Innovation Systems and Technologies*, 422, pp. 99–112, doi: https://doi.org/10.1007/978-981-96-0147-9_9
12. Sharma, R., Goel, A.K., Sharma, M.K., Dhiman, N., Mishra, V.N. (2023), “Modified Round Robin CPU Scheduling: A Fuzzy Logic-Based Approach”, L*ecture Notes in Operations Research*,, Part F3787, pp. 367–383, doi: https://doi.org/10.1007/978-981-19-8012-1_24

Received (Надійшла) 29.10.2025
Accepted for publication (Прийнята до друку) 11.02.2026
Publication date (Дата публікації) 27.02.2026

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Андрусенко Юлія Олександрівна – асистентка кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Yuliia Andrusenko – Assistant Lecturer of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: yuliia.andrusenko@nure.ua; ORCID Author ID: https://orcid.org/0000-0001-7844-2042;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=59412400500&origin=resultslist.

Радченко В'ячеслав Олексійович – старший викладач кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Viacheslav Radchenko – Senior Lecturer of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: viacheslav.radchenko@nure.ua; ORCID Author ID: https://orcid.org/0000-0001-5782-1932;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57189376280.

Parallel implementation of voice signal processing methods on multicore CPU and GPU

Viacheslav Radchenko, Yuliia Andrusenko

Abstract. The relevance of this study is driven by the growing scale of using heterogeneous grid systems to execute computationally intensive tasks under conditions of a dynamic and non-stationary input flow. The heterogeneity of computing resources and uneven workload distribution lead to reduced efficiency of the computing infrastructure and necessitate the use of adaptive load-balancing methods. **The aim of this work** is to improve the efficiency of resource utilization in heterogeneous grid systems by developing and studying load-balancing algorithms based on simulation modeling under conditions of a non-stationary task input flow. The object of the study is the processes of management and allocation of computing resources in heterogeneous grid systems. The subject of the study is methods and algorithms for load balancing and optimization of resource utilization in heterogeneous grid systems with a non-stationary task input flow based on simulation modeling. **Results.** A simulation model of a heterogeneous grid system was developed, taking into account the heterogeneity of computing nodes and the variable intensity of task arrivals. Experimental studies of the efficiency of load-balancing algorithms were conducted under normal and peak operating modes of the system. **Conclusion.** The obtained results demonstrated a reduction in the average waiting time for task execution, improved uniformity of resource utilization, and increased overall system throughput. Future research directions include extending the simulation model to account for task prioritization, quality of service (QoS) constraints, energy consumption of computing nodes, as well as applying intelligent and predictive load-balancing methods in heterogeneous distributed computing environments.

Keywords: grid system, heterogeneity, non-stationarity, load balancing, resource allocation, computing node.