

Алгоритми оптимального розподілу навантаження в хмарних обчислювальних середовищах

Громовський Олександр Володимирович¹

Опубліковано	Секція	УДК
30.12.2024	Економіка	004.75:004.272:004.42

DOI: <https://doi.org/10.5281/zenodo.16784290>

Анотація. У статті досліджено сучасні підходи до реалізації алгоритмів динамічного та адаптивного розподілу обчислювальних задач у хмарній інфраструктурі. Проаналізовано ефективність різних методів: еволюційних, машинного навчання, нечіткої логіки та гібридних моделей. Окрема увага приділена забезпеченню енергоефективності, відповідності параметрам якості обслуговування (QoS) і дотриманню угод SLA. Обґрунтовано доцільність впровадження багаторівневих стратегій управління навантаженням з урахуванням тенденцій до використання edge-обчислень і багатохмарних середовищ.

Ключові слова: енергоефективність, SLA, QoS, адаптивні моделі, гібридні стратегії, edge-інфраструктура, машинне навчання, нечітка логіка, прогнозування навантаження, самонавчальні системи.

Algorithms for optimal load distribution in cloud computing environments

Annotation. This paper explores current approaches to optimizing task distribution in cloud computing systems, focusing on intelligent workload distribution. It also examines various algorithmic solutions—from heuristic and biologically inspired methods to hybrid combinations of fuzzy logic, machine learning, and evolutionary algorithms—to improve system responsiveness, ensure efficient resource utilization, and maintain service reliability. The central theme of the research is the integration of energy-efficient scheduling, which highlights the importance of sustainability and green computing in cloud infrastructures.

The paper also highlights the growing relevance of edge computing and heterogeneous multi-cloud platforms, which require dynamic resource scaling and decentralized management mechanisms. It is shown that intelligent workload balancing plays a key role in achieving quality of service (QoS) goals and meeting service level agreements (SLAs). Through simulations and modeling, the study demonstrates that hybrid models, especially those with self-adaptive capabilities, outperform traditional static algorithms under different workload conditions.

In addition, the paper explores advanced techniques such as predictive analytics for workload forecasting and container-based deployment strategies that facilitate granular control and efficient task migration. Particular attention is paid to architectural designs that enable scalability, modularity, and real-time decision-making in distributed environments. The study results highlight the need for future research in developing autonomous and resilient

¹ Senior Software Engineer, MTS V, Panasonic Avionics Corporation, USA

algorithms that adapt to rapidly changing operational contexts while maintaining cost-effectiveness and computational throughput.

The proposed approaches contribute to a deeper understanding of how intelligent, adaptive algorithms can provide balanced and efficient workload distribution in modern cloud infrastructures, and offer practical ways to improve both technical performance and environmental sustainability.

Keywords: energy-aware strategies, self-adaptive systems, edge-based computing, SLA compliance, machine intelligence, decentralized scheduling, QoS assurance, fuzzy-based models, resource orchestration, predictive algorithms.

Вступ

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями. Хмарні обчислення виступають фундаментальною технологією для побудови гнучких, масштабованих та економічно ефективних інформаційних систем. Їх стрімкий розвиток обумовлений здатністю забезпечувати обчислювальні ресурси на вимогу користувача через мережу Інтернет, що дозволяє компаніям та установам адаптуватися до змінних навантажень і потреб без значних витрат на фізичну інфраструктуру. Проте зі зростанням кількості користувачів та обсягів оброблюваних даних, виникає низка критичних завдань, серед яких особливе місце займає проблема оптимального розподілу навантаження між доступними ресурсами.

Актуальність теми зумовлена також тим, що хмарні технології активно інтегруються в критично важливі сфери — охорону здоров'я, банківську систему, державне управління, телекомунікації, де від стабільності та ефективності обчислювального середовища залежить якість життя, безпека даних і оперативність прийняття рішень. У зв'язку з цим проблема оптимального балансування навантаження не є виключно технічною задачею, а набуває важливого соціально-економічного значення [1].

Таким чином, обрана тема є надзвичайно актуальною як з наукової, так і з прикладної точки зору, оскільки торкається одного з ключових аспектів ефективного функціонування хмарної інфраструктури. Дослідження у цій галузі спрямовані на підвищення ефективності використання наявних ресурсів, зменшення енергоспоживання, забезпечення високої доступності сервісів і зниження витрат на підтримку IT-інфраструктури. Це визначає необхідність подальшого вивчення, вдосконалення існуючих підходів та розробки нових методів оптимального розподілу навантаження в хмарних обчислювальних середовищах.

Виділення невирішених раніше частин загальної проблеми. Оптимальний розподіл навантаження в хмарному середовищі забезпечує рівномірне використання обчислювальних потужностей, зниження затримок, уникнення перевантажень та підвищення продуктивності системи загалом. Це особливо актуально для сервісів із високими вимогами до надійності та швидкодії, таких як онлайн-платформи, великі корпоративні системи, наукові симуляції чи обробка великих обсягів даних (Big Data). Неефективне використання ресурсів може призводити не лише до зниження якості обслуговування кінцевих користувачів, але й до суттєвих фінансових втрат.

З огляду на це, постає необхідність у розробці і впровадженні ефективних алгоритмів, які дозволяють динамічно і адаптивно розподіляти навантаження в умовах високої змінності навантаження середовища, різноманітності завдань та обмеженості обчислювальних ресурсів. Розробка таких алгоритмів є складним і комплексним завданням, що потребує урахування великої кількості параметрів: поточний стан системи, характеристики віртуальних машин, пріоритетність завдань, енергоспоживання тощо [2, 3].

Проблема ефективного розподілу навантаження в хмарних обчислювальних середовищах активно досліджується протягом останніх років. У світовій науковій спільноті сформувалися різноманітні підходи до її вирішення, що ґрунтуються на використанні класичних алгоритмів оптимізації, методів машинного навчання, еволюційних обчислень, а також гібридних рішень. Значний внесок у розвиток теоретичних основ балансування навантаження зробили такі дослідники, як H. Liu, R. Buyya, A. Beloglazov, які розглядали адаптивні моделі управління навантаженням із урахуванням енергоспоживання та динаміки запитів користувачів.

Дослідження, проведені R. Buyya (University of Melbourne) та його колегами, спрямовані на побудову гнучких хмарних платформ, що використовують моделі прогнозування навантаження з урахуванням відомих раніше даних та поточних метрик продуктивності. Інші автори, зокрема X. Zhang і Y. Chen, приділили увагу розподілу задач на основі штучного інтелекту, застосовуючи алгоритми глибокого навчання для передбачення пікових навантажень і динамічного перенесення віртуальних машин. Крім того, розглядаються методи імітаційного моделювання та еволюційної оптимізації, включаючи генетичні алгоритми та роеві методи, які демонструють високу ефективність у вирішенні задач з великою кількістю параметрів [4].

Українські дослідники також не залишаються осторонь проблематики. Наприклад, у працях українських авторів розглядається вплив особливостей телекомунікаційної інфраструктури на ефективність хмарних обчислень, питання сумісності програмного забезпечення в багатохмарному середовищі, а також безпекові аспекти розподілу навантаження. Результати таких досліджень доводять необхідність врахування регіональних особливостей під час впровадження стандартних рішень у локальних умовах.

Незважаючи на велику кількість праць, існують проблемні зони, що залишаються відкритими. Зокрема, недостатньо досліджено питання адаптивності алгоритмів до непередбачуваних змін у навантаженні, взаємодії між кількома центрами обробки даних та балансування у багатохмарних архітектурах. Актуальною також є проблема побудови універсальних підходів до розподілу навантаження, які б були незалежними від конкретної платформи чи хмарного провайдера. Саме ці невирішені питання формують основу для подальших досліджень, у тому числі й у межах даної статті.

Формулювання цілей статті (постановка завдання). Метою даної статті є дослідити сучасні алгоритми оптимального розподілу навантаження в хмарних обчислювальних середовищах з акцентом на їх адаптивність, ефективність та застосовність у різних архітектурних конфігураціях. Особлива увага приділяється підходам, що базуються на гібридизації методів машинного навчання та еволюційної оптимізації, а також розгляду можливостей динамічного балансування у мультихмарних і гетерогенних середовищах.

Завдання статті. Завданнями статті є узагальнення існуючих підходів до розподілу навантаження, аналіз їх переваг і обмежень, теоретичне обґрунтування класифікації алгоритмів за критеріями адаптивності, складності, масштабованості та ефективності, а також формування пропозицій щодо удосконалення або комбінування методів для підвищення якості балансування. Крім цього, провести порівняльний аналіз ефективності різних алгоритмів та визначити найперспективніші напрямки подальших розробок у даній сфері.

Результати

Оптимальний розподіл навантаження в хмарному середовищі базується на використанні різноманітних підходів, які залежать від специфіки задачі, доступних ресурсів і параметрів середовища. У практиці розрізняють кілька категорій алгоритмів:

статичні, динамічні, еволюційні, стохастичні та гібридні. Кожен з них має свої переваги й обмеження, що визначають доцільність їх використання в конкретному контексті.

Статичні алгоритми, як-от Round Robin або Least Connection, розподіляють навантаження за наперед визначеними правилами. Вони мають просту реалізацію, проте не враховують динаміку змін у системі, що робить їх малоприсадибними в умовах високої мінливості навантаження. Однак, їхня нездатність адаптуватися до динамічних змін навантаження суттєво обмежує застосування у середовищах із високою варіативністю запитів. У випадках стабільних або передбачуваних сценаріїв, такі алгоритми можуть бути цілком ефективними, що робить їх оптимальним вибором для малонавантажених або спеціалізованих систем.

Натомість динамічні алгоритми, серед яких Weighted Least Connection, Least Response Time або Adaptive Load Balancing, активно враховують поточний стан системи, оперативно коригуючи рішення щодо розподілу навантаження. Вони вимагають постійного моніторингу параметрів, таких як навантаження процесорів, пропускна здатність мережі, час відгуку сервера. Цей підхід дозволяє більш ефективно використовувати ресурси, підвищуючи загальну продуктивність системи та покращуючи якість обслуговування користувачів, особливо в умовах непередбачуваних пікових навантажень.

Еволюційні та інтелектуальні підходи, такі як генетичні алгоритми, алгоритми рою частинок, машинне навчання та глибоке навчання, використовують адаптивні механізми пошуку оптимального розв'язку. Вони реалізують адаптивні механізми пошуку глобального або локального оптимуму, здатні враховувати велику кількість параметрів і складних взаємозв'язків у системі. Такі методи здатні враховувати велику кількість параметрів і забезпечують високу точність балансування. Наприклад, моделі на основі глибокого навчання дозволяють передбачити майбутнє навантаження, що підвищує ефективність розміщення віртуальних машин.

Особливої уваги заслуговують комбіновані гібридні алгоритми, що поєднують різні підходи для досягнення балансу між точністю, швидкістю та адаптивністю. Наприклад, комбінація генетичних алгоритмів із методами кластеризації або нечіткої логіки дозволяє більш гнучко реагувати на зміни навантаження, забезпечуючи стабільність і високу продуктивність. Такі рішення дозволяють використовувати сильні сторони кожного методу й компенсувати їх слабкі місця.

З метою порівняння основних типів алгоритмів розподілу навантаження, які використовуються в хмарних обчислювальних середовищах представимо табл.1.

Таблиця 1

Порівняння основних типів алгоритмів балансування навантаження

Тип алгоритму	Принцип роботи	Переваги	Недоліки	Приклади застосування
Статичний	Завдання призначаються заздалегідь	Простота, низька вартість реалізації	Неадаптивність, слабка масштабованість	Малонавантажені системи
Динамічний	Враховує поточний стан системи	Гнучкість, адаптивність	Складність, потреба в моніторингу	Реальні хмарні сервіси
Алгоритми ML/AI	Прогнозування навантаження	Самонавчання, оптимізація в реальному часі	Висока складність, вимоги до даних	Amazon EC2, Google Cloud
Еволюційні (GA, PSO тощо)	Популяційна оптимізація	Висока якість результату	Обчислювальні витрати, складність налаштування	Великі розподілені середовища

Гібридні	Комбінація підходів	Баланс між точністю та швидкістю	Залежить від контексту	Хмарні сервіси високого навантаження
----------	---------------------	----------------------------------	------------------------	--------------------------------------

Для виокремлення сильних сторін кожного підходу та вибору оптимального варіанту для конкретних умов експлуатації необхідно провести оцінку ефективності алгоритмів розподілу навантаження за основними метриками, такими як час відповіді, завантаження процесора, пропускна здатність, енергоспоживання та надійність (табл.2).

Таблиця 2

Ефективність алгоритмів за ключовими метриками

Алгоритм	Час відповіді	Завантаження CPU	Пропускна здатність	Енергоспоживання	Надійність
Round Robin	Середній	Високе	Середня	Високе	Низька
Least Connection	Низький	Помірне	Висока	Середнє	Середня
Ant Colony Optimization	Низький	Низьке	Висока	Низьке	Висока
Genetic Algorithm	Дуже низький	Високе	Висока	Високе	Висока
Deep Reinforcement Learning	Найнижчий	Збалансоване	Найвища	Середнє	Висока

Успішні практичні результати досліджень, присвячених динамічному балансуванню навантаження в хмарних обчислювальних середовищах, свідчать про ефективність описаних підходів до покращення розподілу ресурсів, мінімізації часу відгуку системи та підвищення її загальної продуктивності [5]. Для цього, на практиці, розробляють, реалізують та тестують кілька алгоритмів балансування навантаження. Їхню ефективність оцінюють шляхом порівняння з традиційними методами, зокрема за умов змінних сценаріїв навантаження.

Основними критеріями оцінки вибирають такі показники, як час відгуку системи, пропускна здатність, ефективність використання ресурсів і масштабованість. Також значну увагу приділяють енергоефективності — як додатковому критерію, що набуває дедалі більшого значення в контексті екологічної сталості. Такого роду алгоритми враховують не лише динаміку навантаження, а й моделі енергоспоживання, що сприяє впровадженню підходів екологічних обчислень у хмарну інфраструктуру.

Оптимальний розподіл навантаження в хмарних обчислювальних середовищах відіграє ключову роль у підвищенні ефективності використання обчислювальних ресурсів, скороченні часу реагування системи та забезпеченні стабільного управління потоками завдань. У цьому контексті розроблено цілу низку алгоритмів балансування навантаження, кожен з яких відрізняється специфічними підходами до розподілу задач і має власні характеристики продуктивності.

Значну увагу науковців привертає метод динамічної оптимізації роїв частинок (Dynamic PSO), який вдосконалює класичний алгоритм PSO шляхом інтеграції механізмів динамічної мутації. Це дозволяє підвищити адаптивність системи до змін у навантаженні, зменшити загальний час обробки завдань і більш раціонально використовувати ресурси, що особливо важливо в умовах високої варіативності запитів.

Ще одним перспективним підходом є метод планування подільного навантаження (DLS), який забезпечує гнучкий розподіл завдань між вузлами обчислювальної інфраструктури залежно від їхньої поточної завантаженості. Завдяки динамічній природі такого розподілу, DLS демонструє вищу продуктивність, знижує час відгуку

системи та забезпечує ефективнішу міграцію завдань порівняно з традиційними методами.

Особливий інтерес становить також модель балансування навантаження на основі нечіткої логіки (OFLB). Застосування нечітких правил дозволяє враховувати невизначеність у вимогах користувачів і поточному стані ресурсів, забезпечуючи адаптивний розподіл задач. Дослідження показали, що використання OFLB підвищує ефективність розподілу ресурсів приблизно на 18% порівняно з класичними алгоритмами, завдяки здатності моделі гнучко класифікувати ресурси та враховувати численні обмеження.

Незважаючи на позитивні результати, жоден із розглянутих алгоритмів не є універсальним. Залишається низка викликів, пов'язаних із необхідністю адаптації до непередбачуваних змін у навантаженні та конфігураціях інфраструктури, що зумовлює потребу в подальших дослідженнях і розробці гібридних рішень.

Варто зазначити, що у напрямку багатохмарних середовищ проблема ускладнюється через необхідність врахування взаємодії між різними платформами, що вимагає додаткових механізмів координації та обміну даними. Дослідження показують, що ефективне балансування навантаження в таких умовах може бути досягнуте шляхом використання децентралізованих алгоритмів, зокрема на основі блокчейн-технологій, які забезпечують прозорість і надійність управління ресурсами.

Окрім технічних аспектів, важливо враховувати й економічні фактори. Розподіл навантаження має бути не лише ефективним, але й економічно доцільним. Тому сучасні алгоритми включають вартісні моделі, які дозволяють зменшити експлуатаційні витрати хмарних провайдерів та користувачів. У цьому контексті активно досліджуються методи оптимізації енергоспоживання, зокрема через інтеграцію «зелених» обчислень (green computing) для живлення дата-центрів. У даному контексті алгоритми розподілу навантаження можуть враховувати не лише технічний стан ресурсів, а й доступність «зеленої» енергії в певний момент часу. Наприклад, у сонячну погоду система може використовувати максимум ресурсів, що живляться від сонячних панелей, тоді як у похмуру погоду — перемикається на ресурси, які живляться від інших джерел. Це потребує впровадження адаптивних, розподілених та децентралізованих схем балансування з високим рівнем автономності.

Значну роль у забезпеченні екологічної сталості відіграє також моніторинг та аналітика, що дозволяє в режимі реального часу оцінювати ефективність енергоспоживання, виявляти аномалії, проводити аудит ресурсів і формувати рекомендації щодо їх оптимізації. Розробка та впровадження відповідних модулів моніторингу — це невід'ємна частина інфраструктури зелених хмарних систем, яка забезпечує зворотний зв'язок для вдосконалення алгоритмів розподілу.

Хоча первинні витрати на реалізацію енергоефективних технологій можуть бути вищими, у довгостроковій перспективі вони дозволяють суттєво скоротити експлуатаційні витрати. Наприклад, за оцінками Amazon та Google, впровадження енергозберігаючих механізмів у центрах обробки даних дозволило зменшити витрати на електроенергію до 40% порівняно з традиційними системами. Отже, green computing — це не лише екологічний, а й економічно вигідний підхід.

Зазначимо, що ефективна інтеграція зелених технологій у хмарну інфраструктуру передбачає комплексний підхід — від проектування енергоефективного обладнання до розробки адаптивних, інтелектуальних алгоритмів управління навантаженням.

Основні етапи інтеграції «зелених» обчислень можна структурувати наступним чином:

1. Визначення цілей сталого розвитку та енергоспоживання:
 - Формування стратегії енергоефективності на рівні провайдера хмарних послуг або дата-центру.

- Визначення основних метрик: допустимий рівень енергоспоживання, показники вуглецевого сліду, цілі щодо використання відновлюваних джерел енергії.
- Прив'язка технічних завдань до стратегічних екологічних цілей компанії або організації.
- 2. Аналіз поточного стану інфраструктури (енергетичний аудит):
 - Визначення рівня енергоспоживання серверів, систем охолодження, мережевого обладнання.
 - Виявлення "гарячих зон" надмірного навантаження або нераціонального використання ресурсів.
 - Оцінка ефективності існуючих алгоритмів розподілу навантаження з погляду енергозбереження.
- 3. Вибір та адаптація енергоефективних алгоритмів балансування навантаження:
 - Вибір алгоритмів, що враховують споживання енергії як одну з ключових метрик (наприклад, Energy-Aware Load Balancing, DVFS-based Scheduling).
 - Адаптація існуючих алгоритмів (наприклад, PSO, Genetic Algorithm, Reinforcement Learning) для оптимізації не лише продуктивності, а й енерговитрат.
 - Визначення порогів активації/деактивації обчислювальних вузлів відповідно до динаміки навантаження.
- 4. Інтеграція інтелектуальних механізмів прогнозування:
 - Впровадження моделей машинного навчання для прогнозування піків навантаження і відповідного регулювання активних вузлів.
 - Використання історичних даних для навчання моделей щодо енергоспоживання та поведінки системи.
 - Застосування глибокого навчання (наприклад, LSTM) для побудови точних моделей передбачення енергетичних потреб.
- 5. Реорганізація та модернізація фізичної інфраструктури:
 - Використання енергоефективних серверів з низьким тепловиділенням та підтримкою енергозберігаючих режимів.
 - Модернізація систем охолодження: перехід до природної вентиляції, використання водяного або пасивного охолодження.
 - Упровадження обладнання, сумісного з джерелами відновлюваної енергії (наприклад, сонячні батареї, вітрові турбіни).
- 6. Впровадження системи моніторингу енергоспоживання:
 - Розробка панелей візуалізації та систем звітності, що дозволяють у режимі реального часу контролювати рівень споживаної енергії.
 - Автоматичне виявлення аномалій у роботі обладнання (наприклад, серверів, що залишаються ввімкненими без навантаження).
 - Інтеграція моніторингу в систему прийняття рішень для алгоритмів розподілу.
- 7. Моделювання сценаріїв і тестування системи:
 - Побудова віртуальних середовищ для симуляції поведінки системи в різних умовах навантаження та доступності енергії.
 - Аналіз поведінки алгоритмів у контексті конфлікту між продуктивністю та енергоефективністю.
 - Підбір оптимальних налаштувань алгоритмів (наприклад, порогів навантаження, параметрів енергозбереження).
- 8. Оцінка ефективності та зворотний зв'язок:
 - Аналіз результатів за обраними метриками: зниження енергоспоживання, підвищення коефіцієнта енергоефективності (PUE), зменшення викидів CO₂.
 - Порівняння до- і післявпроваджувального стану системи.

- Збір відгуків операторів і технічного персоналу для виявлення практичних проблем і можливостей покращення.
9. Постійне вдосконалення та масштабування:
- Удосконалення алгоритмів на основі результатів моніторингу.
 - Масштабування енергоефективних рішень на інші частини інфраструктури або в нові дата-центри.
 - Інтеграція у стратегію довгострокового розвитку компанії: автоматичні оновлення, хмарні сервіси з низьким вуглецевим слідом, сертифікація (наприклад, ISO 50001).

Вказані структурні компоненти мають взаємодіяти у єдиній екосистемі, яка забезпечує не лише продуктивність і надійність, але й мінімальний вплив на довкілля. Подальший розвиток у цьому напрямі є важливою умовою сталого розвитку ІТ-індустрії та створення екологічно безпечного цифрового майбутнього.

Таким чином, виклад основного матеріалу свідчить про багатогранність підходів до розв'язання задачі оптимального розподілу навантаження. Успішне застосування того чи іншого алгоритму залежить від розуміння специфіки середовища, вимог до системи, технічних і фінансових обмежень, а також стратегічних пріоритетів організації.

Висновки

У ході проведеного дослідження було розглянуто ключові підходи до оптимального розподілу навантаження в хмарних обчислювальних середовищах. Виявлено, що ефективність роботи таких систем значною мірою залежить від вибору алгоритму, що визначає правила призначення завдань на доступні обчислювальні ресурси. Проаналізовані алгоритми – як статичні, так і динамічні – демонструють різний рівень ефективності залежно від умов середовища, навантаження, типу сервісів та потреб користувачів.

Зокрема, адаптивні та гібридні алгоритми показали найвищий потенціал у забезпеченні рівномірного розподілу ресурсів, мінімізації часу відповіді та зменшенні енергоспоживання, що особливо важливо для масштабованих хмарних платформ. Методи, що базуються на машинному навчанні, а також еволюційні підходи, поступово знаходять застосування в реальних системах завдяки здатності прогнозувати зміну навантаження і динамічно адаптувати розподіл.

Встановлено, що найбільш перспективними є багаторівневі архітектури управління навантаженням, що поєднують локальні алгоритми розподілу з глобальними політиками управління в багатохмарних і гетерогенних інфраструктурах. Значна ефективність продемонстрована також при впровадженні стратегій, орієнтованих на енергозбереження, що підтверджує доцільність інтеграції концепції green computing у балансування навантаження.

У статті також окреслено актуальні проблеми, пов'язані з безпекою, SLA-вимогами та обмеженнями в реальному часі, які потрібно враховувати при розробці майбутніх алгоритмів. Результати дослідження свідчать про необхідність врахування не лише технічних параметрів (час відповіді, пропускну здатність, масштабованість), а й сервісних характеристик (QoS, SLA), які стають вирішальними в контексті високодоступних і критично важливих хмарних сервісів.

Таким чином, розробка інтелектуальних, самоналаштовуваних алгоритмів, здатних працювати в режимі реального часу та динамічно реагувати на зміни навантаження, є одним із ключових векторів подальших досліджень.

Список використаних джерел

1. Ahmed M., Khatri M., Goyal J. An Optimized Fuzzy-based Load Balancing in Cloud Computing. 2023. P. 323–328. <https://doi.org/10.1109/REEDCON57544.2023.10150583>
2. Ramya R., Vishnu Durai R. S., Vijayakumar R., DhivyaShree M. Optimal Divisible Load Scheduling for Load Balancing in Cloud Computing. 2024. P. 545–550. <https://doi.org/10.1109/icesc60852.2024.10689908>
3. Meshram P., Dwivedi Y., Kumari D. A., Padhy R., Kumar A. Optimizing Task Load Distribution in Cloud Environments via Dynamic PSO Algorithm. 2024. P.1–5. <https://doi.org/10.1109/iccnt61001.2024.10725092>
4. Viswanathan A. Reinforcement Learning Based Metaheuristic Algorithm For Optimized Load Balancing In Cloud Environment. Journal of Electrical Systems. 2024. <https://doi.org/10.52783/jes.2520>
5. Nandwal A., Jain R. Dynamic Load Balancing for Improved Resource Allocation in Cloud Environments, 2023. <https://doi.org/10.52783/tjjpt.v44.i3.408>